

# **Estudo das Limitações Energéticas em Sensores Inteligentes sem Fios Compatíveis com a Norma IEEE1451**

**João Luís Dâmaso Pereira**

Tese para obtenção do Grau de Doutor em  
**Engenharia Eletrotécnica e de Computadores**  
(3<sup>o</sup> ciclo de estudos)

Orientador: Doutor António Eduardo Vitória do Espírito Santo

**setembro de 2023**

**Provas públicas realizadas em 12 de setembro de 2023**

**Composição do júri**

**Presidente**

**Doutor António João Marques Cardoso**

Professor Catedrático da Universidade da Beira Interior

**Vogais**

**Doutor Gil Manuel Magalhães de Andrade Gonçalves**

Professor Auxiliar da Faculdade de Engenharia da Universidade do Porto

**Doutor Paulo Jorge Pinto Leitão**

Professor Coordenador Principal do Instituto Politécnico de Bragança

**Doutor António Eduardo Vitória do Espírito Santo**

Professor Auxiliar da Universidade da Beira Interior

**Doutor José António da Costa Salvado**

Professor Auxiliar da Universidade da Beira Interior

## **Declaração de Integridade**

Eu, João Luís Dâmaso Pereira, que abaixo assino, estudante com o número de inscrição d2589 do 3º Ciclo em Engenharia Eletrotécnica e de Computadores da Faculdade de Engenharia, declaro ter desenvolvido o presente trabalho e elaborado o presente texto em total consonância com o **Código de Integridades da Universidade da Beira Interior**.

Mais concretamente afirmo não ter incorrido em qualquer das variedades de Fraude Académica, e que aqui declaro conhecer, que em particular atendi à exigida referência de frases, extratos, imagens e outras formas de trabalho intelectual, e assumindo assim na íntegra as responsabilidades da autoria.

Universidade da Beira Interior, Covilhã 10/10/2023

(assinatura conforme Cartão de Cidadão ou preferencialmente  
assinatura digital no documento original se naquele mesmo formato)



Este trabalho de tese foi desenvolvido no Laboratório de Instrumentação e Medida na Universidade da Beira Interior com o apoio do projeto INDTECH 4.0 – Novas tecnologias para fabricação, tem como objetivo geral a conceção e desenvolvimento de tecnologias inovadoras no contexto da Indústria 4.0/Factories of the Future (FoF), under the number POCI-01-0247-FEDER-026653.

Cofinanciado por:



UNIÃO EUROPEIA  
Fundo Europeu  
de Desenvolvimento Regional



# **Dedicatória**

Dedico às duas pessoas mais importantes da minha vida, à minha mulher Cristina Soares e ao meu filho Eduardo Pereira.



# Agradecimentos

Quero agradecer a todas aquelas pessoas que direta ou indiretamente me acompanharam e ajudaram no decorrer destes três anos letivos de doutoramento.

Em particular à minha família, Cristina Soares e Eduardo Pereira, pelo apoio que me permitiu ter a resiliência necessária e pela ajuda que me permitiu ter tempo para realizar todo o trabalho descrito nesta tese.

Aos companheiros de luta, Rita Pinto e Helbert da Rocha. Em especial ao Helbert da Rocha que me acompanhou nesta batalha de quase quatro anos, de muito trabalho, conhecimento partilhado, trabalho de equipa, cumplicidade e amizade. Helbert amigo, o meu muito obrigado.

Agradeço ao Joel Bonifácio a amizade de tantos anos, amigo a resiliência é o segredo.

Ao Professor Doutor António Eduardo de Vitória Espírito Santo pela oportunidade da realização deste estudo.

Finalmente, um agradecimento a todos os intervenientes do projeto INDTECH 4.0 – Novas tecnologias para fabricação inteligente.

Lutai sempre pelos vossos ideais, com perseverança, honestidade e frontalidade.

Que a vida vos sorria.

Possuir conhecimento, por muito que seja, sem lhe dar uso para o benefício da ciência e da Humanidade, de nada serve. Para ser útil é necessário dar-lhe um significado da vida real, que nos permita evoluir como sociedade.

João Pereira



# Resumo

Este estudo de investigação científica e tecnológica, surgiu inicialmente com a integração do projeto INDTECH 4.0 - Novas tecnologias para fábrica inteligente. Projeto que tem como principal objetivo a conceção e desenvolvimento de tecnologias inovadoras no contexto da Indústria 4.0. Do projeto fazem parte um conjunto de empresas e universidades, destacando-se a PSA – Peugeot Citroen em Mangualde.

Com a massificação de sensores nas IoT (*Internet of Things*) há a necessidade de normalizar sistemas de rede de sensores sem fios. Com este estudo provou-se que é possível utilizar uma norma, já existente, numa rede de sensores inteligente sem fios. Neste estudo houve a necessidade de criar ferramentas de desenvolvimento que permitam de uma forma simplificada desenvolver um sensor sem fios. Foi ainda estudada a forma de como se integrar esta norma numa rede de sensores sem fios com limitações energéticas.

A temática da proposta do trabalho centra-se no estudo da disponibilidade energética em sensores inteligentes sem fios, sendo estes compatíveis com a norma IEEE1451. A relevância e necessidade deste estudo justifica-se pelo facto de a recolha de energia ser, cada vez mais, uma solução tecnologicamente viável capaz de fornecer a energia necessária ao funcionamento de sensores inteligentes sem fios. A obtenção de resultados requereu a construção de plataformas de *hardware* e *software*, inexistentes, que suportam o desenvolvimento de sensores e atuadores que respeitam a norma IEEE1451. Foi construída uma rede de sensores inteligentes sem fios, que utiliza a energia recolhida do ambiente para operar. O estudo de diferentes cenários de funcionamento por parte do sensor inteligentes sem fios, integrado na rede sem fios, permite definir critérios de funcionamento e estabelecer os valores a assumir, em conformidade com a disponibilidade energética no nodo sensor num dado momento. O resultado deste trabalho terá um forte impacto na família da norma IEEE1451, em particular, nos resultados do grupo de trabalho IEEEP21451.002. Nas ferramentas de desenvolvimento especialmente concebidas para a norma, o estudo energético de uma TIM sem fios e uma proposta de adição de uma nova TEDS de energia para facilitar a implementação de TIM sem fios e com restrições energéticas.

## Palavras-chave

IoT;IEEE1451;Wireless;Sensor;Energia;Energia*Harvesting*.



# Abstract

This scientific and technological research study initially emerged with the integration of the INDTECH 4.0 project - New technologies for smart factory. Project whose main objective is the design and development of innovative technologies in the context of Industry 4.0. A group of companies and universities are part of the project, with emphasis on PSA – Peugeot Citroen in Mangualde.

With the massification of sensors in the IoT (Internet of Things) there is a need to standardize wireless sensor network systems. With this doctoral work it was proved that it is possible to use an existing standard in a wireless intelligent sensor network. In this study there was a need to create development tools that allow a simplified way to develop a wireless sensor. It was also studied how to integrate this standard in a wireless sensor network with energy limitations.

The theme of the work proposal is centered on the study of energy availability in intelligent wireless sensors, which are compatible with the IEEE1451 standard. The relevance and necessity of this study is justified by the fact that energy collection is increasingly, a technologically viable solution capable of supplying the energy necessary for the operation of wireless smart sensors. Obtaining results requires the construction of non-existent hardware and *software* platforms that support the development of sensors and actuators that comply with the IEEE1451 standard. A wireless smart sensor network was built, which uses energy collected from the environment to operate. The study of different operating scenarios by the intelligent wireless sensor, integrated in the wireless network, allows defining operating criteria and establishing the values to be assumed, in accordance with the energy availability in the sensor node at a given moment. The result of this work will have a strong impact on the IEEE1451 standard family, in particular, on the results of the IEEE P21451.002 working group. In the development tools specially designed for the standard, the energy study of a wireless TIM and a proposal to add a new energy TEDS to facilitate the implementation of wireless TIM with energy restrictions.

## Keywords

IoT;IEEE1451;Wireless;Sensor;Energia;Energia*Harvesting*.



# Índice

Dedicatória.....	VII
Agradecimentos .....	IX
Resumo .....	XI
Abstract.....	XIII
Índice .....	XV
Lista de figuras.....	XIX
Lista de tabelas .....	XXIII
Nomenclatura .....	XXVII
CAPÍTULO 1.....	1
METODOLOGIA DO PLANO DE TRABALHO .....	1
1.1. INTRODUÇÃO.....	1
1.2. O PROBLEMA EM ESTUDO E A SUA RELEVÂNCIA .....	2
1.3. OBJETIVOS E CONTRIBUIÇÃO DA DISSERTAÇÃO.....	2
1.4. VISÃO GERAL E ORGANIZAÇÃO .....	4
1.5. REFERÊNCIAS BIBLIOGRÁFICAS.....	7
CAPÍTULO 2.....	9
ESTADO DA ARTE .....	9
2.1. INTRODUÇÃO.....	9
2.2. SENSORES .....	13
2.2.1. <i>Tipos de sensores</i> .....	13
2.2.2. <i>Precisão e Exatidão de Sensores e sua Problemática</i> .....	14
2.2.3. <i>Sensores Utilizados neste Estudo</i> .....	16
2.3. SENSORES INTELIGENTES .....	18
2.4. REDE DE SENSORES SEM FIOS.....	20
2.4.1. <i>Comunicação Sem Fios</i> .....	20
2.4.2. <i>Aplicações Disponíveis</i> .....	21
2.5. ENERGIA HARVESTING.....	23
2.6. FONTES DE ENERGIA HARVEST.....	23
2.6.1. <i>Fotovoltaico</i> .....	24
2.6.2. <i>Termoelétrico (Gradiente de Temperatura)</i> .....	24
2.6.3. <i>Piezoelétrico (Vibração)</i> .....	24
2.6.4. <i>Eletromagnética (Vibração)</i> .....	25
2.6.5. <i>Rádio Frequência Ambiente</i> .....	25

2.6.6.	<i>Geradores Puramente Eletrostáticos</i> .....	25
2.7.	O IMPACTO DAS RESTRIÇÕES ENERGÉTICAS NOS SENSORES INTELIGENTES.....	26
2.8.	NORMA IEEE1451.....	26
2.8.1.	<i>Breve História da Norma IEEE1451</i> .....	26
2.8.2.	<i>Estrutura da Norma IEEE1451</i> .....	27
2.8.3.	<i>Estrutura de Comandos/Resposta</i> .....	30
2.8.4.	<i>Comando/Resposta Query TEDS</i> .....	31
2.8.5.	<i>Comando Read TEDS</i> .....	33
2.8.6.	<i>Comando/Resposta Read TransducerChannel Data-set Segment</i> .....	33
2.8.7.	<i>TEDS</i> .....	34
2.8.8.	<i>Codificação da Meta-TEDS</i> .....	37
2.8.9.	<i>Codificação da TransducerChannel TEDS</i> .....	43
2.8.10.	<i>Codificação da End users' application-specific TEDS</i> .....	57
2.8.11.	<i>Codificação da User's Transducer Name TEDS</i> .....	57
2.8.12.	<i>Codificação da PHY TEDS</i> .....	58
2.8.13.	<i>TIM</i> .....	62
2.8.14.	<i>NCAP</i> .....	64
2.9.	REFERÊNCIAS BIBLIOGRÁFICAS .....	65
CAPÍTULO 3.....		71
PLATAFORMAS DE DESENVOLVIMENTO PARA A NORMA IEEE1451 .....		71
3.1.	INTRODUÇÃO.....	71
3.2.	SERVIDOR WEB.....	71
3.2.1.	<i>Configuração do Servidor</i> .....	73
3.2.2.	<i>Base de dados</i> .....	76
3.3.	DESENVOLVIMENTO TIM CABLADA .....	78
3.3.1.	<i>Hardware TIM Cablada</i> .....	79
3.3.2.	<i>Software TIM com fios</i> .....	80
3.4.	DESENVOLVIMENTO DA TIM SEM FIOS .....	81
3.4.1.	<i>Hardware da TIM sem Fios</i> .....	81
3.4.2.	<i>Ponte Wireless</i> .....	83
3.5.	DESENVOLVIMENTO NCAP .....	85
3.5.1.	<i>Hardware NCAP</i> .....	85
3.5.2.	<i>Software NCAP</i> .....	86
3.6.	PLATAFORMA DE DESENVOLVIMENTO IEEE1451 .....	87
3.6.1.	<i>TIM TEDS Editor</i> .....	88

3.6.2.	<i>TIM Project Editor</i> .....	91
3.6.3.	<i>Validador TIM</i> .....	94
3.6.4.	<i>TIM Manager</i> .....	101
3.6.5.	<i>Tutoriais</i> .....	105
3.7.	REFERÊNCIAS BIBLIOGRÁFICAS.....	106
CAPÍTULO 4.....		109
ANÁLISE ENERGÉTICA DE SENSORES INTELIGENTES.....		109
4.1.	INTRODUÇÃO.....	109
4.2.	ESTUDO ENERGÉTICO NECESSÁRIO PARA A TIM.....	110
4.2.1.	<i>Painel fotovoltaico</i> .....	118
4.2.2.	<i>Armazenamento de Energia</i> .....	120
4.2.3.	<i>Energia Disponível</i> .....	121
4.3.	DESENVOLVIMENTO DE TIM COM ENERGIA <i>HARVEST</i> .....	123
4.3.1.	<i>Diagrama do Circuito da TIM Harvest</i> .....	123
4.3.2.	<i>Software da TIM Harvest</i> .....	125
4.3.3.	<i>Algoritmo de controle de energia disponível</i> .....	129
4.4.	PROPOSTA DE ADIÇÃO PARA TEDS-ENERGY.....	131
4.5.	REFERÊNCIAS BIBLIOGRÁFICAS.....	133
CAPÍTULO 5.....		135
TESTE E VALIDAÇÃO DE SENSORES INTELIGENTES.....		135
5.1.	INTRODUÇÃO.....	135
5.2.	TIM SEM FIOS.....	135
5.3.	MONTAGEM EXPERIMENTAL.....	137
5.3.1.	<i>TIM 1 – Caudalímetro</i> .....	140
5.3.2.	<i>TIM 2 – Nível de Água</i> .....	142
5.3.3.	<i>TIM 3 – Controlador da Bomba de Água</i> .....	146
5.3.4.	<i>Aplicação</i> .....	149
5.4.	TIM DE ENERGIA <i>HARVEST</i> .....	151
5.4.1.	<i>Aplicação TIM Harvest</i> .....	152
5.4.2.	<i>Valores obtidos com a TIM Harvest</i> .....	153
5.5.	REFERÊNCIA BIBLIOGRÁFICAS.....	157
CAPÍTULO 6.....		159
CONCLUSÃO.....		159
6.1.	CONCLUSÕES GERAIS.....	159
6.2.	PROPOSTAS DE ADIÇÕES À NORMA.....	161

6.3.	LINHAS ORIENTADORAS PARA FUTURAS CONTRIBUIÇÕES .....	164
6.4.	CONTRIBUIÇÕES RESULTANTES DA PESQUISA DESTE ESTUDO .....	164
6.4.1.	<i>Lista de Publicações</i> .....	165
6.4.2.	<i>Participação em Conferências Científicas, no País e no Estrangeiro</i> .....	166

# Lista de Figuras

## Capítulo 1

Sem Figuras.

## Capítulo 2

Figura 2.1 - Pacote de dados do sensor TPMS .....	10
Figura 2.2 - Precisão vs Exatidão .....	14
Figura 2.3 - Exemplo de histerese de um sensor de pressão .....	15
Figura 2.4 - Estrutura sensor inteligente IEEE1451 .....	18
Figura 2.5 - Estrutura sensor sem fios .....	18
Figura 2.6 - Estrutura de sensor sem fios alimentado com energia <i>Harvest</i> .....	19
Figura 2.7 - Efeito Piezoelétrico .....	24
Figura 2.8 - Diagrama de Blocos IEEE1451 .....	27
Figura 2.9 - Arquitetura IEEE1451 .....	28
Figura 2.10 - Estrutura Canal Transdutor .....	29
Figura 2.11 - Exemplo de mensagem/resposta de um comando .....	31
Figura 2.12 - Organização de uma TEDS .....	35
Figura 2.13 - Estrutura do <i>Globally Unique Identifier</i> .....	38
Figura 2.14 - Estrutura interna de uma TIM .....	39
Figura 2.15 - Estrutura interna de uma NCAP .....	40

## Capítulo 3

Figura 3.1 - Serviços do Servidor .....	72
Figura 3.2 - <i>Backup Script</i> .....	74
Figura 3.3 - Estrutura da Base de Dados para as TEDS .....	76
Figura 3.4 - TIM ↔ NCAP comunicação com fios .....	79
Figura 3.5 - Programa da TIM com fios .....	80
Figura 3.6 - TIM ↔ NCAP comunicação sem fios .....	81
Figura 3.7 - Rede sensores sem fios .....	82
Figura 3.8 - EZ430-RF2500 .....	82
Figura 3.9 - Programa do adaptador UART ↔ RF .....	83
Figura 3.10 - Configuração CC2500 com <i>SmartRF Studio</i> .....	84

Figura 3.11 - Programa NCAP .....	86
Figura 3.12 - Gerenciador de TEDS .....	88
Figura 3.13 - Editor de TEDS .....	89
Figura 3.14 - Editor de projetos TIM .....	91
Figura 3.15 - Adicionar nova TIM .....	92
Figura 3.16 - Editor de Código em Blocos .....	93
Figura 3.17 - Verificador de Código em Bloco .....	94
Figura 3.18 - <i>Hardware</i> de validação de TIM IEEE1451 .....	95
Figura 3.19 - Estrutura interna do <i>plugin</i> .....	96
Figura 3.20 - Interface TIM $\leftrightarrow$ Validador .....	97
Figura 3.21 - Validador TIM IEEE1451 .....	98
Figura 3.22 - Decodificação TEDS .....	99
Figura 3.23 - Leitura e decodificação das TEDS .....	100
Figura 3.24 - <i>Plugin</i> a correr .....	100
Figura 3.25 - Gerenciador de TIM .....	101
Figura 3.26 - Fluxograma do programa <i>TIM Discovery</i> .....	102
Figura 3.27 - Aplicação Online .....	103
Figura 3.28 - Envio de comandos IEEE1451 .....	104

## Capítulo 4

Figura 4.1 - Estrutura TIM <i>Harvest</i> .....	109
Figura 4.2 - Organização dos dados transmitidos pelo CC2500 .....	110
Figura 4.3 - Gráfico $I(t)$ , para a recepção, processamento e transmissão de dados ..	112
Figura 4.4 - Gráfico $I(t)$ , para a configuração a) .....	113
Figura 4.5 - Diagrama de dados transacionados NCAP $\leftrightarrow$ TIM .....	114
Figura 4.6 - Gráfico $I(t)$ , para a configuração b) .....	115
Figura 4.7 - Gráfico $U(Lx)$ , para o painel fotovoltaico utilizado .....	119
Figura 4.8 - Gráfico $U(dia)$ , de descarga do condensador em vazio .....	120
Figura 4.9 - Gráfico $U(t)$ , para descarga do condensador .....	122
Figura 4.10 - Diagrama TIM <i>Harvest</i> .....	124
Figura 4.11 - <i>Software</i> da TIM <i>Harvest</i> .....	126
Figura 4.12 - <i>Handshake</i> .....	127
Figura 4.13 - Comando <i>Read data-set</i> .....	128

Figura 4.14 - Fluxograma do algoritmo de controle de energia .....	130
Figura 4.15 - Editor de TEDS de Energia .....	131

## Capítulo 5

Figura 5.1 - Fluxograma do programa TIM sem fios .....	136
Figura 5.2 - Montagem Experimental .....	137
Figura 5.3 - Aquisição de temperatura .....	138
Figura 5.4 - Conversão da Temperatura .....	138
Figura 5.5 - Medição da tensão da bateria .....	138
Figura 5.6 - Medição RSSI em dbm .....	139
Figura 5.7 - Gráfico caudal em função da frequência .....	140
Figura 5.8 - Diagrama de ligação TIM 1 .....	141
Figura 5.9 - a) Medição do tempo de carregamento .....	142
Figura 5.9 - b) Diagrama de ligação TIM 2 .....	142
Figura 5.10 - Gráfico nível de água em função do tempo de carregamento .....	143
Figura 5.11 - Fluxograma da medição de nível .....	144
Figura 5.12 - Configuração do aplicativo macro .....	146
Figura 5.13 - Diagrama de ligação TIM $\Leftrightarrow$ Inversor .....	147
Figura 5.14 - Canal Transdutor 4, atuador de bomba de água .....	148
Figura 5.15 - Programa de controlo da Montagem Experimental .....	149
Figura 5.16 - <i>Hardware TIM Harvest</i> .....	151
Figura 5.17 - Fluxograma aplicação TIM <i>harvest</i> .....	152
Figura 5.18 - Sistema de medição .....	153
Figura 5.19 - Gráfico de $U(t)$ , tensões do conversor em <i>cold start</i> .....	154
Figura 5.20 - Gráfico de $U(t)$ , tensões do conversor .....	154
Figura 5.21 - Gráfico de $U(t)$ , tempo de descarga na pior situação .....	156
Figura 5.22 - Gráfico $I(t)$ , TIM <i>Harvest</i> .....	156
Figura 5.23 - Gráfico $I(t)$ , cálculo da energia .....	156

## Capítulo 6

Sem Figuras

## **Apêndice E**

Figura 1 - Montagem experimental .....	174
--	-----

## **Apêndice F**

Figura 1 - TIM e NCAP .....	175
Figura 2 - TIM com interface série com fios .....	175
Figura 3 - Ponte sem fios .....	175
Figura 4 - Rede NCAP ⇔ TIM sem fios .....	175
Figura 5 - <i>Raspberry Pi</i> com a aplicação .....	175
Figura 6 - Interface TIM ⇔ Inversor SIEMENS .....	175
Figura 7 - <i>Hardware</i> de validação TIM IEEE1451 .....	176
Figura 8 - Medição de corrente TIM sem fios .....	176
Figura 9 - TIM energia <i>Harvest</i> .....	176
Figura 10 - Sensor nível de água .....	176
Figura 11 - TIM energia <i>Harvest</i> + Programador .....	176
Figura 12 - TIM <i>Harvest</i> + Sistema de aquisição de tensões .....	176

# Lista de Tabelas

## Capítulo 1

Sem Tabelas.

## Capítulo 2

Tabela 2.1 - Exemplo de Normas .....	11
Tabela 2.2 - Medição vs Técnica de medição .....	13
Tabela 2.3 - Rádio utilizados em redes de sensores sem fios .....	21
Tabela 2.4 - Densidade de Potência .....	23
Tabela 2.5 - Família IEEE 1451 .....	26
Tabela 2.6 - Estrutura de mensagem de comando .....	30
Tabela 2.7 - Estrutura de mensagem de resposta .....	31
Tabela 2.8 - Estrutura de mensagem do comando <i>Query TEDS</i> .....	31
Tabela 2.9 - Octeto <i>Atributos</i> .....	32
Tabela 2.10 - Estrutura de mensagem da resposta <i>Query TEDS</i> .....	33
Tabela 2.11 - Estrutura de Mensagem do Comando <i>Read TEDS</i> .....	33
Tabela 2.12 - Estrutura de Mensagem do comando <i>Read Data-Set</i> .....	33
Tabela 2.13 - Estrutura de Mensagem da Resposta <i>Read data-set</i> .....	34
Tabela 2.14 - Formato geral TEDS .....	34
Tabela 2.15 - Codificação <i>Globally Unique Identifier</i> .....	38
Tabela 2.16 - Exemplo Tipo 04 - <i>Globally Unique Identifier</i> .....	39
Tabela 2.17 - Exemplo Tipo 14 - <i>Control Group</i> .....	40
Tabela 2.18 - Exemplo Tipo 16 - <i>Geographic Location Groups</i> .....	41
Tabela 2.19 - Exemplo Tipo 17 - <i>Transducer Channels Proxies</i> .....	42
Tabela 2.20 - Exemplo final <i>Meta-TEDS</i> .....	42
Tabela 2.21 - Exemplo <i>Calibration key</i> .....	45
Tabela 2.22 - Exemplo <i>TransducerChannel Type</i> para um sensor .....	45
Tabela 2.23 - Codificação <i>Physical Units</i> .....	46
Tabela 2.24 - Exemplo do campo <i>Physical Units</i> para a aceleração, em SI .....	47
Tabela 2.25 - Exemplo do campo <i>Low Limit</i> .....	47

Tabela 2.26 - Exemplo do campo <i>HiLimit</i> .....	47
Tabela 2.27 - Exemplo do campo <i>Worst-case Uncertainty</i> .....	47
Tabela 2.28 - Exemplo do campo <i>SelfTest</i> .....	48
Tabela 2.29 - Exemplo do campo <i>Sample</i> .....	48
Tabela 2.30 - Exemplo do campo <i>Dataset</i> .....	49
Tabela 2.31 - Exemplo do campo <i>TransducerChannel Update Time</i> .....	50
Tabela 2.32 - Exemplo do campo <i>TransducerChannel Write Setup Time</i> .....	50
Tabela 2.33 - Exemplo do campo <i>TransducerChannel Read Setup Time</i> .....	50
Tabela 2.34 - Exemplo do campo <i>TransducerChannel Sampling Period</i> .....	50
Tabela 2.35 - Exemplo do campo <i>TransducerChannel Warm-up Time</i> .....	51
Tabela 2.36 - Exemplo do campo <i>TransducerChannel Read Delay Time</i> .....	51
Tabela 2.37 - Exemplo do campo <i>TransducerChannel Self-test Time Requirement</i> ..	51
Tabela 2.38 - Exemplo do campo <i>Attributes</i> .....	52
Tabela 2.39 - Exemplo do campo <i>End-of-data-set Operation Attribute</i> .....	53
Tabela 2.40 - Exemplo do campo <i>Edge-to-report Attribute</i> .....	54
Tabela 2.41 - Exemplo do campo <i>Actuator-halt Attribute</i> .....	55
Tabela 2.42 - <i>TransducerChannel TEDS</i> .....	56
Tabela 2.43 - <i>End users' application-specific TEDS</i> .....	57
Tabela 2.44 - <i>User's Transducer Name TEDS</i> .....	58
Tabela 2.45 - <i>PHY TEDS</i> .....	61

### Capítulo 3

Tabela 3.1 - Características do <i>Raspberry Pi 3</i> .....	72
Tabela 3.2 - Periféricos MSP-EXP430F5529 .....	74
Tabela 3.3 - Pinos de ligação série TIM ⇔ NCAP .....	76

### Capítulo 4

Tabela 4.1 - Lista de carregadores de energia <i>Harvest</i> .....	116
Tabela 4.2 - Comparação <i>Boost Charger</i> .....	117
Tabela 4.3 - Configuração AEM10941 .....	118
Tabela 4.4 - Características do painel fotovoltaico .....	119
Tabela 4.5 - Comparação Supercondensador/Bateria .....	119

Tabela 4.6 - Tensão vs tempo entre janelas .....	130
--	-----

## **Capítulo 5**

Tabela 5.1 - Valores de caudal em função da frequência .....	140
--	-----

Tabela 5.2 - Nível de água e tempo de carregamento .....	143
--	-----

Tabela 5.3 - Configurações da macro Cn006 .....	147
---	-----

## **Capítulo 6**

Tabela 6.1 - Estrutura de mensagem do comando <i>Read Available TEDS</i> .....	162
--	-----

Tabela 6.2 - Estrutura da mensagem da resposta <i>Read Available TEDS</i> .....	162
---	-----

Tabela 6.3 - Estrutura do comando <i>Read Available TEDS</i> .....	162
--	-----



# Nomenclatura

## Lista de Acrónimos

AJAX	-	<i>Asynchronous JavaScript and XML</i>
ADC	-	<i>Analog to Digital Converter</i>
API	-	<i>Application Programming Interface</i>
ASCII	-	<i>American Standard Code for Information Interchange</i>
ASK	-	<i>Amplitude-Shift Keying</i>
ASME	-	<i>The American Society of Mechanical Engineers</i>
BD	-	<i>Base de Datos</i>
BLE	-	<i>Bluetooth Low Energy</i>
CENELEC	-	<i>European Committee for Electrotechnical Standardization</i>
CoE	-	<i>Center of Expertise Consortium</i>
COM	-	<i>Communication Port</i>
CPU	-	<i>Central Processing Unit</i>
CSS	-	<i>Cascading Style Sheets</i>
DARPA	-	<i>Defense Advanced Research Projects Agency</i>
DSP	-	<i>Digital Signal Processor</i>
ECU	-	<i>Engine Control Unit</i>
EEPROM	-	<i>Electrically-Erasable Programmable Read-Only Memory</i>
ESR	-	<i>Equivalent Series Resistance</i>
FIFO	-	<i>First In, First Out</i>
FSK	-	<i>Frequency-Shift Keying</i>
GPIO	-	<i>General-Purpose Input/Output</i>
GPU	-	<i>Graphics Processing Unit</i>
GUI	-	<i>Graphical User Interface</i>
HBES	-	<i>Home and Building Electronic System</i>
HDMI	-	<i>High-Definition Multimedia Interface</i>
HTML	-	<i>HyperText Markup Language</i>
HTTP	-	<i>Hypertext Transfer Protocol</i>
I <sup>2</sup> C	-	<i>Inter-Integrated Circuit</i>
IEC	-	<i>International Electrotechnical Commission</i>
IEEE	-	<i>Institute of Electrical and Electronic Engineers</i>
IIoT	-	<i>Industrial Internet of Things</i>
IoT	-	<i>Internet of Things</i>

IP	-	<i>Internet Protocol</i>
ISO	-	<i>International Organization for Standardization</i>
JSON	-	<i>JavaScript Object Notation</i>
LCD	-	<i>Liquid Crystal Display</i>
LDO	-	<i>Low-dropout Regulator</i>
LED	-	<i>Light-Emitting Diode</i>
LGA	-	<i>Land Grid Array</i>
LLC	-	<i>Controle Lógico de Ligação</i>
LSB	-	<i>Least Significant Bit</i>
MAC	-	<i>Media Access Control Address</i>
MCU	-	<i>Micro-controller Unit</i>
MIPS	-	<i>Milhões de Instruções por Segundo</i>
MISO	-	<i>Master In - Slave Out</i>
MMC	-	<i>MultiMediaCard</i>
MOP	-	<i>Modo de Potenciômetro Motorizado</i>
MOSI	-	<i>Master Out - Slave In</i>
MPC	-	<i>Model Predictive Control</i>
MPPC	-	<i>Maximum Power Point Control</i>
MPPT	-	<i>Maximum Power Point Tracking</i>
MSB	-	<i>Most Significant Bit</i>
MSK	-	<i>Minimum Shift Keying</i>
NAWMS	-	<i>Nonintrusive Autonomous Water Monitoring System</i>
NCAP	-	<i>Network-Capable Application Processor</i>
NIST	-	<i>National Institute of Standards and Technology</i>
NMPC	-	<i>Nonlinear Model Predictive Control</i>
OSI	-	<i>Open System Interconnection</i>
PHP	-	<i>Hypertext Pre-processor</i>
PVDF	-	<i>Polyvinylidene Fluoride</i>
PWM	-	<i>Pulse-Width Modulation</i>
PZT	-	<i>Lead Zirconate Titanate — Pb Zi Ti</i>
RF	-	<i>Radio frequency</i>
RFID	-	<i>Radio Frequency Identification</i>
RSSI	-	<i>Received Signal Strength Indicator</i>
RxD	-	<i>Receive Data</i>
SD	-	<i>Secure Digital</i>
SDIO	-	<i>Secure Digital Input Output</i>
SDRAM	-	<i>Synchronous Dynamic Random Access Memory</i>

SDU	-	<i>Service Data Unit</i>
SH	-	<i>Shell Scripts</i>
SI	-	<i>International System of Units</i>
SoC	-	<i>System-on-a-Chip</i>
SPI	-	<i>Serial Peripheral Interface</i>
SQL	-	<i>Structured Query Language</i>
SSH	-	<i>Secure Shell</i>
TEDS	-	<i>Transducer Electronic Data Sheets</i>
TEG	-	<i>Thermoelectric Generation</i>
TII	-	<i>Transducer Independent Interface</i>
TIM	-	<i>Transducer Interface Module</i>
TLV	-	<i>Type/Length/Value</i>
TPMS	-	<i>Tire Pressure Monitoring System</i>
TxD	-	<i>Transmit Data</i>
UART	-	<i>Universal Asynchronous Receive Transmit</i>
USB	-	<i>Universal Serial Bus</i>
VPUP	-	Tensão de <i>pullup</i> Operacional Mínima
WI-FI	-	<i>Wireless Fidelity</i>
WSN	-	<i>Wireless Sensor Network</i>

### **Símbolos gregos**

Símbolo		Grandeza	Unidade
$\Delta t$	-	Intervalo de tempo	s
$\Delta T$	-	Variação de temperatura	K
$\tau$	-	Constante de tempo	s

### **Símbolos latinos**

Símbolo		Grandeza	Unidade
$a$	-	Aceleração	$\text{m s}^{-2}$
$U$	-	Tensão	V
$R$	-	Resistência	$\Omega$
$t$	-	Tempo	s
$P$	-	Potência	W
$x(k)$	-	Estado atual	

$u(k)$	-	Entrada de controle	
$E$	-	Energia	J
$C$	-	Capacidade elétrica	F ou C V <sup>-1</sup>
$I$	-	Intensidade da corrente elétrica	A
$Lux$	-	Iluminância	lx ou ln m <sup>-2</sup>
$Candela$	-	Intensidade Luminosa	Cd

# Capítulo 1

## Metodologia do Plano de Trabalho

### 1.1. Introdução

Todos os sistemas eletrônicos existentes são passíveis de melhorias. Quanta mais informação acerca do que se pretende controlar e quanto melhor se conseguir correlacionar a diversa informação recolhida, mais eficaz será o sistema. Se conseguíssemos controlar todas as variáveis do mundo, seríamos capazes de prever o futuro.

Segundo as fontes [1] e [2], desde os anos 70 do século passado, são implementados sistemas utilizando modelos de previsão linear, MPC (*Model Predictive Control*). Um sistema controlado por MPC é descrito por um modelo de sistema de tempo discreto linear (eq. 1.1).

$$x(k + 1) = A x(k) + B u(k), \quad x(0) = x_0, \quad (\text{eq. 1.1})$$

onde  $x(k)$  corresponde ao estado atual e  $u(k)$  à entrada de controle. Neste tipo de sistemas o sinal de saída depende sempre de um sinal de entrada de controle, realizado num sistema em malha fechada. A partir dos anos 90 houve uma evolução nos modelos teóricos de controle, e surgiram modelos de controle de previsão não linear, NMPC (*Nonlinear Model Predictive Control*). Estes sistemas são complexos e embora estejam bem formulados, do ponto de vista teórico, a sua implementação prática é consequentemente complexa.

Por exemplo, os modelos de previsão meteorológica têm a árdua tarefa de previsão da meteorologia, pois há inúmeras variáveis que ainda não temos a capacidade de medir em tempo real. Num sistema de condução autónoma se se controlasse a velocidade, a posição de todos os carros, os semáforos, condições atmosféricas, etc... em conjunto com algoritmos que permitissem gerir toda a informação em tempo real, seria possível prever e evitar situações de perigo [3]. Assim a densificação da aquisição, tratamento e correlação de variáveis tem um tremendo potencial para a sociedade (desde que empregue para o bem). Portanto, é necessário adquirir informação, e quanta mais se recolhe e analisa melhores resultados se poderão obter.

## **1.2. O problema em estudo e a sua relevância**

Desde o processo de aquisição da grandeza física, que se pretende medir, até ao seu tratamento e análise, existem atualmente inúmeras técnicas que o permitem fazer. Existem empresas que possuem os seus sistemas próprios, não normalizados, que dificultam a interligação com equipamentos de outras empresas. Neste sentido há a necessidade de encontrar uma norma que consiga lidar com toda a heterogeneidade de sensores, protocolos de comunicação e sistemas já implementados.

A aquisição da informação poderá ser feita com sensores em que a alimentação e a comunicação é realizada por fios ou então a comunicação poderá ser feita via rádio e a alimentação por baterias, o que se traduz num problema quer a nível da manutenção das baterias dos sensores, quer nos próprios resíduos criados.

Neste último caso há a necessidade de se estudar algoritmos que permitam a redução do “consumo” de energia por parte dos sensores sem fios, estudar formas de adquirir e armazenar energia por forma a prolongar o tempo de vida das baterias e, na melhor das soluções, eliminar por completo o uso de baterias.

Sistemas de aquisição de dados equipados com esta tipologia podem trazer elevados benefícios para a sociedade, desde a domótica à automação industrial, desde a produção agrícola à distribuição dos alimentos, em tudo é possível aumentar a eficiência dos sistemas. No entanto, estes sistemas precisam de ser profundamente estudados e otimizados antes de chegarem a um produto final. Assim, torna-se de grande importância o estudo e desenvolvimento de técnicas experimentais que permitam o avanço na redução do consumo energético dos sensores sem fios, na forma como a energia é adquirida e armazenada e na forma como os sensores lidam com a escassez de energia, sem descurar uma forma normalizada de o fazer.

## **1.3. Objetivos e contribuição da dissertação**

O trabalho conducente ao grau de doutor, e que se descreve de seguida neste documento de tese, tem como objetivo principal o estudo, a concetualização e a criação de mecanismos de aquisição, de armazenamento de energia e de algoritmos de economia de energia que permitam a uma rede de sensores sem fios operarem de forma autónoma e fiável, respeitando uma norma, IEEE1451, que permita a interoperabilidade com outros sistemas já existentes. Terá ainda o objetivo do estudo da norma que melhor se adequará a estes requisitos.

Como resultado desse estudo pretende-se alcançar uma metodologia que permita a rápida prototipagem de sensores inteligentes, e que permita definir um modo de operação compatível com a norma IEEE1451. A metodologia deve ter sempre presente que a operação do sensor ocorre sempre em condições de restrições energéticas, seja porque o

sensor é alimentado por uma bateria, ou porque obtém a energia de que necessita para trabalhar de um qualquer mecanismo de recolha de energia (Energia “*Harvest*”). Um enquadramento adequado para o trabalho é possível ser alcançado com a realização de uma análise crítica das temáticas associadas a dispositivos de baixo consumo energético, tais como: dispositivos *low power*, redes de sensores *low power*; *Energy Harvest*; IoT (*Internet of Things*); IIoT (*Industrial Internet of Things*); *Cyber-Physical Systems*; e Indústria 4.0.

Dada a diversidade de elementos que são passíveis de integrar uma rede de sensores, que materialize os conceitos descritos no parágrafo anterior, a adoção de normas torna-se uma mais-valia. A adoção da família de normas IEEE1451 suporta, por exemplo, a construção de uma fábrica inteligente ou a aplicação do conceito do IoT em contextos com a *smart cities*, *smart water*, *smart grid*, ou *smart transportation*.

Um estudo aprofundado da norma IEEE1451 permite identificar a natureza e as características das ferramentas de trabalho que é necessário desenvolver e construir. Até ao dia de hoje, ainda não se encontravam disponíveis as ferramentas capazes de suportar o desenvolvimento, teste e validação de sensores inteligentes, essenciais para alcançar a metodologia de desenvolvimento que se pretende. Para que o objetivo geral seja concretizável é necessário desenvolver e construir as seguintes ferramentas:

- plataforma de desenvolvimento composta por ferramentas de *software* e *hardware* que suportem a implementação de sistemas que respeitem, na íntegra, a norma IEEE1451.
- plataforma de teste e validação capaz de suportar os elementos comumente encontrados numa solução industrial.
- instalação piloto composta por uma rede de transdutores, que possuam restrições energéticas, demonstrativa de uma aplicação funcional que implementa a norma IEEE1451.

Com o desenvolvimento destas ferramentas de trabalho é possível alcançar os seguintes objetivos particulares:

- desenvolver uma metodologia de análise que permita quantificar a energia mínima necessária para que um sensor inteligente possa operar segundo a norma IEEE1451, tornando viável a coexistência deste dispositivo na rede de sensores inteligentes.
- sistematizar o desenvolvimento de sensores inteligentes, capazes de operar segundo a norma IEEE1451, e que apresentem restrições energéticas por a sua energia ser proveniente de energia recolhida do ambiente onde operam.

- estudar cenários de operação, onde o consumo de energia de um sensor inteligente possa ser avaliado. Uma análise dos resultados obtidos permite ajustar os mecanismos de recolha de energia ao modo de operação pretendido.
- A validação de um sistema a partir de um sistema experimental com medições reais de energia “produzida”, energia “consumida”. O estudo aprofundado, de *hardware* e *software*, de um sensor sem fios alimentado por energia *Harvest*.

Do trabalho desenvolvido deverá ser possível comprovar que, partindo de uma metodologia de análise de mecanismos de recolha de energia, será possível dimensionar um circuito eletrónico que forneça a energia necessária para o funcionamento do sensor inteligente. Ao mesmo tempo, o sensor inteligente irá adotar modos de funcionamento que permitam a sua coexistência dentro da rede de sensores IEEE1451.

É possível concluir-se que a realização destes dispositivos, por teste e validação em instalação piloto, contribuirão para o Grupo Trabalho IEEE P21451.002, para a norma de novas TEDS (*Transducer Electronic Data Sheets*) e métodos de operação que tornem possível ao sensor inteligente operar em circunstâncias onde se observem restrições energéticas.

Do apresentado, deverá poder-se concluir que a norma IEEE1451 receberá um forte contributo deste trabalho. O impacto será visível em aplicações ligadas à IoT (Internet das Coisas); IIoT (Internet das Coisas na Indústria); *Cyber-Physical Systems*; e Indústria 4.0.

#### **1.4. Visão geral e organização**

O trabalho de tese está organizado em 6 capítulos. No capítulo 1, dando uma introdução ao tema do estudo e termina no capítulo 6 com as conclusões. No final de cada capítulo são apresentadas as referências bibliográficas referentes a esse mesmo capítulo.

No final da tese, encontram-se a secção dos apêndices, que foram colocados no final por apresentarem um formato diferente da restante tese. Todas as referências cruzadas apresentadas ao longo desta tese possuem atalhos que permitem uma rápida acessibilidade aos diferentes capítulos, subcapítulos, imagens e tabelas, a combinação de teclas “Alt+seta esquerda” permite regressar à página de origem.

As imagens apresentadas, com a exceção das que se encontram devidamente referenciadas, são originais e elaboradas com ferramentas de código aberto: *Inkscape* e *Gimp*.

O estudo aqui apresentado requereu obrigatoriamente a escrita de programas informáticos. O código desses programas encontra-se disponível em [github.com](https://github.com) sob a licença de código aberto. Os programas foram escritos, na sua grande maioria, com linguagens de código aberto.

O Capítulo 2, Estado da Arte, faz um levantamento das problemáticas em relação aos sensores sem fios. Apresenta uma definição de sensor, diferentes tipos de sensores, caracterização e toda a problemática à volta dos sensores. Caracteriza a diferença entre sensor e sensor inteligente.

Apresenta uma breve história da norma IEEE1451, quando e porque surgiu. Descreve as diferentes normas que fazem parte da IEEE1451.

Neste capítulo, são apresentados os módulos internos constituintes de uma NCAP e de uma TIM, bem como uma descrição das diferentes interfaces de comunicação entre as duas. Apresenta o módulo de comunicação IEEE1451.2, bem como a descrição da interface de comunicação TII (*Transducer Independent Interface*).

Faz um estudo em pormenor em relação à norma IEE1451.0 e IEEE1451.1, apresentando a estrutura da norma e a forma como trocam mensagens. É explicitado o conceito de TEDS e como se encontra organizado, apresentando exemplos das TEDS mandatárias.

É ainda apresentada a organização teórica de funcionamento de uma TIM e de uma NCAP segundo a norma.

É feito um levantamento das diferentes técnicas de comunicação sem fios: protocolos, modulações e frequências mais utilizados.

Foi feito um estudo de diferentes fontes de energia, proporcionando a escolha de uma delas para a implementação de uma TIM com restrições energéticas, o estudo de diferentes formas de armazenar energia, bem como a pesquisa de soluções já existentes para este tipo de sistemas.

O Capítulo 3, Plataforma de Desenvolvimento para a Norma IEEE1451, descreve o estudo e a implementação de ferramentas de desenvolvimento para a criação completa de um sistema de sensores e atuadores, a TIM, NCAP e aplicação respeitando a norma IEEE1451. Neste capítulo é apresentada uma panóplia de ferramentas disponíveis online e de acesso livre, desde a construção do sensor até à sua validação.

É explicitada a forma como se lidou com a idealização e a concretização da infraestrutura de base que permitiu o estudo do desenvolvimento das ferramentas: desde a configuração do servidor à criação das ferramentas de desenvolvimento e à criação das diferentes TIM (com e sem fios).

Neste capítulo são ainda descritos os tutoriais que foram escritos para melhor entendimento da norma e para a utilização das ferramentas.

O Capítulo 4, Análise Energética de Sensores Inteligentes, descreve a análise realizada acerca da energia necessária para que uma TIM sem fios funcione bem como a análise da melhor forma de adquirir a energia a partir de fontes e energia *Harvest*. Descreve o estudo realizado relativamente à quantificação da energia necessária para a TIM desenvolvida para os diferentes regimes de funcionamento, mais concretamente em relação à energia

necessária para o MCU (*Micro-controller Unit*), nos diferentes modos de energia e em relação ao rádio nos modos de TxD (*Transmit Data*), RxD (*Receive Data*) e em modo adormecido. É apresentado o estudo para a escolha da forma de recolha de energia *Harvest*, no estudo escolhido e o estudo da forma de como armazenar essa energia.

É descrito o desenvolvimento de uma TIM alimentada com energia *Harvest*, onde se apresenta o diagrama do circuito, a descrição do *software* desenvolvido e em pormenor o algoritmo que permite gerenciar a energia disponível. Neste sentido, é ainda proposta uma TEDS de energia, Apêndice G, que está de acordo com o algoritmo de gestão de energia descrito em 4.3.3.

O Capítulo 5, Teste e Validação de Sensores Inteligentes, descreve o funcionamento de uma implementação experimental como forma de validação de três sensores (TIM) sem fios: caudalímetro, nível de água e controle da bomba de água. A implementação dos sensores foi desenvolvida com a plataforma de desenvolvimento descrita em 3.6 e validada com a aplicação descrita em 5.3.4. Além dos sensores é descrita a aplicação que monitoriza os três sensores.

Neste capítulo são ainda apresentados os resultados obtidos experimentalmente para as energias necessárias para o sensor *Harvest*, descritas em 5.4.2.

No Capítulo 6, Conclusão, são apresentadas as conclusões gerais de todos os estudos realizados neste trabalho de tese. São apresentadas algumas propostas de adições à norma, que considero uma mais-valia para futuras revisões à norma, pois proporcionaria uma melhor compatibilidade com sensores alimentados a partir de energias *Harvest*.

São ainda indicadas algumas propostas para futuros trabalhos tendo por base o desenvolvido aqui.

É apresentada uma lista com os artigos publicados, bem como as participações em conferências científicas.

## 1.5. Referências Bibliográficas

- [1] Manfred Morari, Jay H. Lee, “Model predictive control: past, present and future”, ELSEVIER - Computers and Chemical Engineering 23, pag. 667 – 682, 1999. [Online], Disponível: [https://www.itk.ntnu.no/fag/fordypning/TK16-filer/Samling1\\_MorariLee.pdf](https://www.itk.ntnu.no/fag/fordypning/TK16-filer/Samling1_MorariLee.pdf)
  
- [2] R. Findeisen, F. Allgöwer, “An Introduction to Nonlinear Model Predictive Control”, Engineering, Mathematics, 2002. [Online], Disponível: <https://api.semanticscholar.org/CorpusID:59790631>
  
- [3] Y. Liu and H. Wu, "Prediction of Road Traffic Congestion Based on Random Forest," 2017 10th International Symposium on Computational Intelligence and Design (ISCID), 2017, pp. 361-364, doi: 10.1109/ISCID.2017.216.



# Capítulo 2

## Estado da Arte

### 2.1. Introdução

Atualmente existe a crescente necessidade de monitorizar parâmetros físicos, químicos e biológicos. É cada vez mais importante, para garantir a eficácia dos sistemas, recolher informação, medindo cada vez mais parâmetros.

O grande desafio da atualidade surge no desenvolvimento de sistemas onde a densidade de informação é grande, onde se pretende diminuir a dependência energética dos nodos e onde não é possível ou desejável a ligação cablada.

Os estudos que a seguir se apresentam, discutem a implementação de sistemas de monitorização nas mais diversas áreas do IoT. Nestes, são abordados os obstáculos encontrados no decorrer da sua implementação, dos quais se destacam a fonte de energia, a normalização e a interoperabilidade com outros sistemas.

Em “*A framework for wireless sensor networks management for precision viticulture and agriculture based on IEEE1451 standard*” [1] é demonstrada uma rede de sensores sem fios aplicada à agricultura, mais propriamente a uma produção vinícola. Segundo os autores, a agricultura de precisão depende da aquisição, transmissão e processamento de uma vasta aquisição de dados provenientes de uma rede de sensores com uma grande heterogeneidade. Afirmam que, sistemas com estas características apresentam duas barreiras. A primeira, relacionada com a energia, levanta o desafio de como obter energia suficiente e gerenciá-la de modo que todos os dispositivos possam funcionar de uma forma ininterrupta. A segunda tem a ver com a normalização e interoperabilidade relacionadas à multiplicidade de sensores, tipos de saída de sensor, plataformas de aquisição de dados de vários fabricantes e suporte de rede.

Em [2] “*An interoperable multi-sensor system for healthcare*” os autores referem que a população idosa está a aumentar, principalmente nos países ditos desenvolvidos, aumentando conseqüentemente as doenças associadas à idade. Existem, no mercado, sistemas de monitorização de saúde em casa. Dispositivos esses que estão em constante evolução, tornando-se menos invasivos para os pacientes e energeticamente mais eficientes. Afirmam que projetar um sistema de saúde vinculado a dispositivos fixos não permitiria atualizações quando surgem avanços tecnológicos e, como tal, a interoperabilidade (que é a capacidade de trabalhar com qualquer conjunto de sensores)

deve ser considerada como um requisito chave pelos sistemas de monitorização de saúde. Para isso, há a necessidade de fornecer uma interface de sensor normalizada que permitam a ligação de qualquer sensor à rede.

Na indústria automóvel são utilizadas redes de sensores sem fios. O TPMS (*Tire Pressure Monitoring System*) é um dispositivo que se coloca dentro do pneu e envia informação da pressão do pneu, rotação e temperatura da roda e avisa o ECU (*Engine Control Unit*) de bateria “fraca” [3]. O objetivo deste sistema é adicionar um elemento de segurança aos condutores, alertando-os para uma possível baixa de pressão de um pneu, evitando acidentes rodoviários. Embora este tipo de sistemas já exista há algum tempo continuam a ser alimentados por baterias, aumentando a necessidade de manutenção.

Neste sistema a informação é comunicada ao ECU via radio. Na Europa a frequência utilizada é (normalmente) 433 MHz e modulação ASK (*Amplitude-Shift Keying*) ou FSK (*Frequency-Shift Keying*). A estrutura do pacote de dados enviado pode depender das marcas e modelos dos carros, mas geralmente apresenta o formato seguinte:

Preamble	SensorID	Pressure	Temperature	Flags	Checksum
----------	----------	----------	-------------	-------	----------

Figura 2.1 - Pacote de dados do sensor TPMS

O *SensorID* é um número de 28 a 32 bits, único para cada sensor e registado no ECU. Os campos *Pressure* e *Temperature* contêm os valores das leituras do dispositivo para a pressão do pneu e temperatura da roda respetivamente. O campo *Flags* pode conter informações extra, como o aviso de bateria do sensor “fraca”.

Neste tipo de sensores a alimentação realizada a partir de baterias e a transmissão de dados não é normalizada. No artigo “*Battery-and wireless tire pressure measurement systems (TPMS) sensor*” [3] é demonstrado o desenvolvimento de um sensor TPMS, sensor alimentado a partir de energia *Harvest* utilizando um elemento piezoelétrico.

São inúmeros os exemplos onde podemos encontrar estes constrangimentos. Para os satisfazer é necessário encontrar uma norma e uma fonte de energia *Harvest*, fonte essa capaz de fornecer energia superior ao balanço entre a energia necessária e a disponível.

As normas são documentos com regras publicadas por organizações, tais como: ASME (*The American Society of Mechanical Engineers*), IEC (*International Electrotechnical Commission*), IEEE (Instituto de Engenheiros Eletrotécnicos e Eletrónicos), ISO (*International Organization for Standardization*), CENELEC (*European Committee for Electrotechnical Standardization*) entre outros, que estabelecem diretrizes para as diversas atividades, processos e dispositivos [4].

A seguir apresentam-se diferentes normas, relacionadas com o tema e das quais serviram como ponto de partida a escolha de uma que mais se adequa à situação em estudo.

Tabela 2.1 - Exemplo de Normas

<b>Norma</b>	<b>Descrição</b>
ISO/IEC 14543-3-10	Especifica um protocolo para dispositivos sem fios de baixa potência, como dispositivos de recolha de energia em um ambiente doméstico [5].
CENELEC EN50090	Norma europeia para aplicações de controle para o Sistema de Comunicação Aberta HBES ( <i>Home and Building Electronic System</i> ) residencial e de prédios que abrange qualquer combinação de dispositivos eletrônicos ligados por meio de uma rede de transmissão digital [6].
CEN EN 13321-1 e 13321-2	Norma que especifica, os sistemas eletrônicos domésticos ou em prédios (HBES) para o domínio de automação e aplicação de sistemas de controle e gestão (BACS) [6].
ANSI/ASHRAE AE 135	BACnet, é um protocolo de rede de controle e automação em prédios da ASHRAE, que foi projetado especificamente para atender às necessidades de comunicação de sistemas de automação e controle de prédios para aplicações como controle de aquecimento, ventilação e ar condicionado [8].
IEC 61449	Define a arquitetura para sistemas distribuídos, na automação industrial. [9]
IEEE 488	Norma de comunicação digital com barramento paralelo de 8-bits, utilizado para ligar instrumentos de teste em rede, podendo suportar até 15 dispositivos. [10]
IEEE 2888	A norma IEEE 2888 está em desenvolvimento e pretende fornecer uma plataforma comum para <i>Digital Twin Space</i> ou <i>Metaverse Plus</i> [11].
IEC 62832-1:2020	Define os princípios gerais do <i>Digital Factory framework</i> , que é um conjunto de elementos de modelo de referência e regras para modelagem de sistemas de produção. [12]
IEEE 802	As normas IEEE 802 são referentes às redes locais e redes urbanas, e dentro desta família existem algumas sub-normas conhecidas: [13]
IEEE 802.1	Norma que especifica a relação entre os padrões IEEE e a sua interação com os modelos OSI ( <i>Open System Interconnection</i> ).
IEEE 802.2	LLC (Controle Lógico de Ligação).
IEEE 802.3	Norma que especifica as camadas de ligação de dados do modelo OSI para as interligações de redes locais.
IEEE 802.11	Norma para redes sem fios.
IEEE1451	Norma que define um conjunto de regras para sensores e atuadores. Detalhadamente descrita em 2.8

Para satisfazer a normalização e interoperabilidade, o estudo é realizado em torno da família de normas IEEE1451, visto que é a norma que consegue cumprir estes requisitos,

tendo apenas em falta considerações energéticas as quais foram estudadas e apresentadas como uma possível adição numa futura revisão.

De acordo com [14] e [15] as grandes vantagens da norma IEEE1451 são:

- Tempo de configuração mais rápido e, portanto, redução dos custos de fabrico.
- Melhora a integridade dos dados eliminando erros de configuração devido à seleção do sensor incorreto ou à inserção de valores de calibração incorretos.
- A existência de *Transducer Electronic Data Sheets* - TEDS que permite o reconhecimento e identificação das características do dispositivo.
- Quanto mais canais um teste tiver, maior será a vantagem que o TEDS oferece.
- Permite a manutenção de registos de calibração em conformidade com ISO 9001 e QS 9000 (gestão de qualidade)
- Permite a interoperação de transdutores e sistemas de controle de diferentes fabricantes.
- Permite o uso de instalações de sistema de controle já existentes.
- Permite que os transdutores compartilhem um barramento comum, reduzindo o custo e a complexidade da implementação.
- Aumenta o uso das redes existentes.

## 2.2. Sensores

Um sensor pode ser definido como um dispositivo que recebe e responde a um sinal ou um estímulo. Considera-se um estímulo como uma quantidade, propriedade ou condição que é recebida do meio exterior e é convertida num sinal elétrico.

Os sensores, fazem parte integrante de qualquer equipamento eletrônico moderno e é com eles que é possível tornar sistemas mais eficientes.

Os sensores são dispositivos que dão informação de entrada ao sistema acerca do que se pretende controlar. Sistemas que poderão ser tanto mais complexos quanto o maior número de sensores que possui.

Existem diferentes sensores, com diferentes características, diferentes modos de funcionamento, mas todos carecem de calibração, todos sofrem dos mesmos problemas de, por exemplo, histerese e saturação.

### 2.2.1. Tipos de sensores

Os sensores podem ser de dois tipos: passivos e ativos. Um sensor passivo não necessita de uma fonte de energia, ele próprio produz um sinal elétrico em resposta a um estímulo exterior. Por exemplo: termopares, fotodíodos e piezoelétricos são sensores passivos.

Os sensores ativos necessitam de uma fonte de energia externa para funcionarem, a qual se dá o nome de sinal de estímulo. Para produzirem o sinal de saída, este tipo de sensores, alteram este sinal de entrada de acordo com a alteração física que pretendem medir.

A Tabela 2.2 apresenta, para cada tipo de medição, a(s) técnica(s) que se utilizam para a realizar [16].

Tabela 2.2 - Medição vs Técnica de medição

<b>Medição</b>	<b>Tipos de sensores/transdutores</b>
Deslocamento/posição	Efeito de Hall, Optoelectrónico
Temperatura	Termístor, transistor base-emitter voltage
Pressão	Piezoresistivo, capacitivo
Velocidade (linear/angular)	Efeito de Hall, optoelétrico
Aceleração	Piezoresistivo, capacitivo, piezoelétrico
Força	Piezoresistivo
Resistência Elétrica de materiais	Piezoresistivo
Fluxo	Varição de pressão
Humidade	Resistivo, capacitivo
Distância e posição	Radar/ sonar

### 2.2.2. Precisão e Exatidão de Sensores e sua Problemática

Existe, muitas vezes, a confusão entre os termos: precisão e exatidão. Segundo a fonte [18] “...exatidão fornece informações sobre o quão próximo o valor medido está em relação ao valor verdadeiro, enquanto a precisão indica a qualidade da medição, sem garantir que a medição esteja correta.”

A exatidão não é um valor calculável, é apenas uma forma de avaliar a qualidade de uma medição, faz a comparação entre o valor medido e o valor real.

A precisão de uma medição é uma característica importante de um sensor, está relacionada com a proximidade entre os valores obtidos em medições sucessivas (a fidelidade da medição). Esta pode ser calculada em termos de erro absoluto e erro relativo. Erro absoluto de medição é definido como a diferença entre o valor medido e o valor tabelado.

$$\text{Erro Absoluto} = |\text{Valor Medido} - \text{Valor Tabelado}| \quad (\text{eq. 2.1})$$

Onde o *Valor Medido* é a média dos valores de todas as medições efetuadas e o *Valor Tabelado* é aquele que é considerado como verdadeiro e que se encontra tabelado.

$$\text{Erro Relativo} = \frac{|\text{Valor Medido} - \text{Valor Tabelado}|}{\text{Valor Tabelado}} \quad (\text{eq. 2.2})$$

A Figura 2.2, apresenta uma distinção clara da diferença entre estes dois conceitos [18].

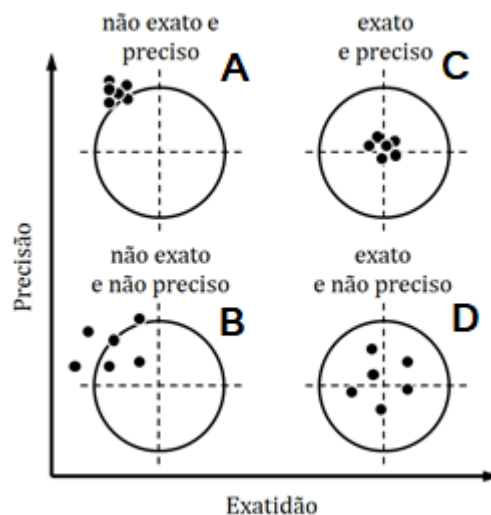


Figura 2.2 - Precisão vs Exatidão

Um conjunto de medições poderão ser exatas, se os seus valores estão muito próximos uns dos outros, mas poderão não ser precisos se estiverem afastados do valor tabelado. Na

situação A, temos um valor preciso, pois os valores obtidos são todos muito próximos, mas não é um valor exato, pois está afastado do valor considerado como real. Na situação B, os valores não são exatos, pois estão afastados do valor considerado como real, não são precisos porque estão afastados uns dos outros. Na situação C, os valores são exatos e precisos, pois estão próximos uns dos outros e próximos do valor real. Contudo, na situação C, os valores são pouco precisos pois estão afastados uns dos outros.

A precisão de um sensor pode ser influenciada por diversos fatores como: histerese, linearidade, saturação, calibração, erros repetitivos entre outros.

Existem muitos sensores disponíveis, mas para obter o melhor sensor possível com o valor ideal de precisão, é necessário que o sensor seja calibrado no dispositivo onde será implementado [19]. A calibração do sensor é realizada a partir de uma curva de calibração. No Capítulo 5.3.1 realiza-se a calibração de um sensor de caudal e no Capítulo 5.3.2, realiza-se a calibração de um sensor de nível de água.

A histerese é um fenómeno ou característica comum que é causada pela alteração das propriedades de um material. O erro de histerese é a diferença entre dois valores de saída que correspondem à mesma entrada dependendo da direção seguida pelo sensor, conforme mostrado na Figura 2.3 [19].

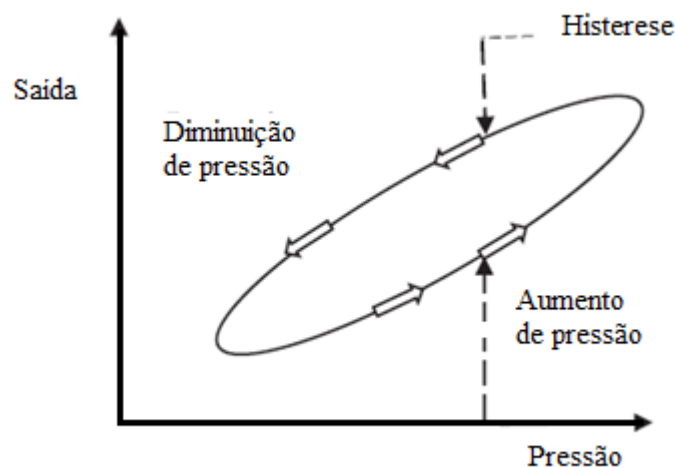


Figura 2.3 - Exemplo de histerese de um sensor de pressão

A Figura 2.3 é o exemplo de um sensor de pressão onde se mostra a diferença entre os valores de saída, na mesma posição do sensor, com a diferença se este está a ser pressionado ou largado. Idealmente o valor deveria ser igual, mas tal não se verifica. A esta diferença de valores dá-se o nome de histerese.

A resolução de um sensor é definida como a mínima alteração do sinal detetável durante a leitura ou medição de alguma quantidade utilizando um sensor adequado. Esta também pode ser definida como a capacidade da medição para obter pequenas alterações na característica do resultado da medição [19].

Cada sensor possui um limite máximo de operação, que é o estado em que o sinal de saída do sensor deixará de responder apesar de aumentar os valores dos estímulos de entrada. Esta característica é referida como saturação do sensor. Há um ponto em que a saída do sensor não responde conforme necessário a um aumento nos estímulos de entrada, e esse ponto específico é conhecido como ponto de saturação ou ponto limite do sensor [19].

### **2.2.3. Sensores Utilizados neste Estudo**

Para a medição da temperatura foram utilizados o sensor TMP36 e o sensor DS18B20 que são sensores de temperatura de precisão.

O sensor TMP36 é um sensor alimentado com uma tensão entre 2,7 V e 5,5V. O sinal de saída analógica é diretamente proporcional à temperatura em graus Celcius, variando 10 mV por cada grau Celcius. O sensor TMP36 não necessita qualquer calibração externa para disponibilizar uma precisão de  $\pm 1^{\circ}\text{C}$  a  $+25^{\circ}\text{C}$  e de  $\pm 2^{\circ}\text{C}$  na gama de valor de  $-40^{\circ}\text{C}$  até  $+125^{\circ}\text{C}$ . Aliado com a baixa impedância de saída, facilita a interface do ADC (*Analogic to Digital Converter*) e a programação do microcontrolador.

A corrente de alimentação é inferior a 50  $\mu\text{A}$  em modo ativo, em modo de *shutdown* a corrente de alimentação é inferior a 0,5  $\mu\text{A}$ .

Existem disponíveis três encapsulamentos: 3-pinos TO-92 (o utilizado neste estudo), 8-pinos SOIC\_N e 5-pinos SOT-23 [20].

O sensor DS18B20 é um termómetro digital que fornece medições de temperatura em Celsius com uma resolução configurável de 9 bits a 12 bits. Possui uma função de alarme com pontos de disparo superior e inferior, programáveis pelo utilizador e guardados em EEPROM (*Electrically-Erasable Programmable Read-Only Memory*) não volátil. Mede valores de temperaturas na gama de valores de  $-55^{\circ}\text{C}$  a  $+125^{\circ}\text{C}$ , com uma precisão de  $\pm 0,5^{\circ}\text{C}$  para valores entre  $-10^{\circ}\text{C}$  a  $+85^{\circ}\text{C}$ . O DS18B20 utiliza a interface de comunicação 1-Wire. Este sensor possui a vantagem de não necessitar utilizar um ADC, libertando recursos do microcontrolador. No entanto requer a implementação, no microcontrolador, do protocolo 1-Wire.

A corrente de alimentação é de a 1 mA em modo ativo e de 0,75  $\mu\text{A}$  em modo de *standby*.

Existem disponíveis três encapsulamentos: 3-pinos TO-92 (o utilizado), 8-pinos SO e 8-pinos  $\mu\text{SOP}$  [21].

O sensor CMA3000-A01 é um acelerómetro de saída analógica de 3 eixos. É um sensor alimentado com uma tensão entre 1,7 V e 3,6V com gama de valores programáveis de  $\pm 2g$  ou  $\pm 8g$ .

A corrente de alimentação é de a 180  $\mu A$  em modo ativo e de 0,1  $\mu A$  em modo de *Power Down* [22].

O sensor LIS3DH é um acelerómetro linear de três eixos de ultra-baixa potência, com interface digital I<sup>2</sup>C/SPI (*Inter-Integrated Circuit*)/(*Serial Peripheral Interface*) e é alimentado com uma tensão entre 1,71 V e 3,6V. O dispositivo possui modos operacionais de energia ultrabaixa, até 2  $\mu A$ , que permitem economia de energia avançada e funções integradas inteligentes. O LIS3DH possui escalas completas dinamicamente selecionáveis pelo utilizador de  $\pm 2g/\pm 4g/\pm 8g/\pm 16g$  e é capaz de medir acelerações com taxas de dados de saída de 1 Hz a 5,3 kHz. O dispositivo pode ser configurado para gerar sinais de interrupção usando dois eventos inerciais independentes de despertar/queda livre, bem como pela posição do próprio dispositivo. Os limites e o tempo dos geradores de interrupção são programáveis pelo utilizador final em tempo real. O LIS3DH possui um *buffer* integrado FIFO (*First In First Out*) de 32 níveis 16-bits, permitindo que o utilizador armazene dados minimizando a intervenção do microcontrolador. O LIS3DH está disponível num encapsulamento de plástico LGA (*Land Grid Array*) e funciona numa faixa de temperaturas entre -40 °C a +85 °C [23].

Relativamente à medição do caudal de água, existem diversas técnicas para a sua medição. Não querendo fazer um estudo exaustivo, indico apenas as técnicas mais utilizadas [24]:

- Caudalímetros eletromagnéticos;
- Caudalímetros ultrassónicos;
- Caudalímetros deprimogéneos.

A escolha do melhor sensor depende sempre da aplicação onde se pretende utilizar. Neste estudo foi escolhido o sensor YF-S201, que é um sensor de fluxo de água, caudalímetro, que consiste num corpo de válvula de plástico, um rotor de água e um sensor de efeito *hall*. Quando a água passa através do rotor, este gira. A velocidade do rotor é alterada conforme a taxa de fluxo do fluido. O sensor de efeito *hall* produz um sinal de onda quadrada cuja frequência é proporcional ao fluxo de água. De acordo com a folha de características [25] pode ser alimentado com uma tensão entre 4,5V a 24V, consumindo uma corrente de 15 mA (quando alimentado com uma tensão contínua de 5V). Este suporta uma temperatura máxima da água até 120 graus Celcius e uma pressão até 1,75 MPa.

O sensor BH1750 é um circuito integrado sensor de luz ambiente digital com interface de barramento I<sup>2</sup>C. Possui uma resolução de 16 bits e uma gama de leituras entre 1 e 65535 lx [52].

### 2.3. Sensores Inteligentes

Um sensor ou transdutor inteligente é a integração de um sensor analógico ou um atuador, uma unidade de processamento e uma interface de comunicação [26].

Um transdutor compreende um dispositivo de *hardware* ou *software* como uma unidade pequena e compacta, contendo um elemento sensor/atuador, um microcontrolador, um controlador de comunicação e *software* para condicionamento de sinais, calibração, diagnóstico e comunicação. A norma IEEE1451 define um transdutor inteligente como sendo um sensor e/ou atuador que pode identificar-se e descrever-se a si mesmo. Este possui capacidade de processamento de dados (inteligente) para apresentar dados do sensor ou aceitar valores de atuação, respetivamente, nas unidades de medição. Possui ainda capacidade de comunicação em rede, e é fácil de usar (*plug-and-play*).

A figura seguinte apresenta um modelo de transdutor inteligente baseado na norma IEEE1451 (sensor e atuador). Os transdutores inteligentes IEEE1451 possuem recursos que incluem auto-identificação, auto-descrição, reconhecimento de local, processamento e junção de dados, formatos de dados e protocolos de comunicação [26].

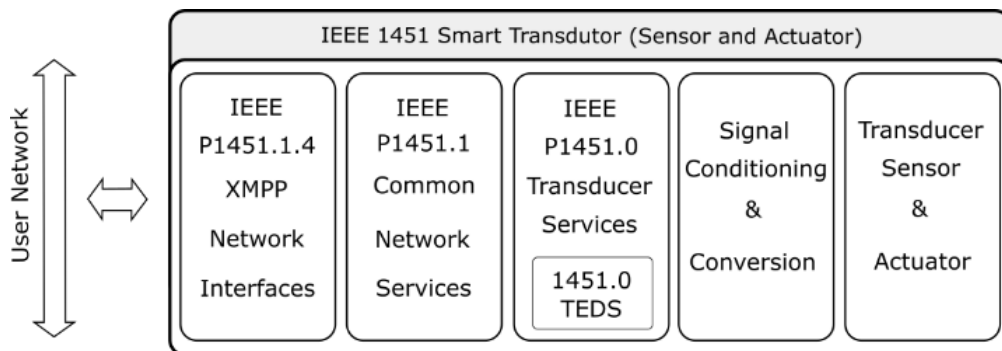


Figura 2.4 - Estrutura sensor inteligente IEEE1451

A Figura 2.5 apresenta o diagrama bloco de um sensor inteligente sem fios. O sensor inteligente sem fios consiste tipicamente em quatro subunidades [27]: elemento sensor; circuito de aquisição; microcontrolador e comunicação rádio.

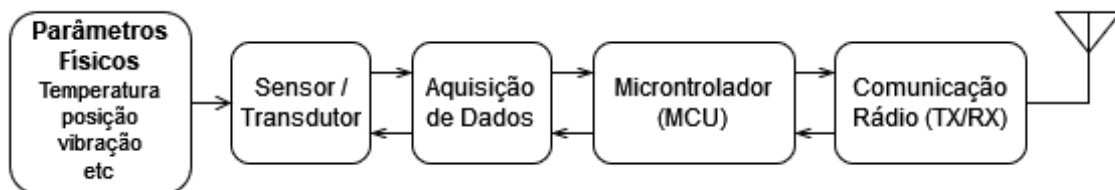


Figura 2.5 - Estrutura sensor sem fios

A Figura 2.6, apresenta a estrutura de um sensor sem fios alimentado por energia *harvest* [28]. Este tipo de sensor inteligente possui uma unidade que permite a recolha e armazenamento de energia. Essa energia poderá ser obtida de diferentes formas para depois ser convertida e armazenada. Este processo é controlado por uma unidade de gestão de energia.

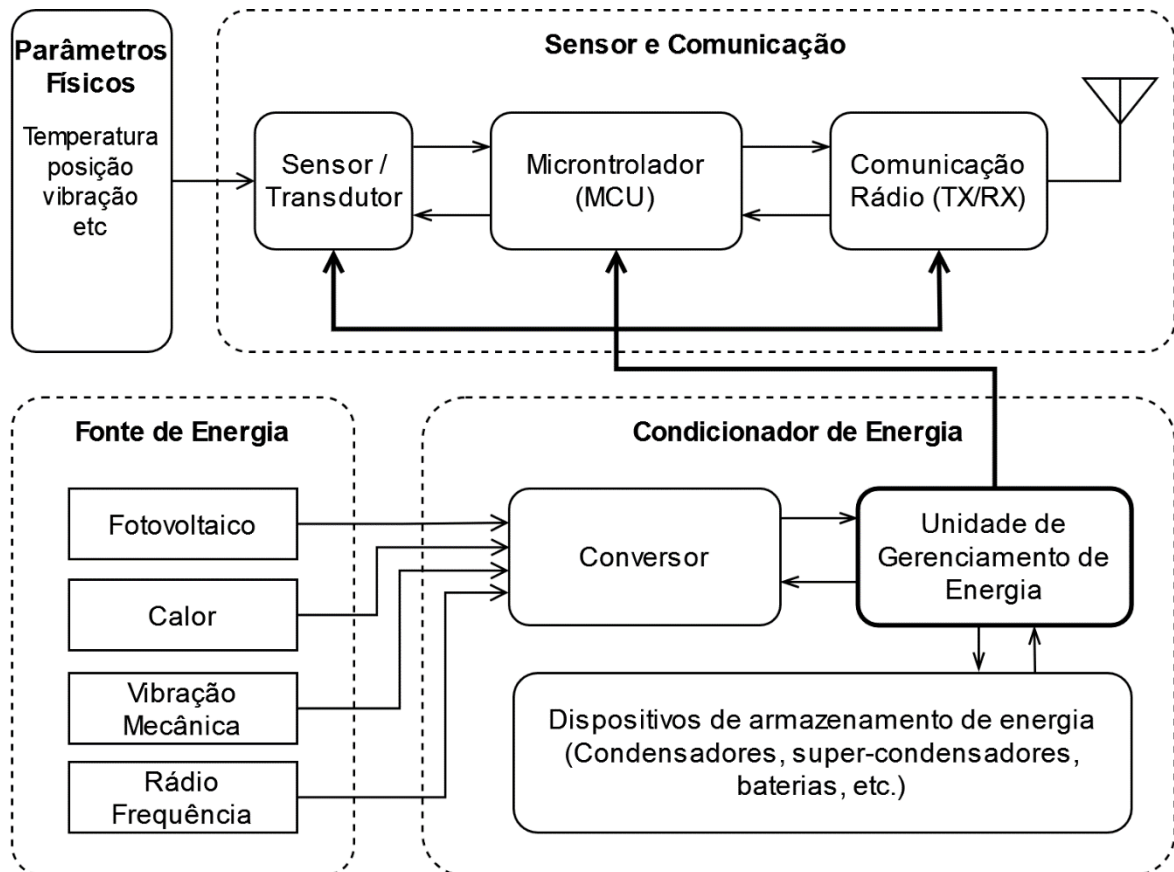


Figura 2.6 - Estrutura de sensor sem fios alimentado com energia *Harvest*

## 2.4. Rede de Sensores Sem Fios

De acordo com a fonte [28] as redes de sensores sem fios WSN (*Wireless Sensor Network*) são dispositivos computacionais de tamanho compacto e de baixo preço que medem as condições ambientais locais ou outros parâmetros e encaminham essas informações para um ponto central para processamento apropriado. Os dispositivos WSN podem detectar alterações no ambiente, podem-se comunicar com dispositivos vizinhos e podem, em muitos casos, realizar cálculos sobre os dados recolhidos.

Ainda de acordo com a mesma fonte, a classificação das WSN é feita em duas categorias:

- Categoria 1: quase invariavelmente sistemas baseados em malha com conectividade de rádio *multihop* entre nós sem fios, utilizando roteamento dinâmico nas partes sem fio e com fio da rede. Os sistemas militares normalmente pertencem a esta categoria.
- Categoria 2: *point-to-point* ou *multipoint-to-point* (em estrela) sistemas geralmente com comunicação rádio de *single-hop* para nós sem fios, utilizando roteamento estático pela rede sem fio. Normalmente, haverá apenas um caminho dos nós sem fios para o nó de encaminhamento terrestre/com fio complementar. Os sistemas de controle residencial normalmente pertencem a esta categoria.

### 2.4.1. Comunicação Sem Fios

A comunicação sem fios poderá ser realizada recorrendo a diferentes frequências, modelações e potência de transmissão. Existem atualmente protocolos de comunicação sem fios que permitem minimizar a perda de informação a transmitir.

Numa comunicação sem fios bidirecional há a necessidade de se utilizar dois rádios (emissor/recetor).

A norma IEEE1451 suporta diferentes protocolos de comunicação sem fios, tais como: Wi-Fi, Bluetooth, ZigBee e 6LoWPAN definidos na norma IEEE1451.5. Suporta ainda o sistema RFID (*Radio Frequency Identification*) definido na norma IEEE1451.7 [29].

Na Tabela 2.3 [30] apresenta-se uma lista de rádios utilizados em sistemas de sensores sem fios. Algumas destas plataformas/rádios não possuem protocolos de comunicação suportados pela IEEE1451.5. O rádio utilizado neste estudo foi o CC2500, que embora não seja suportado pela norma IEEE1451.5 é aquele que satisfaz os objetivos pretendidos. O CC2500 funciona na mesma faixa de frequências é possível alterar a frequência para valores fora dos utilizados no WI-FI.

Tabela 2.3 - Rádios utilizados em redes de sensores sem fios

Rádio	RFM TR1000	Infineon TDA5250	TI CC1000	TI CC2420	TI CC2500	Zeevo ZV4002
Plataformas	WeC, Rene, Dot, Mica	eyesIFX	Mica2Dot, Mica2, BTnode	MicaZ, TelosB, SunSPOT, Imote2	eZ-RF2500	Imote, BTnode
Protocolo	N/A	N/A	N/A	IEEE 802.15.4	N/A	Bluetooth
Velocidade de transmissão (Kbps)	2,4 a 115,2	19,2	38,5	250	1,2 a 500	733,2
Modulação	OOK/ASK	ASK/FSK	FSK	O-QPSK	OOK/ 2-FSK/ GFSK/ MSK	FHSS- GFSK
Frequência (MHz)	916	868	315/433/868/915	2400	2400	2400
Alimentação (V)	2,7 a 3,5	2,1 a 5,5	2,1 a 3,6	2,1 a 3,6	1,8 a 3,6	0,85 a 3,3
TX máximo (mA/dBm)	12/ -1	11,9/9	26,7/10	17,4/0	11,1/21,5	32/4
RX mínimo (mA/dBm)	N/A	4,9/ -22	5,3/ -20	8,5/ -25	13,3	N/A
RX (mA)	1,8-4,5	8,6-9,5	7,4-9,6	18,8	17,0	32
Sleep ( $\mu$ A)	5	9	0,2-1	0,02	0,4	3300
Startup time (ms)	12	0,77-1,43	1,5-5	0,3-0,6	0,24	N/A

#### 2.4.2. Aplicações Disponíveis

Existem diversas aplicações de WSN em diferentes campos de utilização, desde aplicações militares à saúde. Segundo a fonte [30] existem, à data da publicação, as aplicações:

- Militares:
  - *Smart Dust* - Foi uma das primeiras aplicações em WSN, foi financiado pela *Defense Advanced Research Projects Agency (DARPA)*. Tem como principal objetivo fornecer tecnologias de redes de sensores que são utilizadas em contexto de operações militares em ambientes hostis.
  - *Sniper Detection System* - O sistema *Boomerang Sniper Detection System* foi desenvolvido para deteção precisa da localização de atiradores.
  - *VigilNet* - O *VigilNet* é uma rede de vigilância projetada para procurar alvos furtivos com eficiência energética, em ambientes hostis. O projeto é baseado em 70 nós Mica2, equipados com sensores magnéticos que detetam o campo magnético gerado pelo movimento de veículos e objetos magnéticos.
- Ambientais:
  - *Great Duck Island* - O projeto *Great Duck Island* foi desenvolvido numa colaboração entre o *College of the Atlantic* e a *Intel Research Laboratory* em Berkeley para estudar a distribuição e abundância de aves marinhas. Uma rede de sensores de Mica é utilizada para medir a ocupação de tocas de nidificação.

- CORIE - O projeto CORIE é constituído por 24 estações de sensores no estuário do Rio Columbia, estações essas constituídas por vários sensores ambientais que permitem medir a velocidade da água, da temperatura da água, a salinidade e a profundidade.
- ZebraNet - ZebraNet é um sistema, implantado no Quénia, de localização de animais, desenvolvido para investigar os padrões de deslocação a longo prazo das zebras, suas interações dentro e entre espécies, bem como os impactos do desenvolvimento humano.
- *Volcano Monitoring* - é um sistema de observação de vulcões ativos, onde uma rede de sensores é implantada perto de vulcões ativos para monitorizar continuamente as suas atividades.
- *Early Flood Detection* – é um sistema de redes de sensores sem fios que permite a deteção precoce de inundações, mais concretamente monitorizando o aumento do caudal dos rios.
- Saúde:
  - *Artificial Retina* - O projeto *Retina Artificial* apoiado pelo Departamento de Energia dos EUA visa construir uma retina artificial para implementar em pessoas com deficiência visual.
  - *Patient Monitoring* - O projeto CodeBlue da Universidade de Harvard utiliza sensores embutidos no vestuário, os quais permitem a medição de sinais vitais dos pacientes nas suas vidas diárias.
  - *Emergency Response* - as WSN também são utilizadas em situações de emergências, quando um sensor de monitorização faz uma avaliação dos sinais vitais das vítimas e pede automaticamente ajuda.
- Domótica:
  - Water Monitoring – O projeto NAWMS (*Nonintrusive Autonomous Water Monitoring System*) para residências tem como principal objetivo identificar o desperdício no consumo da água e informar os consumidores para uma utilização mais eficiente.
- Industrial Applications:
  - *Preventive Maintenance* - a manutenção preventiva é utilizada na indústria, uma vez que fornece soluções económicas detetando precocemente a falha de um equipamento.
  - *Structural health monitoring* - as WSN têm sido utilizadas para avaliação das condições do estado estrutural de edifícios, avaliando o *stress* causado pela vibração constante de máquinas.

## 2.5. Energia *Harvesting*

Uma rede de sensores é o conjunto de vários sensores inteligentes. Pode ser utilizada em muitos exemplos do dia-a-dia. Nas cidades, a medição da quantidade de ozono permite inferir a qualidade do ar, identificar fontes de poluição atmosférica para posterior análise e correção [31]. Na agricultura, a leitura da temperatura e humidade do ar ou dos solos, o *pH* dos solos e da água de rega e posterior análise e correção destes permite recriar as condições ideais para um crescimento eficiente de uma plantação.

Conforme a densificação do número de nodos, surgem dificuldades: no aumento de energia necessária; no aumento de ligações por fios ou baterias em sistemas sem fios, com a agravante da necessidade de substituir as baterias.

## 2.6. Fontes de Energia *Harvest*

Para obter a energia necessária para o sensor inteligente funcionar, existem como fontes de energia possíveis: a vibração mecânica, o gradiente térmico, ou a radiação eletromagnética. Estão disponíveis várias técnicas para conversão da energia cinética, da vibração mecânica em energia elétrica. As mais utilizadas são: electrostática, eletromagnética e conversão piezoelétrica [32].

A Tabela 2.4 apresenta a densidade de potência fornecida das diferentes fontes de energia, de acordo com duas fontes: coluna do meio [33] e coluna da direita [34].

Tabela 2.4 - Densidade de Potência

<b>Tecnologia</b>	<b>Densidade de Potência</b>	
Vibração - eletromagnética	4,0 $\mu\text{W cm}^{-3}$	1 - 2 $\mu\text{W cm}^{-3}$
Vibração - piezoelétrica	500 $\mu\text{W cm}^{-3}$	100 - 200 $\mu\text{W cm}^{-3}$
Vibração - eletrostática	3,8 $\mu\text{W cm}^{-3}$	-
Termoelétrica	60 $\mu\text{W cm}^{-2}$ ( $\Delta T = 5 \text{ }^\circ\text{C}$ )	50-100 $\mu\text{W cm}^{-2}$ ( $\Delta T = 1 \text{ }^\circ\text{C}$ )
Fotovoltaico - luz solar	3700 $\mu\text{W cm}^{-2}$	100 000 $\mu\text{W cm}^{-2}$
Fotovoltaico - interior	3,2 $\mu\text{W cm}^{-2}$	<100 $\mu\text{W cm}^{-2}$
RF ( <i>Radio Frequency</i> )	-	0,0002 – 1 $\mu\text{W cm}^{-2}$

Da análise da Tabela 2.4 é possível verificar que a tecnologia fotovoltaica é a mais eficiente, mas apresenta o inconveniente de ser intermitente. A escolha da fonte de energia depende sempre do estudo das fontes disponíveis e do local de implementação do sistema de sensores. Um exemplo muito diferente do de Portugal é a Islândia, onde no inverno há pouca quantidade de luz solar, mas há contraste térmico entre a temperatura das fontes

geotermiais e a temperatura do gelo. Aqui pode ser implementado sensores com uma fonte de energia termoelétrica.

### 2.6.1. Fotovoltaico

O efeito fotovoltaico é um fenómeno em que surge uma diferença de potencial, num material, após a sua exposição à luz. Está diretamente relacionado com o efeito fotoelétrico, mas a partir de processos diferentes. No efeito fotoelétrico, os eletrões são “arrancados” da superfície de um material após exposição à radiação com energia suficiente. O efeito fotovoltaico é diferente pois os eletrões são transferidos entre diferentes níveis de energia dentro do próprio material, resultando numa diferença de potencial entre dois elétrodos [34].

### 2.6.2. Termoelétrico (Gradiente de Temperatura)

Os geradores de energia termoelétrica TEG utilizam o efeito termoelétrico, ou de Seebeck, dos semicondutores para converter energia térmica em energia elétrica. Existem há muitos anos, tendo a descoberta do efeito termoelétrico, sido alcançada em 1821 por Thomas J. Seebeck.

Os geradores termoelétricos são dispositivos de estado sólido, projetados para gerar eletricidade diretamente do calor, o que é conhecido como efeito Seebeck [35].

### 2.6.3. Piezoelétrico (Vibração)

Alguns materiais possuem a propriedade de acumular cargas elétricas quando sujeitos a forças. Os dispositivos piezoelétricos são equipamentos que convertem a sua distensão mecânica em energia elétrica. A Figura 2.7, ilustra o seu funcionamento. Na indústria a maioria destes dispositivos são feitos de PZT (*lead zirconate titanate* – Pb Zi Ti) e PVDF (*polyvinylidene fluoride*) [36].

Piezoelétricos com poucos centímetros de tamanho, conseguem gerar milliwatts de potência elétrica utilizando vibrações ambientais para frequências abaixo de 1 KHz [32].

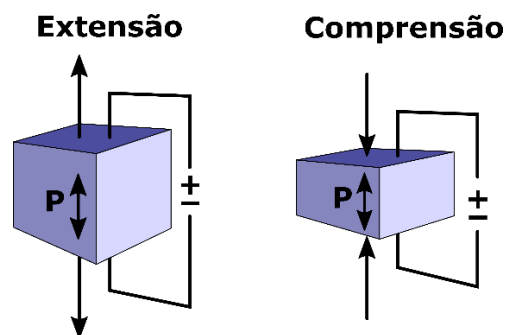


Figura 2.7 - Efeito Piezoelétrico

#### **2.6.4. Eletromagnética (Vibração)**

A indução eletromagnética é um fenômeno em que surge uma diferença de potencial nos terminais de um circuito elétrico, quando há variação do fluxo do campo magnético. Esta variação de fluxo do campo magnético pode ocorrer quando a intensidade do campo magnético varia ou quando varia a posição do circuito em relação ao campo magnético. A diferença de potencial será tanto maior quanto maior for a variação de fluxo de campo magnético.

#### **2.6.5. Rádio Frequência Ambiente**

Como a eletricidade e as comunicações de rádio são utilizadas diariamente, a energia pode ser encontrada na forma de campos eletromagnéticos. A recolha dessa energia possibilita a instalação de WSN alimentadas a partir de energia numa grande variedade de casos.

Devido à sua natureza ondulatória, a energia recolhida do campo eletromagnético cria uma tensão alternada, que é então retificada e armazenada para ser utilizada pelo nó WSN.

Exemplos de utilização desta tecnologia normalmente incluem sistemas de identificação por radiofrequência (RFID) e comunicação por proximidade de campo (NFC), em que um leitor ativo é usado para alimentar e ler medições de uma etiqueta passiva. Como algumas dessas etiquetas só precisam ser alimentadas quando solicitadas para medição, tais sistemas podem ser implementados sem recurso a bateria. Embora a recolha de energia de campo próximo seja geralmente utilizada para transmissão de energia de curto alcance, o alcance pode atingir até 5 m.

#### **2.6.6. Geradores Puramente Eletrostáticos**

Geradores puramente eletrostáticos baseados em um único condensador variável foram descritos em muitos trabalhos. Geralmente, estes exigem um conversor DC/DC para extrair a energia dos dispositivos que em alguns casos depende da sincronização do conversor em certos instantes do ciclo da vibração. Como esses pequenos geradores não produzem muita energia, a alimentação do sistema de controle pelo gerador é um problema significativo. Além disso, a maioria dos dispositivos também exige uma bateria para inicialização, retirando energia e devolvendo-a aumentada após um ciclo de operação [36][37].

## 2.7. O Impacto das Restrições Energéticas nos Sensores Inteligentes

As restrições energéticas podem ocorrer por diferentes motivos. Um sensor que obtenha a energia de que necessita para operar no ambiente onde está instalado, ou que seja alimentado por uma bateria, possui limitações operacionais. Esta tarefa pretende realizar um levantamento das atuais tecnologias utilizadas para recolher energia do ambiente de operação e utilizar essa energia para fazer funcionar um sensor.

Do mesmo modo, a tarefa pretende realizar uma análise crítica da norma IEEE1451, no sentido de identificar limitações e oportunidades desta norma, para que possa ser utilizada em aplicações industriais, ou outras áreas de aplicação, que recorram à *Energy Harvest* para alimentar sensores [38].

## 2.8. Norma IEEE1451

### 2.8.1. Breve História da Norma IEEE1451

Em setembro de 1993 o NIST (*National Institute of Standards and Technology*) e o IEEE (*Institute of Electrical and Electronic Engineers*) realizaram uma reunião para debater acerca das tecnologias existentes na área dos transdutores inteligentes. Nesta reunião reconheceu-se a necessidade de criar uma interface de comunicação comum para transdutores inteligentes. Inicialmente foram criados quatro grupos de trabalho o P1451.1, P1451.2, P1451.3 e P1451.4.

Tabela 2.5 - Família IEEE1451

Norma	Descrição	Estado	Ano	Revisto
1451.0	Formato de: <i>Common Functions, Communication Protocols</i> e <i>Transducer Electronic Data Sheet</i> (TEDS)	Aprovado	2007	
1451.1	Modelo de Informação <i>Network Capable Application Processor</i>	Aprovado	1999	Requer
1451.2	Protocolos de comunicação do Transdutor para o Microprocessor e formato das TEDS	Aprovado	1997	2002
1451.3	Comunicação digital e formato das TEDS	Aprovado	2003	Requer
1451.4	Protocolo de comunicação em <i>Mixed-Mode</i> e formato da TEDS	Aprovado	2004	
1451.5	Protocolo de comunicação sem fios e formato da TEDS	Aprovado	2007	
1451.6	CAN Bus	Pendente		
1451.7	Protocolo de comunicação para RFID ( <i>Radio Frequency Identification</i> ) e formato das TEDS	Substituída	2010	

A Tabela 2.5 [38] apresenta a descrição de cada uma das normas que constituem a IEEE1451.

### 2.8.2. Estrutura da Norma IEEE1451

A norma IEEE1451 propõem três módulos principais: Aplicação do utilizador; NCAP (*Network-Capable Application Processor*) e TIM (*Transducer Interface Module*). Cada um deles possui uma função específica, que podem ser interligadas com recurso a diferentes interfaces, quer sejam cablados quer sejam sem fios. A Figura 2.8 apresenta o diagrama de blocos da arquitetura para a norma IEEE1451 [39]. Atualmente, esta família é constituída, tal como se representa, pelas várias normas.

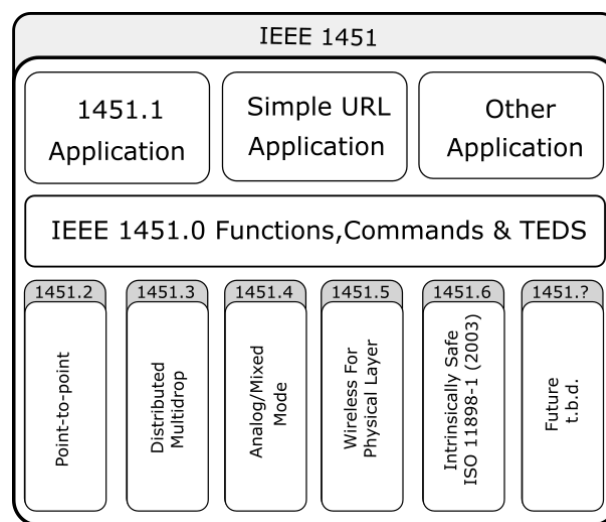


Figura 2.8 - Diagrama de Blocos IEEE1451

A norma IEEE1451.0 define funções, comandos e TEDS (*Transducer Electronic Data Sheet*) comuns para toda a família IEEE1451. Uma TEDS é descrita através de um método normalizado que permite guardar de forma estruturada a informação relevante para a utilização do sensor inteligente. Exemplos de TEDS disponibilizam informação relativa à organização da TIM, modo de operação dos canais de aquisição, calibração e correção de dados de transdutor (sensor ou atuador) e informação do fabricante [39].

A norma IEEE1451.1 define um modelo, independente de rede, que define o modo como um transdutor faz a interface com a NCAP. Fornece a definição de um transdutor e dos seus componentes, tendo por base um modelo orientado a objetos, que consiste num conjunto de classes de objetos com atributos, ações e comportamentos especificados, escolhidos para fornecer uma descrição clara e abrangente de um transdutor [40].

A arquitetura de *software* IEEE1451 é definida utilizando três diferentes modelos [41]:

- *Object Model*, que define objetos ou classes com atributos, métodos e estados do transdutor.
- *Data Model*, que define regras de codificação de informações para transmissão das mesmas nas interfaces de objeto local e remoto.
- *Network Communication Model*, que suporta comunicações cliente/servidor e *publish/subscribe* entre diferentes NCAP.

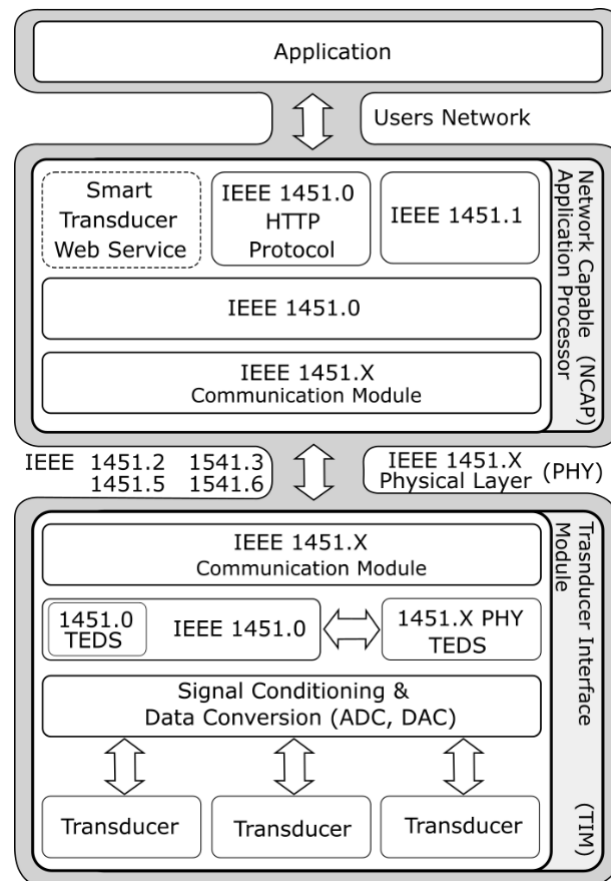


Figura 2.9 - Arquitetura IEEE1451

A NCAP tem a função de gerir todas as TIM que estão ligadas a si e servir de gestor de mensagens e respostas das mensagens trocadas entre a aplicação do utilizador e as TIM. Ao executar um procedimento de *discovery*, com o objetivo de encontrar uma nova TIM, lerá de seguida todas as TEDS que se encontram na TIM, descodificá-las e guardar a informação para esta ser disponibilizada no sistema.

A TIM é composta por um, ou mais, canais transdutores. Se ao canal transdutor estiver ligado um elemento sensor, a TIM é responsável pela aquisição da grandeza e de a converter para o formato digital. Se ao canal estiver ligado um atuador, então a TIM é responsável por controlar o estado de operação desse atuador. Para que possa ser reconhecida automaticamente a sua estrutura interna, a informação das suas

características está contida nas TEDS. A operação da TIM, e respetivos canais transdutores, é gerida de forma normalizada em resultado dos pedidos efetuados pela NCAP.

A TIM é um módulo que contém uma interface de comunicação com uma NCAP, um conjunto de TEDS de configuração, um ou mais canais transdutores, a capacidade de converter o sinal adquirido pelo transdutor num sinal digital e acondicionar o sinal adquirido.

A interface entre a TIM e a NCAP poderá ser realizada através de protocolos suportados pela norma, tais como [11]:

- IEEE1451.2 (RS232);
- IEEE1451.3 (Wired Multi-drop);
- IEEE1451.5 (WiFi, Bluetooth, ZigBee, 6LoWPAN);
- IEEE1451.6 (CAN Bus);
- IEEE1451.7 (RFID).

Neste estudo é utilizada inicialmente a tecnologia com fios, uma ligação série, e posteriormente a tecnologia sem fios, com um protocolo próprio, em que se utiliza um mecanismo de envio, verificação e resposta, otimizado para “consumir” a menor quantidade de energia possível.

Um canal transdutor (*TransducerChannel*) pode assumir três configurações diferentes:

- atuador;
- sensor;
- sensor de eventos.

Numa TIM cada canal transdutor possui duas TEDS de configuração individual, uma que configura o canal transdutor (*TransducerChannel TEDS*) e outra que lhe atribui um nome (*User’s Transducer Name TEDS*).

Quando um canal transdutor assume a configuração de sensor é constituído por três módulos, como apresentado na Figura 2.10:

- sensor;
- processamento de sinal, que realiza operações de amplificação e filtragem do sinal;
- conversor analógico/digital;

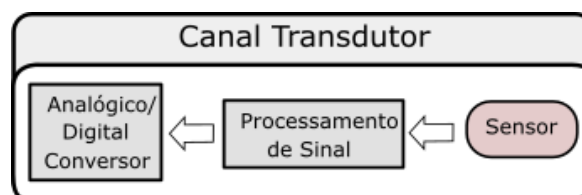


Figura 2.10 - Estrutura Canal Transdutor

Um sistema de aquisição de dados é formado por um sensor (normalmente, que converte a grandeza física numa grandeza elétrica), um conversor analógico/digital e um computador com *software* programável.

O sensor começa por realizar a medição de uma grandeza física, que pode ser: a temperatura, a humidade, a pressão, ou uma força aplicada. O sensor, ou transdutor, converte a grandeza física num sinal elétrico mensurável. Conforme o sensor em causa, a sua saída elétrica pode ser uma característica de tensão, corrente, resistência ou outro atributo elétrico que varie, por exemplo, com o tempo ou o espaço [42].

Um transdutor que seja um atuador é acionado a partir de uma mensagem de comando proveniente da aplicação do utilizador (reencaminhado da NCAP), de outra TIM ou canal transdutor. A estrutura das mensagens enviadas através da interface de comunicação do módulo, pode ser classificada em três tipos: estruturas de mensagem do comando, estrutura de mensagem gerada pela TIM e estrutura de mensagem de resposta [41].

### 2.8.3. Estrutura de Comandos/Resposta

Como já foi referido, a comunicação entre a TIM e a NCAP poderá ser realizada através de diferentes protocolos suportados pela norma, mas independentemente da tecnologia utilizada, esta tem como única e exclusiva função realizar a interface de comunicação, é imperativo a utilização da estrutura de mensagens dos comandos e das respetivas respostas utilizando o conjunto de regras da norma. De acordo com a norma IEEE1451.0 [39], a estrutura da mensagem enviada da NCAP para a TIM tem a seguinte forma:

Tabela 2.6 - Estrutura de Mensagem de Comando

Estrutura de Mensagem de Comando								
ID de Destino		ID do Canal Transdutor		Classe de Comando	Função de Comando	Tamanho	Tamanho	Comando Dependente
1 Octeto	1 Octeto	1 Octeto	1 Octeto	1 Octeto	1 Octeto	1 Octeto	1 Octeto	N Octetos

A estrutura de mensagem do comando enviado da NCAP para a TIM tem a seguinte forma: dois octetos (bytes) para o endereço da TIM de destino, dois octetos para o canal do transdutor, um octeto para a classe de comando, um octeto para a função do comando, dois octetos para o tamanho do comando dependente e o comando dependente (caso exista).

A norma suporta atualmente sete classes de comandos e cada um destes possui diferentes comandos de função. No Apêndice A estão representadas as classes de comandos, as suas funções e as variáveis dependentes.

A estrutura de mensagem de resposta proveniente da TIM para a NCAP tem a seguinte forma:

Tabela 2.7 - Estrutura de Mensagem de Resposta

<b>Estrutura de Mensagem de Resposta</b>			
Sucesso/Falha	Tamanho	Tamanho	Valor da Resposta
1 Octeto	1 Octeto	1 Octeto	N Octeto

Dos comandos enviados pela NCAP que requerem resposta, existem duas possibilidades: se o comando gerar um erro na TIM, esta responde com um octeto “o”, mas se o comando for executado com sucesso na TIM esta responde com octetos “1”. Se a resposta requerer uma mensagem, adiciona mais dois octetos para o tamanho da resposta e a respetiva resposta. O formato do valor da resposta dependerá do comando solicitado à TIM.

Um exemplo de uma mensagem de comando de um atuador está representado na Figura 2.11. Este comando não requer resposta por parte da TIM. (A imagem faz parte da plataforma de desenvolvimento descrito em 3.6.4).

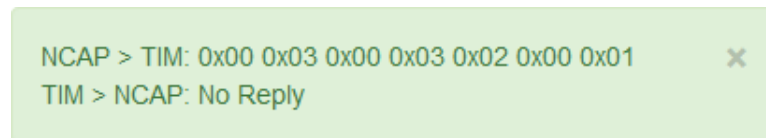


Figura 2.11 - Exemplo de mensagem/resposta de um comando

#### 2.8.4. Comando/Resposta *Query TEDS*

De acordo com a norma IEEE1451.0 [39], o comando *Query TEDS* permite à NCAP verificar que TEDS estão disponíveis numa TIM e recolher informações de: atributos, estado de operação, tamanho da TEDS, soma de verificação e tamanho máximo da TEDS permitido pela TIM.

O comando *Query TEDS* possui a seguinte estrutura:

Tabela 2.8 - Estrutura de Mensagem do Comando *Query TEDS*

<b>Estrutura de Mensagem do Comando <i>Query TEDS</i></b>								
ID de Destino		ID do Canal Transdutor		Classe de comando	Função de comando	Tamanho	Tamanho	TEDS Código de Acesso
0x00	0x01	0x00	0x01	0x01	0x01	0x00	0x01	0x01

No exemplo, apresentado na tabela anterior, o comando é enviado para a TIM com o endereço 0x01 e para o canal transdutor 0x01. O *Command Class*, 0x01, corresponde à classe de comandos *Commands Common to TIM and TransducerChannel* e o *Command Function*, 0x01, à função de comandos *Query TEDS*. O último octeto *TEDS Access Code* permite selecionar qual das TEDS se pretende aceder, no caso do exemplo, 0x01, corresponde à *Meta-TEDS*.

A resposta a este comando tem sempre um tamanho de 12 octetos, onde está contida a informação da TEDS:

- 1 octeto para os atributos;
- 1 octeto para o estado;
- 4 octetos para o tamanho atual da TEDS;
- 2 octetos para a soma de verificação;
- 4 octetos para o tamanho máximo da TEDS.

O octeto de **atributos** é codificado por bits, em que cada bit representa uma característica da TIM. Na Tabela 2.9 está indicado o significado de cada bit.

Tabela 2.9 - Octeto “Atributos”

lsb	0	<i>ReadOnly</i>	Se verdadeiro a TEDS pode ser lida, mas não é possível escrever.
	1	<i>NotAvail</i>	Se verdadeiro a TEDS não é suportada.
	2	<i>Invalid</i>	Se verdadeiro a imagem atual da TEDS é inválida.
	3	<i>Virtual</i>	Se verdadeiro a TEDS é virtual e está localizada na NCAP.
	4	<i>TextTEDS</i>	Se verdadeiro a TEDS é uma TEDS de texto.
	5	<i>Adaptive</i>	Se verdadeiro a TEDS pode ser modificada pela TIM ou pelo canal transdutor sem a necessidade de comando da NCAP.
	6	<i>MfgrDefine</i>	Se verdadeiro o conteúdo deste TEDS é definido pelo fabricante e só estará em conformidade com as estruturas definidas na norma se o atributo <i>TextTEDS</i> também estiver definido.
msb	7	<i>Reserved</i>	Reservado

O octeto de “estado” possui apenas o bit menos significativo codificado. Se for verdadeiro, indica que o tamanho da TEDS que se pretende gravar na TIM é demasiado grande para o tamanho que a TIM suporta. Este bit é alterado pela TIM quando recebe o comando de *Update TEDS* proveniente da NCAP. Os três seguintes bits estão reservados e os quatro últimos estão abertos para os fabricantes.

A resposta a este comando terá o seguinte formato:

Tabela 2.10 - Estrutura de Mensagem da Resposta *Query TEDS*

<b>Estrutura de Mensagem da Resposta <i>Query TEDS</i></b>														
Sucesso /Falha	Tamanho	Tamanho	Resposta											
			Attr	Stat	Tamanho atual da TEDS				Checksum		Tamanho Máximo da TEDS			
0x01	0x00	0x0C	0x00	0x00	0x00	0x00	0x00	0x37	0x3E	0xEF	0x00	0x00	0x00	0xFF

A aplicação terá de possuir mecanismos que lhe permita decodificar a resposta proveniente da TIM.

### 2.8.5. Comando *Read TEDS*

Este comando é usado pela NCAP para solicitar leitura das TEDS. Há um único argumento para este comando e esse é o Código de Acesso TEDS que identifica a TEDS, a ser acedida com os valores de 1 a 15.

Tabela 2.11 - Estrutura de Mensagem do Comando *Read TEDS*

<b>Estrutura de Mensagem do Comando <i>Read TEDS</i></b>								
ID de Destino		ID do Canal Transdutor		Classe de Comando	Função de Comando	Tamanho	Tamanho	Código de Acesso TEDS
0x00	0x01	0x00	0x01	0x01	0x02	0x00	0x01	0x01

Em resposta a este comando é devolvida a respetiva TEDS, caso exista.

### 2.8.6. Comando/Resposta *Read TransducerChannel Data-set Segment*

Este comando é utilizado para ler o valor de um canal transdutor 1 da TIM com endereço 1.

Tabela 2.12 - Estrutura de Mensagem do Comando *Read Data-Set*

<b>Estrutura de Mensagem do Comando <i>Read Data-set</i></b>							
ID de Destino		ID do Canal Transdutor		Classe de Comando	Função de Comando	Tamanho	Tamanho
0x00	0x01	0x00	0x01	0x03	0x01	0x00	0x00

Tabela 2.13 - Estrutura de Mensagem da Resposta *Read Data-set*

<b>Estrutura de Mensagem da Resposta <i>Read Data-set</i></b>						
Sucesso/Falha	Tamanho	Tamanho	Resposta			
0x01	0x00	0x04	0x3D	0xCC	0xCC	0x CD

A resposta é recebida com sucesso, pois o primeiro octeto possui o valor de 0x01. Em seguida, surge o tamanho do valor da resposta de 4 octetos. A resposta representada no exemplo anterior, depois de decodificado, tem um valor de 0,1.

### 2.8.7. TEDS

De acordo com a norma IEEE1451.0 [39], as TEDS são um conjunto de informações codificadas num formato bem definido pela norma. Informações acerca das funcionalidades que uma TIM possui.

As TEDS podem ser armazenadas na memória não volátil da TIM, mas em situações onde não existe espaço suficiente, estas podem ser guardadas noutros locais. Quando guardadas fora da TIM, são chamadas de TEDS “virtuais”. A resposta ao comando *Query TEDS* contém sinalizadores que indicam se a TEDS é suportada e se é virtual.

Por regra, uma TEDS não é alterada depois do seu fabrico. Mas está previsto na norma a alteração às TEDS. Quando o sistema envia um comando de *Query TEDS* à TIM, descrito em 2.8.4, esta responde no Octeto de “Atributos”. O bit *Adaptive* identifica se a TEDS pode ou não ser alterada por via de comandos. O bit de *Status* é definido se o conteúdo de um TEDS for alterado sem escrever a TEDS.

Uma TEDS é composta por um conjunto de octetos de acordo com a Tabela 2.14. Os primeiros quatro octetos representam o tamanho da TEDS (que inclui dois octetos reservados para o CHECKSUM) seguido pelo DATA BLOCK, que possui um número total de octetos dependendo da configuração escolhida pelo fabricante, esta utiliza a tipologia TLV (*TYPE/LENGTH/VALUE*) seguido de dois octetos para o CHECKSUM.

Tabela 2.14 - Formato geral TEDS

<b>Campo</b>	<b>Descrição</b>	<b>Número de Octeto</b>
--	<i>TEDS length</i>	4
1 a N	<i>Data block</i>	Variável
--	<i>Checksum</i>	2

O CHECKSUM é o mecanismo que permite à entidade que lê a TEDS verificar a correta recepção dos dados. Calcula-se a partir da equação 2.3, onde ao somatório dos valores de todos os octetos (com a exceção dos dois últimos, que corresponde ao próprio CHECKSUM) é subtraído o valor de 0xFFFF.

$$\text{CHECKSUM} = 0xFFFF - \sum_{i=1}^{\text{TotalOctets}-2} \text{TEDS OCTET}(i) \quad (\text{eq. 2.3})$$

O DATA BLOCK é composto por um octeto que define o parâmetro (TYPE). Este define um campo específico de informação, seguido pela informação sobre o comprimento do valor (LENGTH), e por fim pelo valor correspondente (VALUE).

Na Figura 2.12 apresenta-se a estrutura de uma TEDS.

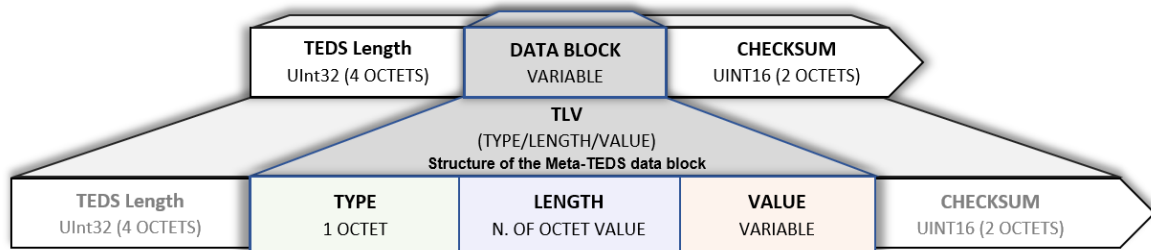


Figura 2.12 - Organização de uma TEDS

A norma conta com um total de quinze TEDS sendo quatro delas de carácter obrigatório [44]. Uma TIM possui obrigatórias as TEDS:

- 1 - *Meta-TEDS*
- 3 - *TransducerChannel TEDS*
- 12 - *User's Transducer Name TEDS*
- 13 - *PHY TEDS*

A **Meta-TEDS** é a TEDS lida em primeiro lugar (pela NCAP ou aplicação) e nela encontram-se codificados os campos relacionados com:

- *Manufacture Globally Unique Identifier* - informação acerca do fabricante como, por exemplo: data de fabrico, local de fabrico.
- *Timing-related information* - informação acerca dos tempos máximos de resposta.
- *Number of implemented TransducerChannels* – informação acerca do número de canais transdutores que a TIM suporta, bem como a forma como se podem agrupar entre eles.

A *TransducerChannel TEDS* fornece informações acerca de um transdutor específico, é obrigatória para cada canal transdutor existente na TIM. Nela é definida que tipo de canal transdutor é, se for um sensor qual a grandeza física que está a medir e a unidade utilizada.

A *User's Transducer Name TEDS* é uma TEDS obrigatória para a TIM e é recomendada para todos os canais transdutores. Esta TEDS tem como função guardar o nome da TIM e do(s) seus canais transdutores.

A *PHY TEDS* informa acerca da interface de comunicação utilizada para ligar a TIM à NCAP.

As restantes 11 TEDS são de caráter opcional e são incluídas conforme a especificidade da função a que se destina a TIM:

- 2 - *Meta-identification TEDS*;
- 4 - *TransducerChannel Identification TEDS*;
- 5 - *Calibration TEDS*;
- 6 - *Calibration identification TEDS*;
- 7 - *End users' application-specific TEDS*
- 8 - *Frequency Response TEDS*;
- 9 - *Transfer Function TEDS*;
- 10 - *Commands TEDS*;
- 11 - *Location and title TEDS*;
- 14 - *Geographic location TEDS*;
- 15 - *Units extension TEDS*.

Como se pode depreender, com o apresentado em 2.8.7, 2.8.9, 2.8.10, 2.8.11 e 2.8.12, a codificação manual das TEDS é um trabalho complexo, moroso e requer o conhecimento profundo da norma. Qualquer pequena alteração no projeto a desenvolver requer obrigatoriamente a reconfiguração das TEDS, tornando o seu desenvolvimento um trabalho complexo. Estas condicionantes podem ser minimizadas utilizando um conjunto de ferramentas de desenvolvimento, projetadas para o efeito. As TEDS apresentadas para a concretização das TIM, deste estudo, foram criadas de acordo com a norma e com a utilização da plataforma de desenvolvimento, criada para o efeito e que se descreve no Capítulo 3.6.

### 2.8.8. Codificação da *Meta-TEDS*

A *META-TEDS* é uma TEDS binária, isto é, os diferentes valores são combinados em bits e não em octetos, só depois são agrupados em octetos. A *Meta-TEDS* é constituída por 10 tipos de informação distintos:

1. Tipo 03 - *Header*
2. Tipo 04 - *Globally Unique Identifier*
3. Tipo 10 - *Operational Time-Out*
4. Tipo 11 - *Slow-Access Time-Out*
5. Tipo 12 - *Self-Test Time*
6. Tipo 13 - *Number of TransducerChannels*
7. Tipo 14 - *Control Group*
  - Tipo 20 - *Group Type*
  - Tipo 21 - *MemList*
8. Tipo 15 - *Vector Group*
  - Tipo 20 - *Group Type*
  - Tipo 21 - *MemList*
9. Tipo 16 - *Geographic Location Groups*
  - Tipo 24 - *LocEnum*
  - Tipo 20 - *Group Type*
  - Tipo 21 - *MemList*
10. Tipo 17 - *Transducer Channels Proxies*
  - Tipo 22 - *ChanNumber*
  - Tipo 23 - *Organiz*
  - Tipo 21 - *MemList*

O *DATA BLOCK* é iniciado pelo Tipo 03, com o nome *Header*, que é a constante que identifica a TEDS. No caso da *META-TED*, assume o valor de “1”, expresso num tamanho de 4 octetos.

O segundo Tipo 04 corresponde ao *Globally Unique Identifier*, com um tamanho de 10 octetos (80 bits), que corresponde ao tamanho total do *Value*. Este *Value* contém várias informações que possuem tamanhos diferentes, requerendo assim a manipulação de bits: o primeiro bit indica o Ponto Cardeal Norte ou o Ponto Cardeal Sul; os seguintes 20 bits representam a latitude do local de fabrico, expressa na unidade de arco-segundo; o bit seguinte representa o Ponto Cardeal Este ou o Ponto Cardeal Oeste; os seguintes 20 bits estão reservados para a longitude do local de fabrico, expressa na unidade de arco-segundo. Até aqui estão utilizados 42 bits que correspondem à informação de localização do fabricante. Os restantes 38 bits correspondem à informação do fabricante em que os

primeiros 4 bits estão reservados para o ID do fabricante; os seguintes 12 bits para o ano de fabrico e os últimos 22 bits para a data de fabrico *Manufacturing Time*. Na estrutura da codificação para o *Globally Unique Identifier* está de acordo com a Figura 2.13.

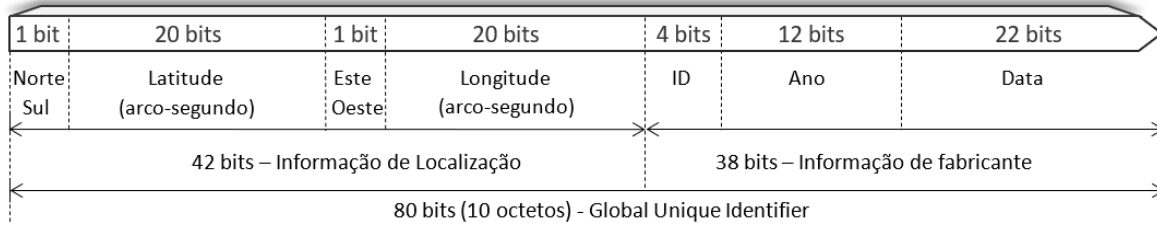


Figura 2.13 - Estrutura do *Globally Unique Identifier*

A data de fabrico é calculada pela equação 2.4.

$$Manufacturing\ Time = Day\ of\ the\ Year \times 1000 + Sequence\ Number \quad (eq.\ 2.4)$$

onde o *Day of the year* é o dia do ano e o *Sequence Number* é o número inteiro na sequência de produção, naquele dia do ano.

Tabela 2.15 - Codificação *Globally Unique Identifier*

Parâmetro	Valor	Valor (Binário)	#Bits
Norte (1) / Sul (0)	Norte	1	1
Latitude	40,27° = 144 972 arco-segundo	0010 0011 0110 0100 1100	20
Este (1) / Oeste (0)	Oeste	0	1
Longitude	7,57° = 27 252 arco-segundo	0000 0110 1010 0111 0100	20
ID fabricante	1	0001	4
Ano	2019	0111 1110 0011	12
Tempo	300001	00 0100 1001 0011 1110 0001	22

Frase Final (Bin)	10010001 10110010 01100000 00011010 10011101 00000101 11111000 11000100 10010011 11100001
Frase Final (Hex)	91 B2 60 1A 9D 05 F8 C4 93 E1

Juntando todos os bits temos uma sequência de 80 bits que podem agora ser separados em grupos de 8 bits, perfazendo os 10 octetos e convertê-los individualmente para o formato de hexadecimal. Podendo então ser colocados à frente do TYPE e do LENGTH.

A latitude e a longitude do local são obtidas normalmente em graus para se converter, em arco-segundo, utiliza-se a correspondência [45]:

$$1 \text{ arco-segundo} = \frac{1}{3600} \text{ de } 1 \text{ grau} \Leftrightarrow 1 \text{ grau} = 3600 \text{ arco-segundos} \quad (\text{eq. 2.5})$$

A Tabela 2.15, apresenta um exemplo prático, atribuindo valores a cada parâmetro, convertê-los em binário e juntá-los no final para obter valor final.

O campo completo ficará da seguinte forma:

Tabela 2.16 - Exemplo Tipo 04 - *Globally Unique Identifier*

T	L	V									
0x04	0x0A	0x91	0xB2	0x60	0x1A	0x9D	0x05	0xF8	0xC4	0x93	0xE1

A informação relacionada com o tempo *Timing-related* é constituída por 3 campos: *Operational Time-Out*, *Slow Access Time-Out* e *Self-Test Time*. Estes três parâmetros utilizam um valor *float32* que é convertido para hexadecimal, com um tamanho de 4 octetos, utilizando a norma IEEE-754 *Floating Point*. Como é definido em [46] pela norma IEEE 754 (Aritmética de vírgula flutuante, possibilitando uma maior precisão em cálculos).

O Tipo 10 corresponde ao campo *Operational Time-Out* que contém o intervalo de tempo máximo, em segundos, para que seja recebida a resposta a um comando enviado. No caso em que o *time-out* seja atingido por falta de resposta, o comando deverá ser interpretado como erro.

O Tipo 11 corresponde ao campo *Slow Access Time-Out* que contém o intervalo de tempo, em segundos, após o qual uma ação para a qual a falta de resposta, após o recebimento de um comando, pode ser interpretada como uma operação com falha.

O Tipo 12 corresponde ao campo *Self-Test Time* que contém o tempo máximo, em segundos, necessário para executar o autoteste. Se nenhum autoteste for implementado, este campo deverá ser zero.

A informação acerca do número de canais transdutores é constituída por 5 campo: *Number of channels*, *Control Group*, *Vector Group*, *Geographic Location Groups* e *Transducer Channels Proxies*.

O Tipo 13 corresponde ao campo *Number of Transducer Channels*, deve ser numerada começando em 1 e ser contínua até ao número de canais transdutores que a TIM possui.

Neste exemplo de TIM, esta possui apenas um canal transdutor, não sendo necessário configurar os parâmetros seguintes.

O Tipo 14 corresponde ao campo *Control Group* que é um campo obrigatório quando numa TIM é implementado um *ControlGroups*. Se não for implementado na TIM, deverá ser omitido da TEDS. O *ControlGroups* identifica os canais transdutores que fazem parte do mesmo grupo. Por exemplo em vez de se fazer um *trigger* individual a cada canal pode-se assim fazer um único comando de *trigger* ao grupo. Possui dois subcampos:

- Tipo 20 - *Group Type* (são suportados 8 tipos *Control Group*)
- Tipo 21 - *MemList* (Lista de canais transdutores que fazem parte do *Control Group*)

O *Control Group* possui um endereço específico [47], este campo é codificado seguindo a estrutura TLV, para o exemplo:

Tabela 2.17 - Exemplo Tipo 14 - *Control Group*

T	L	V						
		T	L	V	T	L	V	
0x0E	0x07	0x14	0x01	0x08	0x15	0x02	0x01	0x02

É iniciado com o 0x0E que corresponde ao campo Tipo 14 seguido do tamanho do campo, que neste exemplo é de 0x07 octetos. O valor 0x14 corresponde ao sub-campo Tipo 20 ou seja ao *Group type* seguido do tamanho de 0x01 octeto e com um valor de 0x08 que segundo a norma corresponde à sequência de: *TransducerChannel (any type)* para o canal transdutor 1 e *Embedded time delay actuator TransducerChannel* para o canal transdutor 2. Seguida surge o octeto de valor 0x015 campo Tipo 21 que corresponde ao *Member List*, seguido do tamanho de 0x02 e o número dos respetivos canais transdutores 0x01 e 0x02.

O Tipo 15 corresponde ao campo *Vector Group* que é um campo obrigatório quando numa TIM é implementado um *Vector Group*. Se não for implementado na TIM, deverá ser omitido da TEDS. Este campo identifica os relacionamentos entre os conjuntos de dados dentro de um módulo transdutor. A definição de *Vector Group* possui dois subcampos:

- Tipo 20 - *Group Type* (são suportados 8 tipos *Control Group*)
- Tipo 21 - *MemList* (Lista de canais transdutores que fazem parte do *VectorGroup*)

O Tipo 16 corresponde ao campo *Geographic Location Groups* que é um campo obrigatório quando numa TIM são implementadas informações de localização dinâmica. Se não for implementado na TIM, deverá ser omitido da TEDS. O campo *Geographic Location Groups* possui três subcampos:

- Tipo 24 - *LocEnum*, disponibiliza 4 opções distintas, de acordo com os valores:
  - Se “0”, nenhuma informação de localização geográfica é fornecida pela TIM;
  - Se “1”, as informações de localização geográfica são estáticas e fornecidas por meio da *Geographic location TEDS*;

- Se “2”, são fornecidas informações dinâmicas de localização geográfica;
- Se “3”, são fornecidas informações dinâmicas de localização geográfica em relação à localização especificada na *Geographic location TEDS*;
- Tipo 20 - *Group Type* (são suportados 8 tipos *Control Group*)
- Tipo 21 - *MemList* (Lista de canais transdutores que fazem parte do *VectorGroup*)

O exemplo mais simples, indica que a TIM não possui informação acerca da localização. Poderia ser omitida, mas nesta situação é dada a certeza que não possui informação acerca da localização.

Tabela 2.18 - Exemplo “Tipo 16 - *Geographic Location Groups*”

T	L	V		
		T	L	V
0x10	0x03	0x18	0x01	0x00

O Tipo 17 corresponde ao campo *Transducer Channels Proxies*. Este campo é requerido apenas numa TIM que implemente *Transducer Channels Proxies* nos seus canais transdutores. Se não for implementado nos canais transdutores da TIM, deverá ser omitido da TEDS.

Com a implementação de um *proxy* é possível combinar as saídas de vários sensores ou a entrada de vários atuadores numa única estrutura. A um canal transdutor *proxy* é atribuído um número de canal transdutor e pode ser acionado, lido ou escrito, mas não possui as outras características de um canal transdutor. Os canais transdutores *proxies* não possuem as TEDS: *TransducerChannel TEDS*, *Calibration TEDS*, *Frequency Response TEDS* e *Transfer Function TEDS*. Está previsto que pode haver uma TEDS baseada em texto, específica da aplicação do utilizador final ou TEDS com o nome do canal transdutor associados a um *proxy*. O campo *Transducer Channels Proxies* possui três subcampos:

- Tipo 22 - *ChanNumber* – Este campo contém o número do canal transdutor *proxy* para o qual se poderá endereçar.
- Tipo 23 - *Organiz* - Este campo indica a forma como os diferentes valores dos canais transdutores, que fazem parte do *proxy* podem ser transmitidos das seguintes formas:
  1. Se “0”, é utilizado o método *Block*;
  2. Se “1”, é utilizado o método *Interleave 1*;
  3. Se “2”, é utilizado o método *Interleave 2*;

No primeiro método deve ser utilizado o valor do *data-set* dos diferentes canais transdutores por possuírem diferentes tamanhos. Os diferentes valores são colocados sequencialmente por ordem do número do canal transdutor.

O segundo método deve ser utilizado quando o *data-set* dos diferentes canais transdutores possuem mais do que um valor. Os diferentes *data-set* são colocados sequencialmente por ordem do número do canal transdutor e por ordem da medição.

O terceiro método é igual ao segundo com a exceção que a lista de canais transdutores intercalados devem ser precedidos por um conjunto de dados de amostra.

- Tipo 21 - *MemList* (Lista de canais transdutores que fazem parte do *ProxyGroup*)

Na Tabela 2.19, apresenta-se um exemplo da implementação de um canal transdutor proxy com o endereço 0x03. Este possui dois canais transdutores: o 0x01 e o 0x02 e os dados são devolvidos pelo método *Interleave 1*.

Tabela 2.19 - Exemplo “Tipo 17 - *Transducer Channels Proxies*”

T	L	V									
		T	L	V		T	L	V	T	L	V
0x11	0x0A	0x15	0x02	0x01	0x02	0x16	0x01	0x03	0x17	0x01	0x01

Colocando tudo junto e calculado o número total de octetos TEDS e o somatório do valor de cada octeto, obtemos a *META-TEDS* apresentada graficamente na Tabela 2.20. Relembrando que uma TEDS é uma sequência de octetos, na figura estão separados de forma a simplificar a identificação do formato TLV, bem como o *TEDS LENGTH* e o *TEDS CHECKSUM*.

Tabela 2.20 - Exemplo final *Meta-TEDS*

Length	Header	DATA-BLOCK										Checksum
00	00 00 2A											
03	04	00	00	00	01							
04	0A	91	B2	60	1A	9D	05	F8	C4	93	E1	
0A	04	3E	99	99	9A							
0B	04	3D	CC	CC	CD							
0C	04	00	00	00	00							
0D	02	00	01									
F5	47											

**Legenda**

- TEDS LENGTH
- T TYPE
- L LENGTH
- V VALUE
- TEDS CHECKSUM

A *Meta-TEDS* indica o número de canais transdutores que cada TIM possui. No exemplo anterior a TIM possui apenas um canal transdutor. Para guardar os parâmetros deste canal transdutor utiliza-se a *TransducerChannel TEDS* que é uma TEDS obrigatória. Caso a TIM possua mais do que um canal transdutor, terá de possuir uma *TransducerChannel TEDS* para cada canal transdutor. Indica que a TIM possui um *Operational Time-out* de 0,3 segundos, um *Slow-access Time-out* de 0,1 segundos e não foi implementado o *Self-Test Time*.

### 2.8.9. Codificação da *TransducerChannel TEDS*

De acordo com a norma IEEE1451.0 [39], a *TransducerChannel TEDS* é uma TEDS binária. Cada canal transdutor necessita obrigatoriamente de uma *TransducerChannel TEDS*. Esta é constituída por 31 campos de configuração distintos.

1. Tipo 03 - *Header*
2. Tipo 10 - *Calibration Key*
3. Tipo 11 - *TransducerChannel Type Key*
4. Tipo 12 - *Physical Units*
  - Tipo 50 - *Physical Units interpretation enumeration*
  - Tipo 51 - *The exponent for Radians*
  - Tipo 52 - *The exponent for Steradians*
  - Tipo 53 - *The exponent for Meters*
  - Tipo 54 - *The exponent for Kilograms*
  - Tipo 55 - *The exponent for Seconds*
  - Tipo 56 - *The exponent for Amperes*
  - Tipo 57 - *The exponent for Kelvins*
  - Tipo 58 - *The exponent for Moles*
  - Tipo 59 - *The exponent for Candelas*
  - Tipo 60 - *TEDS access code for units extension*
5. Tipo 13 - *Design operational lower range limit*
6. Tipo 14 - *Design operational upper range limit*
7. Tipo 15 - *Worst-case uncertainty*
8. Tipo 16 - *Self-test key*
9. Tipo 17 - *Multi-range capability*
10. Tipo 18 - *Sample*
  - Tipo 40 - *Data model*
  - Tipo 41 - *Data model length*
  - Tipo 42 - *Model significant bits*
11. Tipo 19 - *DataSet*

- Tipo 43 - *Maximum data repetitions*
- Tipo 44 - *Series origin*
- Tipo 45 - *Series increment*
- Tipo 46 - *Series units*
- Tipo 47 - *Maximum pre-trigger samples*
- 12. Tipo 20 - *TransducerChannel update time*
- 13. Tipo 21 - *TransducerChannel write setup time*
- 14. Tipo 22 - *TransducerChannel read setup time*
- 15. Tipo 23 - *TransducerChannel sampling period*
- 16. Tipo 24 - *TransducerChannel warm-up time*
- 17. Tipo 25 - *TransducerChannel read delay time*
- 18. Tipo 26 - *TransducerChannel self-test time requirement*
- 19. Tipo 27 - *Source for the time of sample*
- 20. Tipo 28 - *Incoming propagation delay through the data transport logic*
- 21. Tipo 29 - *Outgoing propagation delay through the data transport logic*
- 22. Tipo 30 - *Trigger-to-sample delay uncertainty*
- 23. Tipo 31 - *Sampling attribute*
  - Tipo 48 - *Sampling mode capability*
  - Tipo 49 - *Default sampling mode*
- 24. Tipo 32 - *Data transmission attribute*
- 25. Tipo 33 - *Buffered attribute*
- 26. Tipo 34 - *End-of-data-set operation attribute*
- 27. Tipo 35 - *Edge-to-report attribute*
- 28. Tipo 36 - *Actuator-halt attribute*
- 29. Tipo 37 - *Sensitivity direction*
- 30. Tipo 38 - *Direction angles*
- 31. Tipo 39 - *Event sensor options*

O *DATA BLOCK* é iniciado pelo Tipo 03, com o nome *Header*, que é a constante que identifica a TEDS. Como o exemplo é de uma *TransducerChannel TEDS*, assume o valor de “3”, expressado num tamanho de 4 octetos.

O Tipo 10 corresponde ao campo *Calibration key*, este é constituído apenas por um octeto onde se codifica o tipo de calibração que o canal transdutor suporta. Se o valor for:

- “0” indica que não é necessária informação acerca da calibração do canal.
- “1” a informação da calibração terá de estar disponível numa *Calibration TEDS*.
- “3” a informação da calibração é disponibilizada, mas não no formato que é descrito nesta norma.

- “4” a informação da calibração terá de estar disponível numa *Calibration TEDS*.
- “5” a informação da calibração terá de estar disponível numa *Calibration TEDS*. Neste caso a TIM terá de possuir a capacidade de realizar uma auto calibração.
- “6” a informação da calibração é disponibilizada, mas não no formato que é descrito nesta norma. Neste caso, se houver uma tentativa de leitura da *Calibration TEDS* do canal transdutor a resposta deverá retornar zero como tamanho da TEDS.

Para os valores “1” e “3” a informação, da calibração do canal, facultada pela TIM será sempre manipulada por outra entidade como por exemplo a aplicação. Para os valores “4”, “5” e “6” é da responsabilidade da TIM aplicar a informação acerca da calibração do canal transdutor. Os valores “2” e do “7” ao “255” estão reservados, para futuras adições à norma. Este campo é obrigatório.

Tabela 2.21- Exemplo *Calibration key*

T	L	V
0x0A	0x01	0x00

O Tipo 11 corresponde ao campo *TransducerChannel Type*. Este campo utiliza um octeto para indicar a função do canal transdutor. Este pode assumir três opções: “0” se for um sensor; “1” se for um atuador e “2” se for um sensor de eventos. Este campo é obrigatório.

Tabela 2.22 - Exemplo *TransducerChannel Type* para um sensor

T	L	V
0x0B	0x01	0x00

O Tipo 12 corresponde ao campo *Physical Units* que é codificado numa sequência de 10 octetos.

A codificação *Physical Units* é realizada de acordo com a Tabela 2.23, onde se apresenta o exemplo da codificação da unidade da grandeza aceleração,  $m s^{-2}$ . Cada unidade começa por um valor inteiro de “128” para o caso de existir um expoente negativo o valor resultante seja sempre positivo. Se o valor do expoente for zero, o valor do campo deverá ser “128”. Este campo é obrigatório.

Tabela 2.23 - Codificação *Physical Units*

Octeto	Descrição	Unidade	Tipo de variável	Exemplo (m s <sup>-2</sup> )	
1	Tipo de unidades	--	UInt8	0	0x00
2	Expoente para radianos = 0	Radianos	UInt8	128	0x80
3	Expoente para steradianos = 0	Steradianos	UInt8	128	0x80
4	Expoente para metros = 1	Metros	UInt8	$2^1 + 128 = 130$	0x82
5	Expoente para quilogramas = 0	Quilogramas	UInt8	128	0x80
6	Expoente para segundos = -2	Segundos	UInt8	$2^{-2} + 128 = 124$	0x7c
7	Expoente para amperes = 0	Amperes	UInt8	128	0x80
8	Expoente para kelvins = 0	Kelvins	UInt8	128	0x80
9	Expoente para moles = 0	Moles	UInt8	128	0x80
10	Expoente para candelas = 0	Candelas	UInt8	128	0x80

O primeiro octeto está destinado para indicar o tipo de unidade que se pretende. Estão previstos 6 tipos de unidades diferentes:

- se o valor for “0” a unidade é descrita pelo produto das unidades de base do SI (*International System of Units*), mais radianos e esterradianos, elevados às potências registadas nos campos de 2 a 10.

Unidades para algumas quantidades, como a contagem do número de carros que passam numa portagem, não podem ser representados utilizando estas unidades. Neste caso a melhor forma será atribuir o valor de 128, aos campos 2 a 10, tornando-a adimensional.

- Se o valor for “1” a unidade é definida como  $\frac{U}{U}$ .
- Se o valor for “2” a unidade é definida em logaritmo de base 10,  $\log_{10}(U)$ .
- Se o valor for “3” a unidade é definida em logaritmo de base 10,  $\log_{10}\left(\frac{U}{U}\right)$ .

$U$  é o produto das unidades do SI, radianos e esterradianos, elevados às potências registadas, nos campos de 2 a 10.

- Se o valor for “4” não possui unidade e está associada a uma quantidade digital, por exemplo o *bit*.
- Se o valor for “5” está associada a uma quantidade física representada por valores numa escala arbitrária.

Para estes dois últimos, os campos de 2 a 10 deverão ser definidos com o valor 128.

Na sequência da codificação da TEDS, este campo ficará, para a unidade do exemplo da seguinte forma:

Tabela 2.24 - Exemplo do campo *Physical Units* para a aceleração, em SI

T	L	V									
0x0C	0x0A	0x00	0x80	0x80	0x82	0x80	0x7c	0x80	0x80	0x80	0x80

O Tipo 13 corresponde ao campo de *Low Limit* com um tamanho de 4 octetos. Se o campo *TransducerChannel Type* for definido como sensor, este deverá apresentar o menor valor que o sensor consegue medir, representado nas unidades escolhidas no campo *Physical Units*. Mas se o campo *TransducerChannel Type* for definido como atuador, este deverá apresentar o menor valor para que o atuador seja ativado. Este campo é obrigatório.

Tabela 2.25 - Exemplo do campo *Low Limit*

T	L	V			
0x0D	0x04	0xXX	0xXX	0xXX	0xXX

O Tipo 14 corresponde ao campo de *HiLimit* com um tamanho de 4 octetos. Se o campo *TransducerChannel Type* for definido como sensor, este deverá apresentar o maior valor que o sensor consegue medir, representado nas unidades escolhidas no campo *Physical Units*. Mas se o campo *TransducerChannel Type* for definido como atuador, este deverá apresentar o maior valor para que o atuador ainda seja ativado. Este campo é obrigatório.

Tabela 2.26 - Exemplo do campo *HiLimit*

T	L	V			
0x0E	0x04	0xXX	0xXX	0xXX	0xXX

O Tipo 15 corresponde ao campo *Worst-case Uncertainty* com um tamanho de 4 octetos. Este campo é utilizado para descrever a incerteza que poderá existir na saída deste canal transdutor, devido a fatores externos. Este campo é obrigatório.

Tabela 2.27 - Exemplo do campo *Worst-case Uncertainty*

T	L	V			
0x0F	0x04	0xXX	0xXX	0xXX	0xXX

O Tipo 16 corresponde ao campo de *SelfTest* que define se o canal transdutor possui a capacidade, ou não, de realizar um autoteste. Utiliza apenas um octeto com o valor de “0”

se não necessitar ou não suportar o autoteste, ou o valor “1” se suportar esta função. Este campo é obrigatório.

Tabela 2.28 - Exemplo do campo *SelfTest*

T	L	V
0x10	0x01	0x00

O Tipo 17 corresponde ao campo de *Multi-range capability* com um tamanho de 1 octeto. Define se o canal transdutor possui a capacidade de operar em diferentes escalas de medição. Se o valor for “1” possui esta funcionalidade se for “0” não possui. Este campo é opcional, se não estiver na TEDS significa que o canal transdutor não possui esta funcionalidade.

A partir do campo 18 a informação está relacionada com a *Data converter-related information*.

O Tipo 18 corresponde ao campo de *Sample*, que informa do tipo de dados, tamanho do valor e da resolução suportada. Para definir esta informação, é necessário codificar três subcampos, que são obrigatórios para todas as *TransducerChannel TEDS*:

- 40. *Data Model* - Este campo descreve o modo como lê e escreve o *data-set*. (UInt8, 1 octeto)
- 41. *Data Model Length* - Este campo contém o número de octetos especificados no campo *Data model*. (UInt8, 1 octeto)
- 42. *Model Significant Bits* - Este campo indica o número de bits que fazem parte do valor. (UInt16, 2 octetos)

Por exemplo, se o canal transdutor for um sensor que converte um sinal analógico a partir de um ADC de 16-bits:

- *Data Model* = 0 (N-octeto inteiro)
- *Data Model Length* = 2 (são necessários 2 octetos para conseguir alocar 16 bits)
- *Model Significant Bits* = 16 (0x10)

Tabela 2.29 - Exemplo do campo *Sample*

T	L	V								
		T	L	V	T	L	V	T	L	V
0x12	0x09	0x28	0x01	0x03	0x29	0x01	0x02	0x2A	0x01	0x10
		<i>Data Model</i>			<i>Data Model Length</i>			<i>Model Significant Bits</i>		

O Tipo 19 corresponde ao campo de *DataSet*, este é um campo obrigatório. Caso seja omitido o valor do subcampo *Maximum data repetitions* e *Maximum pre-trigger samples* deve ser assumido como zero. Caso não seja omitido deverão ser definidos os seguintes campos:

- 43. *Maximum Data Repetitions* contém o número máximo de repetições que o canal transdutor pode repetir aquando de um *trigger*. (UInt16, 2 octetos)
- 44. *Series Origin* representa o valor da variável independente associada ao primeiro dado em um conjunto de dados. Se o campo anterior *Maximum Data Repetitions* tiver um valor diferente de zero e este campo for omitido, a NCAP deverá assumir o valor de zero para *Series Origin*. (Float32, 4 octetos)
- 45. *Series Increment* representa o espaçamento mínimo entre os valores da variável associada a membros sucessivos do *DataSet*. O incremento da série é expresso em unidades definidas pelo campo de *Series Units* na TEDS do canal transdutor. (Float32, 4 octetos)
- 46. *Series Units* contém as *Physical Units* associadas aos campos *Series Origin* e *Series increment* na TEDS do canal transdutor.
- 47. *Maximum pre-trigger samples* contém o número de repetições do valor do canal transdutor que pode ser amostrado e armazenado antes de um *Trigger* ao operar nos modos de *Free-running* com *Pre-trigger*. Cada repetição representa um valor de medição adicional produzido pelo canal transdutor, que deve ser alocado para além do valor inicial (por exemplo, tempo) por um valor definido nos campos *Series Increment* e *Series Units* da TEDS do canal transdutor.  
Quando o número de repetições é zero, o sensor não pode ser utilizado no modo de *Free-running* com *Pre-trigger*. (UInt16, 2 octetos)

Tabela 2.30 - Exemplo do campo *Dataset*

T		L		V												
T	L	T	L	V				T	L	V				T	L	V
0x13	0x11	0x2C	0x04	0xXX	0xXX	0xXX	0xXX	0x2D	0x04	0xXX	0xXX	0xXX	0xXX	0x2E	0x04	...
		<i>Series Origin</i>						<i>Series Increment</i>						<i>Series Units</i>		

Dos campos 20 até ao 26 estão relacionados com *Timing-related information*.

O Tipo 20 corresponde ao campo de *TransducerChannel Update Time* com um tamanho de 4 octetos (Float32, 4 octetos). Define o tempo máximo, expresso em segundos, entre o *Trigger Event* e a falta (falha) da primeira amostra do *data-set* para este canal transdutor.

Se o canal transdutor estiver configurado como sensor em modo de *Free-running Sampling*, este parâmetro só se aplica quando está no estado de *Transducer Operating*. Este campo é obrigatório.

Tabela 2.31 - Exemplo do campo *TransducerChannel Update Time*

T	L	V			
0x14	0x04	0xXX	0xXX	0xXX	0xXX

O Tipo 21 corresponde ao campo *TransducerChannel Write Setup Time* com um tamanho de 4 octetos (Float32, 4 octetos). Define o tempo mínimo, expresso em segundos, entre o final da escrita de dados e a aplicação de um *trigger*. Este campo é obrigatório para todos os canais transdutores configurados como atuadores.

T	L	V			
0x15	0x04	0xXX	0xXX	0xXX	0xXX

Tabela 2.32 - Exemplo do campo “*TransducerChannel Write Setup Time*”

O Tipo 22 corresponde ao campo *TransducerChannel Read Setup Time* com um tamanho de 4 octetos. Define o tempo máximo, expresso em segundos, entre um *trigger* ao canal transdutor e o tempo em que os dados estão disponíveis para serem lidos. Se o valor for zero, o canal transdutor poderá ser lido a qualquer momento após ser acionado. Este campo é obrigatório para canais transdutores configurados como sensor. Poderá ser omitido para sensores de eventos e atuadores.

T	L	V			
0x16	0x04	0xXX	0xXX	0xXX	0xXX

Tabela 2.33 - Exemplo do campo *TransducerChannel Read Setup Time*

O Tipo 23 corresponde ao campo *TransducerChannel Sampling Period* com um tamanho de 4 octetos. Define o período mínimo de amostragem, expresso em segundos, do canal transdutor, livre para leitura ou escrita. Este campo só é obrigatório para os canais transdutores.

T	L	V			
0x17	0x04	0xXX	0xXX	0xXX	0xXX

Tabela 2.34 - Exemplo do campo *TransducerChannel Sampling Period*

O Tipo 24 corresponde ao campo *TransducerChannel Warm-up Time* com um tamanho de 4 octetos. Define o intervalo de tempo, expresso em segundos, que o canal transdutor estabiliza o seu desempenho, conforme especificado em *Uncertainty Under Worst-case Conditions*, após a aplicação de energia ao canal transdutor. Este campo é obrigatório.

T	L	V			
0x18	0x04	0xXX	0xXX	0xXX	0xXX

Tabela 2.35 - Exemplo do campo *TransducerChannel Warm-up Time*

O Tipo 25 corresponde ao campo *TransducerChannel Read Delay Time* com um tamanho de 4 octetos. Define o intervalo de tempo máximo, expresso em segundos, entre a recepção do comando *Read TransducerChannel Data-set Segment* e o início da transmissão de dados. Este campo é obrigatório se o canal transdutor for um sensor, mas poderá ser omitido se for um atuador ou um sensor de eventos.

T	L	V			
0x19	0x04	0xXX	0xXX	0xXX	0xXX

Tabela 2.36 - Exemplo do campo *TransducerChannel Read Delay Time*

O Tipo 26 corresponde ao campo *TransducerChannel Self-test Time Requirement* com um tamanho de 4 octetos. Define o tempo máximo, expresso em segundos, necessário para executar o autoteste. Este campo só é obrigatório para os canais transdutores que implementam a funcionalidade de *self-test* indicado no campo *Self-test key*.

T	L	V			
0x1A	0x04	0xXX	0xXX	0xXX	0xXX

Tabela 2.37 - Exemplo do campo *TransducerChannel Self-test Time Requirement*

Do 27 ao 30 estão relacionados com *Time of the Sample Information*.

O Tipo 27 corresponde ao campo *Source for the Time of Sample* com um tamanho de 1 octeto. Este campo é opcional e pode ser omitido. Se for utilizado poderá ter os valores:

- 0 - *NoHelp*;
- 1 - *Incoming* - Se selecionado, o campo Tipo 28 deve ser incluído;
- 2 - *Outgoing* - Se selecionado, o campo Tipo 29 deve ser incluído;
- 3 - *Minterval*;

- 4 - *SInterval*;
- 5 - *ToDSense*.

Os valores 3 - *MInterval* e 4 - *SInterval* requerem um *embedded sensor* para fornecer as mesmas informações que o campo *TEDS* para *Incoming*.

Para o valor 5 - *ToDSense*, o módulo transdutor fornece a hora do dia em que a amostra foi adquirida. A operação com *MInterval*, *OInterval* ou *ToDSense* requerem que um *ControlGroup* esteja definido na *TEDS Meta-TEDS* para identificar o sensor de intervalo de tempo ou *Sensor TimeInstance* a ser utilizado com este canal transdutor.

O Tipo 28 corresponde ao campo *Incoming propagation delay through the data transport logic* com um tamanho de 4 octetos. Define o *time delay*, expresso em segundos, entre a recepção de um *trigger* dado pela *physical layer* e a aquisição ou aplicação de uma amostra de dados. Este campo só é obrigatório se no campo *Source for the time of sample* for selecionado o valor de 1 - *Incoming*.

O Tipo 29 corresponde ao campo *Outgoing propagation delay through the data transport logic* com um tamanho de 4 octetos. Define o *time delay*, expresso em segundos, entre o último sinal amostrado e o sinal da *data transport logic* de uma mensagem de dados que foi transmitida. Este campo só é obrigatório se no campo *Source for the time of sample field* for selecionado o valor de 2 - *Outgoing*.

O Tipo 30 corresponde ao campo *Trigger-to-sample delay uncertainty* com um tamanho de 4 octetos. Define o valor por defeito da incerteza, expressa em segundos, do atraso entre o *trigger* e a obtenção da amostra que está a ser obtida ou aplicada. Este campo é opcional para todos os tipos de canais transdutores.

Os restantes correspondem aos *Attributes*.

O Tipo 31 corresponde ao campo *Sampling attribute* com um tamanho de 1 octeto.

Os subcampos que compõem este tipo são os seguintes:

48. *Sampling Mode Capability*

49. *Default Sampling Mode*

Este campo é obrigatório para todos os canais transdutores.

T	L	V					
		T	L	V	T	L	V
0x1F	0x06	0x30	0x01	0x01	0x31	0x01	0x01
		Sampling Mode Capability			Default Sampling Mode		

Tabela 2.38 - Exemplo do campo “Attributes”

O Tipo 32 corresponde ao campo *Buffered attribute* com um tamanho de 1 octeto. Este atributo, define os modos de operação de *Buffered* disponíveis para este canal transdutor. Os valores permitidos para este atributo:

- 0 - O canal transdutor não possui *buffer*.
- 1 - O canal transdutor possui vários *buffers* só pode ser utilizado no modo de *buffered*.
- 2 - O canal transdutor possui vários *buffers* e pode ser utilizado com *buffer* ou sem *buffer* (por defeito).
- 3 - O canal transdutor possui vários *buffers* e pode ser utilizado com *buffer* (por defeito) ou sem *buffer*.

Este campo é opcional, se omitido a NCAP deve interpretar que o canal transdutor não possui *buffer*.

O Tipo 33 corresponde ao campo *End-of-data-set Operation Attribute* com um tamanho de 1 octeto.

Este atributo, define os modos de operação de *End-of-data-set* disponíveis para este canal transdutor. Os valores permitidos para este atributo:

- 0 - Não aplicável.
- 1 - Este canal transdutor mantém o último valor no *data-set* até que outro *trigger* seja recebido.
- 2 - Este canal transdutor recircula pelo último *data-set* até que outro *trigger* seja recebido.
- 3 - Este canal transdutor pode ser utilizado com o modo *hold* (por defeito) ou o modo *recirculate*.
- 4 - Este canal transdutor pode ser utilizado com o modo *hold* ou o modo *recirculate* (por defeito).

Este campo é obrigatório para canais transdutores configurados como atuadores. Poderá ser omitido para sensores e sensores de evento.

T	L	V
0x0	0x01	0x00

Tabela 2.39 - Exemplo do campo *End-of-data-set Operation Attribute*

O Tipo 34 corresponde ao campo *Data transmission attribute* com um tamanho de 1 octeto. Este atributo define os modos de *Data Transmission* suportados pelo canal transdutor. Os valores permitidos para este atributo:

- 0 - Reservado;
- 1 - Este canal transdutor só funciona no modo *only when commanded*.
- 2 - Este canal transdutor é capaz de funcionar nos modos de *Streaming when a buffer is full* ou *Only when commanded*.
- 3 - Este canal transdutor é capaz de funcionar nos modos de *Streaming at a fixed interval* ou *Only when commanded*.
- 4 - Este canal transdutor é capaz de funcionar nos modos de *only when commanded*, *Streaming when a buffer is full* ou *Streaming at a fixed interval*.

Este campo é opcional para canais transdutores de sensores ou sensores de evento, poderá ser omitido para atuadores.

Os campos Tipo 32, Tipo 33 e Tipo 34 suportam os seguintes modos de operação, definidos na TIM em *Data Transmission Mode*:

- 0 - Reservado;
- 1 - Apenas quando é recebido um comando de *Read Data-set*;
- 2 - Envia quando o *buffer* está cheio;
- 3 - Envia em intervalo de tempo fixo.

O Tipo 35 corresponde ao campo *Edge-to-report Attribute* com um tamanho de 1 octeto. Este atributo descreve os modos de operação *Edge-to-report* suportados por este sensor de eventos. Este campo é obrigatório para um canal transdutor configurado como sensor de eventos, mas poderá ser omitido em sensores e atuadores.

T	L	V
0x23	0x01	0x00

Tabela 2.40 - Exemplo do campo *Edge-to-report Attribute*

O Tipo 36 corresponde ao campo *Actuator-halt Attribute* com um tamanho de 1 octeto. Este atributo descreve os modos de *Actuator-halt* suportados por este atuador. Este modo determina o que o atuador faz quando recebe um comando *TransducerChannel Idle*. Assume os valores:

- 0 - Não se aplica;
- 1 - Para imediatamente;
- 2 - Para no final do *data-set*;
- 3 - Vai para um estado predefinido.

Este campo é obrigatório se o canal transdutor for um atuador, mas poderá ser omitido se for um sensor ou um sensor de eventos.

T	L	V
0x24	0x01	0x00

Tabela 2.41 - Exemplo do campo *Actuator-halt Attribute*

O Tipo 37 corresponde ao campo *Sensitivity Direction* com um tamanho de 1 octeto.

Para sensores que medem fenômenos físicos com propriedades espaciais tridimensionais, como aceleração, velocidade ou deslocamento, isso deve identificar a direção em relação a um sistema de coordenadas definido pelo fabricante do sensor. O sensor deve produzir uma saída positiva quando o fenômeno físico é aplicado ao sensor na direção especificada. Este campo é opcional.

O Tipo 38 corresponde ao campo *Direction Angles* com um tamanho de 2 octetos.

As duas palavras reais de precisão simples definem dois ângulos medidos a partir do plano de referência e da direção de referência marcada no canal transdutor pelo fabricante. O primeiro número representa  $\rho$  em coordenadas espaciais cilíndricas à direita expressas em radianos. O segundo número representa  $\varphi$  em coordenadas espaciais cilíndricas à direita expressas em radianos. Este campo é opcional.

O Tipo 39 corresponde ao campo *Event sensor options* com um tamanho de 1 octeto.

Um sensor de eventos tem a opção de padrão alterável, limite superior e/ou histerese. Ele também tem a opção de detetar inconsistências nas configurações desses parâmetros. Este campo é obrigatório apenas para canais transdutores configurados como sensores de evento. Poderá ser omitido para sensores e atuadores.

A TEDS final terá a seguinte aparência:

	00	00	00	4A																
Header	03	04	00	00	00	03														
Calibration Key	0A	01	01																	
Type Key	0B	01	01																	
Physical Units	0C	09	32	01	00	33	01	82	34	01	82									
Design lower limit	0D	04	3F	80	00	00														
Design upper limit	0E	04	3F	80	00	00														
Worst-case uncertainty	0F	04	3F	80	00	00														
Self-test key	10	01	01																	
Multi-range capability	11	01	01																	
Sample	12	09	28	01	03	29	01	02	2A	01	10									
DataSet	13	11	2C	04	00	00	00	00	2D	04	00	00	00	00	2E	04	00	00	00	00
Update time	14	04	3F	80	00	00														
write setup time	15	04	3F	80	00	00														
Read setup time	16	04	3F	80	00	00														
Sampling period	17	04	3F	80	00	00														
Warm-up time	18	04	3F	80	00	00														
Read delay time	19	04	3F	80	00	00														
Self-test time	1A	04	3F	80	00	00														
Source time of sample	1B	01	00																	
Incoming delay	1C	04	3F	80	00	00														
Outgoing delay	1D	04	3F	80	00	00														
Trigger-to-sample delay	1E	04	3F	80	00	00														
Sampling attribute	1F	03	01	00	31															
Data transmission	20	01	01																	
Buffered attribute	21	01	01																	
End-of-data-set	22	01	01																	
Edge-to-report	23	01	01																	
Actuator-halt attribute	24	01	00																	
Sensitivity direction	25	04	00	00	00	01														
Direction angles	26	08	00	00	00	00	00	00	00	01										
Event sensor options	27	01	01																	
Checksum	FF	94																		

Legenda

- TEDS LENGTH
- T TYPE
- L LENGTH
- V VALUE
- TEDS CHECKSUM

Tabela 2.42 - TransducerChannel TEDS

### 2.8.10. Codificação da *End users' application-specific TEDS*

De acordo com a norma IEEE1451.0 [39], a *End users' application-specific TEDS* é constituída por 2 campos de configuração distintos:

1. Type 03 - Header
2. Type 10 - EndUserData

O DATA BLOCK é iniciado pelo Type 03, com o nome *Header*, que é a constante que identifica a TEDS. Para a *End users' application-specific TEDS*, assume o valor inteiro de “7”, expressado num tamanho de 4 octetos.

No campo *EndUserData*, o conteúdo do bloco de dados fica ao critério do utilizador e não está definido nesta norma.

A TEDS final terá a seguinte aparência:

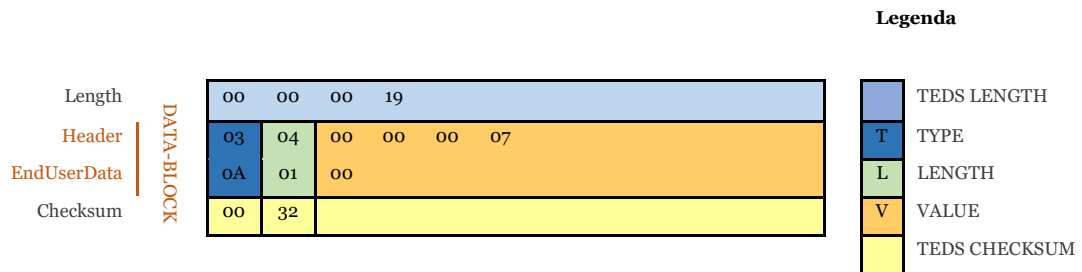


Tabela 2.43 - “*End users' application-specific TEDS*”

### 2.8.11. Codificação da *User's Transducer Name TEDS*

De acordo com a norma IEEE1451.0 [39], a *User's Transducer Name TEDS* é uma TEDS obrigatória para a TIM e recomendada para todos os canais transdutores existentes na TIM. Nesta TEDS fica guardada o nome da TIM e o nome de todos os canais transdutores.

A *TransducerChannel TEDS* é constituída por 3 campos de configuração distintos:

1. Tipo 03 - *Header*
2. Tipo 04 - *Format*
3. Tipo 05 - *TransducerChannel Name*

O DATA BLOCK é iniciado pelo Tipo 03, com o nome *Header*, que é a constante que identifica a TEDS. Para a PHY TEDS, assume o valor inteiro de “12”, expressado num tamanho de 4 octetos.

O Tipo 04 corresponde ao campo *Format* onde é possível indicar o formato da descrição desta TEDS, “0” para ser definido pelo utilizador e “1” para definir uma TEDS de texto.

O Tipo 05 corresponde ao campo *TransducerChannelName*. É codificado convertendo cada uma das letras, individualmente, segundo a tabela ASCII (*American Standard Code*



A PHY TEDS é uma TEDS obrigatória. A função do PHY TEDS será disponibilizar a informação da interface e de todas as informações necessárias para o acesso a qualquer canal, além das informações comuns a todos os canais. Os octetos constituintes da PHY TEDS são constantes e são apenas de leitura.

O DATA BLOCK é iniciado pelo Tipo *03*, com o nome *Header*, que é a constante que identifica a TEDS. Como o exemplo é de uma *PHY TEDS*, assume o valor inteiro de “13”, expresso num tamanho de 4 octetos.

O Tipo *10* corresponde ao campo *Radio Type* onde se indica o tipo de ligação sem fios que se utiliza. O valor é um número inteiro com o valor máximo de um octeto.

Assume o valor de:

- 0 para IEEE 802.11;
- 1 para Bluetooth;
- 2 para ZigBee;
- 3 para LoWPAN;
- de 4 a 254 estão reservados para futuras revisões da norma.

O Tipo *11* corresponde ao campo *MaxBPS Max data throughput*. Este contém a máxima taxa de transferência de dados que o rádio suporta em bits por segundo (bps). O valor deverá ser um número inteiro com o valor máximo de 4 octetos. No exemplo foi utilizado um valor de 250000 bps.

O Tipo *12* corresponde ao campo *MaxCDev Max Connected Devices*. Este campo contém o número máximo de dispositivos que podem estar simultaneamente ligados com este dispositivo. O valor deverá ser um número inteiro com o valor máximo de dois octetos.

O Tipo *13* corresponde ao campo *MaxRDev Max Registered Devices*. Este campo contém o número máximo de dispositivos que se podem registar simultaneamente no nó de rádio. O valor deverá ser um número inteiro com o valor máximo de dois octetos.

O Tipo *14* corresponde ao campo *Encrypt Encryption*. Este campo contém dois octetos, o primeiro octeto, MSB (*Most Significant Byte*), indica se a criptografia de nível MAC é suportada pelo rádio, pela enumeração do tipo de criptografia. Assume o valor de:

- 0 se não tiver qualquer tipo de encriptação;
- 1 para a encriptação AES;
- 2 para a encriptação SAFER;
- 3 para a encriptação 3DES;
- de 4 a 254 reservado para futuras expansões;
- 255 reservado para especificações de fabricante.

O segundo octeto, LSB (*Least Significant Byte*), contém o tamanho da chave de encriptação em bits.

O Tipo 15 corresponde ao campo *Authent Authentication Boolean* que indica se a autenticação de nível MAC (*Media Access Control Address*) é ou não suportada pelo rádio.

O Tipo 16 corresponde ao campo *Min Key Length* que contém o comprimento mínimo da chave de autenticação ou encriptação suportada pelo rádio.

O Tipo 17 corresponde ao campo *MaxKeyL Max Key Length* que contém o comprimento máximo da chave de autenticação ou encriptação suportada pelo rádio.

O Tipo 18 corresponde ao campo *Max SDU Size* que contém o tamanho máximo da unidade de dados de serviço, SDU (*Service Data Unit*), em octetos aceites para transferência entre este dispositivo e outro dispositivo.

O Tipo 19 corresponde ao campo *Min Access Latency* que contém o período mínimo, em nanossegundos, para iniciar uma primeira transmissão para um dispositivo desligado.

O Tipo 20 corresponde ao campo *Min Transmit Latency* que contém o tempo mínimo, em nanossegundos, para transmitir o menor pacote de dados completo de/para o nó de rádio.

O Tipo 21 corresponde ao campo *MaxXact Max Simultaneous Transactions* que contém o número máximo de transações simultâneas que podem estar pendentes na API da WTIM IEEE1451.0.

O Tipo 22 corresponde ao campo *Battery Device is battery powered* que indica se o nó de rádio é alimentado por bateria. Um valor diferente de zero indica que o dispositivo é alimentado por bateria.

O Tipo 23 corresponde ao campo *RadioVer Radio Version* que indica a versão do *software* do código de nível de rádio. Um valor de zero é o padrão e indica que a versão do rádio é desconhecida. Um valor diferente de zero é interpretado com os dois dígitos superiores indicando o número da versão principal do rádio e os dois dígitos inferiores indicando o número da versão secundária. Por exemplo, um rádio com o número de versão 1.2 seria indicado com dois octetos com os valores, respetivamente de 0x01 e 0x02.

O Tipo 24 corresponde ao campo *MaxRetry Maximum Retries before Disconnect* que indica o número máximo de tentativas que o rádio fará antes de executar automaticamente uma operação de desvincular o link do dispositivo. O valor padrão é zero e indica que o rádio nunca será desconectado sem ser instruído.

Do Tipo 32 até ao Tipo 41 estão indicadas as especificações do Protocolo Bluetooth.

Do Tipo 48 até ao Tipo 54 estão indicadas as especificações 802-11 Radio.

Do Tipo 64 até ao Tipo 80 estão indicadas as especificações do protocolo ZigBee.

Do Tipo 96 até ao Tipo 103 estão indicadas as especificações do protocolo 6LoWPAN.

A TEDS final terá a seguinte aparência:

		00	00	00	46	
Length		00	00	00	46	
Header		03	04	00	00	0D
Radio Type		0A	01	04		
Max data throughput		0B	04	00	03	Do 90
Max Connected Devices		0C	02	00	01	
Max Registered Devices		0D	02	00	01	
Encryption		0E	02	04	64	
Authentication		0F	01	01		
Min Key Length		10	02	00	08	
Max Key Length		11	02	00	10	
Max SDU Size		12	02	00	80	
Min Access Latency		13	04	00	00	01
Min Transmit Latency		14	04	00	00	01
Max Simultaneous Transactions		15	01	01		
Device is battery powered		16	01	00		
Radio Version		17	02	01	02	
Maximum Retries before Disconnect		18	02	00	04	
Checksum		FD	B5			

DATA-BLOCK

**Legenda**

- TEDS LENGTH
- T TYPE
- L LENGTH
- V VALUE
- TEDS CHECKSUM

Tabela 2.45 - “PHY TEDS”

### 2.8.13. TIM

A estrutura de funcionamento de uma TIM, está baseada em estados, estados esses que respondem a diferentes comandos provenientes da NCAP. Existem três estados de funcionamento: *TIM INITIALIZATION*, *TIM ACTIVE* e *SLEEP*. O estado de *TIM ACTIVE* é, por sua vez, composto por outros três estados: *TRANSDUCER INITIALIZATION*, *TRANSDUCER IDLE* e *TRANSDUCER OPERATING*. Existem comandos específicos para cada um destes estados e existem comandos que são comuns a todos ou a grupos de estados. O único comando comum a todos os estados é o *Reset* que pode ser utilizado em qualquer altura e que coloca a TIM no estado de *TIM INITIALIZATION* [49].

O processo de inicialização de uma TIM pode ocorrer quando se liga ou quando se executa um *reset* ao *hardware*. Assim que a TIM arranca passa para o estado *TIM INITIALIZATION STATE* onde lê todas as TEDS, coloca na memória toda a configuração contidas nas mesmas e habilita os comandos *Common Commands to the TIM and TransducerChannel*, dando-se o processo de inicialização terminado.

Quando a TIM recebe um comando dos que fazem parte *Common Commands to the TIM and TransducerChannel* passa para o estado *TIM ACTIVE STATE*, nesses comandos são especificados o endereço do transdutor para que se destina, bem como a classe e função do comando. Quando entra no estado de *TIM ACTIVE STATE* são habilitados novos comandos *Transducer Channel Commands*.

Quando entra no estado de *Transducer Idle* é possível enviar comandos que permitem alterar as configurações de funcionamento (anteriormente definidas nas TEDS) como por exemplo o *Sampling mode*, *Data transmission mode*, *Bufered State*, etc. Aquando da leitura das TEDS a TIM carrega toda a configuração de funcionamento, mas esta poderá ser alterada quando se encontra no estado de *Transducer Idle*.

Existem 4 comandos quando a TIM se encontra no estado de *Transducer Operating*: *Trigger*, *Trigger Abort*, *Read TansducerChannel data-set segment* e *Write TansducerChannel data-set segment*.

O comando de *Trigger* pode ser endereçado para um canal transdutor, para um *Proxy* de canais transdutores, um Grupo de endereços ou mesmo globalmente para toda a TIM. Quando a TIM recebe um comando de *Trigger Abort* cancela o *Trigger*.

Quando a TIM recebe um comando de *Read TansducerChannel data-set segment* dependendo do *sample mode* configurado, irá realizar uma leitura do sensor (analógico ou digital), converter o sinal e enviar a resposta para a NCAP. Existem 5 *sample mode*: *Trigger-initiated*, *Free-running without pre-trigger*, *Free-running with pre-trigger*, *Continuous sampling*, *Immediate operation*. Um Transdutor sensor configurado com *Immediate operation* irá adquirir de imediato o valor do sensor e responder a um

comando de *Read TransducerChannel data-set*. A recepção deste comando irá funcionar como *trigger*.

Quando a TIM recebe um comando de *Write TansducerChannel data-set segment*, dependendo se é um atuador analógico ou digital, converte o sinal e coloca-o na porta correspondente ao canal Transdutor.

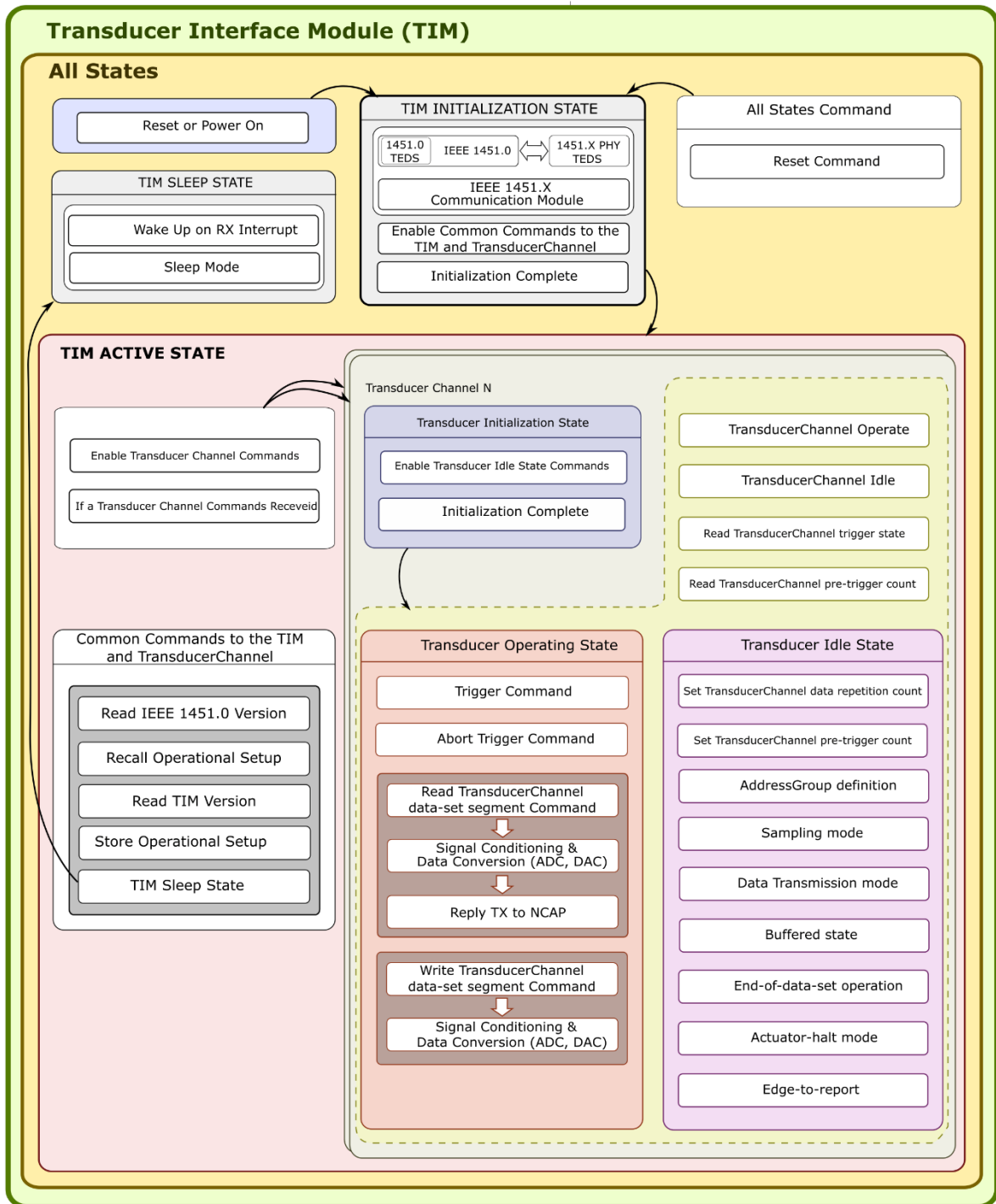


Figura 2.14 - Estrutura interna de uma TIM

A Figura 2.14 apresenta, de uma forma gráfica, os estados, as suas transições e alguns dos comandos suportados por cada estado.

A implementação de um sensor numa TIM pressupõe a adição de um canal transdutor, canal esse definido como sensor, pelo que terá de ser definido um canal transdutor na *meta-TEDS* e adicionar duas novas TEDS, a *Transducer Channel TEDS* e a *Users TransducerChannel Name TEDS*. Todas essas TEDS são configuradas na ferramenta de desenvolvimento *TIM TEDS EDITOR* descrita em 3.6.1 [50].

### 2.8.14. NCAP

A norma que define o conjunto de regras para a NCAP é a norma IEEE1451.1 [51]. A NCAP é uma entidade intermediária entre a TIM e a aplicação. Embora se tenha construído uma NCAP simplificada, esta sai do âmbito deste estudo e não será descrita em pormenor. Apenas é apresentada, na figura seguinte, que mostra a complexa estrutura interna de uma NCAP.

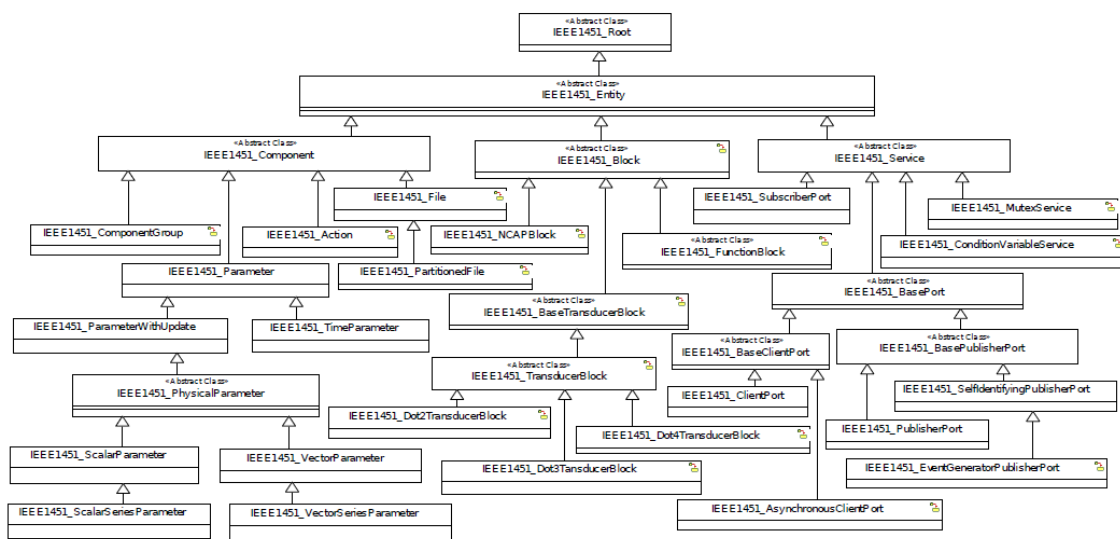


Figura 2.15 - Estrutura interna de uma NCAP

## 2.9. Referências Bibliográficas

- [1] M. A. Fernandes et al., “A framework for wireless sensor networks management for precision viticulture and agriculture based on IEEE1451 standard,” *Computers and Electronics in Agriculture*, vol. 95, pp. 19–30, 2013.
- [2] B. Selim, Y. Iraqi, and H. Choi, “An interoperable multi-sensor system for healthcare,” in *2013 7th IEEE GCC Conference and Exhibition (GCC)*, 2013, pp. 22–27. doi: 10.1109/IEEEGCC.2013.6705742.
- [3] Smith, C., “The car hacker's handbook: a guide for the penetration tester”. San Francisco No Starch Press, 2016. ISBN 9781593277031
- [4] IPQ (Instituto Português da Qualidade), “Módulo 1 - O que são Normas e o que é a Normalização?”, p. 16, Caparica, Portugal, 2015.
- [5] ISO, “ISO/IEC 14543-3-10:2020 - Information technology — Home electronic systems (HES) architecture — Part 3-10: Wireless short-packet (WSP) protocol optimized for energy harvesting — Architecture and lower layer protocols”, 2020 [online], Disponível: <https://www.iso.org/standard/80934.html>, [Acedido: 13-12-2021]
- [6] CENELEC, “EN 50090-1, Home and Building Electronic Systems (HBES) - Part 1: Standardization structure”, 2011, [online], Disponível: <https://standards.globalspec.com/std/1386212/EN%2050090-1>, [Acedido: 15-12-2019]
- [7] CEN, “EN 13321-1, Open data communication in building automation, controls and building management - Home and building electronic system - Part 1: Product and system requirements”, [online], Disponível: <https://standards.globalspec.com/std/14369769/EN%2013321-1>, [Acedido: 15-11-2021]
- [8] ASHRAE, “ASHRAE 135, BACnet - A Data Communication Protocol for Building Automation and Control Networks”, 2020 [online], Disponível: [https://www.techstreet.com/standards/ashrae-135-2020?product\\_id=2191852](https://www.techstreet.com/standards/ashrae-135-2020?product_id=2191852), [Acedido: 17-11-2021]
- [9] IEC, “IEC 61449”, 2005, [online], Disponível: <https://iec61499.com/>, [Acedido: 21-11-2021]

- [10] IEEE, “IEEE Standard Codes, Formats, Protocols, and Common Commands for Use With IEEE Std 488.1-1987, IEEE Standard Digital Interface for Programmable Instrumentation,” in IEEE Std 488.2-1992 , vol., no., pp.1-254, 30 Nov. 1991, doi: 10.1109/IEEESTD.1991.8727628.
- [11] K. Yoon, S. -K. Kim, S. P. Jeong and J. -H. Choi, "Interfacing Cyber and Physical Worlds: Introduction to IEEE 2888 Standards," 2021 IEEE International Conference on Intelligent Reality (ICIR), 2021, pp. 49-50, doi: 10.1109/ICIR51845.2021.00016.
- [12] IEC, “Industrial-process measurement, control and automation - Digital factory framework”, 2020, [online], Disponível: <https://standards.iteh.ai/catalog/standards/iec/cfef54c6-c080-4e1d-980a-a7d53a784409/iec-62832-1-2020>, [Acedido: 25-11-2021]
- [13] IEEE, “IEEE 802 - Ethernet, Bridging and Virtual Bridged LANs Wireless LAN, Wireless PAN, Wireless MAN, Wireless Coexistence, Media Independent Handover Services, and Wireless RAN”, [online], Disponível: <https://standards.ieee.org/featured/ieee-802/>, [Acedido: 23-12-2021]
- [14] DEWESoft, “O que é a tecnologia de sensor TEDS IEEE 1451.4?”, [online], Disponível: <https://dewesoft.com/br/aquisicao-de-dados/o-que-e-tecnologia-de-sensor-teds-ieee-1451>, [Acedido: 23-09-2021]
- [15] John Mark, Paul Hufnagel, “The IEEE1451.1 Standard for Smart Transducer”, Kistler Instrument Corporation, [online], Disponível: <https://standards.ieee.org/content/dam/ieee-standards/standards/web/documents/tutorials/1451d4.pdf>, [Acedido: 25-09-2021]
- [16] Fraden, Jacob, “*Handbook of Modern Sensors: Physics, Designs, and Applications*”, Springer Science+Business Media, LLC, 2010, ISBN 978-1-4419-6465-6, DOI 10.1007/978-1-4419-6466-3
- [17] "IEEE Std 1451.1™-1999", IEEE Standard for a Smart Transducer Interface for Sensors and Actuators—Network Capable Application Processor (NCAP) Information Model.
- [18] Focus Metrologia, “Diferença entre exatidão e precisão de medição”, 2020, [online], Disponível: <https://focusmetrologia.com/2020/06/25/exatidao-vs-precisao-de-medicao/>, [Acedido: 27-02-2020]

- [19] SINHA, G. R. “*Advances in Modern Sensors; Physics, design, simulation and applications*”, IOP Publishing, Bristol, UK, 2020, ISBN 978-0-7503-2707-7, DOI 10.1088/978-0-7503-2707-7
- [20] Analog Devices, “Low Voltage Temperature Sensors”, TMP36 folha de características, 1996, [Revisto maio 2015], [online], Disponível: [https://www.analog.com/media/en/technical-documentation/data-sheets/TMP35\\_36\\_37.pdf](https://www.analog.com/media/en/technical-documentation/data-sheets/TMP35_36_37.pdf), [Acedido: 28-02-2019]
- [21] Maxim Integrated, “Programmable Resolution 1-Wire Digital Thermometer”, DS18B20 folha de características, 2007, [Revisto julho 2019], [online], Disponível: <https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>, [Acedido: 28-02-2019]
- [22] muRata, “3-Axis Low Power Analog Output Accelerometer”, CMA3000-A01 folha de características, 2007, [Revisto agosto 2012], [online], Disponível: [https://pt.mouser.com/datasheet/2/281/murata\\_cma3000\\_a01\\_datasheet\\_8278700a.03-1186067.pdf](https://pt.mouser.com/datasheet/2/281/murata_cma3000_a01_datasheet_8278700a.03-1186067.pdf), [Acedido: 28-02-2019]
- [23] ST, “MEMS digital output motion sensor: ultra-low-power high-performance 3-axis "nano" accelerometer”, LIS3DH folha de características, 2010, [Revisto dezembro 2016], [online], Disponível: <https://www.st.com/resource/en/datasheet/lis3dh.pdf>, [Acedido: 02-02-2021]
- [24] Jim E. Hardy, Jim O. Hylton, Tim E. McKnight, C. J. Remenyik, and Frances R. Ruppel. “Flow Measurement Methods and Applications”, Wiley-Interscience: New York, 1999. ix + 254 pp. ISBN 0-471-24509-7
- [25] Sea “YF-S201, “Water Flow Sensor”, YF-S201 folha de características, [online], Disponível: [http://www.mantech.co.za/datasheets/products/yf-s201\\_sea.pdf](http://www.mantech.co.za/datasheets/products/yf-s201_sea.pdf), [Acedido: 09-03-2021]
- [26] E. Song, M. Burns, A. Pandey e T. Roth, “IEEE1451 Smart Sensor Digital Twin Federation for IoT/CPS Research,” IEEE1451 Smart Sensor Digital Twin Federation for IoT/CPS Research, pp. 1-2, 6 maio 2019.
- [27] TAN YEN KHENG, “Analysis, Design and implementation of Energy Harvesting Systems for Wireless Sensor Nodes”, singapore, 2010, [online], Disponível: <http://scholarbank.nus.edu.sg/handle/10635/20996>, [Acedido: 19-03-2021]

- [28] Nicu Bizon, Naser Mahdavi Tabatabaei, Frede Blaabjerg, Erol Kurt, “Energy Harvesting and Energy Efficiency”, Springer International Publishing, 2017, doi: 10.1007/978-3-319-49875-1
- [29] "IEEE Standard for Smart Transducer Interface for Sensors and Actuators-- Transducers to Radio Frequency Identification (RFID) Systems Communication Protocols and Transducer Electronic Data Sheet Formats," in IEEE Std 1451.7-2010, vol., no., pp.1-99, 26 June 2010, doi: 10.1109/IEEESTD.2010.5494713.
- [30] Ian F. Akyildiz, Mehmet Can Vuran, “Wireless Sensor Networks” 29 junho 2010, John Wiley & Sons, Ltd p. 74, ISBN:9780470036013, DOI:10.1002/9780470515181
- [31] R. Pinto, M. Moreira, J. Bonifácio and A. E. Santo, “An IEEE1451 Vehicular Sensor Network Development”, 2020 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), 2020, pp. 1-6, doi: 10.1109/I2MTC43012.2020.9129224.
- [32] D. Kumar, P. Chaturvedi and N. Jejurikar, "Piezoelectric energy harvester design and power conditioning," 2014 IEEE Students' Conference on Electrical, Electronics and Computer Science, 2014, pp. 1-6, doi: 10.1109/SCEECS.2014.6804491.
- [33] B. H. Calhoun et al., "Design considerations for ultra-low energy wireless microsensor nodes," in IEEE Transactions on Computers, vol. 54, no. 6, pp. 727-740, June 2005, doi: 10.1109/TC.2005.98.
- [34] Nicu Bizon, Naser Mahdavi Tabatabaei, Frede Blaabjerg, Erol Kurt, “Energy Harvesting and Energy Efficiency”, Springer International Publishing, 2017, doi: 10.1007/978-3-319-49875-1
- [35] Z. M. Dalala, "Energy harvesting using thermoelectric generators," 2016 IEEE International Energy Conference (ENERGYCON), 2016, pp. 1-6, doi: 10.1109/ENERGYCON.2016.7514088.
- [36] Philip-Dylan Gleonec “Design and implementation of power management strategies for long range radio modules with energy harvesting.” Université de Rennes 1, 2019, [online], Disponível: <https://tel.archives-ouvertes.fr/tel-02131685/file/gleonec19phd.pdf>, [Acedido: 29-03-2021]

- [37] A. C. M. de Queiroz, "Electrostatic energy harvesting without active control circuits," 2014 IEEE 5th Latin American Symposium on Circuits and Systems, 2014, pp. 1-4, doi: 10.1109/LASCAS.2014.6820274.
- [38] Becerra Gómez, Julio Alberto , “Aplicación a sistemas inteligentes de transporte del estándar ieee 1451 en la integración de sistemas empotrados y sensores. Desarrollo de entidades y librerías software para ncap y tim”, [online], Disponível: <http://bibing.us.es/proyectos/abreproy/11904/>, [Acedido: 09-05-2021]
- [39] IEEE Standard for a Smart Transducer Interface for Sensors and Actuators - Network Capable Application Processor Information Model," in IEEE Std 1451.1-1999 , vol., no., pp.1-480, 18 April 2000, doi: 10.1109/IEEESTD.2000.91313.
- [40] R. Franzl and D. Gurkan, "Implementation of an IEEE 1451.1 compatible multiple NCAP controller using labview," 2009 IEEE Sensors Applications Symposium, 2009, pp. 137-141, doi: 10.1109/SAS.2009.4801794.
- [41] J. Kim and W. Kim, "A study on the Implementation of IEEE1451 for e-Health," 2006 7th International Symposium on Antennas, Propagation & EM Theory, 2006, pp. 1-4, doi: 10.1109/ISAPE.2006.353287.
- [42] N. Ulivieri, C. Distanto, T. Luca, S. Rocchi e P. Siciliano, “IEEE1451.4: A way to standardize gas sensor, Sensors and Actuators” p. 142, 1 Julho 2006, doi: 10.1016/j.snb.2005.04.044
- [43] TAN YEN KHENG, “Analysis, Design and implementation of Energy Harvesting Systems for Wireless Sensor Nodes”, singapore, 2010, [online], Disponível: <http://scholarbank.nus.edu.sg/handle/10635/20996>, [Acedido: 19-03-2021]
- [44] E. Y. Song and K. Lee, "Understanding IEEE 1451-Networked smart transducer interface standard - What is a smart transducer?", in IEEE Instrumentation & Measurement Magazine, vol. 11, no. 2, pp. 11-17, April 2008, doi: 10.1109/MIM.2008.4483728.
- [45] D. A. Fleisch and J. Kregenow, “A student’s guide to the mathematics of astronomy.” New York: Cambridge University Press, 2013, ISBN: 1107610214.
- [46] M. Brain, C. Tinelli, P. Ruemmer and T. Wahl, "An Automatable Formal Semantics for IEEE-754 Floating-Point Arithmetic," 2015 IEEE 22nd Symposium on Computer Arithmetic, 2015, pp. 160-167, doi: 10.1109/ARITH.2015.26.

- [47] R. Pinto, M. Moreira, J. Bonifácio and A. E. Santo, "An IEEE 1451 Vehicular Sensor Network Development," 2020 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), 2020, pp. 1-6, doi: 10.1109/I2MTC43012.2020.9129224.
- [48] "IEEE Standard for a Smart Transducer Interface for Sensors and Actuator -- Wireless Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats," in IEEE Std 1451.5-2007, vol., no., pp.1-225, 5 Oct. 2007, doi: 10.1109/IEEESTD.2007.4346346.
- [49] "IEEE Standard for a Smart Transducer Interface for Sensors and Actuators - Common Functions, Communication Protocols, and Transducer Electronic Data Sheet (TEDS) Formats," in IEEE Std 1451.0-2007, vol., no., pp.1-335, 21 Sept. 2007, doi: 10.1109/IEEESTD.2007.4338161.
- [50] J. L. D. Pereira, H. da Rocha and A. E. Santo, "A Platform for IEEE 1451 Standard's Education, Development and Validation for Industry 4.0," 2020 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), 2020, pp. 1-6, doi: 10.1109/I2MTC43012.2020.9129114.
- [51] K. B. Lee and E. Y. Song, "Object-oriented application framework for IEEE 1451.1 standard," in IEEE Transactions on Instrumentation and Measurement, vol. 54, no. 4, pp. 1527-1533, Aug. 2005, doi: 10.1109/TIM.2005.851225.
- [52] ROHM Semiconductor, "BH1750FVI, Digital 16bit Serial Output Type Ambient Light Sensor IC", BH1750FVI folha de características, 2011 [Revisto novembro 2011]
- [53] IEEE Standards Association, IEEE Standard for a Smart Transducer Interface for Sensors and Actuators (IEEE 1451.2), 1997. [Online]. Available: <http://ieeexplore.ieee.org/xpl/standards.jsp?findtitle=1451&letter=1451>

# Capítulo 3

## Plataformas de Desenvolvimento para a Norma IEEE1451

### 3.1. Introdução

A norma IEEE1451 é uma norma versátil e que conta com vários anos de formulação. Para se construir um sistema, utilizando esta norma, não existem de forma livre e totalmente funcional ferramentas de codificação das TEDS, ferramentas de elaboração das TIM e nem validadores IEEE1451 para as TIM. Os autores [1] descrevem no seu artigo “*An IEEE1451.0-based Platform-Independent TEDS Creator using Open Source Software Components*” uma aplicação para a codificação das TEDS mandatárias, mas não se encontra disponível para utilização.

Este capítulo tem como objetivo, mostrar que é possível a construção de uma plataforma de desenvolvimento gráfica, intuitiva e disponível online de forma gratuita. Será constituída por várias partes: o servidor web onde ficará alojada a plataforma de desenvolvimento, com diversas ferramentas; as TIM com e sem fios (baseadas nas placas MSP-EXP430F5529 e eZ430-RF2500 respetivamente); uma NCAP construída num *Raspberry Pi* utilizando a linguagem Python e uma aplicação escrita na linguagem Python.

### 3.2. Servidor Web

Por forma a desenvolver todas as ferramentas é necessário, em primeiro lugar, criar um conjunto de infraestruturas. A plataforma de desenvolvimento é baseada em aplicações web e, portanto, é necessário escolher um servidor de páginas de internet configurado com os recursos adequados: servidor de páginas, base de dados e servidor de conteúdos dinâmicos.

Os servidores de conteúdo dinâmico utilizam uma arquitetura de três camadas, como se observa na Figura 3.1, que consiste numa camada de servidor web *front-end*, uma camada de servidor de aplicativos que implementa a lógica de negócios e uma camada de banco de dados *back-end* que armazena o conteúdo dinâmico do local [2].

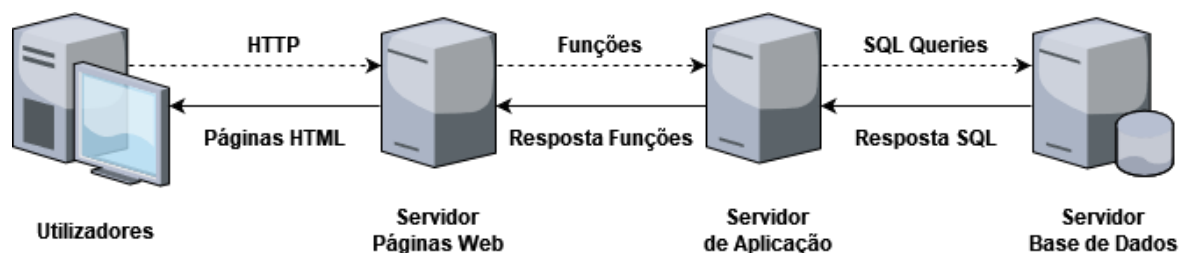


Figura 3.1 - Serviços do Servidor

Por ser uma aplicação web, permite que todas as atualizações realizadas fiquem de imediato disponíveis a todos os utilizadores. O servidor escolhido foi um *Raspberry Pi* versão 3, mas poderia ser qualquer servidor Windows ou Linux. As grandes vantagens do *Raspberry Pi* são: é um microcomputador económico; possui suficiente poder computacional; grande comunidade online. Pelo fato de correr o sistema operativo Linux, rapidamente se pode migrar para servidores profissionais.

Embora tenha um tamanho pequeno, possui características impressionantes, as quais se descrevem na Tabela 3.1 [3].

Tabela 3.1- Características do *Raspberry Pi 3*

Periférico	Características
SoC (BCM2837Bo)	Broadcom BCM2837 [CPU ( <i>Central Processing Unit</i> ), GPU ( <i>Graphics Processing Unit</i> ), DSP ( <i>Digital Signal Processor</i> ) e SDRAM ( <i>Synchronous Dynamic Random Access Memory</i> )]
CPU	1.4GHz 64 bits Quad-Core ARMv8
GPU	Broadcom VideoCore IV 3D graphics core
Memória (SDRAM)	1 GB SDRAM @400 MHz
Portas USB 2.0	4
Saídas de vídeo	HDMI ( <i>High-Definition Multimedia Interface</i> )
Armazenamento	Slot para cartões SD ( <i>Secure Digital</i> ), MMC ( <i>MultiMediaCard</i> ), SDIO ( <i>Secure Digital Input Output</i> )
Saídas de som	Saída 4 Pinos stereo e vídeo composite, HDMI
Rede LAN	Ethernet 10/100 (RJ45)
WI-FI (BCM43438)	Dual-Band (2.4 GHz and 5 GHz) IEEE 802.11.b/g/n/ac
Bluetooth	Bluetooth 4.2 e BLE ( <i>Bluetooth Low Energy</i> )
GPIO	GPIO ( <i>General-Purpose Input/Output</i> ) 40 pinos
Periféricos (baixo-nível)	UART ( <i>Universal Asynchronous Receive Transmit</i> ), I <sup>2</sup> C
Corrente	2,5 A
Alimentação	5 volt via MicroUSB ou GPIO header
Dimensões	85,60 mm × 53,98 mm × 17 mm
Sistemas Operativos	Debian GNU/Linux, Fedora, Arch Linux ...

O *Raspberry Pi* é uma pequena placa do tamanho aproximado de um cartão de crédito com as dimensões de 85,60 mm × 53,98 mm × 17 mm e de massa apenas 45 g. É construído na base de um SoC, (*System-on-a-Chip*), o Processador Broadcom BCM2837Bo de 1,4 GHz 64 bits Quad Core.

### 3.2.1. Configuração do Servidor

A configuração do servidor requer a instalação e configuração de vários programas:

- Apache – Servidor de páginas HTTP (*Hypertext Transfer Protocol*);
- MySQL – Servidor de base de dados SQL (*Structured Query Language*);
- PhpMyAdmin – Ferramenta de administração de base de dados;
- PHP – PHP (*Hypertext Preprocessor*) é uma linguagem de script, que corre no servidor, que permite criar páginas dinâmicas e interativas;
- Bootstrap – É um *framework* de código-fonte aberto para desenvolvimento da interface e *front-end* de sites e aplicações web utilizando as linguagens de programação HTML (*HyperText Markup Language*), CSS (*Cascading Style Sheets*) e JavaScript, tornando o GUI (*Graphical User Interface*) atrativo e de aparência moderna;
- Python – É uma linguagem de programação de alto nível, que interpreta *scripts*, orientada a objetos, disponível para diferentes sistemas operativos;
- Webmin – Ferramenta de administração geral do servidor em ambiente gráfico;
- SSH – Configuração SSH (*Secure Shell*) que permite o acesso remoto ao servidor.

A instalação e configuração deste tipo de servidor encontra-se documentada em muitas fontes, pois a comunidade *Raspberry Pi* está amplamente difundida. O procedimento realizado encontra-se descrita, por exemplo em [4].

O servidor encontra-se, à data da escrita desta tese, no Laboratório de Instrumentação e Medida da Universidade da Beira Interior, pelo que o seu acesso a partir de redes exteriores é impossível devido a restrições de segurança. Para contornar esta situação foi necessário solicitar, ao Centro de Informática, um domínio reencaminhado para o IP (*Internet Protocol*) deste servidor. Ficando assim o servidor acessível através do domínio “<http://iml.ubi.pt/ieee1451>”.

Num *Raspberry Pi* o sistema operativo e todos os dados ficam guardados um cartão de memória SD. Os dados guardados neste tipo de cartões são muito suscetíveis de ficarem corrompidos aquando de uma falha de energia ou quando se atinge o limite de escrita/leitura do cartão. Outras formas bastante frequentes de perder os dados são: ataques informáticos, erros de administração, infiltração de vírus e danos físicos ao equipamento. Por estas situações, os *backups* são cruciais, porque alguns dados não

podem ser substituídos e a sua recriação pode ser dispendiosa em termos de tempo e dinheiro [5]. Para acautelar estas situações é crucial fazer o *backup* dos dados importantes, criando sistemas de *backup* para a eventual perda de dados.

O sistema de *backup*, criado, consiste em dois ficheiros SH (*shell scripts*): *backup\_script.sh* e *dropbox\_uploader.sh*.

```

1  #!/bin/bash
2  # Backup Script
3  # Autor: joadper@gmail.com                                19/02/2019
4
5  #*****
6  # SQL Dump
7  #*****
8  echo "Dump IML Database to sql file..."
9  mysqldump -u _____ -p _____ iml > /home/pi/iml.sql
10 sudo cp /home/pi/iml.sql /var/www/html
11 sudo rm /home/pi/iml.sql
12 echo "Done"
13
14 #*****
15 # Remove all tar.gz files from /var/www/html
16 #*****
17 echo "Remove platform tar.gz files"
18 sudo rm /var/www/html/*tar.gz
19 echo "Done"
20
21 #*****
22 # Create file TAR
23 #*****
24 echo "Compact directory /var/www/html..."
25 FILE_NAME=`date +"backup_%Y%m%d-%H%M".tgz`
26 DIR_ORIG="/var/www/html/"
27
28 tar cvzf $FILE_NAME $DIR_ORIG
29 echo "Done"
30
31 #*****
32 # Dropbox Send file
33 #*****
34 echo "Upload to Dropbox"
35 /home/pi/dropbox_uploader.sh upload $FILE_NAME /
36
37 rm $FILE_NAME
38
39 exit 0

```

Figura 3.2 - Backup Script

A Figura 3.2, apresenta a forma reduzida do primeiro script, este:

- faz uma cópia da base de dados para o ficheiro “*iml.sql*”;
- envia-o para o diretório da página “*/var/html/www*”;
- apaga o *backup* anterior, caso exista;
- cria um ficheiro compactado de todos os ficheiros do diretório “*/var/html/www*”.
- Envia o ficheiro para a nuvem de dados.

O segundo ficheiro é responsável, pelo envio do ficheiro compactado para uma nuvem. O original deste é mantido em [6].

Para tornar o script executável é necessário executar o comando:

```
chmod +x backup_script.sh
```

Estes ficheiros podem ser executados manualmente, no terminal, da seguinte forma:

```
./backup_script.sh
```

Para tornar o processo totalmente automático é possível utilizar o utilitário CRON do Linux, que executa processos em horário agendado. Para isso basta adicionar a seguinte linha ao ficheiro de configuração do CRON.

```
crontab -e
```

```
0 0 * * * /home/pi/backup_script.sh
```

Esta configuração faz com que todos os dias exatamente às 24 horas se efetue um *backup* de todos os ficheiros importantes.

### 3.2.2. Base de dados

A BD (base de dados) é um dos componentes mais importante do sistema, pois é à volta desta que todos os dados são transacionados e manipulados a partir de *scripts PHP* e *Python*. A Figura 3.3 apresenta um exemplo da estrutura de tabelas da BD que foram elaboradas utilizando as ferramentas PHPMyAdmin e MySQL Workbench.

A BD está organizada em quatro secções distintas:

1. Utilizadores;
2. TEDS;
3. TIM Creator;
4. TIM.

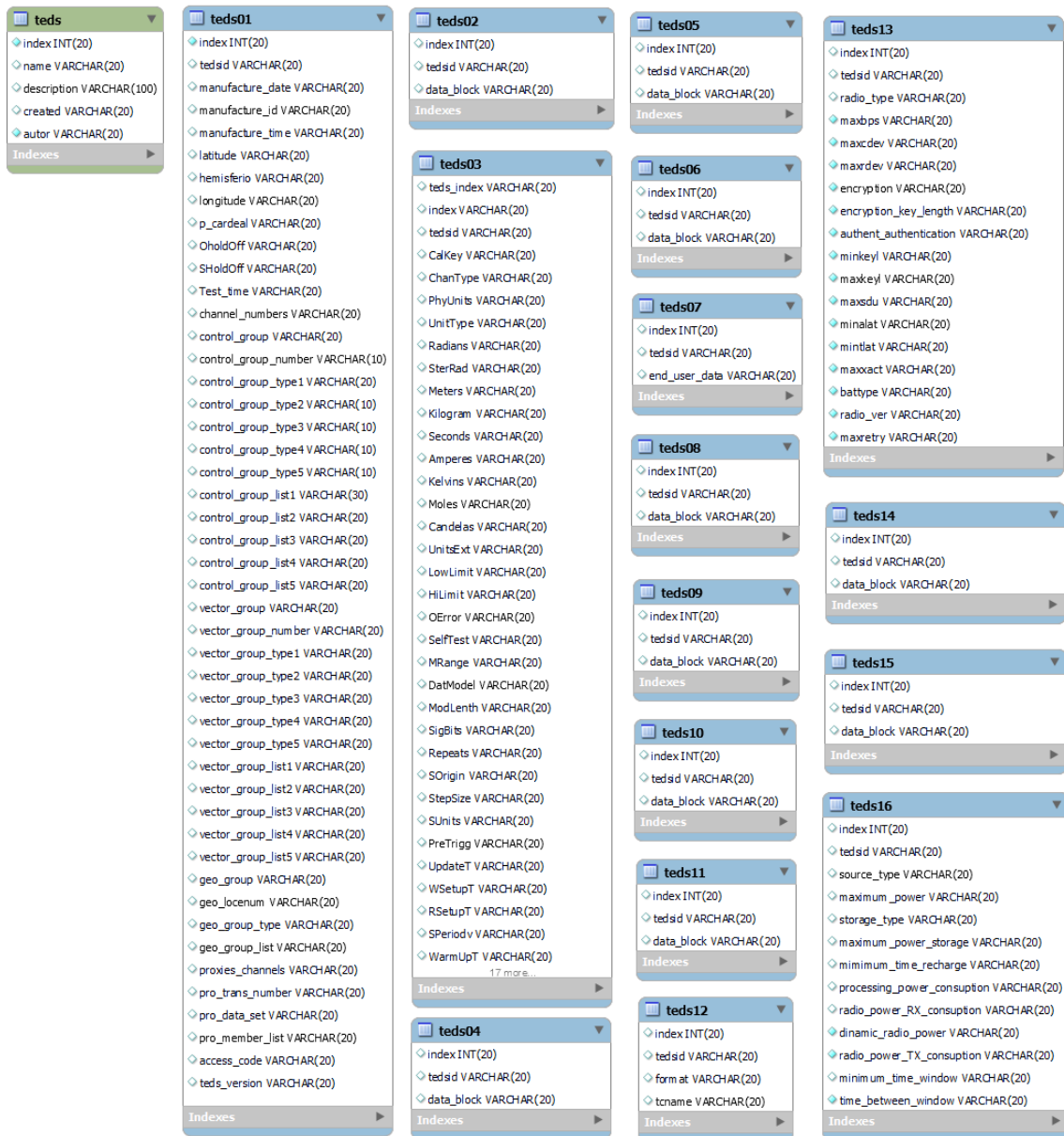


Figura 3.3 - Estrutura da Base de Dados para as TEDS

A primeira secção contém informações acerca dos utilizadores em duas tabelas distintas. A primeira tabela guarda: nome de utilizador; palavra-passe, nome curto; email; nível de utilizador; tema da página e a data do último acesso. A segunda tabela está relacionada com a segurança da página e guarda o número de tentativas de acesso, para que no caso de tentativa abusiva de acesso sem sucesso, a conta fique bloqueada automaticamente.

A segunda secção, está relacionada com a codificação das TEDS. A Figura 3.3 apresenta as tabelas referentes aos dados para a TEDS, uma tabela para cada TEDS e uma para gerir todos os conjuntos de TEDS. Quando é adicionada uma nova TEDS, descrito em 3.6.1, à tabela *teds*, esta é preenchida com os dados inseridos no formulário e automaticamente são inicializadas todas as outras tabelas referentes a cada uma das TEDS. Todos os parâmetros inseridos, na ferramenta Editor de TEDS, são guardados nestas tabelas para posteriormente serem acedidos e para codificar as TEDS a partir de *scripts* PHP.

A terceira secção guarda a informação das TIM desenvolvidas na ferramenta TIM Project Editor, descrita em 3.6.2. Nesta ficam guardados todos os parâmetros que irão permitir elaborar o código para programar a TIM. Parâmetros como: nome e descrição do projeto, endereço da TIM, plataforma escolhida, data de criação, TEDS do projeto, tipo de sensor/atuador e porta onde se liga o sensor/atuador. A informação da ferramenta Editor de TIM fica guardada na BD em formato XML.

A quarta secção contém a informação das TIM que foram adicionadas ao sistema a partir da funcionalidade *TIM Discovery* que se encontra na ferramenta TIM Manager, descrita em 3.6.4. É a partir desta que a aplicação online sabe o endereço da TIM e a interface a que se liga ao sistema.

### 3.3. Desenvolvimento TIM Cablada

O desenvolvimento da TIM requereu necessariamente o desenvolvimento de outras partes, tais como:

- *hardware* da TIM (descrito em 3.3.1);
- *software* da TIM (descrito em 3.3.2);
- *hardware* da NCAP (descrito em 3.5.1);
- *software* da NCAP (descrito em 3.5.2);
- aplicação (descrito em 3.6.4).

A função de TIM é atribuída a uma placa MSP-EXP430F5529, da *Texas Instruments* [7]. Esta placa possui o microcontrolador MSP430F5529 e os seguintes periféricos:

- LCD (*Liquid Crystal Display*) com resolução de 102×64 pixels (EA DOGS102W-6), com luz de fundo controlada por PWM (*Pulse-Width Modulation*).
- Botões de pressão, potenciômetro e LEDs (*Light-Emitting Diode*).

<b>Periférico</b>	<b>Pino</b>
Potenciômetro	P8.0
Interruptor (S1)	P1.7
Interruptor (S2)	P2.2
Interruptor (S3) RESET	$\overline{\text{RST}}/\text{NMI}$
LED1	P1.0
LED2	P8.1
LED3	P8.3
Toque Capacitivo 1	P1.1
Toque Capacitivo 2	P1.2
Toque Capacitivo 3	P1.3
Toque Capacitivo 4	P1.4
Toque Capacitivo 4	P1.5

Tabela 3.2 - Periféricos MSP-EXP430F5529

- Interface para módulos sem fios.
- Interface para o módulo eZ430-RF2500T.
- Acelerômetro de 3-Eixos.
- Pinos de acessos a diversas portas do microcontrolador.

### 3.3.1. Hardware TIM Cablada

O primeiro dispositivo a ser considerado para a TIM foi uma placa de desenvolvimento da *Texas Instruments*: uma MSP-EXP430F5529 por possuir características de baixo consumo. Contém um sensor de temperatura e um acelerómetro. Para implementar a NCAP, foi escolhido o *Raspberry Pi 3* Modelo B. A comunicação entre esses dois dispositivos é assegurada por um protocolo de comunicação física, UART, ligando ao pino TX da TIM ao pino RX da NCAP e o pino RX da TIM com o pino TX da NCAP, de acordo com a Tabela 3.3.

Tabela 3.3 - Pinos de ligação série TIM ↔ NCAP

	TIM	NCAP
TX	Porta 4.5	GPIO14 (pino 8)
RX	Porta 4.4	GPIO15 (pino 10)

Inicialmente, foi implementada a aquisição das componentes escalares  $a_x$ ,  $a_y$  e  $a_z$  da aceleração (sensor CMA3000 na versão 2.0 e o sensor LIS3DH na versão 2.2), da temperatura, com o sensor TMP36, e a tensão do potenciômetro da placa. Estas informações podem ser acedidas a partir dos comandos da norma e simultaneamente apresentados no display da placa, conforme mostra a Figura 3.4. A figura apresenta o diagrama de ligação da TIM ↔ NCAP e do sensor de temperatura à TIM.

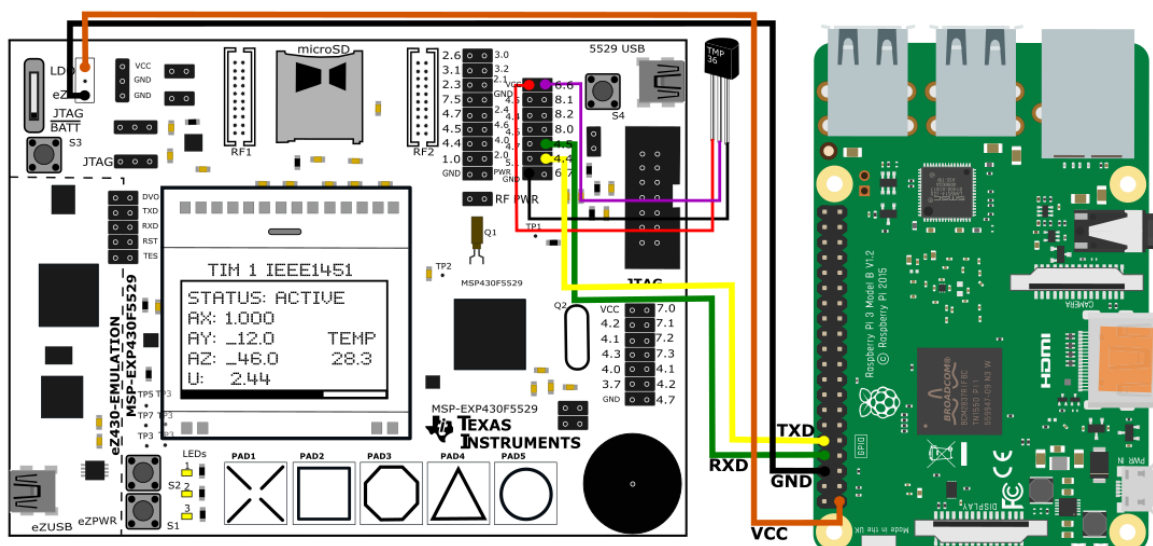


Figura 3.4 - TIM ↔ NCAP comunicação com fios

### 3.3.2. Software TIM com fios

O desenvolvimento do *software* para a TIM foi feito em simultâneo com o *software* desenvolvido para a NCAP, descrito em 3.5.2. O código da TIM foi escrito em C e permite que a TIM forneça respostas aos comandos, suportados pela norma, provenientes da NCAP.

A TIM inicialmente continha os seguintes transdutores:

- acelerómetro de 3 eixos,
- sensor de temperatura
- sensor de entrada analógica (potenciómetro).
- atuador de 5 leds.

No código inicialmente escrito foi implementada parte da norma IEEE1451, incluindo a leitura da Meta-TEDS.

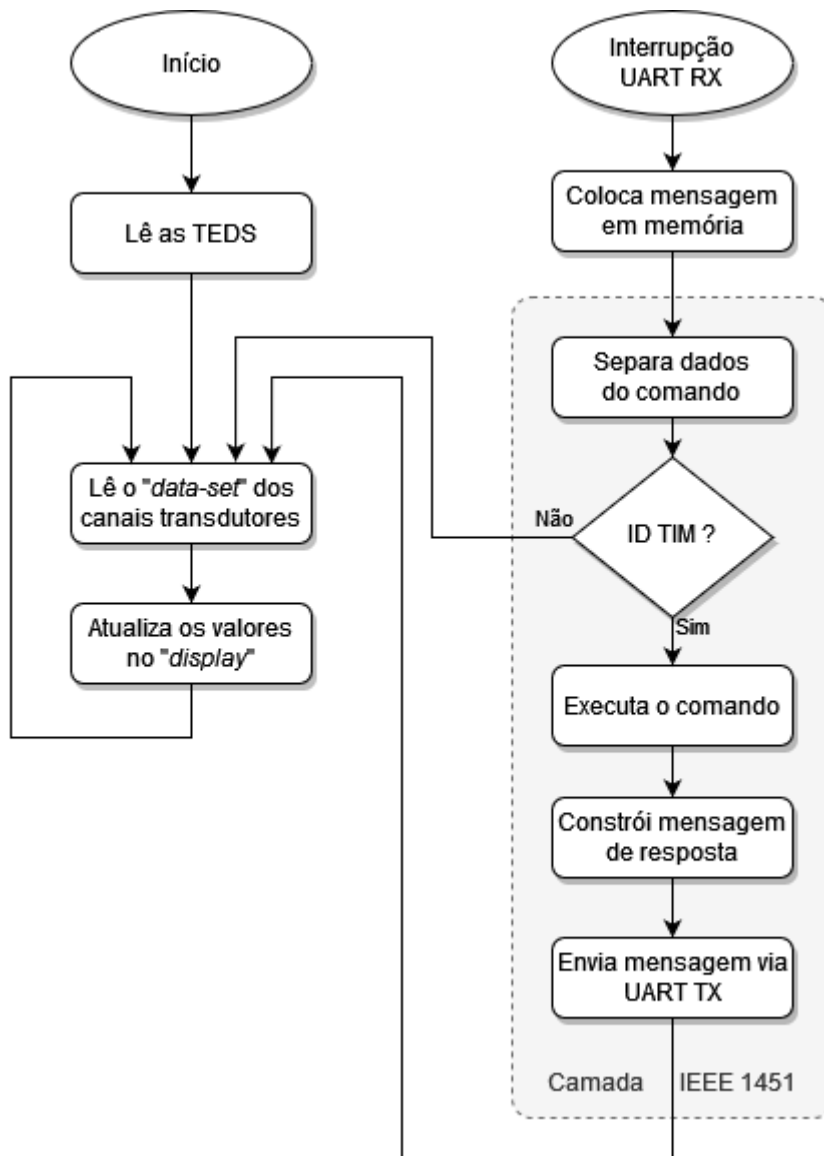


Figura 3.5 - Programa da TIM com fios

A TIM começa por ler as TEDS e configura-se de acordo as informações contidas nestas. Passando a um ciclo onde lê os valores de cada canal transdutor configurado como sensor e apresenta esses valores no display da placa de desenvolvimento. Este ciclo só é interrompido aquando de uma interrupção ao pino RX da UART quando a TIM recebe um comando da NCAP. Quando recebe um comando da NCAP a TIM separa a mensagem pelas diversas partes que a constitui e faz a verificação do endereço da TIM. Se este corresponder ao seu endereço processa o comando e envia a resposta à NCAP. A Figura 3.5 apresenta o fluxograma do programa da TIM.

### 3.4. Desenvolvimento da TIM sem Fios

#### 3.4.1. Hardware da TIM sem Fios

A implementação do sistema é realizada de acordo com a Figura 3.6 e constituída por: um *Raspberry Pi*, um adaptador UART ⇔ RF (descrito em 3.4.2) e três TIM.

Inicialmente a TIM sem fios foi construída utilizando a placa MSP EXP430F5529 à qual se retirou a ligação cablada e adicionou o adaptador UART ⇔ RF (Figura 3.6).

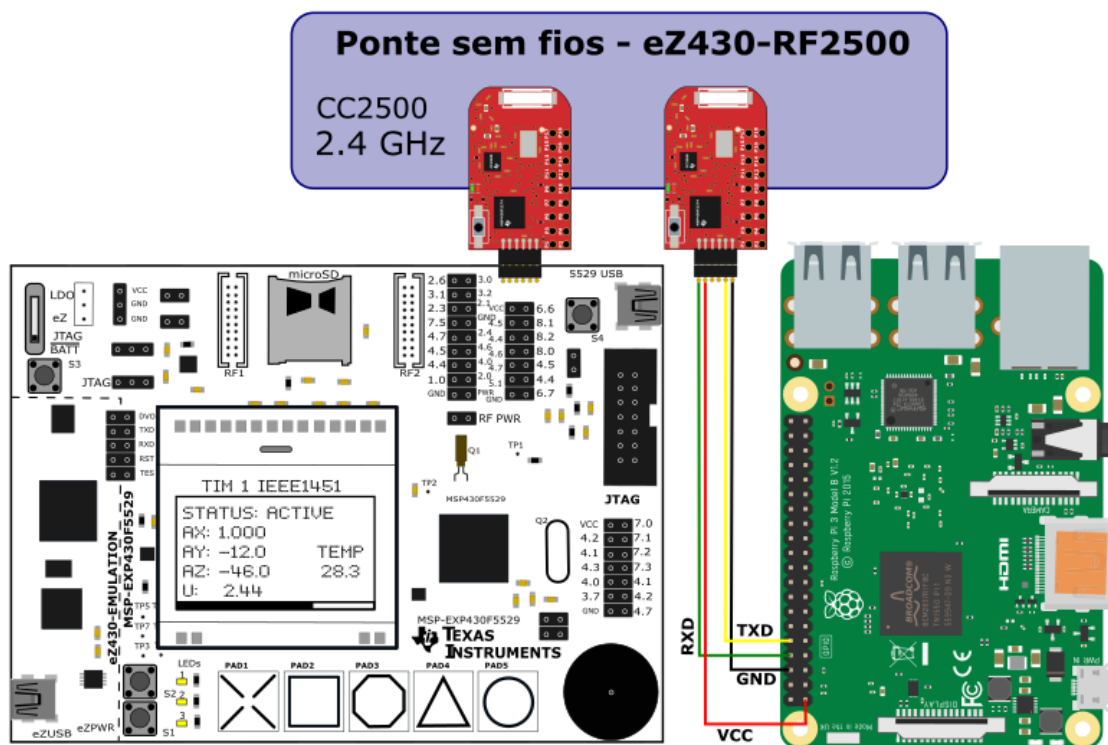


Figura 3.6 - TIM ⇔ NCAP comunicação sem fios

Posteriormente foi substituída apenas pela placa eZ430-RF2500.

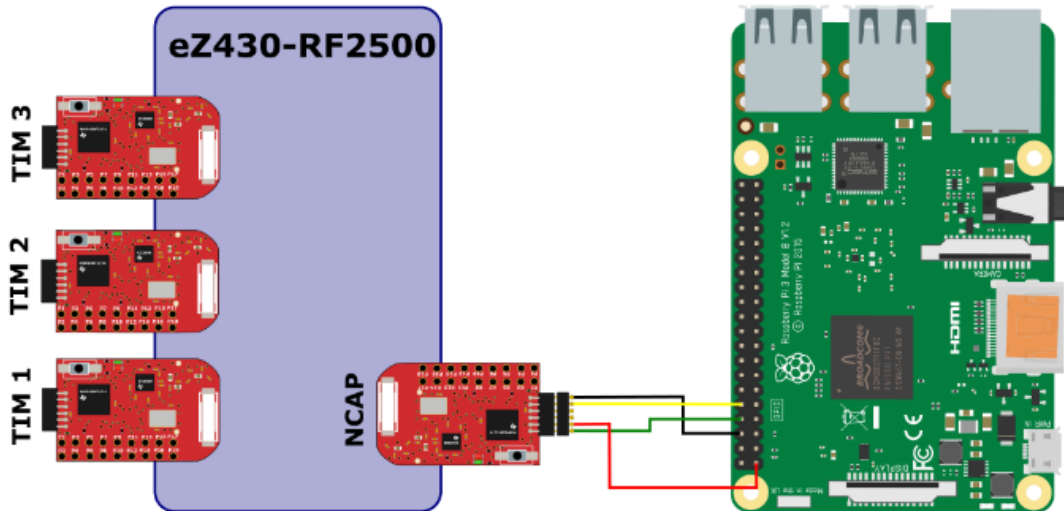


Figura 3.7 Rede de sensores sem fios

A interface entre a TIM e a NCAP é realizada pela utilização de uma placa eZ430-RF2500 programada como conversor UART  $\Leftrightarrow$  RÁDIO bidirecional. Esta permite substituir a ligação cablada, por uma ligação sem fios.

As TIM têm como principal função adquirir as grandezas físicas (caudal, nível de água e atuador) e enviar essa informação para a aplicação pelo intermédio da NCAP que realiza a interface entre a TIM e a aplicação.

Cada uma das TIM é desenvolvida em torno da placa de desenvolvimento EZ430-RF2500 da *Texas Instruments*. É uma ferramenta completa de desenvolvimento para sistemas sem fios, que inclui todo o *hardware* necessário para desenvolver este estudo. Utiliza a interface USB para se programar diretamente com *Code Composer Studio*.

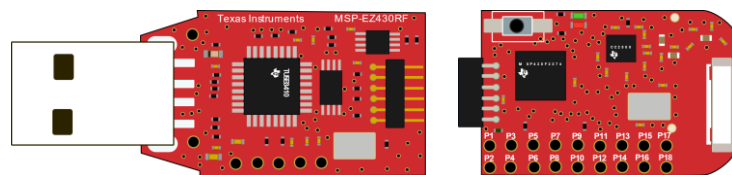


Figura 3.8 - EZ430-RF2500

A ferramenta inclui um emulador alimentado por USB, à direita, para programar e depurar o programa da placa e uma placa sem fios de 2,4 GHz, à esquerda, com o MCU MSP430F2274 de ultrabaixa potência.

O eZ430-RF2500 incorpora o microcontrolador MSP430F2274 de arquitetura RISC de 16-bit, que combina desempenho de 16 MIPS (*Milhões de Instruções por Segundo*) com

um ADC de 10 bits de resolução, com uma amostragem de 200 ksps e um rádio CC2500 projetado para aplicações sem fios de baixa potência [10].

### 3.4.2. Ponte Wireless

A ligação entre o *Raspberry Pi* e a placa eZ430-RF2500 faz-se pela UART nos pinos TX e RX. Como o *Raspberry Pi* e a placa eZ430-RF2500 são alimentados a uma tensão de 3,3V, podem ser ligados diretamente sem a necessidade de haver conversores de tensão.

Este permite que o processador esteja na maioria do tempo em *Sleep Mode* e sempre que ocorre uma interrupção da UART ou do SPI, acorda para realizar a transferência de informação. O programa desenvolvido está representado no fluxograma da Figura 3.9.

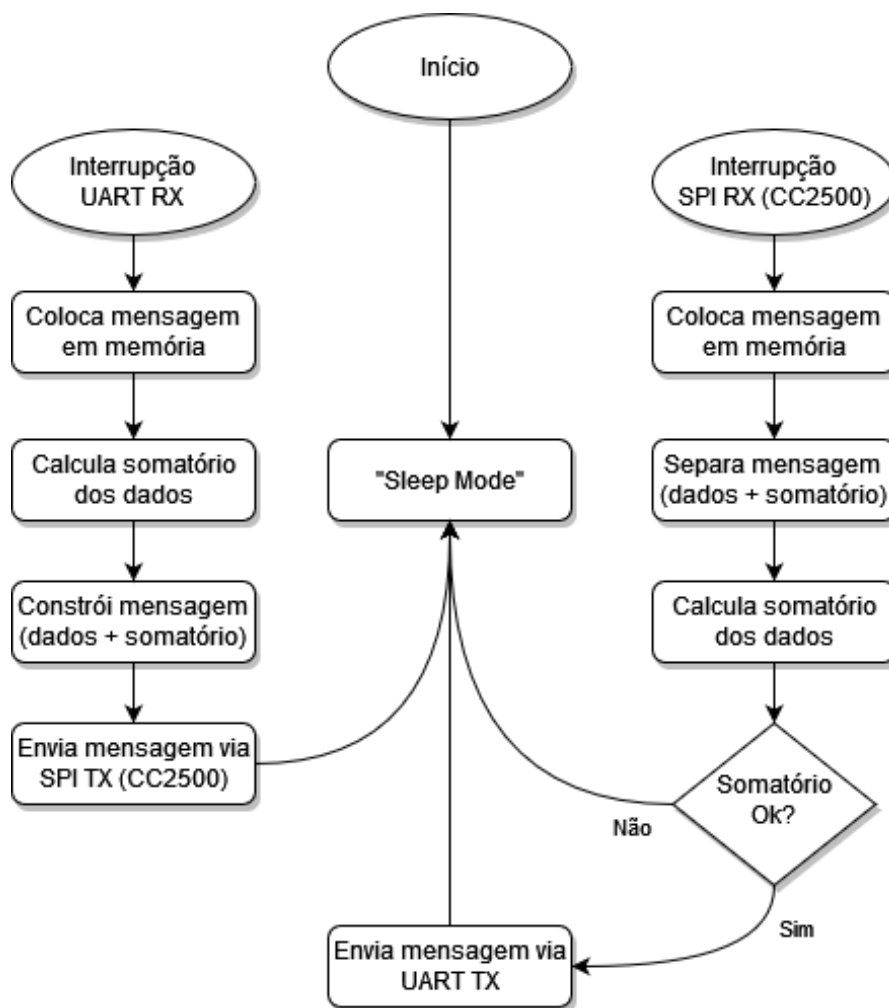


Figura 3.9 - Programa do adaptador UART ↔ RF

Sempre que recebe dados pela UART, o microcontrolador coloca esses dados num *buffer* e realiza o seu somatório. Em seguida constrói uma nova sequência de dados mais o somatório de dados e envia a mensagem para o CC2500 via SPI.

### Capítulo 3 - Plataformas de desenvolvimento para a norma IEEE1451

Sempre que recebe dados do rádio, CC2500, via SPI, o microcontrolador coloca os dados no *buffer* e separa a mensagem e o somatório. Calcula o somatório da mensagem e compara-o com o somatório que recebeu. Se estes forem iguais envia a mensagem pela UART.

A verificação de dados recebidos, com a adição da comparação do somatório de dados com o somatório de dados recebido, permite validar a correta recepção da comunicação.

Para configurar os rádios CC2500 utiliza-se a aplicação *SmartRF Studio 7* da *Texas Instruments*, Figura 3.10. Com esta aplicação foram testadas diferentes configurações por forma a obter a melhor performance e menor consumo energético dos rádios. (Alterando a potência de transmissão e o canal evitando colisões com os routers WI-FI que utilizam frequências próximas).

A taxa de transmissão dos dados entre a TIM e a ponte UART/RF é de 9600 Baud, enquanto a taxa entre os dois rádios é de 249,939 kBaud.

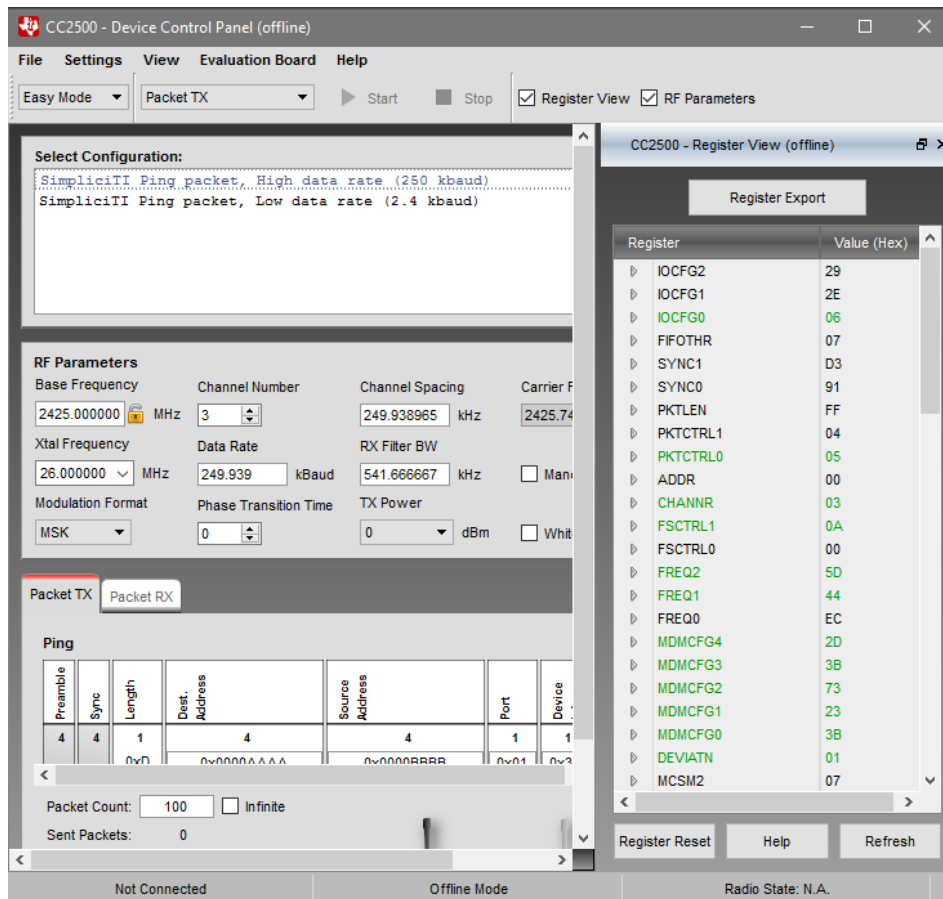


Figura 3.10 - Configuração CC2500 com *SmartRF Studio 7*

Este programa gera um conjunto de registos, à direita na Figura 3.10, que são introduzidos no código de programação do microcontrolador no *Code Composer Studio*. É o microcontrolador que configura o rádio, CC2500, aquando do arranque do programa.

Os principais parâmetros de configuração:

- Frequência Base: 2433,000000 MHz
- Modulação: MSK (*Minimum Shift Keying*)
- Velocidade de Transmissão: 249,939 kBaud
- Espaçamento entre canais: 199,951172 kHz
- Filtro RX BW: 541,666667 kHz
- Potência TX: 1 dBm

Para se utilizar mais do que um sistema é possível utilizar até 256 canais com um espaçamento de 199,95 kHz entre canais. Como o sistema foi instalado no laboratório da Universidade, onde existem muitas Pontes de Acesso Wi-Fi que funcionam em 2,4 GHz, foi necessário analisar o espectro rádio, para encontrar uma frequência livre, de forma a evitar as interferências do Wi-Fi.

## 3.5. Desenvolvimento NCAP

A função de NCAP é atribuída a um *Raspberry pi 3*, um pequeno computador ARM já descrito na Tabela 3.1 no capítulo 3.2.

### 3.5.1. Hardware NCAP

A comunicação entre a NCAP e a TIM é realizada via UART. O sistema operativo do *Raspberry Pi 3*, utiliza por defeito a UART como “consola” login através do serviço *getty*. Para se poder utilizar como interface NCAP ⇔ TIM é necessário desabilitar esta funcionalidade. A configuração [8] para o *Raspberry Pi 3* (que difere de outras versões) refere-se à porta “/dev/ttySo” através dos comandos:

```
sudo systemctl stop serial-getty@ttySo.service
```

```
sudo systemctl disable serial-getty@ttySo.service
```

Assim, é necessário realizar a reinicialização da máquina. É ainda necessário remover o texto “console=serial0,115200” da única linha que está no ficheiro “/boot/cmdline.txt”,

guardar a alteração e voltar a reiniciar o Raspberry. O comando para editar o ficheiro texto:

```
sudo nano /boot/cmdline.txt
```

O pino 8 da NCAP, corresponde ao TX da UART, que liga à porta 4.4 da TIM e corresponde ao RX da UART. O pino 10 da NCAP, corresponde ao RX da UART, liga à porta 4.5 da TIM e corresponde ao TX da UART. Ambas as placas funcionam com uma tensão de 3,3 V pelo que não há necessidade de utilizar conversores de tensão entre os pinos. O diagrama de ligação está apresentado, anteriormente, na Figura 3.4.

### 3.5.2. Software NCAP

O código NCAP foi escrito em *Python* e pode fazer interface entre a TIM e a aplicação. Com o código desenvolvido, a NCAP pode criar uma mensagem que a TIM possa entender e receber a resposta aos comandos vindos da aplicação do utilizador.

O desenvolvimento de uma NCAP fica fora do âmbito do trabalho deste estudo, pelo que se desenvolveu uma simples aplicação que sirva de intermediária entre a aplicação e a TIM.

A NCAP é chamada pela aplicação quando esta necessita de interagir com a TIM. A Figura 3.11 apresenta a organização da NCAP.

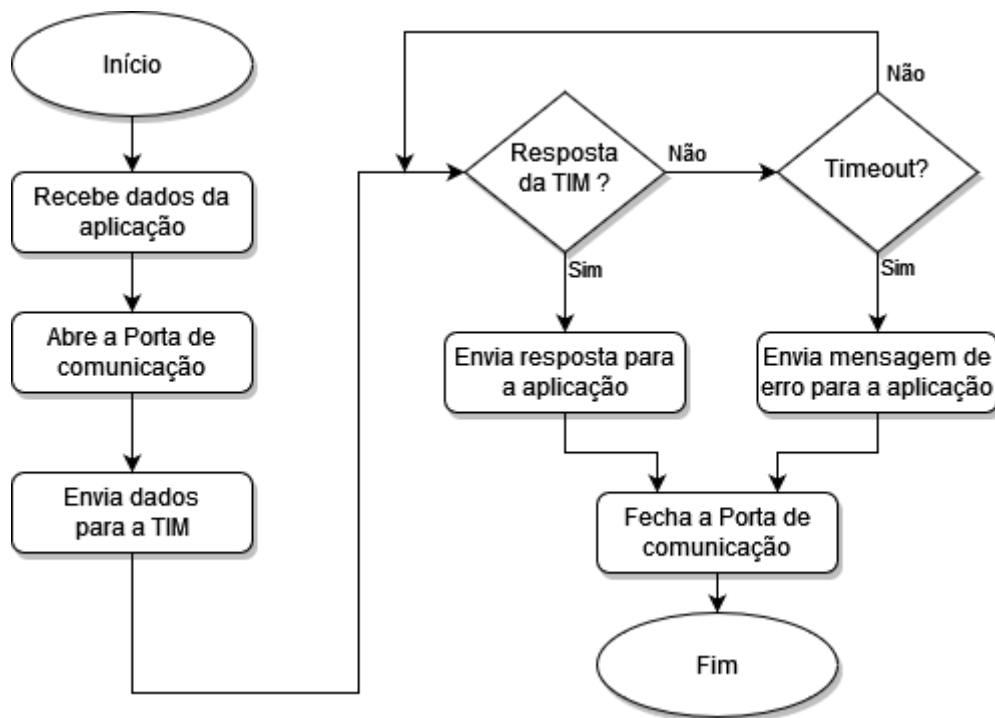


Figura 3.11 - Programa NCAP

Esta começa por receber os comandos provenientes da aplicação, abre a porta de comunicação e envia o comando para a TIM. Aguarda a resposta durante um determinado intervalo de tempo. Caso receba resposta da TIM, envia-a para a aplicação, caso não receba a resposta envia para a aplicação uma mensagem de erro. Por fim fecha a porta de comunicação e termina.

### **3.6. Plataforma de Desenvolvimento IEEE1451**

Uma plataforma de desenvolvimento pressupõe um conjunto de programas que permitam o desenvolvimento de um projeto, mas também que possuam ferramentas que permitam a verificação do correto funcionamento do sistema desenvolvido. No caso em estudo são necessárias as ferramentas:

- *TIM TEDS Editor* – permitindo a configuração rápida de TEDS;
- *TIM Project Editor* – permitindo a programação por blocos;
- *TIM Validation* – permitindo a verificação do correto funcionamento da TIM;
- *TIM Manager* – permitindo gerenciar todas as TIM existentes no sistema;
- Aplicação online;
- Tutoriais.

No capítulo 2.8.7 foi abordada a forma como estão organizadas as TEDS e mostra a complexidade destas, para isto foi criado *TIM TEDS Editor*.

A plataforma é orientada para a utilização de diferentes utilizadores. Foram considerados dois tipos de utilizadores: administrador e utilizador. Dependendo do tipo de utilizador que efetua o *login*, as ferramentas disponíveis adaptam-se dinamicamente ao utilizador. O administrador possui um conjunto de ferramentas extra que lhe permitem adicionar ou apagar novos utilizadores. Todos os utilizadores podem escolher um de cinco temas disponíveis e alterar a palavra-passe de acesso à plataforma.

### 3.6.1. TIM TEDS Editor

O TIM TEDS Editor permite compilar, de forma simples e intuitiva, todos os parâmetros que integram qualquer TEDS. Esta ferramenta é constituída inicialmente por uma página de gestão de TEDS, Figura 3.12, onde é possível:

- Visualizar a lista de Configurações TEDS, anteriormente adicionadas, com a informação: nome, descrição, data de adição, autor e opções.
- Adicionar um novo conjunto de TEDS.
- Gerir TEDS, editar as existentes, renomear ou apagá-las.
- Guardar as TEDS na plataforma.

A página inicial desta ferramenta, inclui vários pormenores que permitem rentabilizar o tempo de configuração de uma TEDS.

The screenshot shows the TIM TEDS Editor interface. At the top, there is a navigation bar with 'IML', 'Development Tools', and 'Management' menus, and a user profile for 'joao.luis.pereira@ubi.pt'. Below the navigation bar is the IML logo and the title 'TIM TEDS Editor'. The main content area is titled 'TEDS List' and contains a table of TEDS configurations. A 'New TEDS' button with a plus icon is located below the table. The table has columns for Name, Description, Created, Autor, and Options. The Options column contains edit, delete, and refresh icons. Below the table is a pagination control showing page 1 of 1 and a dropdown for '10 - Items per page'. At the bottom, there is a footer with 'IML© - 2022'.

Name	Description ↓	Created ↓	Autor	Options
tim teds 1	Example from standard (Appendix O)	2019-04-24 10:50:09	demo	[edit] [refresh] [delete]
tim teds 2	Example for development	2019-04-25 16:30:01	demo	[edit] [refresh] [delete]
tim teds 3	Analog Gauge Reader	2019-06-10 20:49:34	demo	[edit] [refresh] [delete]
My TEDS TIM	My TEDS TIM	2021-07-28 16:50:51	João Pereira	[edit] [refresh] [delete]

Figura 3.12 - Gerenciador de TEDS

Ao clicar no botão de apagar (ícone x) irá surgir um *modal* perguntando a confirmação para a eliminação daquele conjunto de configurações TEDS. Ao clicar no botão editar

nome (ícone asterisco) surge um *modal* onde é possível editar o nome e a descrição do conjunto de TEDS. É possível ordenar a lista por ordem alfabética, da primeira letra da descrição, e por ordem da data de criação do conjunto de TEDS; selecionar o número máximo de TEDS por página e avançar ou retroceder na lista através do paginador. Ao clicar no botão adicionar (ícone de mais) irá surgir um *modal* onde é possível escrever o nome e a descrição do conjunto de configurações TEDS, adicionando esse conjunto de configurações TEDS ao sistema. Ao clicar no botão editar (ícone de lápis) será redirecionado para a página onde é possível editar todas as configurações de todas as TEDS. A Figura 3.13 apresenta a página de configuração da TEDS.

The screenshot displays the TIM TEDS Editor interface. At the top, there is a navigation bar with 'IML', 'Development Tools', and 'Management' menus, and a user profile 'joao.luis.pereira@ubi.pt'. Below the navigation bar is the IML logo and the title 'TIM TEDS Editor'. There are two buttons: 'Download TEDS TXT' and 'Download TEDS XML'. The main content area shows the configuration for a TEDS named 'tim teds 1'. It features several sections: 'Manufacture Globally Unique Identifier' with fields for TEDS Identification Header (1), Latitude (North, 38.9374), Longitude (East, 1.41723), Manufacturing Year (20), and Manufacturing Time (204012); 'Timing-related information' with fields for Operational time-out (0.3), Slow access time-out (0.1), and Self-Test Time (0.0); and 'Output TEDS' showing a list of hexadecimal strings. A dropdown menu is open, listing various TEDS types, with '16 - Energy TEDS' highlighted in green. A tooltip for 'Manufacture Time' explains the formula: 'Manufacture Time = Day of the Year × 1000 + Sequence Number'. A 'Save' button is located at the bottom left.

Figura 3.13 - Editor de TEDS

O Editor de TEDS está organizado por uma barra superior de tabulação. Cada tabulação corresponde a cada uma das TEDS previstas na norma, mais uma para a TEDS de Energia que será uma proposta para adição à norma e descrita em 4.4.

Os campos e subcampos de configuração da TEDS estão organizados da mesma forma que são apresentados na norma IEEE1451.0 [11]. Por forma a minimizar o tempo de configuração de um novo conjunto de TEDS foram automatizados todos os campos dinamicamente. Por exemplo para se preencher os quatro campos de localização, basta apenas clicar no ícone do globo e surge um *modal* com um mapa onde se pode escolher a localização pretendida e automaticamente serão preenchidos esses quatro campos. Outro exemplo é o preenchimento do *Manufacture Time* onde aparece semipreenchido com a data do dia em que se realiza a nova configuração. Os botões com os ícones mais e menos adicionam ou subtraem uma unidade ao valor pré-preenchido. Da mesma forma o campo de “número de canais” aparece automaticamente preenchido com um canal, para aumentar basta clicar no ícone mais. Configurando o este campo para, por exemplo seis canais, a plataforma irá criar de forma dinâmica seis *TransducerChannel TEDS* e seis *User’s Transducer Name TEDS*, que passarão a estar disponíveis na respetiva tabulação.

Todas as informações introduzidas pelo utilizador são codificadas e a qualquer momento, é possível visualizar o resultado dessa codificação na secção *Output TEDS*.

À frente de cada campo editável, existe um ícone de ajuda que apresenta informações adicionais sobre o que se deve inserir naquele campo específico. Quando o utilizador passa o ponteiro do rato sobre o ícone de ajuda, um *pop-up*, com uma breve descrição do campo, é exibido para ajudar o utilizador a preencher esse campo.

O conjunto das configurações TEDS ficam guardadas na base de dados, para serem posteriormente utilizadas na ferramenta de TIM Project Editor descrita em 3.6.2. Mas é também possível o *download* das mesmas em formato TXT ou XML, clicando nos respetivos botões localizados na parte superior da página. No Apêndice H apresenta-se o exemplo do *download* do ficheiro em formato TXT das TEDS. Neste caso é possível adicioná-las ao código da TIM escrito em linguagem C no programa *Code Composer Studio* ou outro, dependendo do microcontrolador que se pretende utilizar.

Com esta ferramenta para além da rápida configuração das TEDS, permite o entendimento da norma para quem a começa a utilizar.

Com todas as TEDS definidas e guardadas é criado um, novo, projeto na ferramenta de desenvolvimento *TIM Project Editor* onde é atribuído o conjunto de TEDS anteriormente definidas ao projeto atual.

### 3.6.2. TIM Project Editor

Com a configuração das TEDS concluída com a ferramenta descrita em 3.6.1, pode agora realizar-se o *download* das TEDS e introduzir diretamente no código de programação do *Code Composer Studio*, ou então pode ser utilizada a ferramenta online *TIM Project Editor* que permite escolher a placa de desenvolvimento e associar-lhe a TEDS desejada, para gerar todo o código da TIM.

A página inicial desta ferramenta apresenta uma lista dos projetos anteriormente criados, Figura 3.14, com a informação:

- nome do projeto;
- descrição do projeto;
- endereço da TIM;
- plataforma de *hardware* utilizado;
- conjunto de configuração TEDS que está implementado;
- data e hora em que foi criado.

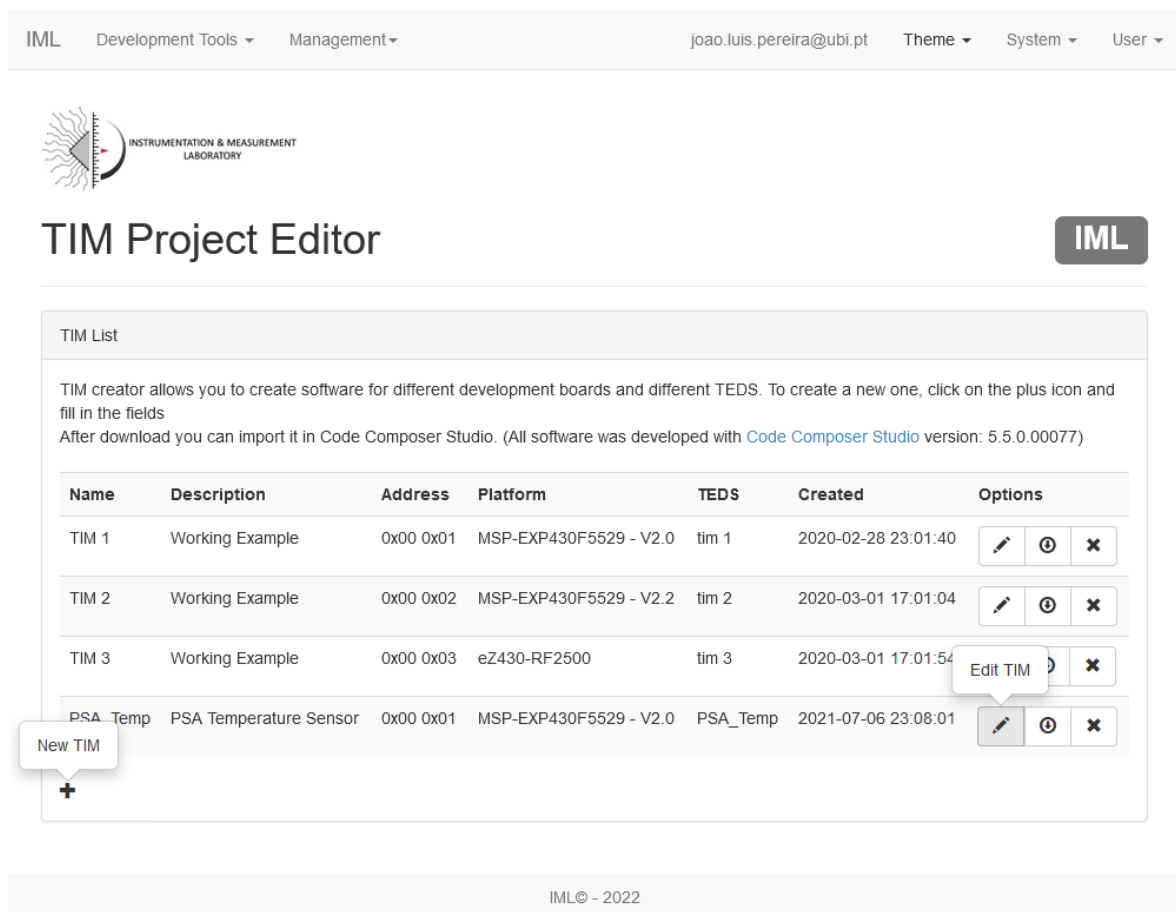


Figura 3.14 - Editor de projetos TIM

Apresenta quatro opções:

- apagar um projeto clicando no ícone x;
- realizar o download do código para programar a TIM (*Code Composer Studio*);
- editar os parâmetros da TIM clicando no ícone do lápis;
- adicionar um novo projeto clicando no ícone mais.

Ao clicar no ícone mais, surge um *modal* para adicionar um novo projeto de TIM, Figura 3.15, onde se pode escolher:

- o nome do projeto;
- a descrição da TIM;
- o endereço da TIM;
- a TEDS que configura a TIM (configurado anteriormente no TIM TEDS Editor);
- a placa onde será programada.

No momento da escrita deste documento estão implementadas três placas de desenvolvimento: MSP-EXP430F5529 v2.0, MSP-EXP430F5529 v2.2 e EZ430-RF2500.

The image shows a 'New TIM' modal window. It has a title bar with the text 'New TIM' and a close button (X). The form contains the following fields:

- Name:** A text input field with the placeholder text 'Enter TIM's Name'.
- Description:** A text input field with the placeholder text 'Enter Description'.
- Destination ID1 (Low Byte):** A text input field with the placeholder text 'Enter Destination ID 1 (Low Byte)'.
- Destination ID2 (High Byte):** A text input field with the placeholder text 'Enter Destination ID 2 (High Byte)'.
- TEDS List:** A dropdown menu with 'tim teds 1' selected.
- Platform:** A dropdown menu with 'MSP-EXP430F5529 - V2.0' selected.

At the bottom right of the modal, there are two buttons: a blue 'Submit' button and a white 'Close' button.

Figura 3.15 - Adicionar nova TIM

Ao clicar no botão editar (ícone lápis) surge uma nova ferramenta de programação em blocos. No menu de “BLOCOS” existem sete blocos distintos: *Board*; *Address*; *TEDS*; *Sensor*; *Event Sensor*; *Actuator* e *Port*. Quando se inicia surgem alguns blocos já preenchidos com informações contidas nas TEDS, os restantes podem ser adicionados a partir do menu BLOCOS arrastando o bloco para a área de desenho.

Podemos ainda visualizar, em forma de blocos, ajustar os parâmetros em falta tais como as portas onde se liga o sensor. Em qualquer momento é possível verificar se o código possui erros, clicando no botão *Verify*, guardar o código de blocos na base de dados ou exportar no formato de XML.

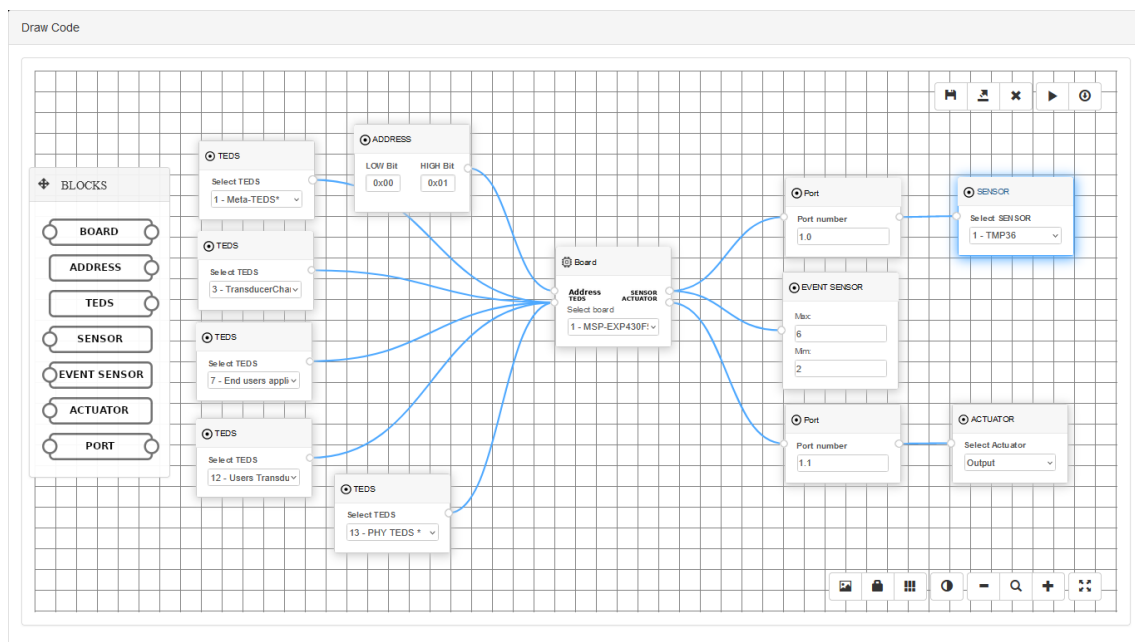


Figura 3.16 - Editor de Código em Blocos

Opções:

- Gravar projeto em blocos na base de dados
- *Download* do projeto em blocos em formato XML
- Limpar blocos
- Verificar erros
- *Download* de código (*Code Composer Studio*)
- *Download* de imagem;
- Ligar/desligar grelha de fundo;
- Bloquear edição;
- Alterar cor de fundo;

- Aumentar/diminuir tamanho dos blocos;
- Colocar em ecrã completo.

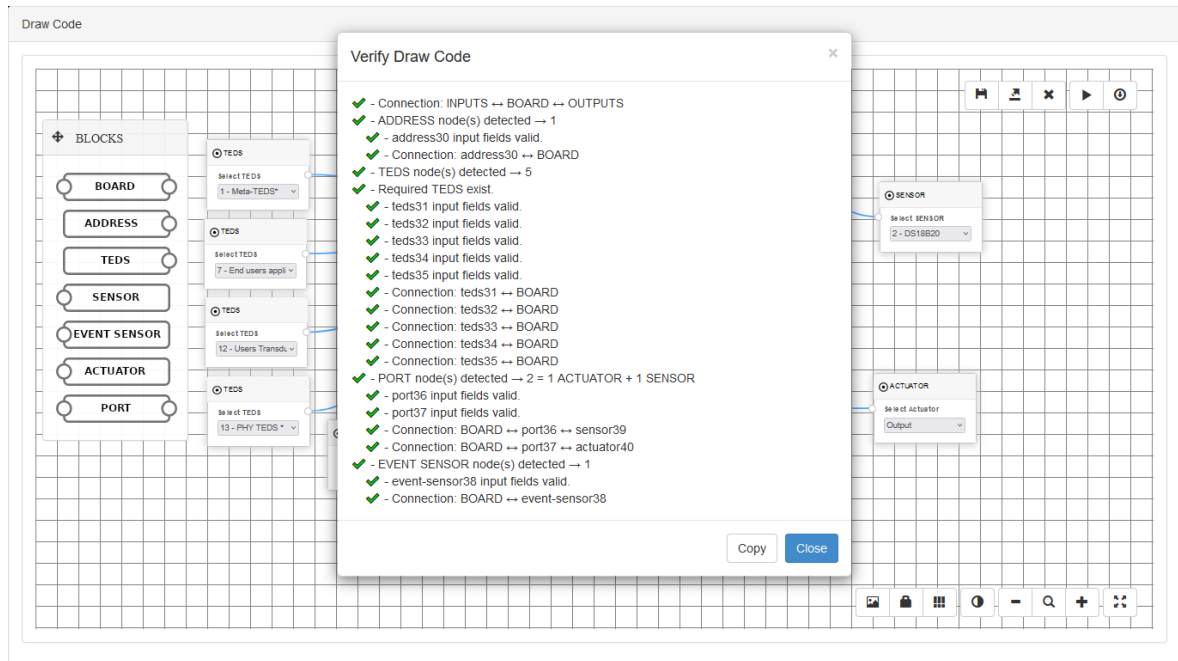


Figura 3.17 - Verificador de Código em Blocos

No final de todos os blocos configurados é possível fazer o *download* do código, para o *Code Composer Studio* e programar a placa escolhida, clicando no botão *Download Code*. Com a placa programada é hora de verificar se tudo está a funcionar corretamente, podendo para isso utilizar a ferramenta *TIM Validation*.

### 3.6.3. Validador TIM

Para completar o conjunto de ferramentas de desenvolvimento, foi criado um validador de TIM, com o objetivo de verificar se uma TIM está a funcionar de acordo com a norma. Para o efeito foram desenvolvidas um conjunto de ferramentas de validação que permitem a verificação do correto funcionamento de uma TIM de acordo com a norma. Com este validador, um utilizador envia comandos através da página de validação. Todos os comandos são suportados, mas o mais importante é permitir enviar comandos de leitura das TEDS. A ferramenta depois de verificar se as mesmas respeitam a tipologia TLV, separa todos os seus parâmetros e descodifica os seus valores, sendo ainda possível adicionar esses valores numa base de dados.

Na Figura 3.18 apresenta-se o diagrama e a Figura 7 apresenta a fotografia, do *hardware* utilizado. Este é constituído pela TIM desenvolvida anteriormente onde se intercalou um conversor UART $\leftrightarrow$ USB permitindo a interface com o computador. O conversor utilizado

foi o FT232RL, que cria uma porta COM virtual no computador a que se liga. Este conversor é muito versátil pois possui drivers para Windows, Mac e Linux.

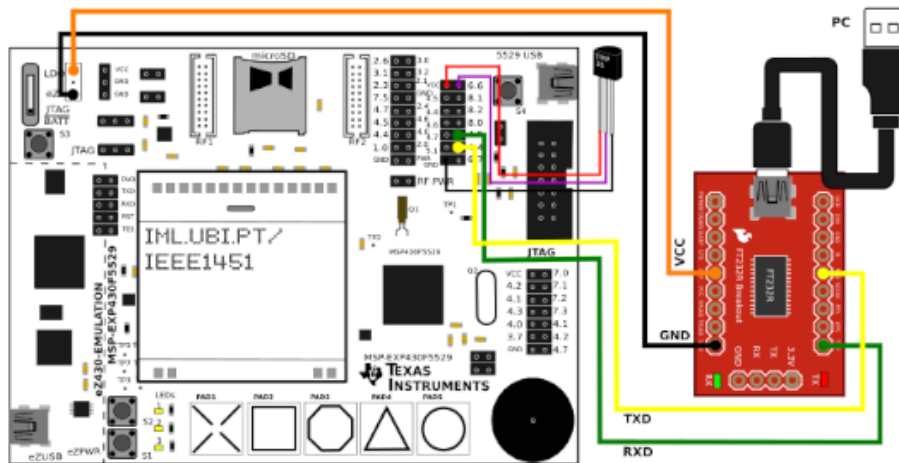


Figura 3.18 - Hardware de validação de TIM IEEE1451

Como um dos objetivos da ferramenta é poder ser utilizada em qualquer lugar e por qualquer pessoa, o interface gráfico de utilização está acessível através do sítio <http://iml.ubi.pt/ieee1451> que corre num browser de internet. Como os browsers atuais apenas funcionam como caixas de areia, não tendo (por motivos de segurança) a capacidade de aceder diretamente aos portos série dos computadores, houve a necessidade de desenvolver um *plugin* de interface que realiza a conexão entre a Página ↔ PC ↔ TIM. Este *plugin* foi escrito em linguagem *Python* e utiliza duas bibliotecas importantes: *pyserial* e *tornado* [9]. A primeira dá acesso às portas séries disponíveis no computador e a segunda cria um *websocket http* que permite a ligação com a página.

A Figura 3.19, apresenta a estrutura interna do *plugin*. Este corre no computador do utilizador e cria um servidor web na porta 8010, aguarda comandos provenientes da página, processa-os e devolve a resposta para a página. A página envia o pedido ao *plugin* utilizando a tecnologia Ajax (o que permite enviar dados sem a necessidade de recarregar a página). O *plugin* recebe o comando e envia para a TIM, esta processa o pedido e envia para o *plugin*, o *plugin* descodifica a resposta da TIM para depois enviar o resultado para a página.

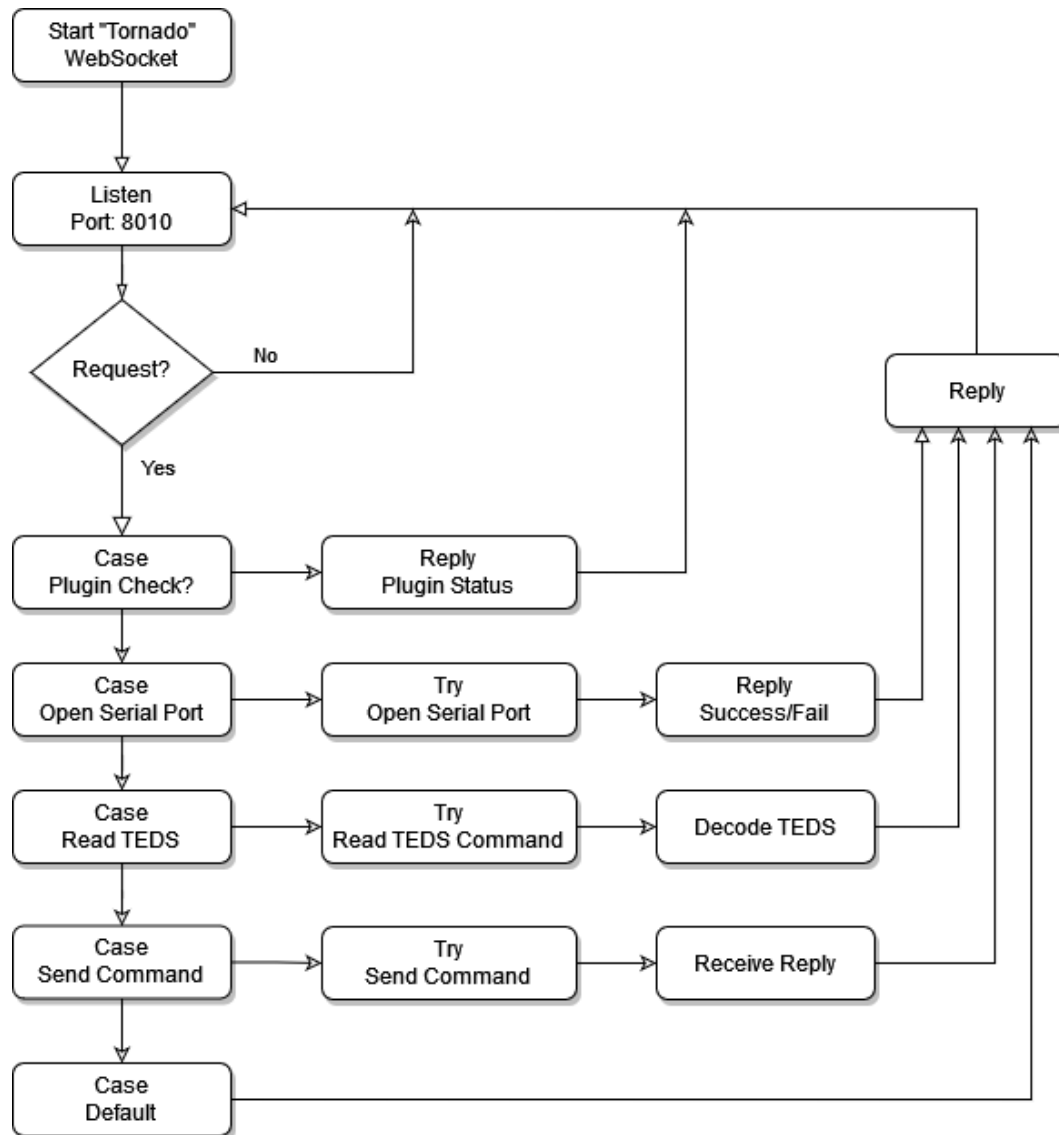


Figura 3.19 - Estrutura interna do *plugin*

Sempre que o utilizador realiza um pedido ao *plugin*, via página, esta envia os seguintes dados em formato JSON (*JavaScript Object Notation*):

- Opções: *plugin check*, *open serial port*; *read TEDS*; *send commands*;
- Velocidade de comunicação *baudrate*;
- Porta COM;
- Endereço TIM (1º octeto);
- Endereço TIM (2º octeto);
- Classe de comando;
- Função de comando;
- Número de identificação da TEDS.

Quando o *plugin* não está a correr, ou quando a TIM não está ligada ao computador, ou mesmo quando a TIM está com problemas de comunicação, são apresentados, na página, erros que indicam qual poderá ser a causa.

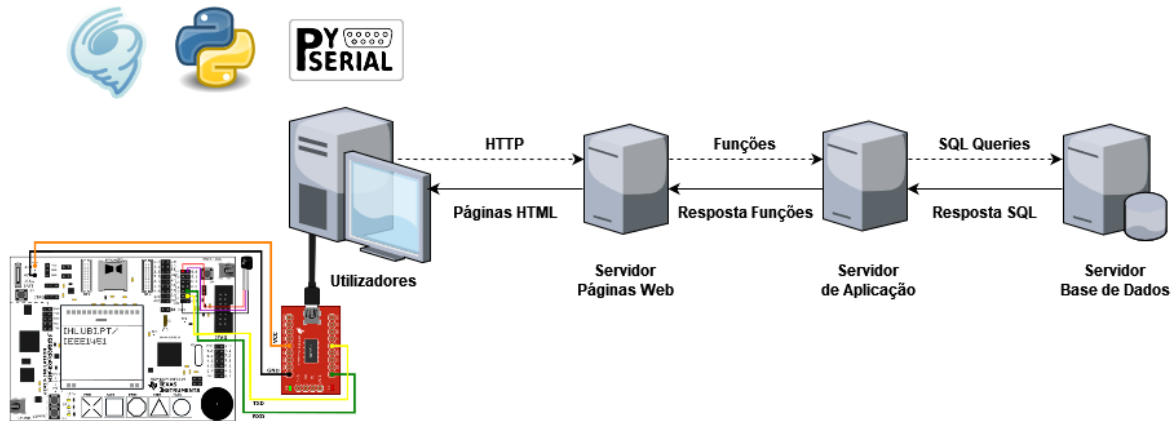


Figura 3.20 - Interface TIM ⇔ Validador

A página foi escrita em HTML5, PHP, Javascript, Ajax e MySql para guardar os dados numa base de dados. Na Figura 3.21, apresenta-se o interface gráfico da ferramenta *TIM Validation* que é constituída pelas secções:

1. *TIM List*;
2. *Interface Configuration*;
3. *MQTT Configuration*;
4. *Plugin Configuration*;
5. *Serial Port Configuration*;
6. *TIM TEDS Validation Commands*;
7. *TIM Validations Commands*.

Na *TIM List* são apresentadas todas as TIM guardadas no sistema, indicando o endereço, o nome, a data de quando foi guardada, a interface na qual se encontra ligada e um menu onde se pode aceder a outras ferramentas: comandos, edição de TEDS, aplicação e apagar a TIM do sistema.

No *Interface Configuration* existe a possibilidade de escolher entre uma ligação por MQTT ou por SÉRIE.

No *MQTT Configuration* indica-se o endereço do servidor de MQTT e o respetivo porto de comunicação.

No *Serial Port Configuration* é possível escolher a porta série do computador onde se encontra ligada a TIM, (qualquer que seja o sistema operativo), bem como a velocidade de transmissão de dados.

Figura 3.21 - Validador TIM IEEE1451

Na secção *Plugin Configuration* é necessário escrever o endereço da máquina onde o *plugin* está a correr, que é normalmente o do próprio computador “127.0.0.1”, bem como o porto de comunicação “8010”. Este plugin também poderá ser executado noutra computador e, neste caso, o endereço deverá ser o do computador onde está a ser executado. Esta funcionalidade torna o sistema muito versátil, num ambiente colaborativo

de ensino/aprendizagem da norma IEEE1451, pois é possível partilhar *hardware* de desenvolvimento.

A secção *TIM TEDS Validation Commands* está dedicada especificamente para a verificação das TEDS. Deve-se indicar o endereço da TIM e seleccionar a TEDS que se pretende ler. Ao clicar no botão *Read* é possível verificar quais as TEDS implementadas na TIM, verificar se a integridade e formato da TEDS está correto, separar as TEDS no formato TLV, decodificar os valores dos diferentes parâmetros, apresentar toda essa informação e por fim guardar a informação na base de dados que será apresentada na *TIM List*. O algoritmo que permite a realização desta decodificação está descrito na Figura 3.22.

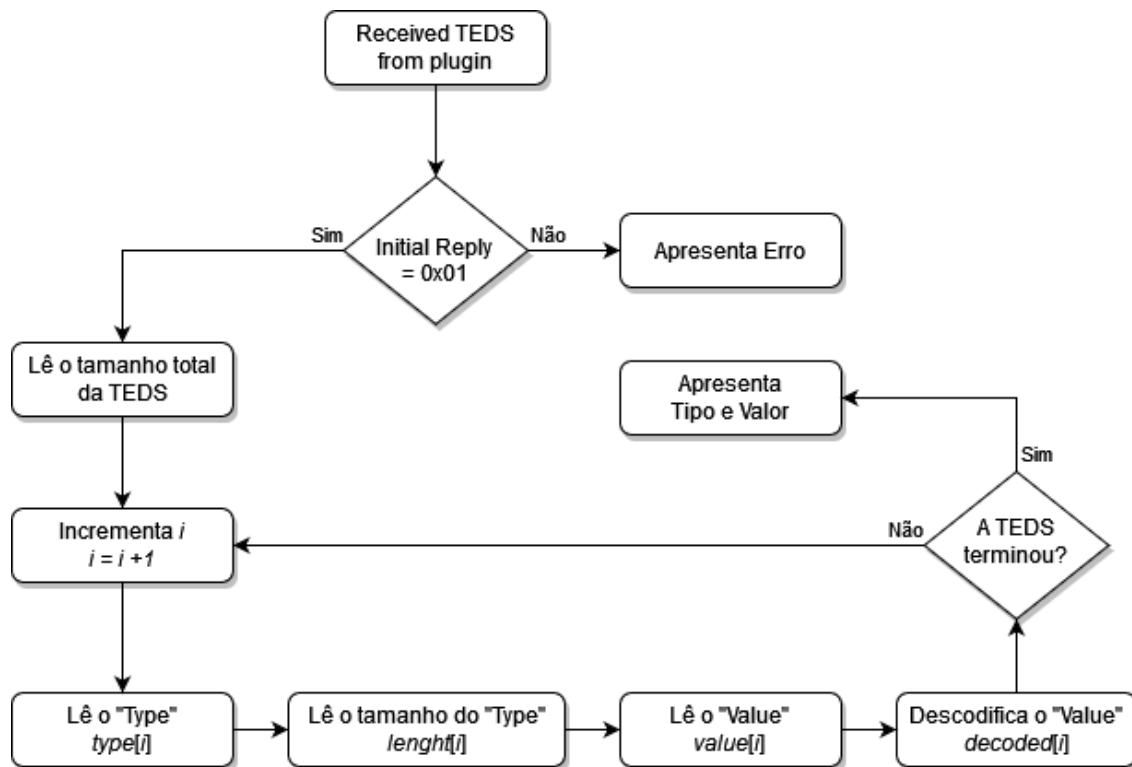


Figura 3.22 - Descodificação TEDS

No *TIM Validations Commands* é possível enviar todos os comandos suportados pela norma IEEE1451, seleccionando a função e a classe de comandos. Com esta ferramenta é possível verificar a implementação existente na TIM. Em caso de resposta ao comando enviado, esta será apresentada em formato hexadecimal e devidamente convertida.

Apresenta-se ainda um tutorial onde se pode realizar o *download* do *plugin* e onde se explica como colocar o *plugin* a funcionar em sistemas Windows, Mac ou Linux.

A Figura 3.23, apresenta o resultado da leitura e decodificação da Meta-TEDS proveniente de uma TIM, a partir do *plugin* que corre no computador, Figura 3.24.

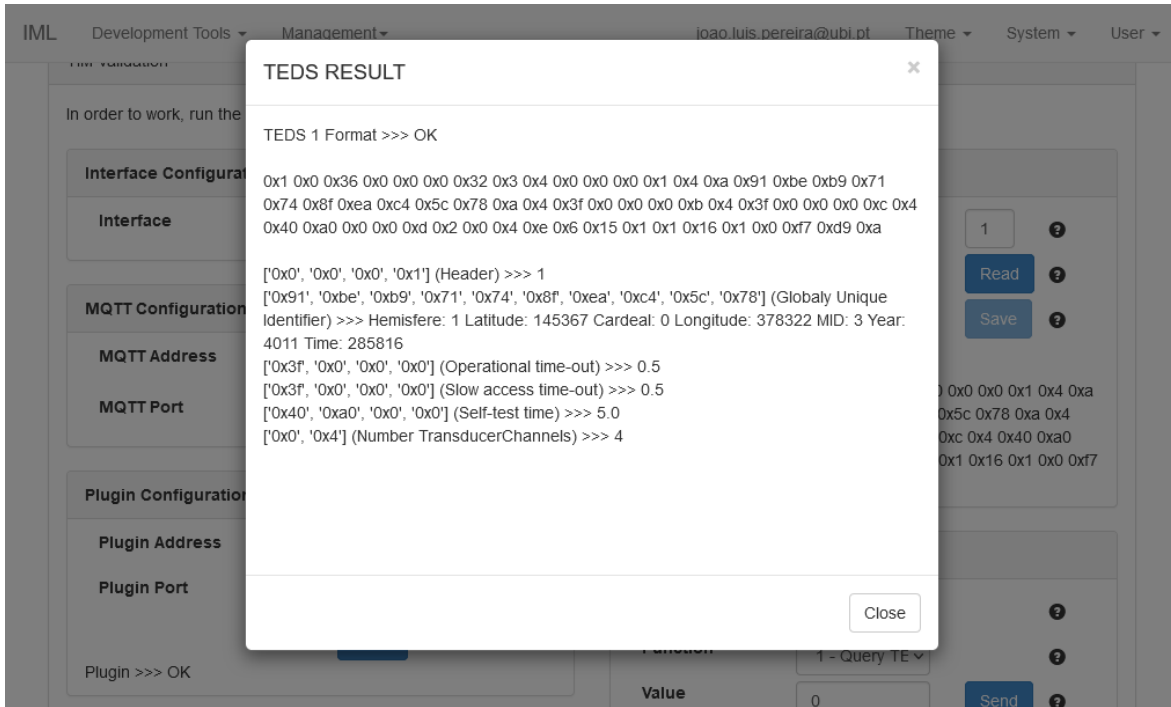


Figura 3.23 - Leitura e descodificação das TEDS



Figura 3.24 - *Plugin* a correr

Esta ferramenta, verifica a existência de uma TIM ligada ao PC, lê as TEDS que essa TIM possui, verifica a integridade e formato dessas TEDS, descodifica cada TEDS decompondo cada uma delas em formato TLV para depois descodificar o valor de cada campo e posteriormente guardar a TIM no sistema. Pode verificar quais as funcionalidades da

norma que estão implementadas e se estão implementadas de acordo com esta. Com esta ferramenta foi possível adicionar mais funcionalidades à TIM e verificar o seu correto funcionamento.

Todas estas ferramentas têm a grande vantagem de poderem ser utilizadas em qualquer sistema operativo e todo o *software* encontra-se disponível para *download* em forma de código livre.

Foram ainda elaborados vídeos tutoriais para explicar a utilização de todas as ferramentas de desenvolvimento disponíveis, ainda que exista uma pequena ajuda em cada um dos campos a preencher. ☞

### 3.6.4. TIM Manager

Depois da TIM programada e validada é possível adicioná-la ao sistema no *TIM Manager*. O *TIM Manager* foi desenvolvido para gerir e verificar as TIM. Esta ferramenta apresenta uma lista das TIM que se encontram guardadas no sistema, com a informação do endereço, descrição, a data de quando foi adicionada e a interface que utiliza, que poderá ser MQTT ou SÉRIE.

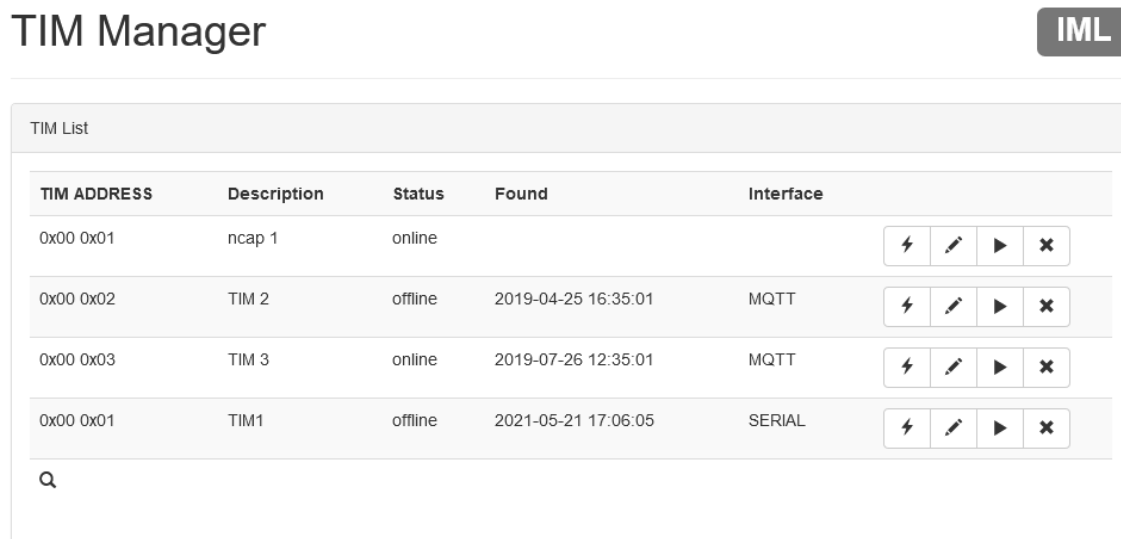


Figura 3.25 - Gestor de TIM

Essa ferramenta permite ao utilizador procurar uma nova TIM, ler todas as TEDS e colocar todas as TEDS nas bases de dados correspondentes à TIM, depois decodificar as TEDS e armazenar as informações nas bases de dados. Pode ser utilizada para editar as informações dentro das TEDS. Além disso, esta ferramenta envia todos os comandos definidos pelo padrão IEEE1451.0 e verifica a correta implementação da TEDS. Poder-se-á alterar as TEDS lidas e realizar uma atualização destas na TIM.

Esta ferramenta possui cinco opções:

- Adicionar nova TIM ao sistema;
- Enviar todos os comandos suportados pela norma;
- Editar as TEDS e futuramente realizar o *update* das mesmas na TIM;
- Correr uma aplicação de demonstração;
- Apagar a TIM do sistema.

Ao clicar no ícone lupa surge um *modal* onde é pedido o endereço da TIM que se pretende adicionar ao sistema e selecionar a interface SÉRIE ou MQTT. Neste, é possível adicionar a TIM ligada via interface SÉRIE e MQTT. A página executa um programa escrito em Python. A Figura 3.26 apresenta a organização desse programa que permite procurar e adicionar a TIM ao sistema.

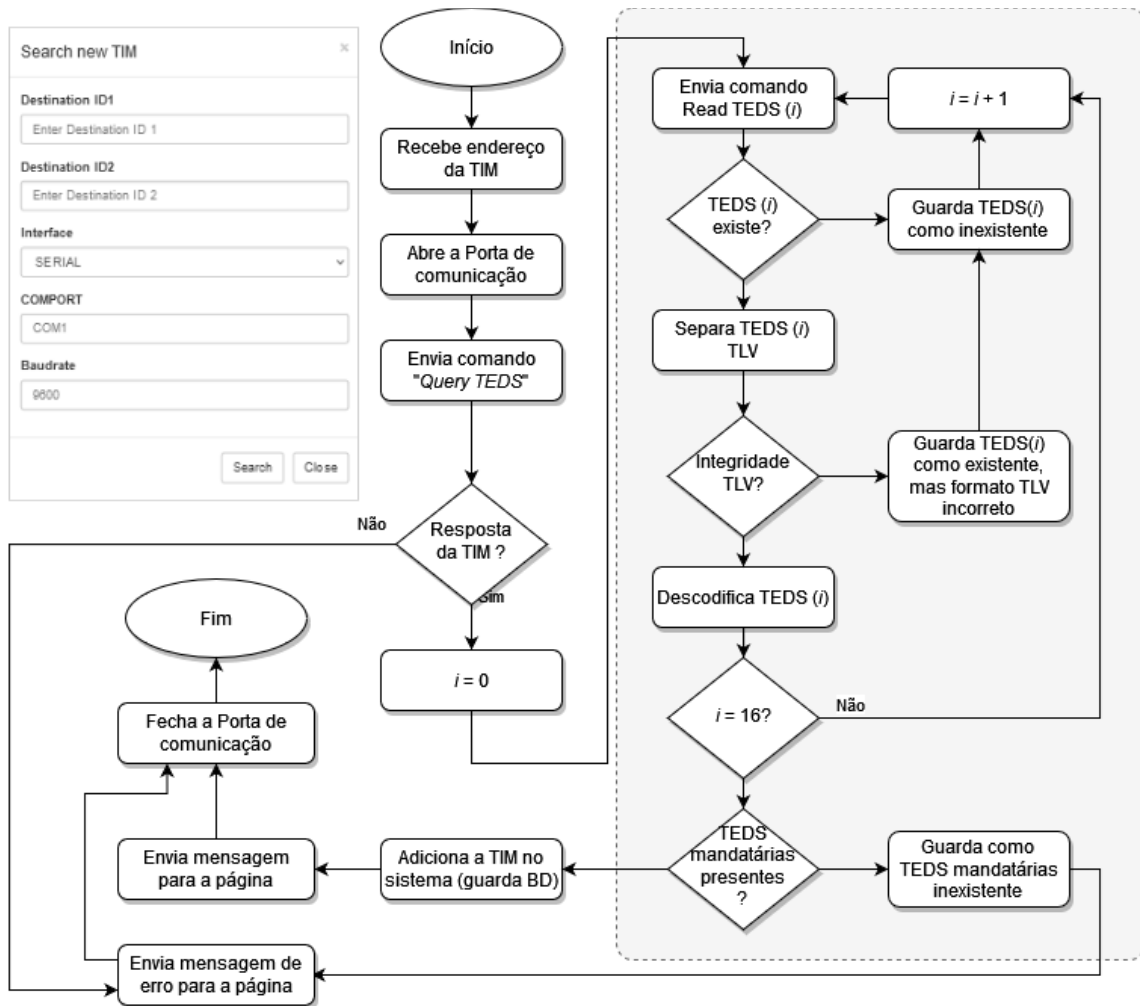


Figura 3.26- Fluxograma do programa *TIM Discovery*

O programa inicia-se recebendo o endereço da TIM que se pretende adicionar, abre a porta de comunicação e envia o comando *Read Query TEDS*. Caso não receba resposta,

por parte da TIM, envia uma mensagem de erro para a página e fecha a porta de comunicação. Caso receba a resposta, inicia um ciclo de leitura de todas as TEDS. Ao ler sequencialmente as TEDS verifica:

- se não estiver presente na TIM (envia mensagem de TEDS não presente, prossegue para a próxima TEDS).
- se estiver presente:
  - verifica a tipologia TLV (se não respeitar a norma, envia mensagem de erro, prossegue para a próxima TEDS),
  - descodifica todas as TEDS;
- Se as quatro TEDS mandatárias estiverem presentes na TIM, adiciona-a no sistema guardando os valores descodificados na base de dados. Envia mensagem de sucesso. Caso contrário envia mensagem de erro.

Depois de a TIM estar inserida no sistema é possível enviar-lhe todos os comandos na secção *TIM Send Commands*. Na Figura 3.28, observamos como são enviados para a TIM todos os comandos suportados pela norma, bem como se recebem todas as respostas implementadas na TIM. Como forma de entender a estrutura das mensagens trocadas entre NCAP/TIM, estas são sempre apresentadas.

Como todas as TEDS estão agora descodificadas e guardadas é possível editá-las, clicando no ícone “lápiz” e futuramente atualizá-las diretamente na plataforma.

Existe ainda uma secção de aplicação de demonstração onde se pode visualizar a TIM em funcionamento. Aplicação online, Figura 3.27, permite visualizar os valores obtidos pelo sensor em tempo real.

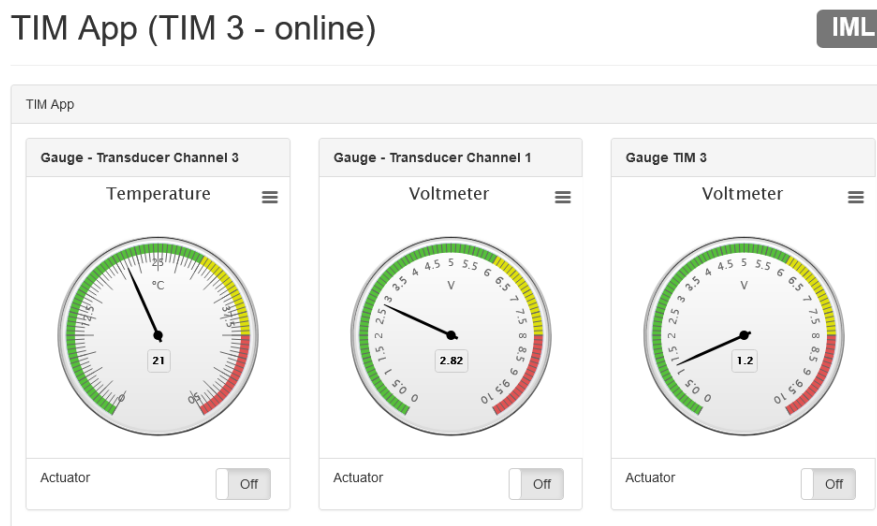


Figura 3.27 - Aplicação Online

Por fim é possível apagar a TIM do sistema, clicando no ícone de “x” apagando todos os valores referentes a essa TIM.

TIM Send Commands (TIM1)



TIM commands

#### Commands common to the TIM and TransducerChannel

- 1 - Query TEDS: 1 - MetaTEDS, Query
- 2 - Read TEDS segment: 1 - MetaTEDS, 0x00, Read
- 3 - Write TEDS segment: 1 - MetaTEDS, 0x01, Write
- 4 - Update TEDS: 1 - MetaTEDS, Send
- 5 - Run self-test: Send
- 6 - Write service request mask: Write
- 7 - Read service request mask: Read
- 8 - Read status-event register: Read
- 9 - Read status-condition register: Read
- 10 - Clear status-event register: Clear
- 11 - Write status-event protocol state: Write
- 12 - Read status-event protocol state: Read

#### Transducer idle state

- 1 - Set TransducerChannel data repetition count: Set
- 2 - Set TransducerChannel pre-trigger count: Set
- 3 - AddressGroup definition: Send
- 4 - Sampling mode: 1 - Trigger-initiate, Send
- 5 - Data Transmission mode: Send
- 6 - Buffered state: Send
- 7 - End-of-data-set operation: Send
- 8 - Actuator-halt mode: Send
- 9 - Edge-to-report: Send
- 10 - Calibrate TransducerChannel: Send
- 11 - Zero TransducerChannel: Send
- 12 - Write corrections state: Write
- 13 - Read corrections state: Read
- 14 - Write TransducerChannel initiate trigger state: Write
- 15 - Write TransducerChannel initiate trigger configuration: Write

#### Transducer operating state

- 1 - Read TransducerChannel data-set segment: TIM, Read
- 2 - Write TransducerChannel data-set segment: Value, TIM, Write
- 3 - Trigger command: TIM, Send
- 4 - Abort Trigger: TIM, Abort

#### Any state

- 1 - Reset: Reset

#### Sleep state

- 1 - Wake-up: Wake

#### Tim active state commands

- 1 - Read TIM version: 0.0, Read
- 2 - TIM Sleep: Sleep
- 3 - Store operational setup: Store
- 4 - Recall operational setup: Recall
- 5 - Read IEEE 1451.0 Version: 0, Read

#### Transducer either idle or operating state

- 1 - TransducerChannel Operate: Send
- 2 - TransducerChannel Idle: Send
- 3 - Write TransducerChannel trigger state: Write
- 4 - Read TransducerChannel trigger state: Read
- 5 - Read TransducerChannel data repetition count: Read
- 6 - Read TransducerChannel pre-trigger count: Read
- 7 - Read AddressGroup assignment: Read
- 8 - Read Sampling mode: Read
- 9 - Read data transmission mode: Read
- 10 - Read buffered state: Read
- 11 - Read end-of-data-set operation: Read
- 12 - Read actuator halt mode: Read
- 13 - Read edge-to-report mode: Read
- 14 - Read TransducerChannel initiate trigger state: Read
- 15 - Read TransducerChannel initiate trigger configuration: Read
- 16 - Device clear: Clear

Figura 3.28 - Envío de comandos IEEE1451

### **3.6.5. Tutoriais**

Com o objetivo de difundir a norma IEEE1451 e a utilização das ferramentas de desenvolvimento, foram escritos uma série de tutoriais onde qualquer pessoa pode entender de forma fácil como funciona. Nestes tutoriais podem encontrar as seguintes seções: Estruturas de mensagens, Comandos, Especificações TEDS, API de serviço do transdutor e API de comunicações do módulo. Todos os tutoriais têm um exemplo prático que demonstra como funciona. Este projeto está em constante evolução e os próximos passos são: implementar todas as TEDS previstas pela norma, com processo de decodificação completo, atualizar e escrever multi TIMs, implementar redes de sensores com comunicação sem fio normalizada.

Foram elaborados diversos vídeos de demonstração de utilização da plataforma que divulgados no canal Youtube - IML Laboratory acessíveis em [12].

### 3.7. Referências Bibliográficas

- [1] Paul Celicourt, Michael Piasecki, “An IEEE 1451.0-based Platform-Independent TEDS Creator using Open Source Software Components”, International Journal of Sensors and Sensor Networks. Vol. 3, No. 1, 2015, pp. 1-11. doi: 10.11648/j.ijssn.20150301.11
- [2] Jin Chen, G. Soundararajan and C. Amza, "Autonomic Provisioning of Backend Databases in Dynamic Content Web Servers", 2006 IEEE International Conference on Autonomic Computing, 2006, pp. 231-242, doi: 10.1109/ICAC.2006.1662403.
- [3] Dennis, A. K., “Raspberry Pi Home Automation with Arduino”, United Kingdom: Packt Publishing, 2013, ISBN-10: 1849695865
- [4] Pereira J., “Sistema de Localização dentro de Edifícios com SimpliciT<sup>T</sup>”, 2º Ciclo em Engenharia Eletrotécnica e de Computadores, 28 de junho 2016, <http://hdl.handle.net/10400.6/5770>
- [5] Avishay Traeger, Nikolai Joukov, Josef Sipek, and Erez Zadok. 2006. Using free web storage for data backup. In “Proceedings of the second ACM workshop on Storage security and survivability” (StorageSS '06). Association for Computing Machinery, New York, NY, USA, 73–78. <https://doi.org/10.1145/1179559.1179574>
- [6] Fabrizi A., “Dropbox Uploader”, January 1, 2016. [Online], Disponível: <https://github.com/andreafabrizi/Dropbox-Uploader>, [Acedido: 19-02-2019]
- [7] Texas Instruments, “MSP-EXP430F5529 Experimenter's Board, User's Guide”, May 2011. [Online], Disponível: <https://www.ti.com/lit/ug/slau330b/slau330b.pdf>
- [8] Garima, N. Agarwal and S. R. N. Reddy, “Design & development of daughter board for Raspberry Pi to support Bluetooth communication using UART”, International Conference on Computing, Communication & Automation, 2015, pp. 949-954, doi: 10.1109/CCAA.2015.7148533
- [9] Michael Dory, Allison Parrish, Brendan Berg, “Introduction to Tornado”, Released March 2012, Publisher(s): O'Reilly Media, Inc. ISBN: 9781449309077
- [10] Texas Instruments, “CC2500, Low-Cost Low-Power 2.4 GHz RF Transceiver”, CC2500 folha de características, janeiro 2005 [Revisto junho 2006].

- [11] "IEEE Standard for a Smart Transducer Interface for Sensors and Actuators - Common Functions, Communication Protocols, and Transducer Electronic Data Sheet (TEDS) Formats," in IEEE Std 1451.0-2007, vol., no., pp.1-335, 21 Sept. 2007, doi: 10.1109/IEEESTD.2007.4338161
  
- [12] IML Laboratory, "IML Laboratory, Youtube Channel", 2022. [Online], Disponível: <https://www.youtube.com/channel/UCxfsY4yXcl9PeULxjAdlgAA>, [Acedido: 11-09-2022]



# Capítulo 4

## Análise Energética de Sensores Inteligentes

### 4.1. Introdução

Existem inúmeros estudos na problemática energética na alimentação dos sensores numa rede de sensores sem fios. No artigo [1] “*An Energy-Efficient Smart Comfort Sensing System Based on the IEEE 1451 Standard for Green Buildings*” é apresentado um estudo/implementação de um nó com um sensor de temperatura e de humidade, bem como o estudo da energia necessária para o seu funcionamento. No artigo [2] “*Smart Wireless Climate Sensor Node for Indoor Comfort Quality Monitoring Application*” tem como foco principal a implementação de um sistema de redes sem fios, IEEE1451, onde foram tidos em conta a portabilidade, o consumo do nó e o preço do sistema.

Segundo a fonte [3] são essenciais três elementos por forma a desenvolver um sistema baseado em energia *harvest*:

- O sistema de recolha e conversão de energia *harvest* em energia elétrica;
- O gestão da energia recolhida para um elemento de armazenamento;
- O elemento de armazenamento que armazena a energia para uso posterior.

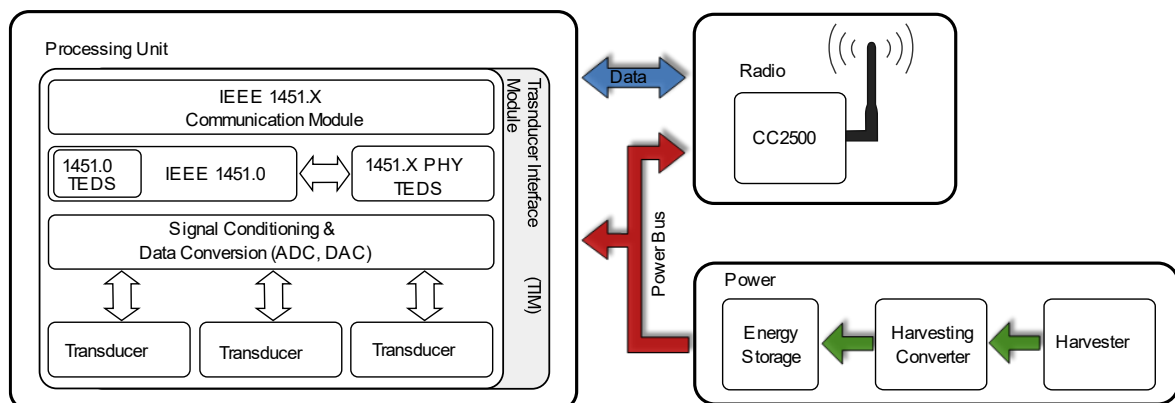


Figura 4.1 - Estrutura TIM Harvest

## 4.2. Estudo Energético Necessário para a TIM

Para garantir a quantidade de energia necessária para a TIM funcionar foi realizada a quantificação de energia consumida, primeiramente calculada e depois medida.

A TIM *Harvest* [4] é constituída por quatro partes: o rádio CC2500; o microcontrolador MSP430F2274; o conversor de energia *booster* AEM10941 e o elemento de armazenamento de energia constituído por dois supercondensadores de 1F cada.

O rádio possui três modos de funcionamento: *Sleep Mode*; TX Mode e RX Mode. O consumo energético depende da configuração escolhida para o rádio, que neste estudo é respetivamente 400 nA, 18,8 mA (250kbps) e 21,2 mA [5].

O microcontrolador possui três modos de energia: *Active Mode*: 490 µA (1 MHz, 3,3 V); *Standby Mode*: 0,7 µA e *Off Mode (RAM Retention)*: 0,1 µA [6].

A quantificação teórica da energia necessária é efetuada para os três modos de energia do rádio e para os dois modos de energia do microcontrolador utilizado:

- Receção de dados do CC2500;
- Transmissão de dados do CC2500;
- *Sleep mode* do CC2500;
- *Active mode* do microcontrolador.
- *Standby mode* do microcontrolador.

A quantificação de energia para a receção de dados é realizada a partir da quantidade de tempo que o rádio está ligado em modo RX.

$$E = P \times \Delta t \quad (\text{eq. 4.1})$$

$$P = UI \quad (\text{eq. 4.2})$$

$$E = U \times I \times \Delta t \quad (\text{eq. 4.3})$$

Para um tempo de receção,  $\Delta t$ , o valor da energia é dado pela função:

$$E = 3,3 \times 1,88 \times 10^{-2} \times \Delta t \quad (\text{eq. 4.4})$$

Para a quantificação de energia na transmissão de dados é necessário determinar a quantidade de tempo que o rádio está ligado em modo TX (250 kbps).

De acordo com a Figura 4.2 [5], o pacote de dados para transmissão é formado por, no mínimo, 1 octeto para o preâmbulo, 2 octetos para *Sync word*, 1 octeto para o *Length Field*, 1 octeto para o *Address Field* e 2 octetos para CRC.

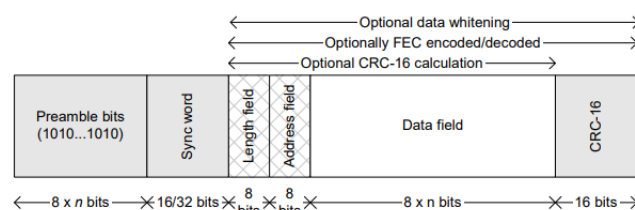


Figura 4.2 - Organização dos dados transmitidos pelo CC2500

Quanto ao campo *Data Field*, segundo a norma IEEE 1451, este é composto por 9 octetos, acrescidos de 6 octetos adicionais para *acknowledge* do *Trigger*, isto é, num total de 15 octetos. Contabilizando todos os campos, verifica-se que o tamanho mínimo de um pacote (trama de bits) é de 22 octetos, isto é, 176 bits. A uma taxa de transmissão de 250kbps obtém-se um tempo de transmissão de:

$$v = \frac{bits}{\Delta t} \quad (\text{eq. 4.5})$$

Para a transmissão de 176 bits a energia obtém-se a partir das expressões:

$$E = 3,3 \times 2,12 \times 10^{-2} \times \Delta t \quad (\text{eq. 4.6})$$

$$\Delta t = \frac{176}{2,5 \times 10^5} \Leftrightarrow \Delta t = 7,04 \times 10^{-4} \text{ s} \quad (\text{eq. 4.7})$$

$$E = 3,3 \times 2,12 \times 10^{-2} \times 7,04 \times 10^{-4} \Leftrightarrow E = 4,9 \times 10^{-5} \text{ J} \quad (\text{eq. 4.8})$$

Para o envio da resposta do comando *Read Data-set* a quantidade de energia é da mesma ordem de grandeza.

A quantidade de energia requerida pelo rádio em *sleep mode* é dada por:

$$E = 3,3 \times 4 \times 10^{-7} \times \Delta t \Leftrightarrow E = 1,320 \times 10^{-6} \times \Delta t \quad (\text{eq. 4.9})$$

A quantidade de energia requerida pelo microcontrolador no *Active Mode* é dada por:

$$E = 3,3 \times 490 \times 10^{-6} \times \Delta t \Leftrightarrow E = 1,617 \times 10^{-3} \times \Delta t \quad (\text{eq. 4.10})$$

A quantidade de energia requerida pelo microcontrolador no *Standby Mode* é dada por:

$$E = 3,3 \times 7 \times 10^{-7} \times \Delta t \Leftrightarrow E = 2,310 \times 10^{-6} \times \Delta t \quad (\text{eq. 4.11})$$

Pelas características do fabricante o processador requer uma corrente de 270  $\mu\text{A}$  a 3,3V em *Active Mode* e o CC2500 requer 13,3 mA a 3,3V no modo de RX (a 250 kBaud). Em *Sleep Mode* a corrente baixa para 400 nA e demora 240  $\mu\text{s}$  a transitar de *Sleep Mode* para *RX mode* ou *TX mode*. Possui ainda a funcionalidade *Wake-on-radio* para *Automatic low-power RX polling* [5].

Para medir o valor real da corrente, utilizou-se uma placa X-NUCLEO-LPM01A da STMicroelectronics, apresentado na Figura 8 do Apêndice F, com a qual se obtiveram os

gráficos de corrente em função do tempo que são mostrados a seguir. Esta placa possui um alcance de 200 mA, com uma menor divisão de escala de 1 nA e 3,2 Msps de amostragem. A Figura 4.3 apresenta o gráfico de corrente em função do tempo para a recepção e transmissão de uma operação de aquisição de dados (RX, medição e TX), tendo-se obtido um consumo total de energia de  $1,040 \times 10^{-3}$  J.

$$\begin{cases} P = \frac{E}{\Delta t} \\ P = UI \end{cases} \Leftrightarrow \begin{cases} UI = \frac{E}{\Delta t} \\ - \end{cases} \Leftrightarrow E = UI\Delta t \quad (\text{eq. 4.12})$$

$$E_{RX} = 3,3 \times 0,020 \times 0,05 \Leftrightarrow E_{RX} = 3,300 \times 10^{-3} \text{ J} \quad (\text{eq. 4.13})$$

$$E_{Processing} = 3,3 \times 0,023 \times 0,05 \Leftrightarrow E_{Processing} = 3,795 \times 10^{-3} \text{ J} \quad (\text{eq. 4.14})$$

$$E_{TX} = 3,3 \times 0,020 \times 0,05 \Leftrightarrow E_{TX} = 3,300 \times 10^{-3} \text{ J} \quad (\text{eq. 4.15})$$

$$E_{total} = E_{RX} + E_{Processing} + E_{TX} \Leftrightarrow E_{total} = 1,040 \times 10^{-3} \text{ J} \quad (\text{eq. 4.16})$$

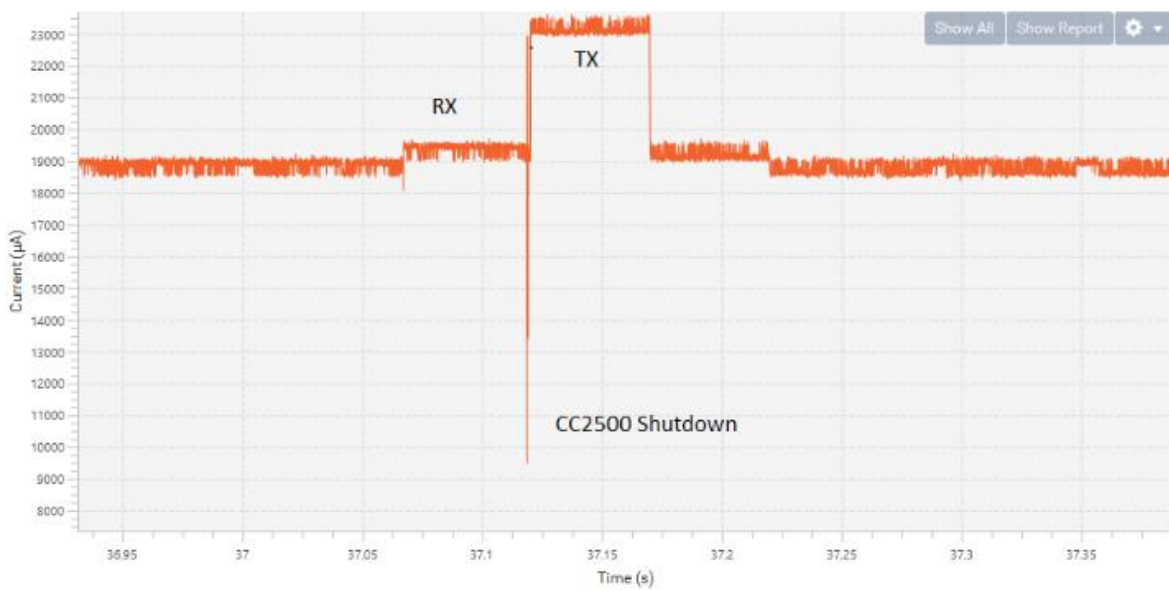


Figura 4.3 - Gráfico  $I(t)$ , para a recepção, processamento e transmissão de dados

Os cálculos são realizados para um segundo, podendo assim dizer que o valor da energia é coincidente com o valor da potência.

Para 1 segundo o total de energia consumida pelo sistema é:

$$E_{total} = E_{RX} + E_{Processing} + E_{TX} + E_{Espera} \quad (\text{eq. 4.17})$$

$$E_{Espera} = 3,3 \times 0,019 \times 0,85 = 5,330 \times 10^{-2} \text{ J} \quad (\text{eq. 4.18})$$

$$E_{total} = 0,0010395 + 0,053295 \Leftrightarrow E_{total} = 5,433 \times 10^{-2} \text{ J} \quad (\text{eq. 4.19})$$

Nesta primeira configuração verifica-se que a quantidade de energia utilizada pelo rádio da TIM é elevada. Por esta razão não poderá estar constantemente ligado, pois irá drenar a fonte de energia rapidamente.

Para diminuir a quantidade de energia necessária, é necessário criar mecanismos, tais como:

- a) Desligando-o periodicamente.
- b) Desligá-lo e acordá-lo só para enviar dados.
- c) Mecanismo de troca de dados consoante a energia disponível.
- d) Reduzir a taxa de transmissão.
- e) Reduzir a potência do transmissor.

Implementação das estratégias e comparação dos resultados obtidos:

- a) Desligando periodicamente.

Nesta metodologia, o rádio liga e desliga periodicamente, permitindo assim reduzir o consumo para metade. A Figura 4.4 apresenta o consumo de corrente, numa fase em que não há receção de dados e a receção de um comando, por parte da NCAP.

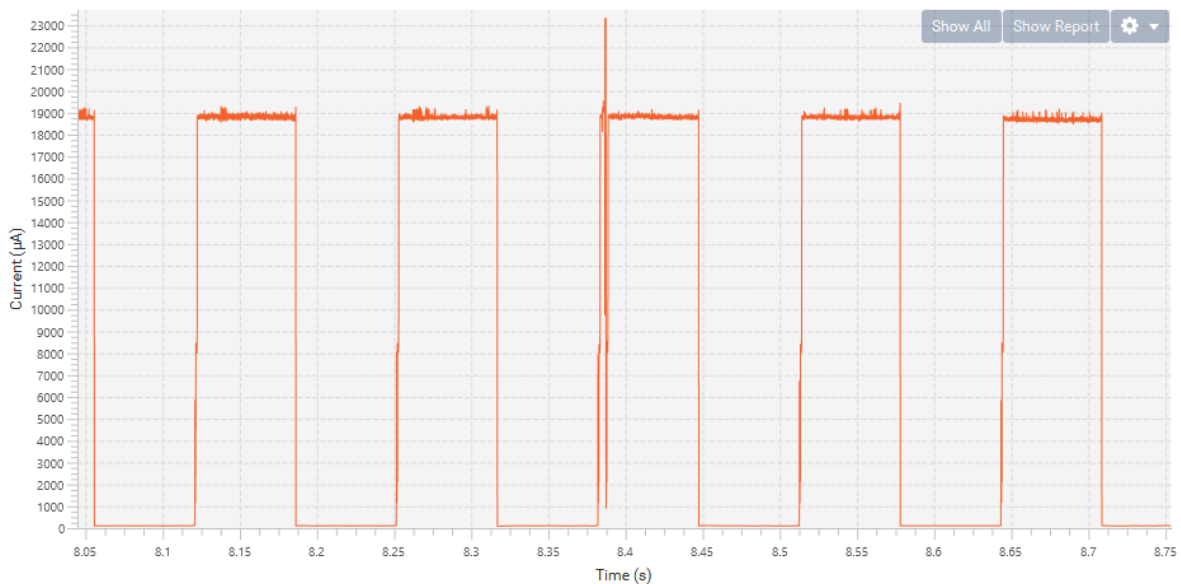


Figura 4.4 - Gráfico  $I(t)$ , para a configuração a)

O facto de o recetor estar desligado torna-se uma desvantagem pois existe grande possibilidade de falhar uma receção. Ao desligar o CC2500 em *Sleep Mode* o sistema passa a consumir 100  $\mu\text{A}$ .

Para 1 segundo o total de energia consumida pelo sistema é:

$$E_{total} = E_{RX} + E_{Processing} + E_{TX} + E_{Espera Low} + E_{Espera High} \quad (\text{eq. 4.20})$$

$$E_{Espera Low} = 3,3 \times 100 \times 10^{-6} \times 0,425 = 1,275 \times 10^{-5} \text{ J} \quad (\text{eq. 4.21})$$

$$E_{Espera High} = 3,3 \times 0,019 \times 0,425 = 2,665 \times 10^{-2} \text{ J} \quad (\text{eq. 4.22})$$

$$E_{total} = 1,040 \times 10^{-3} + 2,665 \times 10^{-2} + 2,665 \times 10^{-2} \Leftrightarrow E_{total} = 2,770 \times 10^{-2} \text{ J} \quad (\text{eq. 4.23})$$

Desta forma foi possível reduzir o consumo de energia para cerca de metade, mas o facto de haver a probabilidade de perder a receção de dados torna-a ineficaz.

b) Desligar o rádio e acordá-lo só para enviar dados.

Nesta configuração a TIM possui dois estados de operar: no primeiro é reconhecido pela NCAP; no segundo o rádio está sempre em *Sleep Mode* e só passa para *TX Mode* para enviar dados e esperar um breve período de tempo em *RX Mode* para receber um possível comando. A informação do formato dos dados enviados da TIM para a NCAP estão nas respetivas TEDS. A TIM possui dois canais transdutor: o primeiro envia o id da TIM; o segundo envia o valor da grandeza física adquirida.

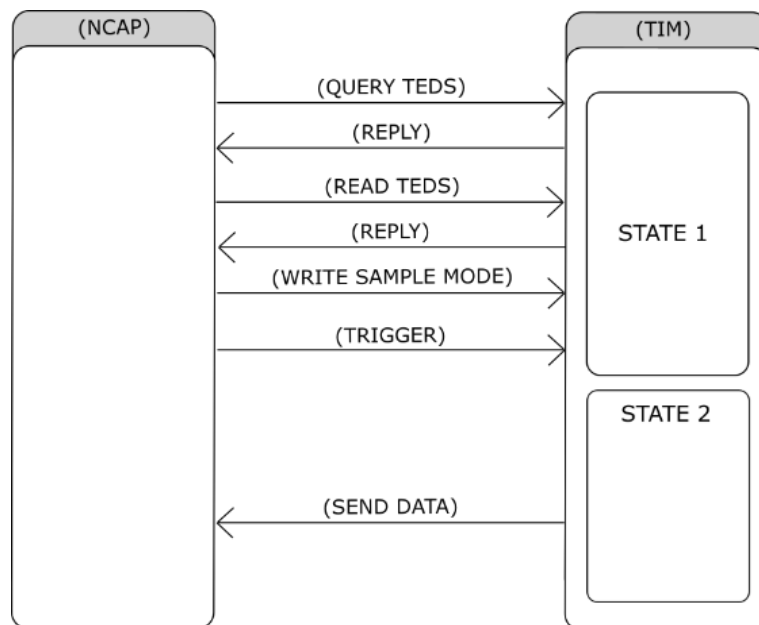


Figura 4.5 - Diagrama de dados transacionados NCAP ↔ TIM

Para 1 segundo o total de energia consumida pelo sistema é:

$$E_{total} = E_{RX} + E_{Processing} + E_{TX} + E_{Espera Low} + E_{Espera High} \quad (\text{eq. 4.24})$$

$$E_{Espera\ Low} = 3,3 \times 100 \times 10^{-6} \times 0,85 = 2,805 \times 10^{-5} \text{ J} \quad (\text{eq. 4.25})$$

$$E_{Espera\ High} = 3,3 \times 0,019 \times 0,05 = 3,135 \times 10^{-3} \text{ J} \quad (\text{eq. 4.26})$$

$$E_{total} = 1,040 \times 10^{-3} + 2,805 \times 10^{-5} + 3,135 \times 10^{-3} \Leftrightarrow E_{total} = 4,203 \times 10^{-3} \text{ J} \quad (\text{eq. 4.27})$$

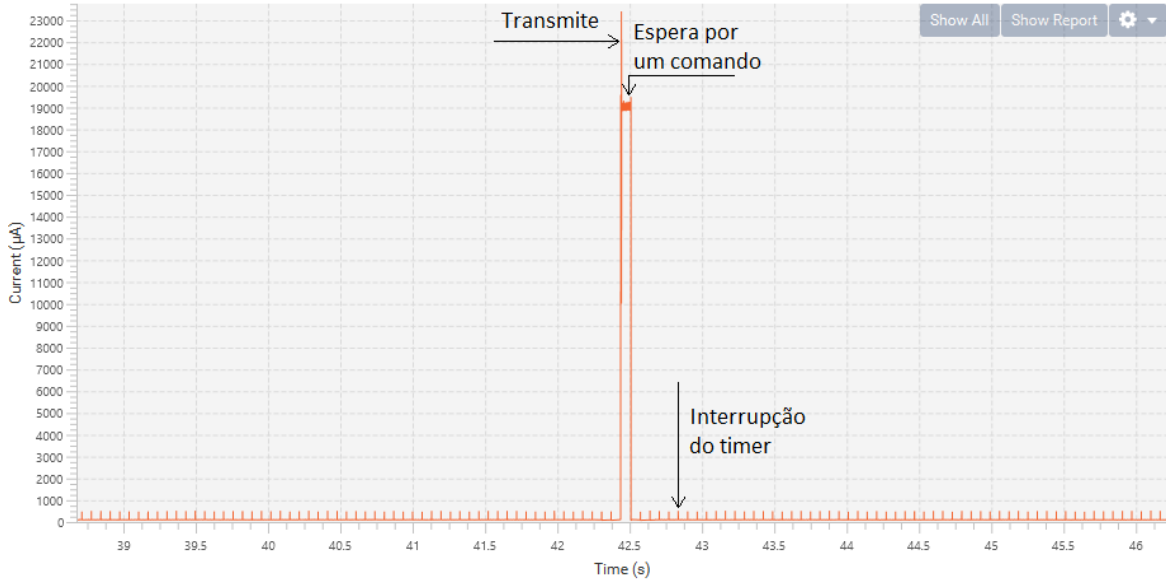


Figura 4.6 - Gráfico  $I(t)$ , para a configuração b)

Nesta configuração o consumo de energia ficou reduzido a 0,004203 J, cerca 13 vezes menos que na configuração inicial. De salientar que esta energia é apenas no segundo que são transmitidos dados, todo o restante tempo consome apenas 0,00001275 J. Pode-se agora abrir caminho para o desenvolvimento de uma fonte de energia.

O dimensionamento de uma fonte de energia para sensores sem fios, que não esteja dependente da rede elétrica, é uma tarefa árdua. As baterias são de imediato a solução mais óbvia. Porém requerem a sua substituição ou recarregamento, aumentando o custo de manutenção.

Para recolher energia *Harvest* da forma mais eficiente possível, a partir de um painel fotovoltaico, existem atualmente várias técnicas como: MPPT (*Maximum Power Point Tracking*) e MPPC (*Maximum Power Point Control*).

Idealmente qualquer painel solar deverá operar na sua máxima potência de saída, mas nem sempre é possível, devido a vários fatores, tais como: se a luz é interior (iluminação) ou exterior (solar); se é de dia ou de noite, se está um dia limpo ou com nuvens, inclusive o tipo de iluminação instalada no edifício, pois os painéis fotovoltaicos possuem rendimentos diferentes para diferentes comprimentos de onda.

A forma mais simples de ligar um painel fotovoltaico é intercalá-lo diretamente ao circuito com um diodo, carregando apenas quando a tensão do painel for superior à tensão da bateria.

Utilizando o algoritmo MPPT, este tem a capacidade de encontrar o ponto máximo de potência do painel de acordo com a quantidade de luz existente num determinado intervalo de tempo. Quando termina o varrimento, força o circuito para que o painel funcione na sua potência máxima.

Uma outra abordagem, MPPC utiliza a premissa de que a tensão máxima de energia de um painel solar não varia, normalmente, tanto quanto a quantidade de luz incidente muda. O que para o caso em estudo desta tese não faz sentido pois o painel é colocado no interior havendo pouca quantidade de luz [8].

Existem várias empresas que disponibilizam circuitos integrados que conseguem lidar com esta situação, ou pelo menos conseguem aumentar o tempo de vida de uma bateria. A Tabela 4.1 apresenta, alguns desses circuitos integrados de acordo com a fonte de energia.

Tabela 4.1 - Lista de carregadores de energia *harvest*

	<b>Linear Technology</b>	<b>e-peas</b>	<b>Texas Instruments</b>	<b>ST</b>
Vibração mecânica	LTC 3588	AEM30300 AEM10330 AEM10340		
TEG Gerador Termoelétrico	LTC3108 LTC3109	AEM20940	BQ25504 BQ25505 BQ25570	
Fotovoltaico	LTC3105	AEM10300 AEM10330 AEM10900 AEM10941	BQ25504 BQ25505 BQ25570	SPV1040 SPV1050
Rádio frequência		AEM30300 AEM10330 AEM10340 AEM40940	BQ25504 BQ25505 BQ25570	

A Tabela 4.2, apresenta a comparação de diferentes circuitos integrados para a utilização de painéis fotovoltaicos, como fonte de energia.

Tabela 4.2 - Comparação *Boost Charger*

Circuito Integrado	<b>BQ25570</b>	<b>AEM10941</b>	<b>LTC3105</b>	<b>SPV1040</b>
$U_{in}$ min (V)	0,1	0,05	0,225	0,3
$U_{in}$ máx (V)	5,5	5	5	5,5
$U_{in}$ cold start (V)	0,6	0,38	0,25	0,8
$P_{in}$ cold start (W)	0,015	0,003	-	-
Suitable for thermal	sim	não	-	-
Balanco com Dual supercondensador	não	sim	não	não
Número de componentes externos	14	7	9	11
Alarme de bateria baixa	sim	sim	sim	não
Alarme de carga desligada	não	sim	não	não
Eficiência máxima (%)	93	94 (a 3,6V)	80 (a 3,3V)	94
Algoritmo	MPPT	MPPT	MPPC	MPPT

Os valores apresentados na tabela foram obtidos das respectivas folhas de características do fabricante. De acordo com a comparação da Tabela 4.2, o circuito integrado mais adequado para o teste recai sobre o AEM10941, pois é o que promete melhores condições para o trabalho em estudo.

O circuito integrado AEM10941 é um *Ultra-Low Power Booster* muito versátil e configurável, que possui um *Cold start* a partir de 380 mV para uma potência de entrada de 3  $\mu$ W. Um *Cold start* acontece quando não existe qualquer energia armazenada, necessária para o AEM10941 iniciar o seu funcionamento.

Possui três pinos de configuração, conferindo-lhe 8 configurações distintas. Com estes é possível configurar o tipo de armazenamento e as tensões de saída. Possui duas saídas de  $U_{LV}$  e o  $U_{HV}$  que permitem utilizar a tensão de 1,8 V para o microcontrolador e 3,3V e para o rádio.

Mede a tensão do painel, em circuito aberto, a cada 5 s para melhor ajustar o algoritmo MPPT.

Possui dois pinos de configuração SELMPP[1:0] que permitem selecionar o rácio MPPT com os valores de 70, 75, 85 ou 90%.

O AEM10941 possui três pinos que informam do estado em que se encontra STATUS[0:3]. O pino de estado STATUS[0] alerta que os LDO (*Low-dropout Regulator*) estão operacionais e podem ser habilitados.

Se a tensão dos condensadores for mais baixa que a  $U_{ovdis}$ , os LDO são desligados e o microcontrolador fica sem energia. Cerca de 600 ms antes de os desligar, o AEM10941, alerta o microcontrolador colocando o pino de status STATUS[1] a nível lógico “1”, para que o microcontrolador possa efetuar um encerramento apropriado. O pino de estado

STATUS[2] é colocado a nível lógico “1” quando o AEM10941 realiza uma avaliação MPPT. Neste estudo foi utilizado o pino STATUS[1].

Possui ainda três pinos de configuração CFG[0:2], da tensão de saída e do tipo de armazenamento utilizado de acordo com três níveis de tensão limite no condensador de armazenamento de energia:

- $U_{ovch}$ : Tensão máxima aceita no elemento de armazenamento antes de desabilitar o conversor;
- $U_{chrdy}$ : Tensão mínima necessária no elemento de armazenamento após um *Cold Start* antes de habilitar os LDO;
- $U_{ovdis}$ : Tensão mínima aceita no elemento de armazenamento antes de considerar o elemento de armazenamento sem energia.

A Tabela 4.3, apresenta as configurações possíveis e assinala a verde a configuração escolhida.

Tabela 4.3 - Configuração AEM10941

#	Pinos de Configuração			Tensões Limite de Armazenamento			Tensão de Saída		Tipo de Armazenamento
	CFG[2]	CFG[1]	CFG[0]	$U_{ovch}$	$U_{chrdy}$	$U_{ovdis}$	$U_{hv}$	$U_{lv}$	
1	1	1	1	4,12 V	3,67 V	3,60 V	3,30 V	1,80 V	Bateria Li-on
2	1	1	0	4,12 V	4,04 V	3,60 V	3,30 V	1,80 V	Bateria estado sólido
3	1	0	1	4,12 V	3,67 V	3,01 V	2,50 V	1,80 V	Bateria Li-on/NiMH
4	1	0	0	2,70 V	2,30 V	2,20 V	1,80 V	1,80 V	Supercondensador
5	0	1	1	4,50 V	3,67 V	2,80 V	2,50 V	1,80 V	Supercondensador Dual
6	0	1	0	4,50 V	3,92 V	3,60 V	3,30 V	1,80 V	Supercondensador Dual
7	0	0	1	3,63 V	3,10 V	2,80 V	2,50 V	1,80 V	Bateria LiFePO <sub>4</sub>
8	0	0	0	Personalizado através de R <sub>i</sub> a R <sub>6</sub>				1,80 V	

Pela análise da melhor configuração, para o caso em estudo, foi escolhida a número 6 da Tabela 4.3. Tensão de saída de 3,3 V necessária para o rádio CC2500, e a utilização de dois supercondensadores para o armazenamento de energia.

#### 4.2.1. Painel fotovoltaico

Para a escolha do painel fotovoltaico deve-se ter em conta o valor máximo, de 5 V, da tensão de entrada do AEM10941, sob o risco da sua destruição (embora o painel seja para ser utilizado com iluminação interior).

Neste estudo apenas foram analisados painéis monocristalinos, pois são aqueles que estão disponíveis no mercado e possuem maior rendimento. Existem painéis com maior

rendimento, mas a sua aquisição é ainda difícil e muitos deles ainda estão em fase de desenvolvimento.

Tabela 4.4 - Características do painel fotovoltaico

Descrição	Valor
Fabricante	<i>Seed Studio 0,5W</i>
Potência	0,5 W
Tensão típica	5,5 V
Corrente típica	100 mA
Tensão Circuito Aberto	8,2 V
Tensão Máxima	6,4 V
Tipo	Monocristalino
Rendimento	17 %
Dimensões	70 x 55 x 3mm

A Tabela 4.4 [12], apresenta as características do painel utilizado. O painel da *Seed Studio* não apresenta diferenças significativas na tensão de saída em relação a três tipos de iluminação interior utilizada, sendo a luz amarela a melhor para pouca iluminação.

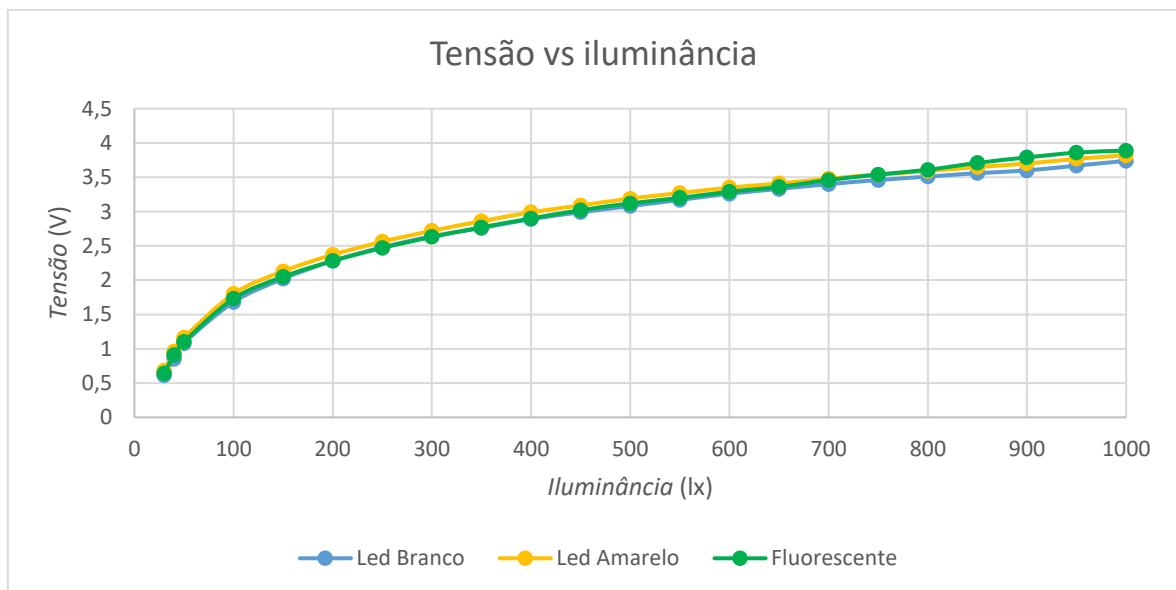


Figura 4.7 - Gráfico  $U(Lx)$  para o painel fotovoltaico utilizado

O gráfico anterior apresenta a tensão de saída do painel em função da intensidade de iluminação, medido em circuito aberto. A unidade de iluminação utilizada é o *Lux* (símbolo lx) que corresponde a um lúmen por metro quadrado ( $1 \text{ lx} = 1 \text{ lm m}^{-2}$ ).

### 4.2.2. Armazenamento de Energia

Dependendo do circuito integrado escolhido, a energia poderá ser armazenada em supercondensadores ou baterias. A escolha do tipo de armazenamento dependerá do sistema que se está a dimensionar. A Tabela 4.5, apresenta a comparação entre supercondensadores e baterias como elemento de armazenamento [3].

Tabela 4.5 - Comparação Supercondensador/Bateria

<b>Supercondensadores</b>	<b>Baterias</b>
A carga pode começar de zero V	Pre-carregamento de fábrica
Baixa densidade de energia	Alta densidade de energia
Baixo ESR	Alto ESR
Maior tempo de vida útil	Tempo de vida útil limitado
Alcance de temperaturas normalizado	Limitações em determinadas temperaturas de funcionamento
Pode ser descarregado até zero V	Não pode ser descarregado até o V
Correntes de fuga interna	Sem correntes de fuga (apreciável)
Sem limite de pico de corrente	Com limite de corrente de pico

Para a escolha do valor do supercondensador foram tidos em conta dois fatores:

- a quantidade de energia necessária para a TIM realizar um *handshake*;
- o que menos descarrega ao longo do tempo.

Para testar o supercondensador foi realizado o seguinte procedimento:

- carregou-se cada condensador durante 24 horas, a uma tensão de 2,7V;
- mediu-se a cada 24 horas a tensão do condensador, durante cinco dias.

O gráfico da Figura 4.8, apresenta a descarga do condensador em vazio.

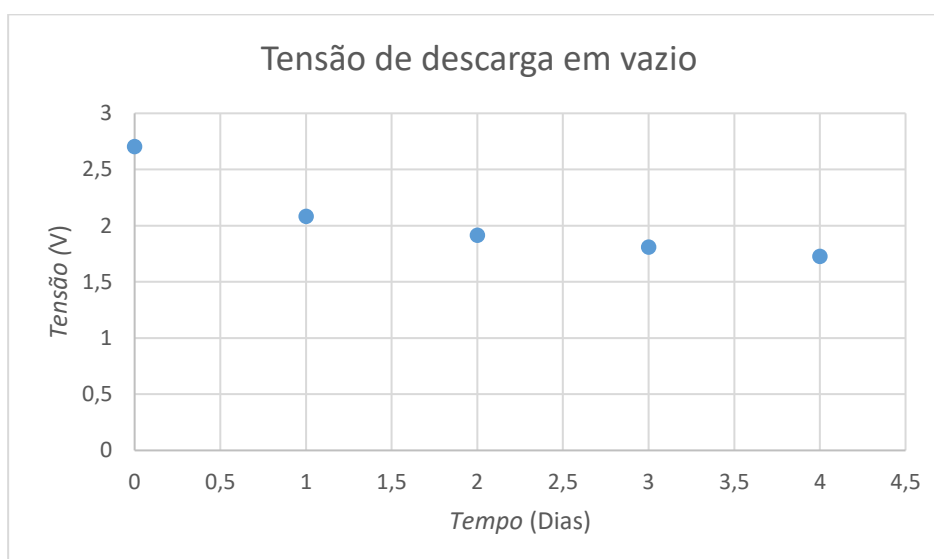


Figura 4.8 - Gráfico  $U(dia)$ , de descarga do condensador em vazio

### 4.2.3. Energia Disponível

No AEM10941 a energia é armazenada com dois condensadores de 1F cada, ligados em série, com uma capacitância equivalente de 0,5F:

$$\frac{1}{C_{eq}} = \frac{1}{C_1} + \frac{1}{C_2} \Leftrightarrow C_{eq} = \frac{1}{\frac{1}{C_1} + \frac{1}{C_2}} \Leftrightarrow C_{eq} = \frac{1}{\frac{1}{1} + \frac{1}{1}} \Leftrightarrow C_{eq} = 0,5 \text{ F} \quad (\text{eq. 4.28})$$

A energia disponível quando os condensadores estão completamente carregados será de:

$$E(U) = \frac{1}{2} \times C \times U^2 \Leftrightarrow E(4,5) = \frac{1}{2} \times 0,5 \times 4,5^2 \Leftrightarrow E = 5,06 \text{ J} \quad (\text{eq. 4.29})$$

O AEM10941 habilita os LDO quando os condensadores estão com tensões entre 4,5V e 3,6 V. A energia armazenada nos condensadores quando estes estão a uma tensão de 3,6 V é de 3,24 J:

$$E(U) = \frac{1}{2} \times C \times U^2 \Leftrightarrow E(3,6) = \frac{1}{2} \times 0,5 \times 3,6^2 \Leftrightarrow E = 3,24 \text{ J} \quad (\text{eq. 4.30})$$

Para uma tensão de saída,  $U$ , de 3,3 V e uma corrente,  $I$ , de aproximadamente 20 mA, a constante de tempo dos condensadores é dada por expressão:

$$\tau = R \times C \Leftrightarrow \tau = \frac{U}{I} \times C \Leftrightarrow \tau = \frac{3,3}{0,02} \times 0,5 \Leftrightarrow \tau = 82,5 \text{ s} \quad (\text{eq. 4.31})$$

O tempo que o AEM10941 consegue disponibilizar uma tensão de saída de 3,3 V, corresponde ao tempo de descarga dos condensadores entre as tensões 4,5V e 3,6V. O tempo é calculado com a expressão:

$$t = \frac{C(U_0 - U_i)}{I} \Leftrightarrow t = \frac{0,5(4,5 - 3,6)}{0,02} \Leftrightarrow t = 22,5 \text{ s} \quad (\text{eq. 4.32})$$

O valor teórico para o tempo de energia disponível é de 22,5 s, como se ilustra na Figura 4.9.

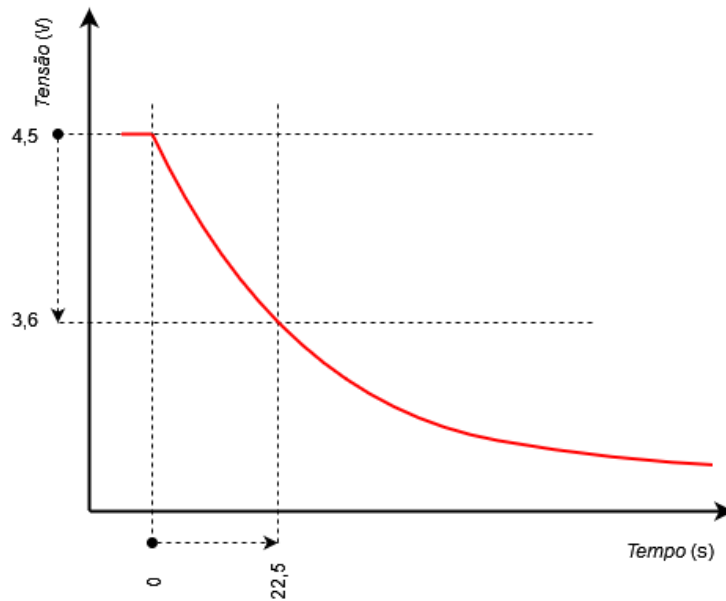


Figura 4.9 - Gráfico  $U(t)$ , para descarga do condensador

Nos cálculos não foram considerados as autodescargas dos condensadores, nem eventuais fugas de corrente no método de medição. A corrente utilizada nos cálculos é um valor médio, pelo que o valor do tempo deverá ser ligeiramente inferior ao calculado.

A energia disponível para o funcionamento pode ser calculada pela equação:

$$E_{\text{disponível}} = E(4,5) - E(3,6) \Leftrightarrow E_{\text{disponível}} = 5,06 - 3,24 \Leftrightarrow E_{\text{disponível}} = 1,82 \text{ J} \quad (\text{eq. 4.33})$$

### 4.3. Desenvolvimento de TIM com Energia *Harvest*

Juntando todas as escolhas que melhor se ajustam às condições deste estudo e tendo por base a fonte [13] realizou-se a TIM alimentada por energia *harvest* que a seguir se descreve.

#### 4.3.1. Diagrama do Circuito da TIM *Harvest*

No seguimento do estudo de diferentes cenários de consumos energéticos, e na escolha da melhor forma de adquirir e armazenar energia, foi construído um protótipo de uma TIM sem fios de energia *Harvest*. Esta utiliza como fonte de energia um painel fotovoltaico monocristalino e um AEM1941, um *power boost charger and buck converter*. A Figura 9 do Apêndice F, apresenta o protótipo e a Figura 4.10 apresenta o diagrama do *hardware* da TIM. A TIM pode ser programada diretamente com o *Code Composer Studio* e realizar o *debugg* em tempo real, pois utiliza a interface de programação original da placa eZ430-RF2500.

Com este protótipo é possível medir individualmente a quantidade de energia necessária para cada módulo, essencial para o estudo e desenvolvimento do algoritmo e gestão de energia da TIM.

Como a quantidade de energia é limitada e como na norma IEEE1451 é a NCAP que inicia a comunicação, é necessário que a NCAP saiba sempre a janela temporal em que a TIM estará com o rádio ligado, já que este terá de estar a maior parte do tempo desligado. Como o rádio não está sempre ligado é necessário garantir a sincronização entre a TIM e a NCAP, para isso é necessário que ambas saibam a quantidade de energia disponível na TIM. Assim sendo, foi escrito um algoritmo na TIM, que lê periodicamente a tensão do supercondensador e traça a linha de tendência de carga/descarga. Com esta linha de tendência calcula/estima o tempo da próxima disponibilidade.

Quando a NCAP faz um pedido, recebe num *proxy* os dados dos sensores e do tempo da próxima leitura. Em caso da NCAP receber o mesmo tempo para duas TIM diferentes descarta aquela que recebeu dados mais recentes.

Para melhor garantir a sincronização, e para que não haja a acumulação de erros na contagem do tempo, sempre que a NCAP recebe uma resposta por parte da TIM, ambas reiniciam a contagem do tempo. Se se perder a sincronização, a NCAP assinala a TIM como “dessincronizada” e passa a enviar pedidos periódicos até que a TIM volte a responder. A TIM terá de garantir energia para que o rádio fique à escuta 1 segundo.

O diagrama do circuito foi elaborado a partir das folhas de características do CC2500 [5], do MSP430F2274 [6], do AEM1941 [14] e do diagrama da placa de desenvolvimento eZ430-RF2500 [15] (Figura 10 da página 18).

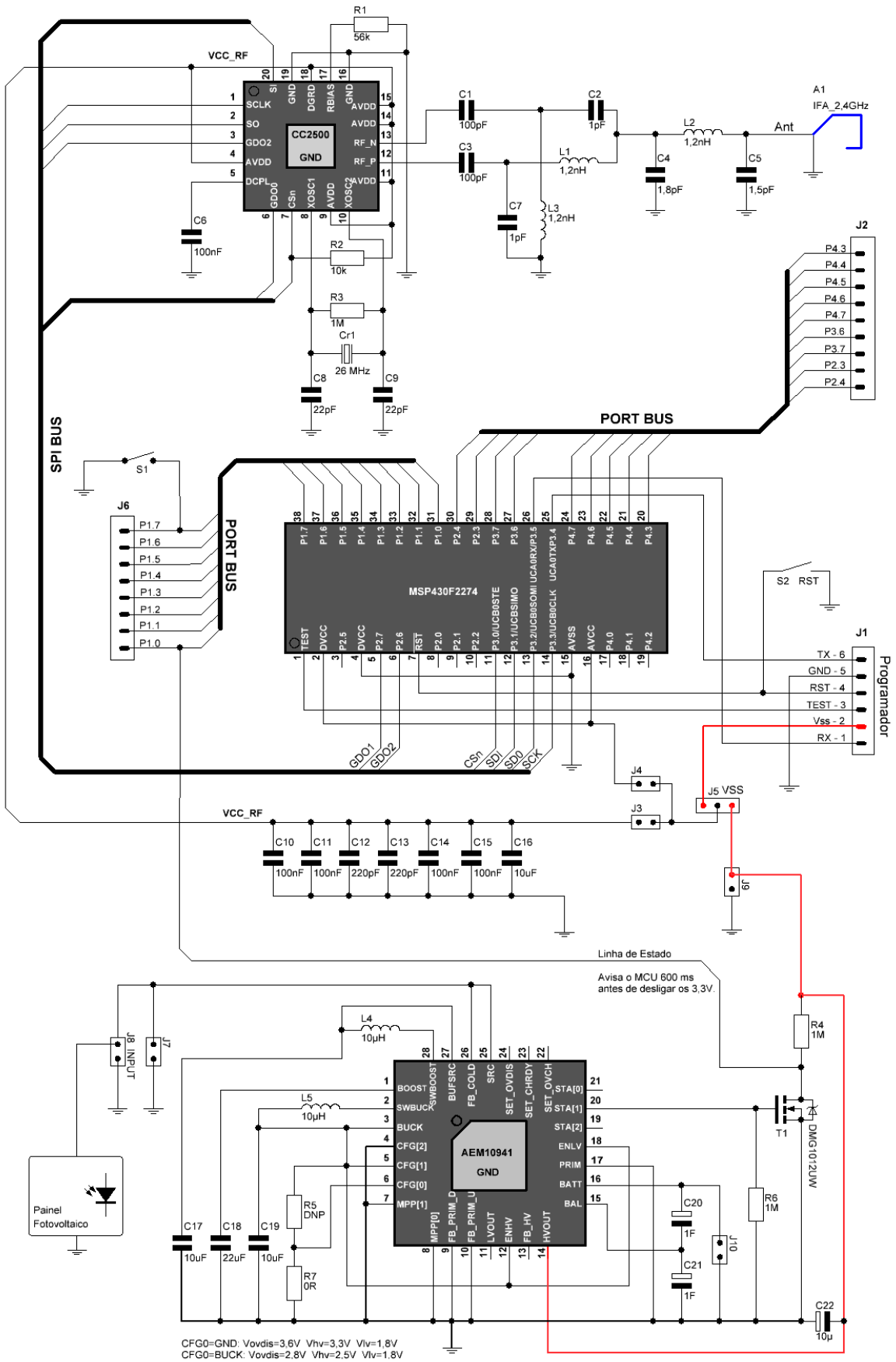


Figura 4.10 - Diagrama TIM Harvest

### 4.3.2. *Software da TIM Harvest*

O sistema possui 5 canais transdutores:

1. Temperatura
2. Tensão do condensador
3. RSSI (potência do sinal recebido)
4. Tempo entre janelas
5. Proxy (temperatura, tensão do condensador, RSSI e tempo entre janelas)

Para configurar este sistema foram utilizadas as seguintes TEDS

- meta-TEDS
- TransducerChannel TEDS 1
- TransducerChannel TEDS 2
- TransducerChannel TEDS 3
- TransducerChannel TEDS 4
- User's Transducer Name TEDS 1
- User's Transducer Name TEDS 2
- User's Transducer Name TEDS 3
- User's Transducer Name TEDS 4
- PHY TEDS
- Energy TEDS

Na meta-TEDS estão configurados 5 canais transdutores. O último é um *proxy*, para que a NCAP possa ler todos os dados de uma só vez.

A TIM ao iniciar carrega todas as configurações contidas nas TEDS, passando logo de seguida, para um estado de menor consumo de energia. Este pode sair deste estado de duas formas distintas: ou através de um *wakeup on port*, com o qual se realiza o *handshake* com a NCAP; ou através *wakeup on timer*, que acorda periodicamente para verificar a receção de comandos provenientes da NCAP. A periodicidade com que acorda está definida na variável *time between slots* que inicialmente é constante e está definida pelo fabricante na TEDS de energia, mas que poderá variar conforme a decisão do algoritmo de gestão de energia. O algoritmo de gestão de energia mede periodicamente a tensão da bateria e o valor do RSSI aquando da receção de um comando proveniente da NCAP. Com estas medições identifica a tendência de carga ou de descarga ajustando o valor *time between slots*. A TIM deverá devolver um *time between slots* com valor zero quando sabe que não terá energia suficiente para voltar a abrir uma *slot*. Neste caso a NCAP perde o *handshake* com esta TIM.

A *slot time* é definida como o tempo de duração em que o rádio está ligado e, portanto, pode receber um comando. Este é um valor constante e é definido pelo fabricante na *Energy TEDS*.

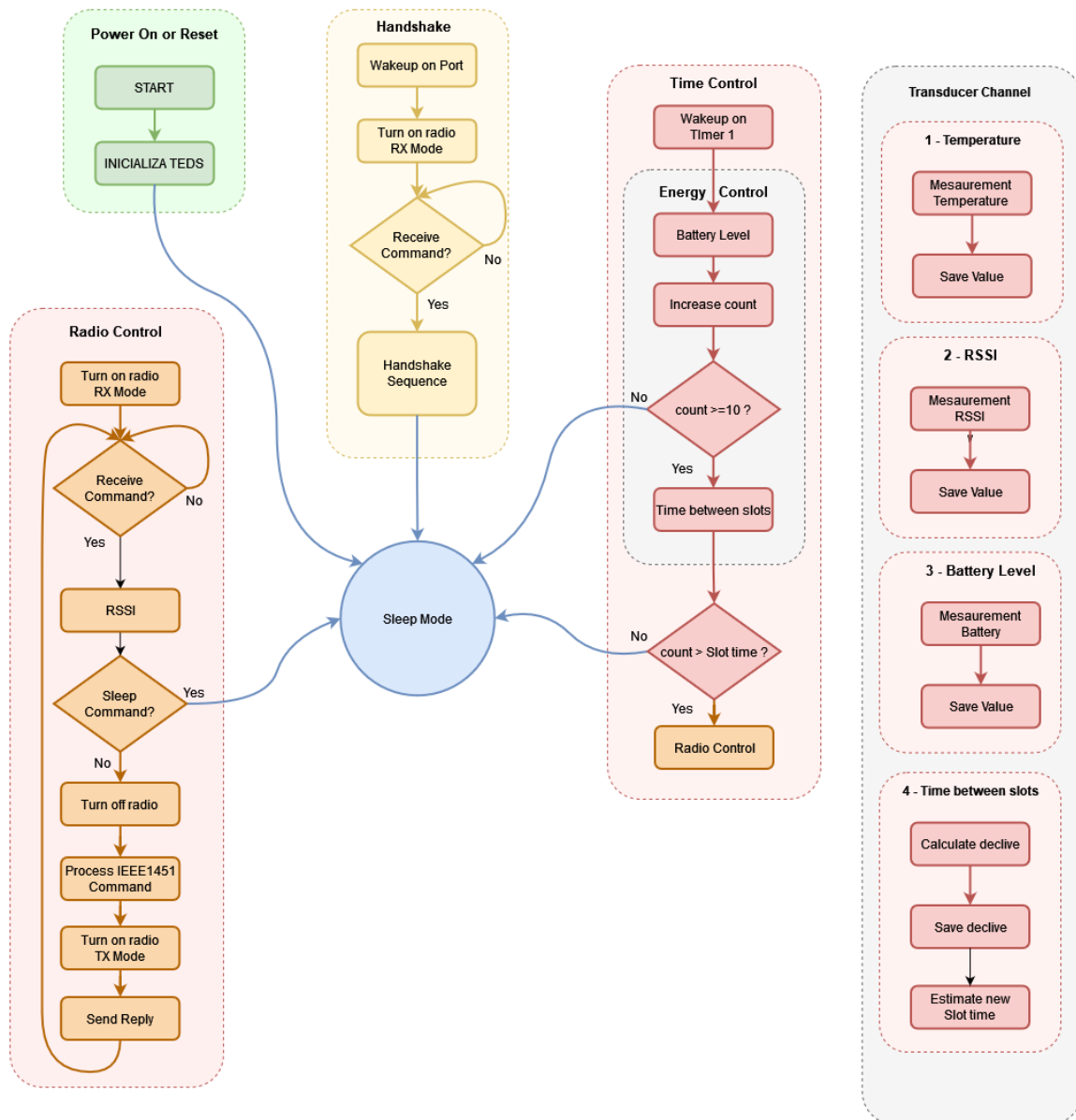


Figura 4.11 – Software da TIM

Quando a TIM é ligada pela primeira vez a NCAP não sabe da existência dela, para tal é necessário um mecanismo de reconhecimento da nova TIM, o *handshake*. A TIM aguarda, num estado de menor consumo de energia, a interrupção deste estado a partir de uma porta definida para essa função. Quando essa porta é ativada a nível lógico 1, liga o rádio e aguarda comandos provenientes pela NCAP. O primeiro comando que a NCAP envia é o comando *Query TEDS*, se não receber volta a enviar o comando até um máximo definido pelo fabricante, se receber a resposta da TIM, fica a saber quais as TEDS que possui. Sabendo as TEDS existentes, irá enviar o comando *Read TEDS* para a leitura de cada uma das TEDS identificadas com o comando anterior. Se tudo estiver correto até aqui, envia um comando de *Read data-set* ao proxy ficando a saber os valores da temperatura, RSSI, tensão do condensador e o tempo entre janelas. Com este último saberá quando é que a

TIM voltará a estar à escuta para a receção de novos comandos. Para terminar o *handshake* envia o comando de *Sleep* e começa a contar o tempo para a próxima janela.

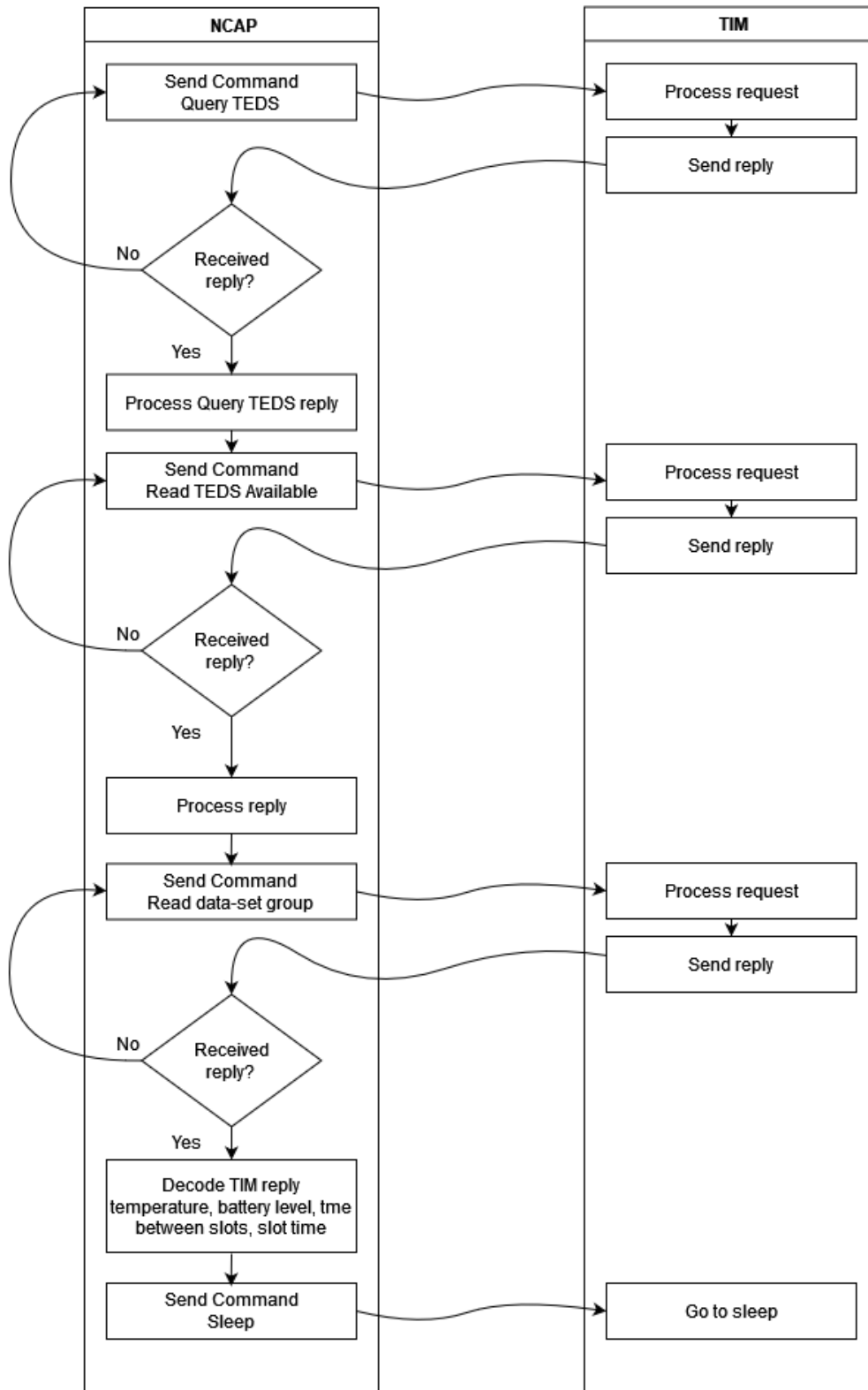


Figura 4.12 - *Handshake*

A Figura 4.12, apresenta o fluxograma de funcionamento do mecanismo de *handshake*. O fabricante da TIM deverá garantir energia suficiente para o *handshake*.

Assim que o *handshake* é concluído com sucesso, a NCAP poderá enviar qualquer comando suportado pela norma (sempre respeitando a janela temporal correspondente àquela TIM). Mais concretamente, neste caso de estudo, envia o comando *Read Data-set* ao *proxy* de canais transdutores. Se não receber volta a enviar o comando até um máximo definido pelo utilizador, se receber descodifica e guarda os valores recebidos. Envia um comando de *Sleep* e aguarda pela próxima janela temporal. A Figura 4.13 mostra o fluxograma de funcionamento deste mecanismo.

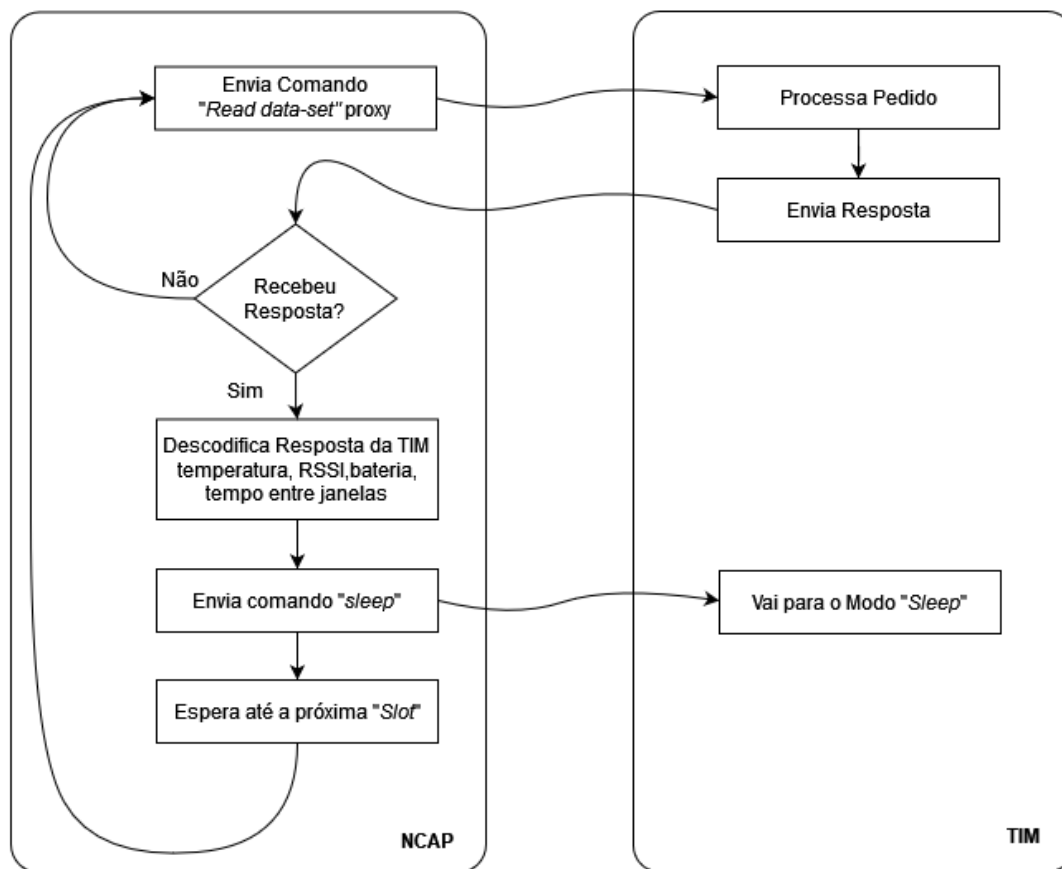


Figura 4.13 - Comando *Read Data-set*

Os canais transdutores estão configurados com um *Sample Mode* de *Single precision real* que representa um *float 32*. Os canais transdutores de temperatura e de RSSI adquirem valores para um *trigger* de *Immediate operations*, ou seja, são medidos aquando da receção do comando *Read Data-set*. O canal transdutor de tensão da bateria é medido no modo de *trigger Continuous Sampling*, pois este é necessário para o algoritmo de energia calcular o tempo entre janelas. O algoritmo de controle de energia é descrito em 4.3.3.

O controlo temporal da TIM, representado no fluxograma como *Time Control*, está à responsabilidade do *timer 1*. Este está configurado para realizar uma interrupção ao processador a cada segundo, um valor que deve ser constante. Em cada segundo executa o algoritmo de controle de energia disponível, que calcula o tempo entre janelas. Esta variável tem sempre um valor que é um múltiplo inteiro do tempo configurado no *timer 1*.

### 4.3.3. Algoritmo de controle de energia disponível

Como a quantidade de energia é limitada e como na norma IEEE1451 é a NCAP que inicia a comunicação, é necessário que a NCAP saiba sempre a janela temporal em que a TIM estará com o rádio ligado, já que este terá de estar a maior parte do tempo desligado. Como o rádio não está sempre ligado, é necessário garantir a sincronização entre a TIM e a NCAP. Para isso é necessário que ambas saibam quando a TIM irá voltar a estar em modo de RX. Quando a NCAP faz um pedido recebe no *proxy* os dados dos sensores e do tempo da próxima leitura. Em caso da NCAP receber ao mesmo tempo dados de duas TIM diferentes, descarta aquela da qual recebeu dados mais recentes.

Para melhor garantir a sincronização e para que não haja a acumulação de erros na contagem do tempo, sempre que a NCAP recebe uma resposta por parte da TIM, envia um comando de *sleep* e ambos reiniciam a contagem do tempo. Se a sincronização se perder, a NCAP assinala a TIM como “dessincronizada” e passa a enviar pedidos periódicos até que a TIM volte a responder.

A TIM mede periodicamente (1 vez por minuto) a tensão do supercondensador e ao fim da segunda medição calcula o declive entre as duas últimas medições.

O declive é calculado a partir da equação, referente a um gráfico  $U(t)$ .

$$U(t) = declive \times t + U_0 \quad (\text{eq. 4.34})$$

$$declive = \frac{U(1) - U(0)}{\Delta t} \quad (\text{eq. 4.35})$$

Se o valor do declive calculado for inferior a zero a TIM, caso ainda tenha energia suficiente, envia o valor zero para o *tempo entre janelas*, notificando a NCAP que irá ficar sem energia e entrar em modo TIM dessincronizada.

Se o valor do declive calculado for igual ou superior a zero a TIM, calcula o valor estimado para o tempo entre janelas de acordo com a equação seguinte:

$$t = \left[ \left( \frac{1}{\frac{tensão(1)}{4,5}} \times ajuste \right) + (TITBW - ajuste) \right] \quad (\text{eq. 4.36})$$

Na eq. 4.36 o TITBW é o valor definido na TEDS de energia, descrito em 4.4, e o *ajuste* é a constante de tempo que o fabricante terá de inserir aquando do fabrico da TIM.

Para um valor de *ajuste* igual a 2000 e um valor de TITTBW igual a 300 obtêm-se os valores apresentados na Tabela 4.6 para o *tempo entre janelas*.

Tabela 4.6 - Tensão vs tempo entre janelas

<b>Tensão (V)</b>	<b>Tempo (s)</b>
3,6	800
3,7	732
3,8	668
3,9	608
4	550
4,1	495
4,2	443
4,3	393
4,4	345
4,5	300

A Figura 4.14 apresenta a organização do algoritmo que permite à TIM estimar o tempo que voltará a estar em modo RX (*tempo entre janelas*).

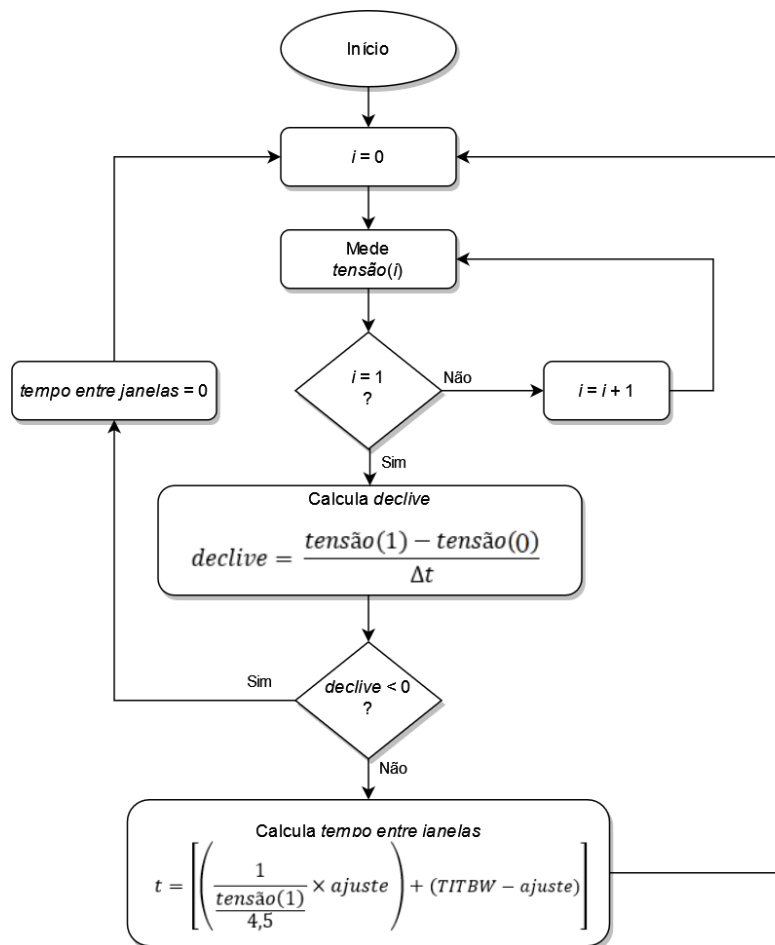


Figura 4.14 - Fluxograma do algoritmo de controle de energia

## 4.4. Proposta de adição para TEDS-Energy

No artigo *TEDS Extensions Toward Energy Management of Wireless Transducers* [16] os autores propõem uma extensão à PHY TEDS para que a norma IEEE1451 possua melhor compatibilidade com sensores sem fios *harvest*. Proponho, em vez disso, a adição de uma nova TEDS especificamente para esse efeito. Esta TEDS será necessária para que o algoritmo descrito em 4.3.3 funcione na sua plenitude.

A Figura 4.15 apresenta a implementação, da TEDS de Energia, na ferramenta de desenvolvimento Editor de TEDS, descrito em 3.6.1.

The Energy TEDS is an experimental TEDS for Harvest Energy Wireless TIM. Energy TEDS octets are constant and read-only.

**Energy Source Type**

- TEDS Identification Header: 16
- Source type: Photovoltaic
- Maximum Energy: 1

**Energy Storage Type**

- Energy Storage Type: Super Capacitor
- Maximum Energy Storage: 5.2
- Minimum Time Recharge: 300

**Timing-related information**

- Minimum time window: 3
- Initial Time between window: 300

**TIM Energy Consumption**

- Processing Energy Consumption: 0.001
- Radio Power RX Mode: 24.0
- Dynamic Radio Power: No
- Radio Power TX Mode: 19.2

**Output TEDS**

```

00 00 00 41 (Length)
03 04 00 00 00 0F (Header)
06 01 01 (Source Type)
07 04 3F 80 00 00 (Maximum Energy)
08 01 02 (Energy Storage Type)
09 04 40 A6 66 66 (Maximum Energy Storage)
0A 04 43 96 00 00 (Minimum Time Recharge)
0B 04 3A 83 12 6F (Processing Energy Consumption)
0C 04 41 C0 00 00 (Radio Energy Consumption RX Mode)
0D 01 00 (Dynamic Radio TX Power)
0E 04 41 99 99 9A (Radio Energy Consumption TX Mode)
0F 04 40 40 00 00 (Minimum Time Window)
10 04 43 96 00 00 (Initial Time Between Window)
F6 1A (Checksum)
    
```

Save

Figura 4.15 - Editor de TEDS de Energia

Nesta proposta a TEDS de energia, possui quatro campos principais e cada campo com os respetivos subcampos editáveis:

- Tipo de fonte de energia:
  - Tipo de fonte de energia;
  - Máxima energia fornecida no melhor dos casos.
- Tipo de armazenamento de energia:
  - Tipo de armazenamento de energia;
  - Máximo de energia armazenada;
  - Tempo mínimo de recarga no melhor cenário.
- Energia consumida pela TIM:

## Capítulo 4 - Análise Energética de Sensores Inteligentes

- Energia consumida pelo microcontrolador;
- Energia consumida pelo rádio na transmissão;
- Energia consumida pelo rádio na recepção;
- Potência de transmissão do rádio dinâmico.
- Informação acerca do tempo:
  - Tempo mínimo em que a TIM voltará a estar ativa;
  - Tempo (inicial) em que a TIM voltará a estar ativa.

No Apêndice G é apresentada uma proposta para a TEDS de energia num formato idêntico ao apresentado na norma IEEE1451.

## 4.5. Referências Bibliográficas

- [1] A. Kumar and G. P. Hancke, "An Energy-Efficient Smart Comfort Sensing System Based on the IEEE 1451 Standard for Green Buildings", in *IEEE Sensors Journal*, vol. 14, no. 12, pp. 4245-4252, Dec. 2014, doi: 10.1109/JSEN.2014.2356651.
- [2] Kuncoro, C. B.D., Arvanida F. Permana, Moch B.Z. Asyikin, and Cornelia Adristi. 2022. "Smart Wireless Climate Sensor Node for Indoor Comfort Quality Monitoring Application" *Energies* 15, no. 8: 2939, doi: 10.3390/en15082939
- [3] e-peas semicondutores, "Photovoltaic energy harvesting with the AEM10941", AEM10941 nota de aplicação, junho 2020 [Online], Disponível: [https://e-peas.com/wp-content/uploads/2020/06/AppNote\\_EH\\_Photovoltaic\\_AEM10941.pdf](https://e-peas.com/wp-content/uploads/2020/06/AppNote_EH_Photovoltaic_AEM10941.pdf) [Acedido: 19-06-2021]
- [4] D. Wobschall and S. Mupparaju, "Low-power wireless sensor with SNAP and IEEE 1451 protocol," 2008 IEEE Sensors Applications Symposium, 2008, pp. 225-227, doi: 10.1109/SAS13374.2008.4472975.
- [5] Texas Instruments, "CC2500, Low-Cost Low-Power 2.4 GHz RF Transceiver", CC2500 folha de características, janeiro 2005 [Revisto junho 2006].
- [6] Texas Instruments, "MSP430F22x2, MSP430F22x4 Mixed Signal Microcontroller", MSP430F2274 folha de características, julho 2006 [Revisto agosto 2012].
- [7] Linear Technology, "Harvest Energy from a Single Photovoltaic Cell", *LT Journal of Analog Innovation*, abril 2011, [Online], Disponível: <http://www.ctimes.com.tw/news/201105/31/Harvest%20Energy%20from%20a%20Single%20Photovoltaic%20Cell%20article.pdf>, [Acedido: 08-01-2022]
- [8] Analog Devices, "Techniques to Maximize Solar Panel Power Output", [Online], Disponível: <https://www.ee.co.za/article/techniques-to-maximise-pv-panel-power-output.html>, [Acedido: 02-12-2021]
- [9] e-peas semicondutores, "Energy Harvesting Ambient Energy Managers", [Online], Disponível: <https://e-peas.com/energy-harvesting/>, [Acedido: 10-01-2022]
- [10] Xie, L., Song, W., Ge, J., Tang, B., Zhang, X., Wu, T., Ge, Z. (2021). Recent progress of organic photovoltaics for indoor energy harvesting. *Nano Energy*, 82, 105770. Doi: 10.1016/j.nanoen.2021.105770

- [11] M. Rasheduzzaman, P. B. Pillai, A. N. C. Mendoza and M. M. De Souza, "A study of the performance of solar cells for indoor autonomous wireless sensors," 2016 10th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP), 2016, pp. 1-6, doi: 10.1109/CSNDSP.2016.7574001.
- [12] Seeed Studio, "Seeed Studio 0.5W Kit solar panel", [Online], Disponível: <https://ie.rs-online.com/web/p/solar-panels/1793740>, [Acedido: 15-01-2022]
- [13] Yongxin, W., Junwu, Z., Yak, S.K. (2019). Demonstration Kit for Battery-Less RF Energy Harvesting Device. In: Guo, H., Ren, H., Bandla, A. (eds) IRC-SET 2018. Springer, Singapore. [https://doi.org/10.1007/978-981-32-9828-6\\_16](https://doi.org/10.1007/978-981-32-9828-6_16)
- [14] e-peas semicondutores, "Highly efficient, regulated dual-output, ambient energy manager for up to 7-cell solar panels with optional primary battery", AEM10941 folha de características, julho 2018, [Revisto novembro 2020], [Online], Disponível: <https://e-peas.com/e-peas-aem10941-datasheet-energy-harvesting/>
- [15] Texas Instruments, "eZ430-RF2500 Development Tool, User's Guide", SLAU227F, setembro 2007 [Revisto Junho 2015].
- [16] H. Zangl, M. Zine-Zine and S. Mühlbacher-Karrer, "TEDS Extensions Toward Energy Management of Wireless Transducers," in IEEE Sensors Journal, vol. 15, no. 5, pp. 2587-2594, May 2015, doi: 10.1109/JSEN.2014.2384276.

# Capítulo 5

## Teste e Validação de Sensores Inteligentes

### 5.1. Introdução

Para a validação IEEE1451 das TIM utilizou-se a ferramenta *TIM Validation*, descrita em 3.6.3. Este capítulo é dedicado ao teste e validação de uma rede de sensores sem fios utilizando a norma IEEE1451, com a particularidade de utilizar uma TIM sem fios utilizando energia *Harvest*.

### 5.2. TIM sem Fios

Segundo o artigo “*Design of the Wireless Temperature Supervisory Control System based on eZ430-RF2500*” [1], os autores propõem um modelo de arquitetura sem fios, não normalizado, e a implementação deste sistema para a monitorização de temperatura utilizando a placa eZ430-RF2500.

Na implementação apresentada neste estudo as TIM foram desenvolvidas também em torno da placa eZ430-RF2500, embora a implementação respeite a norma IEE1451.

O programa desenvolvido está representado, de forma simplificada, no fluxograma da Figura 5.1. A TIM encontra-se, neste sistema, a maioria do tempo em *Sleep Mode*, o que permite incrementar a sua disponibilidade energética. Quando recebe dados provenientes do rádio CC2500 (via SPI), acorda e coloca os dados recebidos em memória, separa a mensagem (dados + somatório), soma os dados de todos os bits recebidos e compara-os com o somatório recebido na mensagem. Caso o somatório da mensagem esteja errado, significa que os dados recebidos estão corrompidos e, volta para o estado de *Sleep Mode* até a receção de novos dados. Caso o somatório da mensagem esteja correto, separa os dados recebidos do comando. Os dois primeiros octetos correspondem à identificação da TIM pelo que compara o ID da TIM com o ID da TIM recebido, se forem diferentes significa que o comando não é destinado para esta TIM e volta para o *Sleep Mode*. Caso sejam iguais procede à execução do comando de acordo com o número do canal transdutor, classe de comando e função de comando. Este módulo respeita a norma IEEE1451, descrita em 2.8. Dependendo do comando recebido e caso seja necessária resposta por parte da TIM esta, depois de executar o comando, constrói a mensagem de

resposta, conforme descrito na Figura 5.1 e de acordo com o Apêndice C e envia-a para o rádio CC2500 (via SPI).

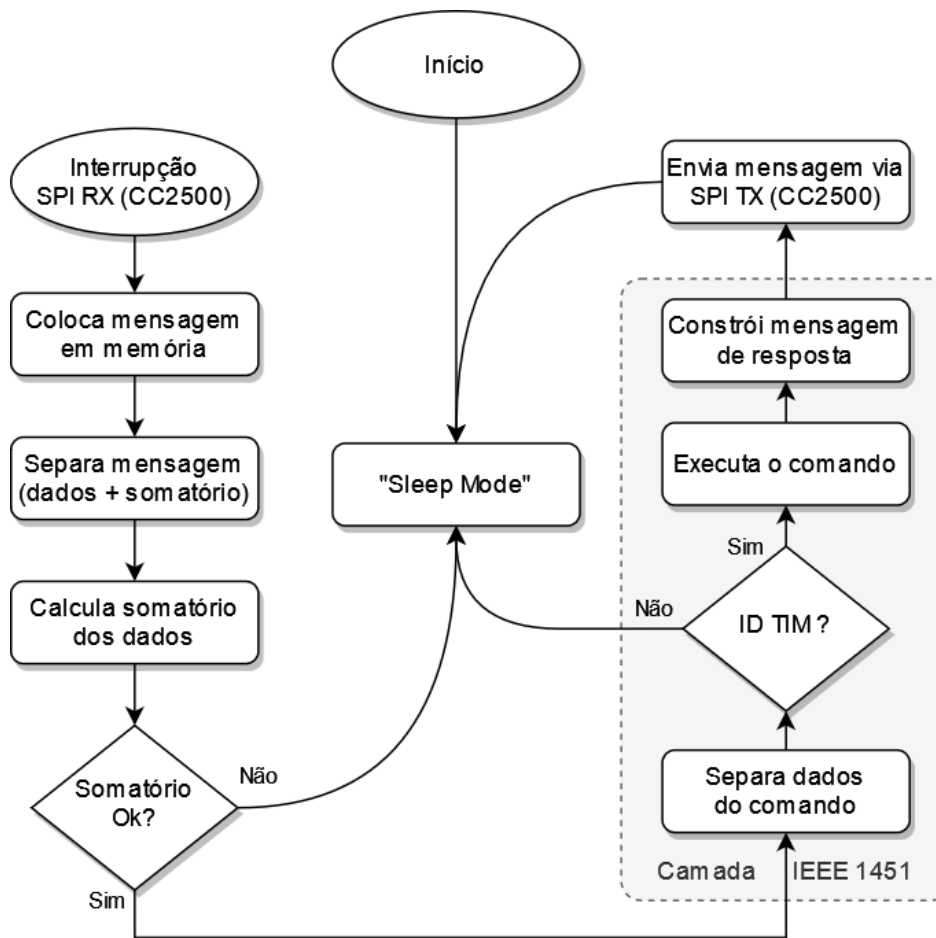


Figura 5.1 - Fluxograma do programa TIM sem fios

### 5.3. Montagem Experimental

A montagem experimental é constituída por uma NCAP, uma aplicação e três TIM:

1. Caudalímetro (sensor simples);
2. Nível de água (sensor de evento analógico);
3. Controlador da bomba de água (atuador simples).

A Figura 5.2 representa de uma forma simplificada a montagem experimental onde se mostra a instalação de um circuito fechado de água e a disposição das TIM. O sistema apresenta-se graficamente na sua totalidade no Apêndice D e uma fotografia da respetiva montagem, em ambiente real, no Apêndice E.

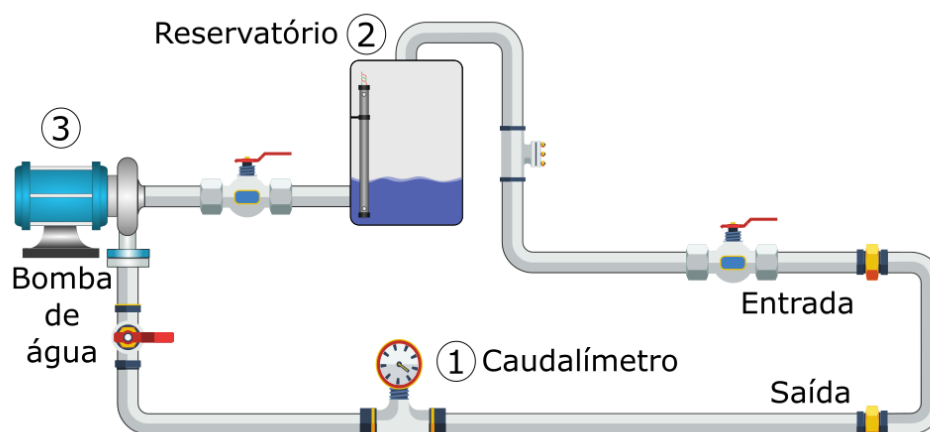


Figura 5.2 - Montagem Experimental

As TIM respondem aos comandos da norma IEEE1451. A enumeração dos comandos dos canais transdutores estão de acordo com a tabela apresentada no Apêndice C.

As configurações das TIM estão contidas nas TEDS:

- Meta-TEDS
- TransducerChannel TEDS 1
- TransducerChannel TEDS 2
- TransducerChannel TEDS 3
- TransducerChannel TEDS 4
- TransducerChannel TEDS 5 (apenas na TIM 2)
- User's Transducer Name TEDS 1
- User's Transducer Name TEDS 2
- User's Transducer Name TEDS 3
- User's Transducer Name TEDS 4
- User's Transducer Name TEDS 5 (apenas na TIM 2)
- PHY TEDS

Os três primeiros canais transdutores, de todas as TIM, são iguais e onde:

- o 1º Canal Transdutor, corresponde à leitura da temperatura interna do processador;
- o 2º Canal Transdutor, corresponde à leitura do RSSI (*Received Signal Strength Indication*);
- o 3º Canal Transdutor, corresponde à leitura da tensão da bateria.

O canal transdutor 1 mede a temperatura interna do microcontrolador. O MSP430F2274 possui um sensor de temperatura interno, de 10-Bit, acessível a partir do canal de entrada 10 (INCH = 0Ah) [2].

```

/* Get temperature */
ADC10CTL1 = INCH_10 + ADC10DIV_4;           // Temperature Sensor ADC10CLK/5
ADC10CTL0 = SREF_1 + ADC10SHT_3 + REFON + ADC10ON + ADC10IE + ADC10SR;
ADC10CTL0 |= ENC + ADC10SC;               // Sampling and conversion start
// reply will be sent from the ADC interrupt
    
```

Figura 5.3 - Aquisição de temperatura

Assim que o conversor ADC obtiver o valor da conversão ativa a interrupção, o MSP430F2274 coloca o valor na variável *temperature* e converte-o para graus Celsius de acordo com a Figura 5.4. Este valor será convertido para hexadecimal, com um tamanho de 4 octetos, utilizando a norma IEEE-754 *Floating Point* para que depois de construída a mensagem de resposta seja enviada para a NCAp.

```

temperature = ADC10MEM;
IntDegC = ((temperature - 673) * 423) / 1024;
    
```

Figura 5.4 - Conversão da Temperatura

O canal transdutor 2 mede a tensão da bateria. O MSP430F2274 mede a tensão da bateria a partir do canal de entrada 11 (INCH = 0Bh) do ADC de 10-Bit. Assim que o ADC obtém o valor converte-o para decimal de acordo com a Figura 5.5.

```

// Get Battery Voltage
ADC10CTL1 = INCH_11;                       // AVcc/2
ADC10CTL0 = SREF_1 + ADC10SHT_2 + REFON + ADC10ON;

ADC10CTL0 |= ENC + ADC10SC;               // Sampling and conversion start
while (ADC10CTL1 & ADC10BUSY);           // ADC10BUSY?

volts = ADC10MEM;
volt = (volts*3.3)/1024;
    
```

Figura 5.5 - Medição da tensão da bateria

O canal transdutor 3 mede o valor RSSI. O valor RSSI é uma estimativa de potência do sinal rádio recebido. De acordo com a folha de características [3] do rádio CC2500, o MSP430F2274 pode ler o valor RSSI obtido diretamente da seguinte forma:

- Lê o registo de estado *RSSI*;
- Converte o valor lido em hexadecimal para decimal (*RSSI\_dec*)
- Se  $RSSI\_dec \geq 128$  então aplica a equação:

$$RSSI\_dBm = \frac{RSSI\_dec - 256}{2} - RSSI\_offset \quad (\text{eq. 5.1})$$

- Se  $RSSI\_dec < 128$  então aplica a equação:

$$RSSI\_dBm = \frac{RSSI\_dec}{2} - RSSI\_offset \quad (\text{eq. 5.2})$$

De acordo com a Tabela 25 da folha de características, o valor do *RSSI\_offset* depende do valor da taxa da transmissão de dados. Para uma velocidade de taxa de 250 kbaud o valor de *RSSI\_offset* deverá ser 72.

```

rssi_dec=TI_CC_SPIReadStatus(TI_CCxxx0_RSSI);
if( rssi_dec >= 128)
{
    rssi_dbm = (rssi_dec-256)/2;
    rssi_dbm = rssi_dbm - 72;
}
else if(rssi_dec < 128)
{
    rssi_dbm = (rssi_dec/2);
    rssi_dbm = rssi_dbm - 72;
}
    
```

Figura 5.6 - Medição RSSI em dBm

Os restantes canais transdutores diferem conforme a função específica de cada TIM e estão descritos em 5.3.1, 5.3.2 e 5.3.3.

Na TIM 1 – Caudalímetro:

- o 4º Canal Transdutor, corresponde à leitura do caudal de água.

Na TIM 2 – Nível de Água:

- o 4º Canal Transdutor, corresponde à leitura do nível de água do reservatório;
- o 5º Canal Transdutor, corresponde à leitura da temperatura da água do reservatório.

Na TIM 3 – Controlador da Bomba de Água:

- o 4º Canal Transdutor, controla a bomba de água, podendo ligar, desligar e ver o seu estado.

### 5.3.1. TIM 1 – Caudalímetro

A TIM 1 tem como funcionalidade principal a medição do caudal de água do circuito fechado. Esta medição corresponde ao canal transdutor 4. Existem diversos métodos para efetuar a medição do caudal como apresentado em 2.2.3.

Para medir o caudal no circuito de água, foi utilizado o sensor YF-S201, com uma configuração semelhante ao descrito no artigo [4]. Este é constituído por uma turbina, que gira com a passagem da água. No centro desta turbina está colocado um íman que gira com o seu movimento. Na parte exterior está colocado um sensor de efeito de hall que atua conforme a alteração do campo magnético, gerando um sinal de onda quadrada [5]. Com o aumento do caudal a frequência do sinal de saída do sensor aumenta de acordo com os valores da Tabela 5.1. Estes foram obtidos na folha de características do fabricante do sensor [6].

Tabela 5.1 - Valores de caudal em função da frequência

<i>Caudal (L/Min)</i>	<i>Frequência (Hz)</i>
2	16
4	32,5
6	49,3
8	65,5
10	82

Com os valores apresentados da Tabela 5.1 foi construído o respetivo gráfico de dispersão do caudal em função da frequência.

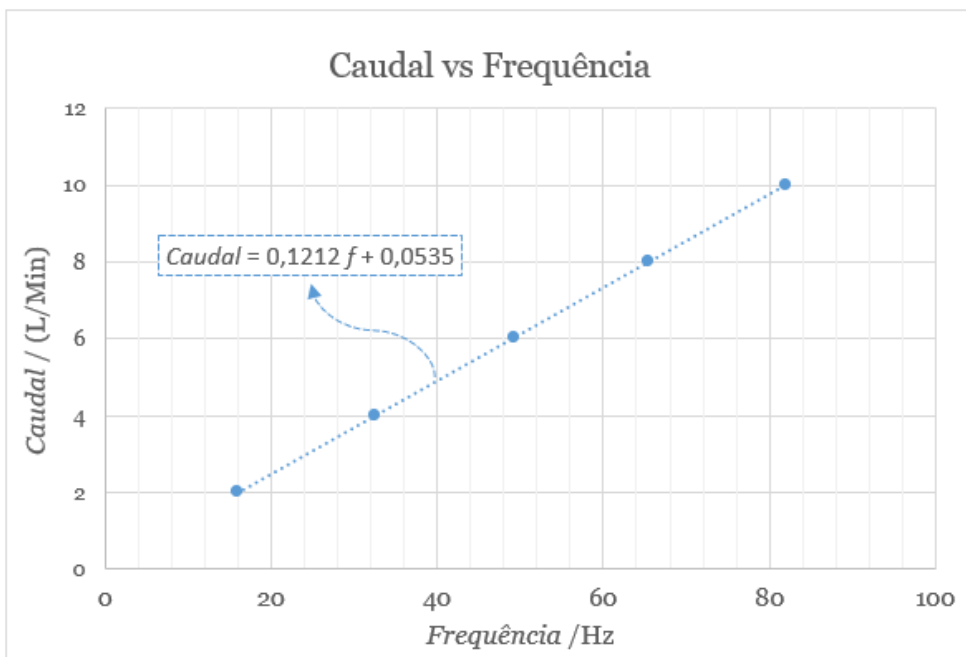


Figura 5.7 - Gráfico do caudal em função da frequência

A Figura 5.7 apresenta o gráfico com a representação dos valores do caudal em litros por minuto em função da frequência em Hertz. Neste, é notória a tendência linear entre as duas variáveis. A curva de calibração do sensor é o resultado de uma regressão linear dos pontos no gráfico de dispersão. Com a curva de calibração, eq. 5.3, é possível ao MSP430F2274 calcular o caudal em litros por minuto.

$$\text{Caudal (frequência)} = 0,1212 \times \text{frequência} + 0,0535 \quad (\text{eq. 5.3})$$

Segundo a folha de características do fabricante o sensor possui um alcance de medição (valor máximo que um aparelho de medida consegue medir) de 30 litros por minuto. Pela equação anterior, 30 litros por minuto, corresponde a uma frequência máxima de aproximadamente de 247 Hz, e um período de 0,0040 segundos. O processador terá de possuir a capacidade de medir tempos de no mínimo 4 milissegundos.

A frequência do relógio do MSP430F2274 está configurada para 1 MHz e o *timer* configurado para contar com a mesma frequência. Como o *timer* é de 16 bits poderá contar até um máximo de 0,065535 segundos, tempo suficiente para a leitura do caudal. Valores superiores a 4 milissegundos serão interpretados como caudal igual a zero.

O MSP430F2274 tem a porta 2.3 configurada como interrupção. Quando o primeiro pulso é gerado pelo sensor, atua a interrupção e liga o *timer*, aquando da seguinte interrupção, segundo pulso, desliga o *timer*. O tempo entre duas cristas sucessivas corresponde ao período do sinal, ao inverso do período corresponde a frequência. Ao saber o valor da frequência o MSP430F2274 calcula o caudal a partir da curva de calibração. A Figura 5.8 apresenta o diagrama de ligação entre o sensor e a TIM.

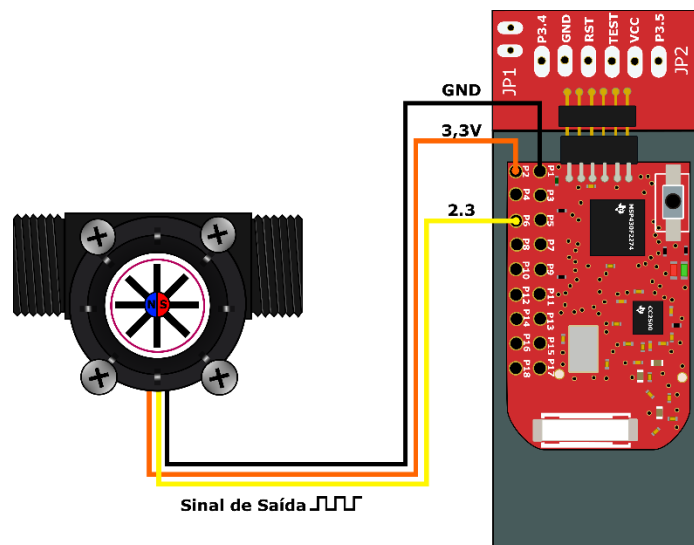


Figura 5.8 - Diagrama de ligação TIM 1

### 5.3.2. TIM 2 – Nível de Água

A TIM 2 tem como funcionalidade principal a medição do nível de água do reservatório. Esta medição corresponde ao canal transdutor 4, e à medição da temperatura da água do reservatório, canal transdutor 5. Existem diversos métodos para efetuar a medição do caudal como apresentado em 2.2.3. Neste estudo foi utilizado um sensor capacitivo como é apresentado em [7] e [8]. Este sensor é constituído por duas varetas de alumínio (os elétrodos) e um isolamento entre os dois, em borracha (o dielétrico).

O MSP430F2274 conta o tempo de carregamento do sensor capacitivo para inferir o nível de água em que o sensor está imerso.

A frequência do relógio do MSP430F2274 está configurada para 1 MHz e o *timer* para contar à mesma frequência, como o *timer* é de 16 bits poderá contar até ao máximo de 0,065535 segundos, bem mais do que o necessário para atingir o alcance de medição, pelo que não irá introduzir erros de leitura.

A Figura 5.9 - a) apresenta a medição efetuada para o tempo de carregamento do sensor, para uma resistência de 500 kΩ. O tempo decorrido até ativar a interrupção do MSP430F2274 foi de 172 μs a qual ocorreu a uma tensão de 2,44 V.

A Figura 5.9 - b) apresenta o diagrama de ligação. O MSP430F2274 ao mesmo tempo que coloca a porta 4.3 no nível lógico a “1”, ativa a contagem do *timer B*. Ao ligar a porta 4.3 carrega o sensor capacitivo através da resistência de 500 kΩ. Assim que o sensor capacitivo atingir a tensão de 2,44 V irá atuar a interrupção configurada na porta 2.3 e parará a contagem do *timer B*. O valor da contagem pode ser relacionado com o nível de água, pois a capacitância muda conforme o nível de água imerso no sensor. Na Figura 10 do Apêndice F, encontra-se uma fotografia do sensor construído.

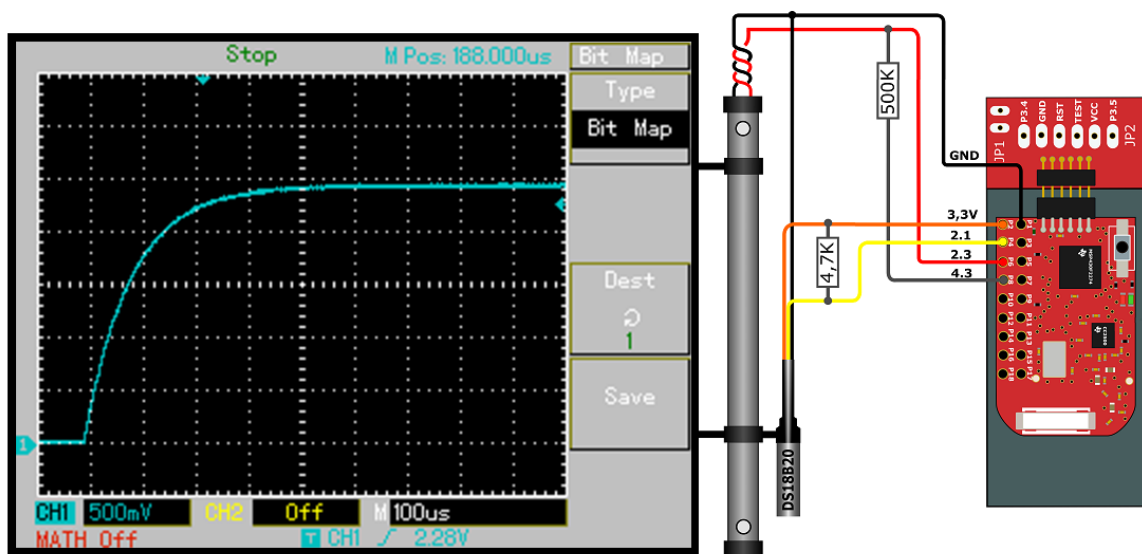


Figura 5.9 - a) Medição do tempo de carregamento - b) Diagrama de ligação TIM 2

A Tabela 5.2, apresenta os valores medidos experimentalmente do tempo de carregamento do sensor (em milissegundos) em função da altura da água (em percentagem).

Tabela 5.2 - Nível de água e tempo de carregamento

Nível de Água (%)	Tempo (ms)	Média (ms)	Nível de Água (%)	Tempo (ms)	Média (ms)
0	98	97	50	168	168
	97			168	
	95			167	
10	112	114	60	182	181
	114			181	
	116			180	
20	127	127	70	196	196
	128			198	
	126			193	
30	139	138	80	210	210
	138			208	
	137			211	
40	153	153			
	153				
	152				

Os tempos de carregamento registados na Tabela 5.2, foram obtidos utilizando o próprio MSP430F2274, programado para esse efeito.

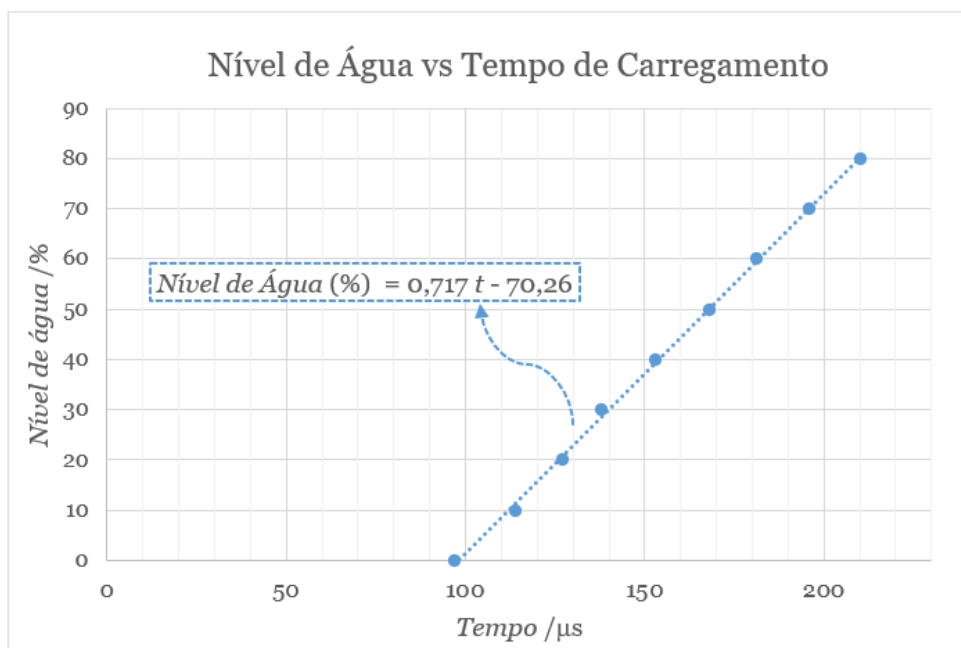


Figura 5.10 - Gráfico do nível de água em função do tempo de carregamento

A Figura 5.10 apresenta o gráfico de dispersão com os valores, tabela anterior, do nível de água em função do tempo de carregamento do sensor capacitivo. Neste, é notória a tendência linear entre as duas variáveis. A curva de calibração do sensor é o resultado de uma regressão linear dos pontos do gráfico.

Com a curva de calibração é possível ao MSP430F2274 calcular o nível de água para qualquer valor de tempo de carregamento medido.

$$\text{Nível de Água (\%)} = 0,717 \times \text{tempo} - 70,26 \quad (\text{eq. 5.4})$$

Este procedimento foi realizado para valores de resistências de 100 kΩ, 500 kΩ 1 MΩ e 2 MΩ, sendo os valores obtidos para os níveis de altura de água igualmente consistentes. O valor do declive é praticamente igual, para todas as resistências, variando apenas o valor da ordenada na origem.

Para minimizar erros de leitura o MSP430F2274 faz 6 leituras, calcula o valor mais provável (a média), para cada medição calcula o desvio padrão, retira os valores mais afastados e volta a fazer a média para os valores mais próximos.

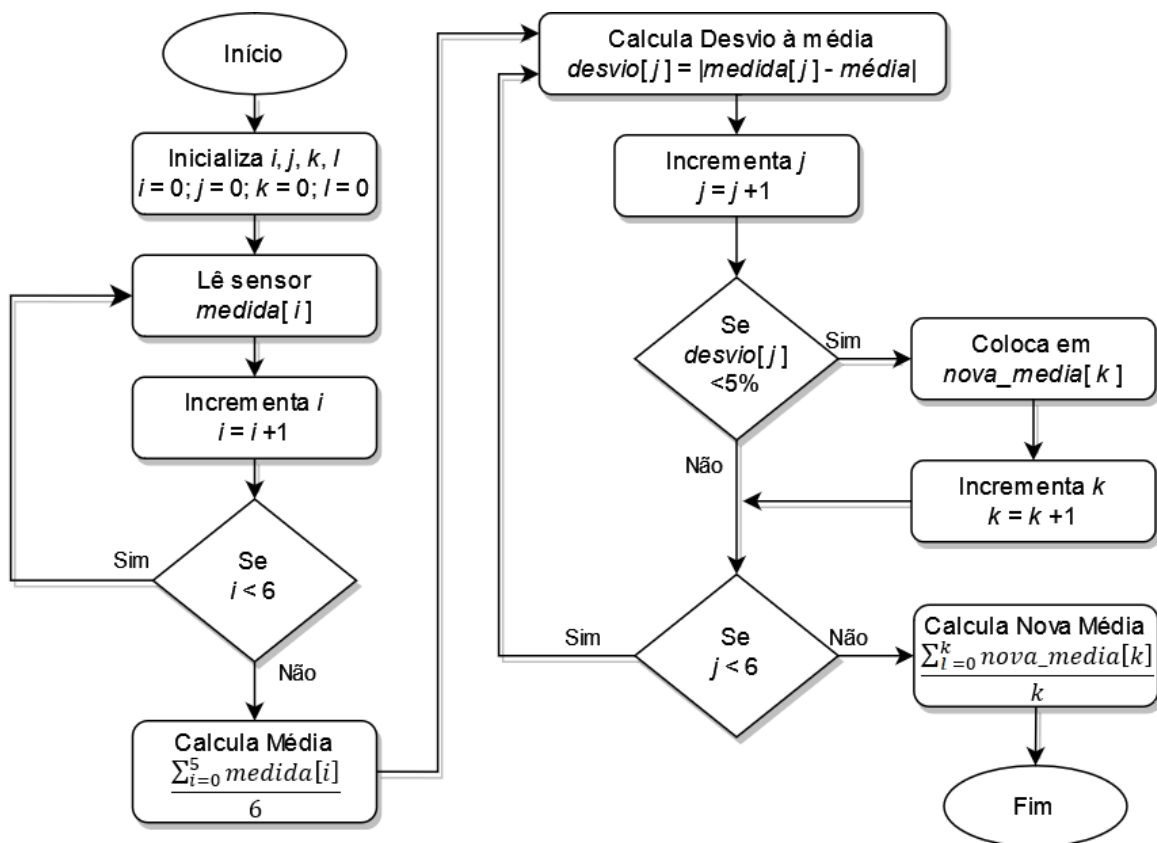


Figura 5.11 - Fluxograma da medição do nível de água

Para a medição da temperatura da água do reservatório foi utilizado o sensor DS18B20, descrito em 2.2.3. Este sensor fornece medições em Celsius e é calibrado pelo fabricante. Possui um conversor ADC que disponibiliza uma resolução configurável de 9 bits a 12 bits. A comunicação, com o microcontrolador, é efetuada através da interface 1-Wire que, por definição, requer apenas dois fios de ligação, um para os dados e outro para o negativo.

A maioria dos dispositivos 1-Wire requer pouca quantidade de energia, na ordem de dezenas de microamperes, para funcionar e comunicar. Mas há outros dispositivos, como o DS18B20, que necessita de mais energia. Durante períodos de aumento de consumo de energia, é importante que a tensão no barramento 1-Wire não desça para valores abaixo da tensão de *pullup* operacional mínima do dispositivo (VPUP). Neste caso foi colocada uma resistência em *pullup* entre os 3,3V e o pino dos dados.

O sensor foi ligado à porta 2.1 do MSP430F2274 a partir do qual efetua a leitura digital do valor da temperatura.





colocar a porta 2.3 a nível lógico “0”. Simultaneamente coloca a variável interna de “estado da bomba” com o valor lógico falso. Se receber o comando de “Lê Estado de Bomba de Água”, lê o valor da variável interna de “estado da bomba” e envia-o para a aplicação.

```

switch (valor){
  case 1:
    MOTOR_OUT ^= MOTOR_OFF;
    TI_CC_Wait(65535);
    TI_CC_Wait(65535);
    TI_CC_Wait(65535);
    TI_CC_Wait(65535);
    TI_CC_Wait(65535);
    TI_CC_Wait(65535);
    TI_CC_Wait(65535);
    TI_CC_Wait(65535);
    TI_CC_Wait(65535);
    MOTOR_OUT ^= MOTOR_OFF;
    estado = 0;
    break;

  case 2:
    MOTOR_OUT ^= MOTOR_ON;
    TI_CC_Wait(65535);
    TI_CC_Wait(65535);
    TI_CC_Wait(65535);
    MOTOR_OUT ^= MOTOR_ON;
    estado = 1;
    break;

  default:
    break;
}

txCC2500Buffer[0] = 0x05;           // Packet length (CC2500)
txCC2500Buffer[1] = 0x01;         // CC2500 channel

txCC2500Buffer[2] = 0x01;         // Success/Fail Flag
txCC2500Buffer[3] = 0x00;         // Length (most significant byte)
txCC2500Buffer[4] = 0x01;         // Length (least significant byte)
txCC2500Buffer[5] = estado;       // Get motor state

RFSendPacket(txCC2500Buffer, 6); // Send data to CC2500

```

Figura 5.14 - Canal Transdutor 4, atuador de bomba de água

### 5.3.4. Aplicação

Como demonstração do sistema do desenvolvimento da montagem experimental, foi escrito um programa, Figura 5.15, em Python com um GUI escrito utilizando a ferramenta QT Creator [10].

Esta aplicação permite o controlo das diversas TIM sem fios constituintes do sistema. A aplicação pode operar de duas formas diferentes para adquirir os diferentes valores: manualmente ou de forma automática, clicando em *Start*.

Existe um conjunto de dois botões para ligar e desligar a bomba de água, um conjunto de dois botões que permitem iniciar e parar a aquisição automática de dados, criando janelas temporais bem definidas, para a recolha de informação de cada TIM.

Para cada TIM existe um conjunto de quatro botões que permitem o envio individual dos comandos de leitura dos valores:

- da temperatura;
- do RSSI;
- do valor da bateria;
- das funções específicas de cada TIM:
  - caudal da TIM 1;
  - nível de água da TIM 2;
  - estado da bomba de água da TIM 3.



Figura 5.15 - Programa de controlo da montagem experimental

No lado direito da aplicação surgem os diferentes valores obtidos. Quando o valor não é recebido, ou recebido mas não corretamente decodificado é apresentado como “erro” e contabilizado na secção das estatísticas. Os comandos enviados são apresentados visualmente, bem como a respetiva resposta (caso seja recebida), permitindo um entendimento de possíveis problemas de comunicação.

Na secção de configurações da aplicação é possível escolher uma ligação por MQTT ou SÉRIE (a partir da ponte SÉRIE ⇔ RF), seleccionar o servidor MQTT ou a porta de comunicação série, bem como a velocidade de transmissão. Os dados recolhidos são guardados numa base de dados MySql para posterior análise. O programa é executado num *Raspberry Pi*, que possui um LCD de 7” com uma resolução de 800×480 *pixels*, tal como se apresenta na Figura 5 do Apêndice F, mas poderá funcionar com qualquer outro sistema operativo.

## 5.4. TIM de energia *Harvest*

O diagrama da placa está descrito em 4.3.1 e o seu funcionamento interno em 4.3.2. A placa de circuito impresso da TIM *Harvest* foi desenhada com o programa *Sprint-Layout 6.0* da *ABACOM*. Esta placa foi provida de um conjunto de pontos de ligação, para facilitar a medição individual da:

- corrente do MSP430F2274;
- corrente do CC2500;
- tensão de saída do painel fotovoltaico;
- tensão dos condensadores de armazenamento.

Estes pontos de ligação permitiram analisar o comportamento do funcionamento do conversor AEM10941, dos painéis fotovoltaicos e do programa que corre no microcontrolador.

A Figura 5.16 apresenta a serigrafia da placa da TIM *Harvest*, onde se indicam os pontos de ligação.

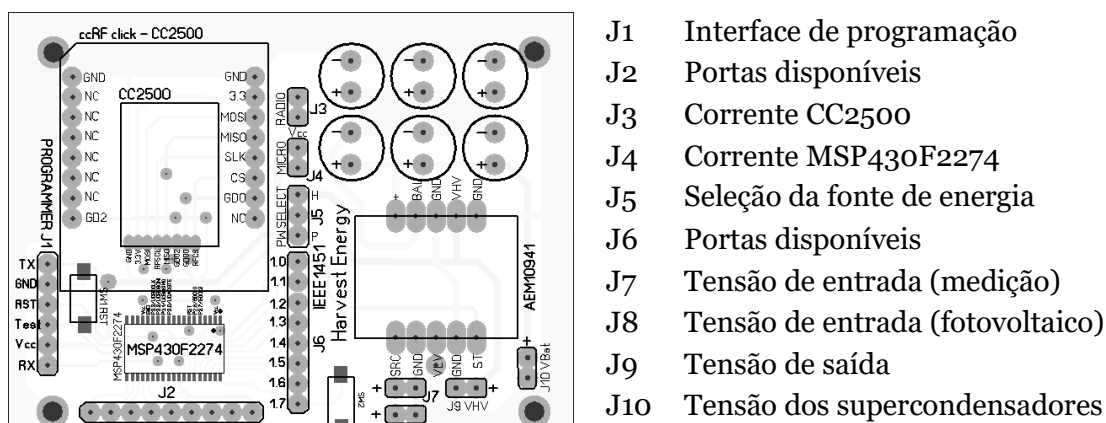


Figura 5.16 - Hardware TIM *Harvest*

Na Figura 9, do Apêndice F, encontra-se a fotografia da TIM *Harvest*.

### 5.4.1. Aplicação TIM Harvest

Para o estudo da TIM sem fios alimentada por energia *harvest* foi necessário escrever um programa em python. Este executa um ciclo fechado que se inicia quando se introduz o endereço da TIM e se clica no botão *Add TIM* (clicando no botão da TIM para iniciar a comunicação). Se a aplicação receber uma resposta por parte da TIM, descodifica os valores da resposta (temperatura, RSSI, bateria, e tempo da próxima vez que vai estar em modo RX) e coloca-os na base de dados. Cria então uma *thread* temporal que é executada ao fim do tempo enviado pela TIM. Envia nova mensagem para a TIM e verifica a sua receção. Caso haja a correta receção de dados, cria *thread* temporal, caso não haja receção de dados coloca essa TIM num estado de “TIM Perdida”. Neste estado a aplicação vai enviar periodicamente mensagens até que, eventualmente, volta a ter energia para poder comunicar (o tempo em procura da TIM perdida pode ser ajustado na aplicação). Assim que a aplicação receber novamente dados da TIM, destrói essa *thread* de “TIM Perdida” e cria uma *thread* temporal para essa TIM colocando-a novamente no sistema. A Figura 5.17 apresenta o fluxograma da aplicação na qual se descreve o seu funcionamento.

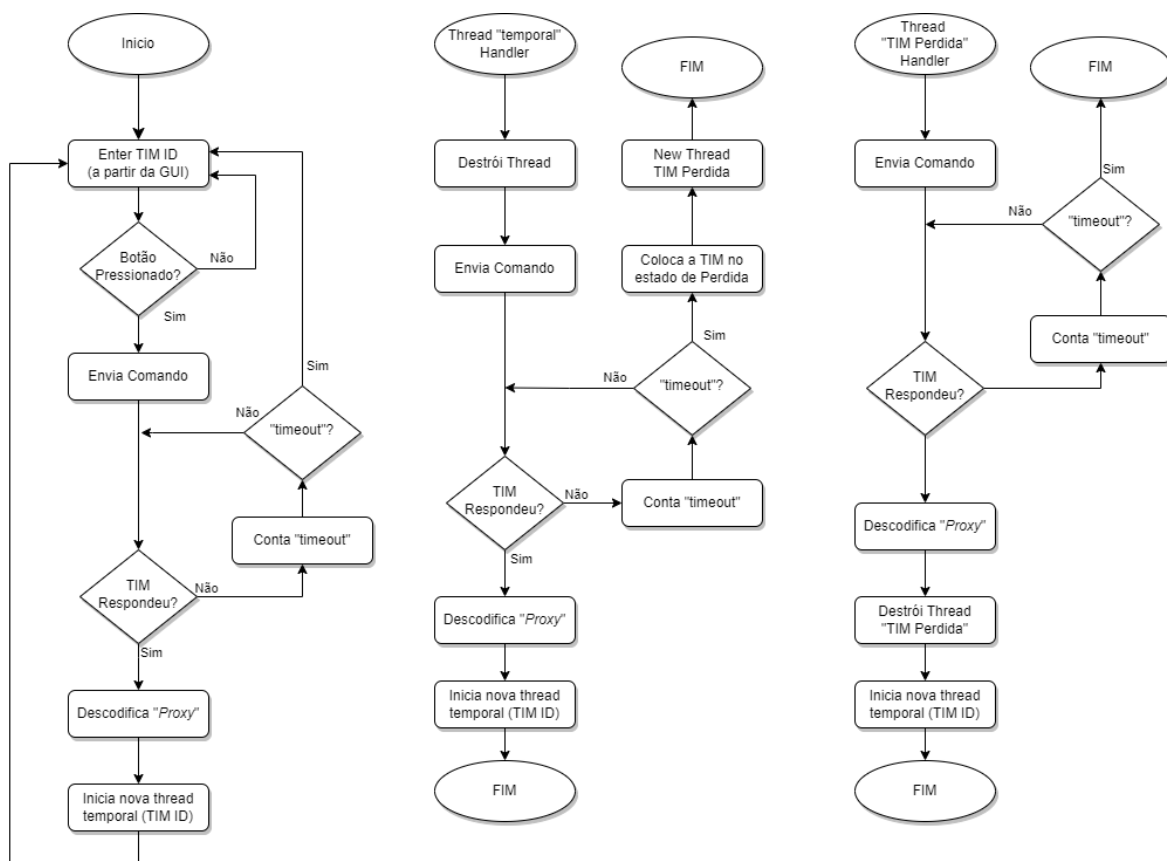


Figura 5.17 - Fluxograma da aplicação TIM harvest

### 5.4.2. Valores obtidos com a TIM Harvest

Os valores obtidos para elaborar os gráficos das tensões em função do tempo e da iluminância foram adquiridos a partir de um *Arduino* [11], Figura 12 do Apêndice F, que foi programado para converter a tensão em valores digitais e enviados para o computador. A iluminância foi obtida utilizando o sensor BH1750 também ligado ao *Arduino*. Os dados foram recolhidos e guardados com o programa *SmartTTY* [12] e analisados com o *Microsoft Excel*.

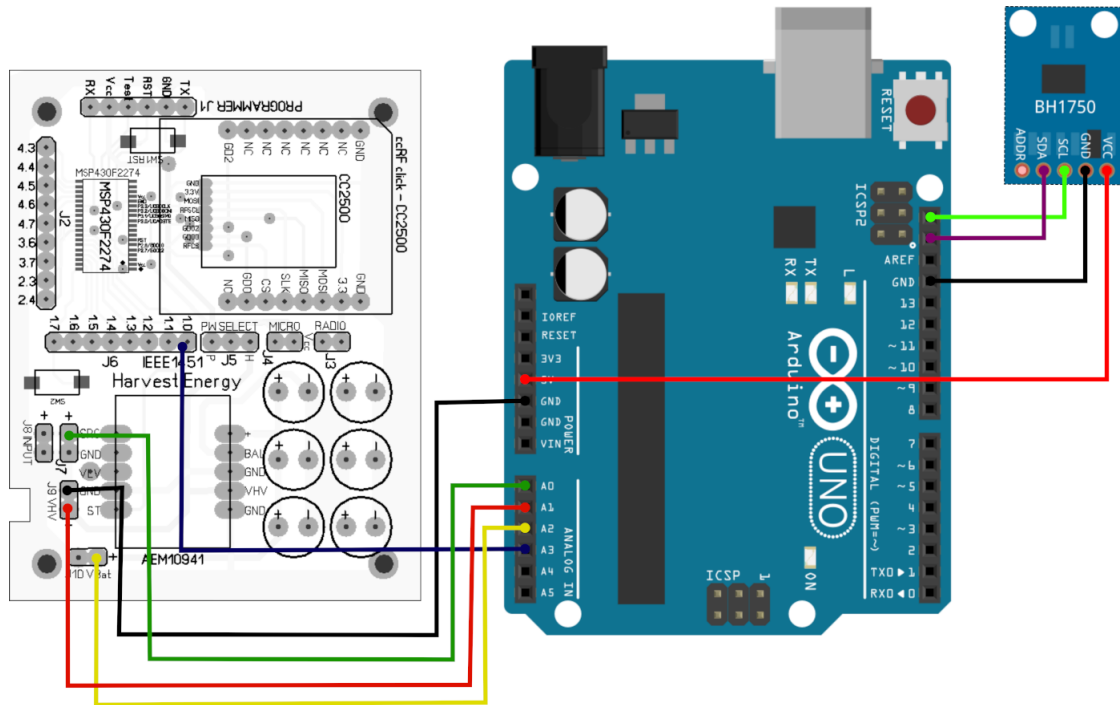


Figura 5.18 - Sistema de medição

Para a medição da corrente foi utilizada a placa X-NUCLEO-LPM01A da *STMicroelectronics*, Figura 8 do Apêndice F. Estas duas formas, de aquisição de dados, não são a melhor solução para o fazer, mas foi a forma possível. De notar que cada entrada analógica possui uma impedância de 100 M $\Omega$ , pelo que irá consumir energia, diminuindo o tempo de descarga e aumentando o tempo de carga.

A Figura 3.19, apresenta o gráfico da tensão em função do tempo para o carregamento dos condensadores em *cold start*. Para uma iluminância com um valor médio de aproximadamente 500 lux ou lm m<sup>-2</sup>, os condensadores demoraram 6420 segundos (~ 107 minutos) a carregar até uma tensão de 4,5 V, ou seja, carga completa.

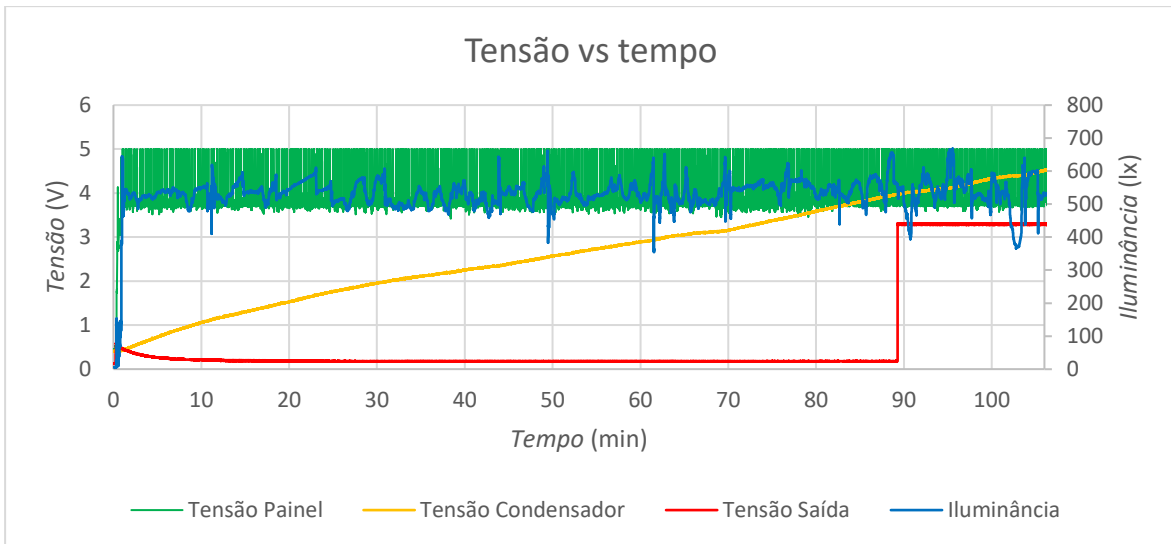


Figura 5.19 - Gráfico de  $U(t)$ , tensões do conversor em *cold start*

A Figura 5.20, apresenta o gráfico para as tensões: do painel, dos condensadores e da saída em função do tempo e em função da iluminância.

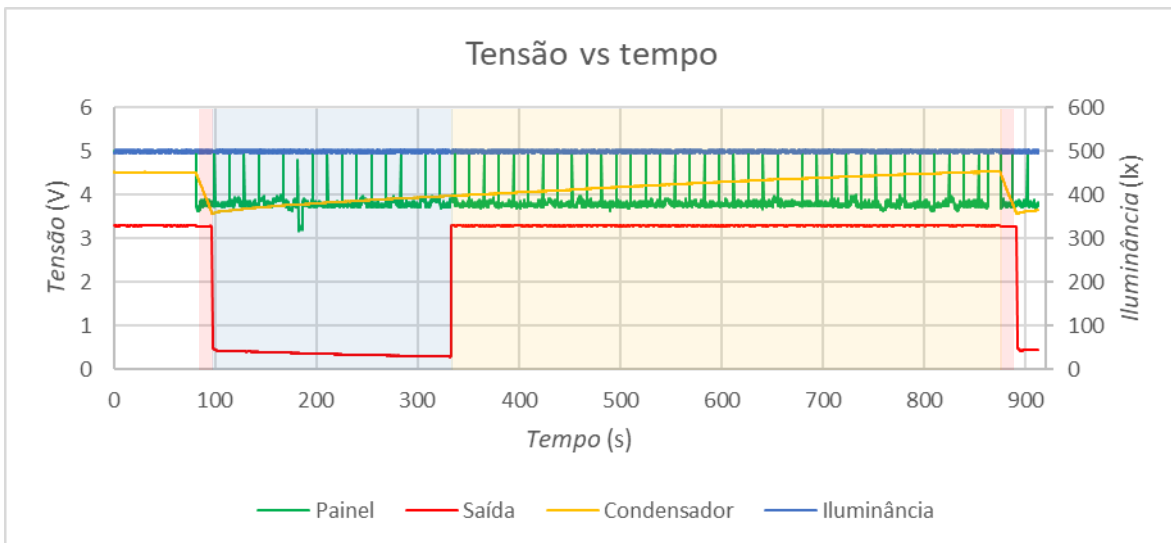


Figura 5.20- Gráfico de  $U(t)$ , tensões do conversor

A linha vermelha representa a tensão de saída (que alimenta o rádio a partir do ponto de ligação J3 e o microcontrolador a partir do ponto de ligação J4) que é ativada quando a tensão dos condensadores atinge o valor de 3,92V e desativada quando a tensão dos condensadores atinge o valor de 3,60V.

A linha amarela representa a tensão dos condensadores e mostra ciclos de carga e descarga ao longo do tempo.

A linha verde apresenta a tensão do painel fotovoltaico. Os picos de tensão surgem quando o conversor deixa temporariamente de carregar para encontrar o melhor regime de funcionamento (MPPT).

Para uma iluminância de aproximadamente 500 lux ou  $\text{lm m}^{-2}$ , os condensadores demoraram 235 segundos ( $\sim 4$  minutos) a carregar até uma tensão de 3,92 V mais 530 segundos ( $\sim 9$  minutos) até uma tensão de 4,5 V, ou seja, carga completa.

No pior cenário de funcionamento, com o microcontrolador e o rádio em modo RX em funcionamento, os condensadores demoraram aproximadamente 17 segundos a descarregar. Na Figura 5.21 é evidenciado o tempo obtido experimentalmente para a descarga dos condensadores entre estas duas tensões.

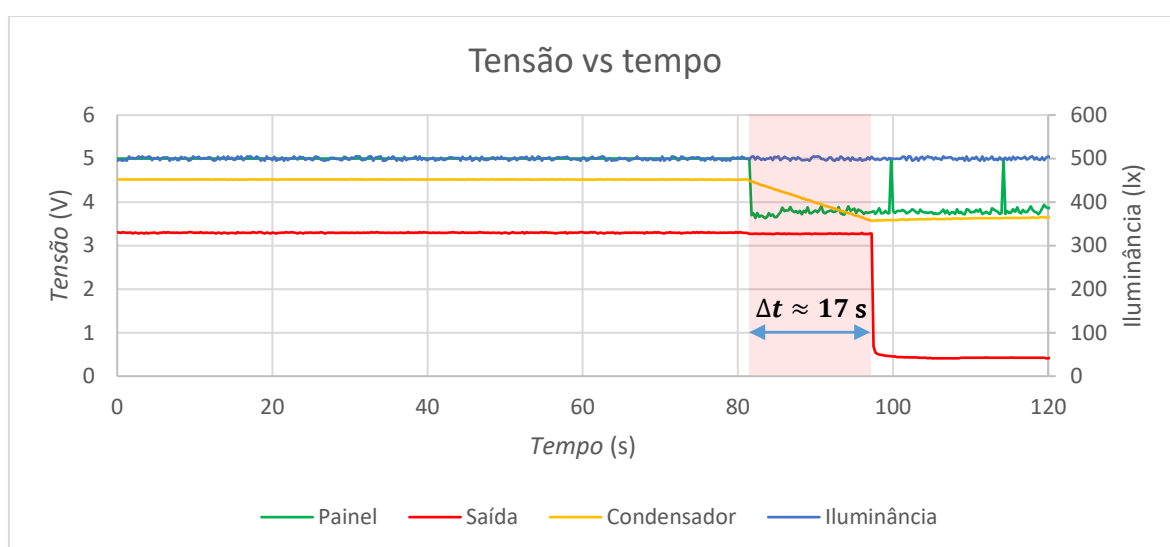


Figura 5.21 - Gráfico de  $U(t)$ , tempo de descarga na pior situação

O valor obtido experimentalmente foi inferior ao calculado em 4.2.3, como seria expectável, pois no cálculo teórico não foram contabilizados os consumos do AEM10941, o autodescarregamento dos condensadores e o próprio consumo da leitura das tensões.

A Figura 5.22 apresenta a corrente “consumida” pela TIM em função do tempo, aquando da receção do comando de *Read Data-set* do *proxy*, da respetiva resposta e da receção do comando *Sleep*. Assim que a TIM recebe este comando envia a resposta de sucesso e entra em modo de *Sleep* desligando o rádio.

As zonas marcadas a azul correspondem ao estado de microcontrolador ativo e rádio em modo RX, as zonas marcadas a violeta correspondem ao estado de microcontrolador ativo e rádio em modo TX.

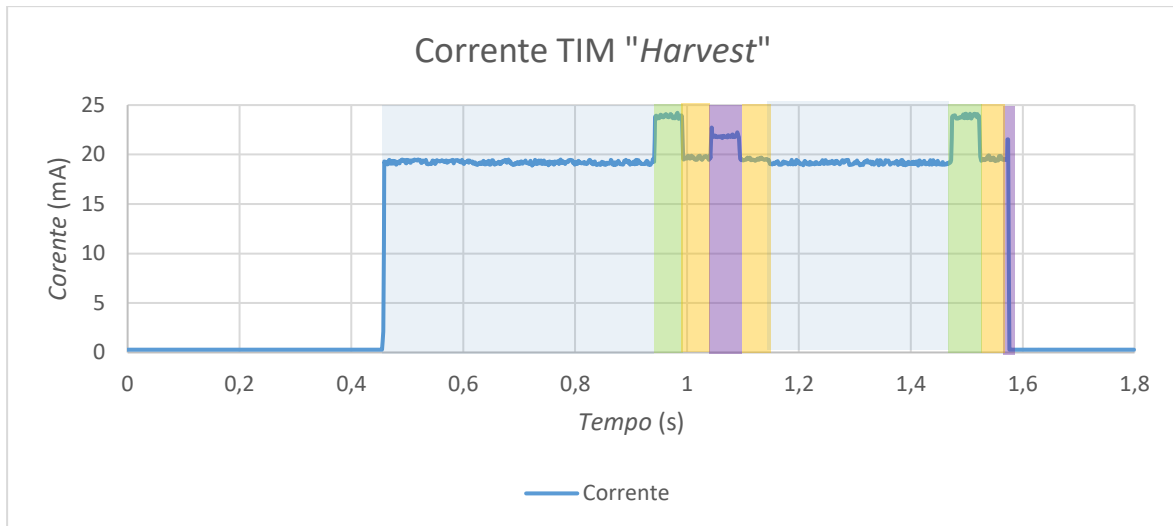


Figura 5.22 - Gráfico  $I(t)$ , TIM *Harvest*

A partir da equação (eq. 4.13) é possível calcular a energia utilizada pela TIM *Harvest*. Onde a área abaixo da linha corresponde ao produto entre a corrente,  $I$ , e o intervalo de tempo,  $\Delta t$ , que multiplicado pela tensão de 3,3V se obteve o valor experimental da energia de 0,0736 J.

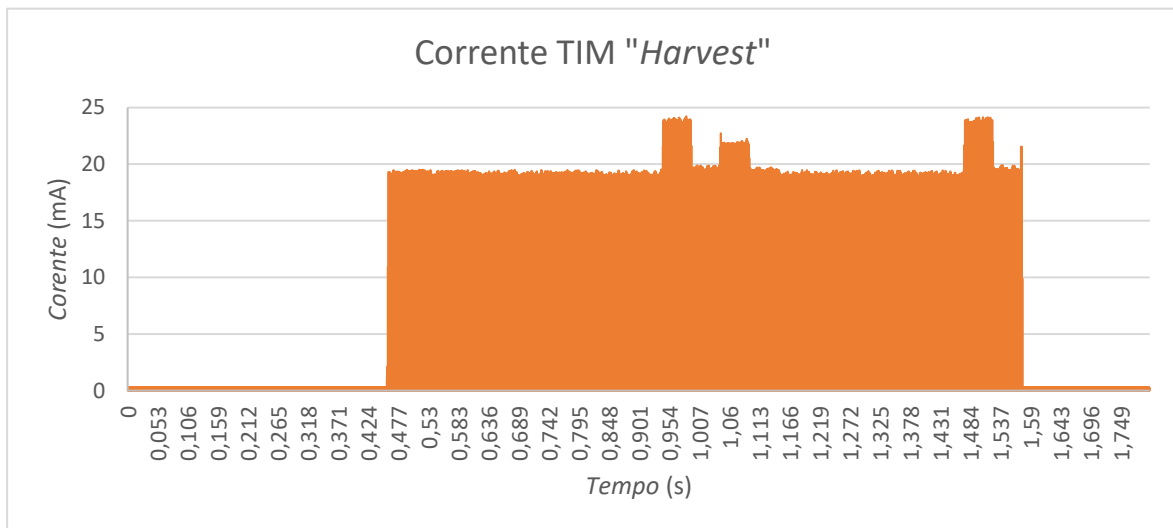


Figura 5.23 - Gráfico  $I(t)$ , cálculo da energia

O valor de energia utilizada neste processo é cerca de 24,7 vezes menor ao valor disponibilizado pelos condensadores. Provando que é possível utilizar este sistema de fonte de energia.

$$\frac{E_{disponivel}}{E_{utilizada}} = \frac{(5,06 - 3,24)}{0,0736} \Leftrightarrow \frac{E_{disponivel}}{E_{utilizada}} = 24,7 \Leftrightarrow E_{utilizada} = \frac{1}{24,7} \times E_{disponivel} \quad (\text{eq. 5.5})$$

## 5.5. Referência Bibliográfica

- [1] X. Cheng, F. Shen and M. Kuga, "Design of the wireless temperature supervisory control system based on eZ430-RF2500," 2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE), 2012, pp. 729-732, doi: 10.1109/CSAE.2012.6272870.
- [2] Texas Instruments, "MSP430F22x2, MSP430F22x4 Mixed Signal Microcontroller", MSP430F2274 folha de características, julho 2006 [Revisto agosto 2012].
- [3] Texas Instruments, "CC2500, Low-Cost Low-Power 2.4 GHz RF Transceiver", CC2500 folha de características, janeiro 2005 [Revisto junho 2006].
- [4] P. Singh and S. Saikia, "Arduino-based smart irrigation using water flow sensor, soil moisture sensor, temperature sensor and ESP8266 WiFi module," 2016 IEEE Region 10 Humanitarian Technology Conference (R10-HTC), 2016, pp. 1-4, doi: 10.1109/R10-HTC.2016.7906792.
- [5] C. Rajurkar, S. R. S. Prabakaran and S. Muthulakshmi, "IoT based water management," 2017 International Conference on Nextgen Electronic Technologies: Silicon to Software (ICNETS2), 2017, pp. 255-259, doi: 10.1109/ICNETS2.2017.8067943.
- [6] Sea "YF-S201, "Water Flow Sensor", YF-S201 folha de características, [online], Disponível: [http://www.mantech.co.za/datasheets/products/yf-s201\\_sea.pdf](http://www.mantech.co.za/datasheets/products/yf-s201_sea.pdf), [Acedido: 09-03-2021]
- [7] K. Loizou, E. Koutroulis, D. Zalikas and G. Liontas, "A low-cost capacitive sensor for water level monitoring in large-scale storage tanks," 2015 IEEE International Conference on Industrial Technology (ICIT), 2015, pp. 1416-1421, doi: 10.1109/ICIT.2015.7125295.
- [8] K. Loizou, E. Koutroulis, "Water level sensing: State of the art review and performance evaluation of a low-cost measurement system", Measurement, 2016, pp. 204-214, 89, doi: 10.1016/j.measurement.2016.04.019.
- [9] SIEMENS, "SIEMENS Inversor SINAMICS V20", SINAMICS V20 folha de características, fevereiro 2013 [Revisto junho 2006].
- [10] Qt Creator, "Qt Creator - A Cross-platform IDE for software development", [online], Disponível: <https://www.qt.io/product/development-tools>, [Acedido: 09-03-2021]

- [11] Arduino, “Arduino, open-source electronic prototyping platform”, [online], Disponível: <https://www.arduino.cc>, [Acedido: 09-05-2021]
- [12] SmarTTY, “SmarTTY, multi-tabbed SSH client”, [online], Disponível: <https://sysprogs.com/SmarTTY>, [Acedido: 05-02-2019]

# Capítulo 6

## Conclusão

### 6.1. Conclusões Gerais

A norma IEEE1451 define um transdutor como um elemento que pode medir e atuar num processo. Esta família de normas estabelece aspetos essenciais na construção de transdutores que operem em rede.

A adoção da norma exige o seu conhecimento e compreensão por parte dos agentes industriais, reconhecendo o seu potencial de aplicação e funcionamento interno.

Para alcançar o objetivo, desta tese, estudou-se toda a norma IEEE1451, mas mais em pormenor a norma IEEE1451.0. A utilização generalizada da norma IEEE1451, permite o desenvolvimento de um sistema de sensores fiável e adaptado às necessidades de utilização. O desenvolvimento manual, com recurso a esta norma, é um trabalho complexo, moroso e requer o conhecimento profundo da mesma. Qualquer pequena alteração no projeto a desenvolver requer, obrigatoriamente, a reconfiguração das TEDS, tornando o seu desenvolvimento um trabalho complexo. Com estas condicionantes, foi necessário criar uma plataforma de desenvolvimento. Com a utilização destas ferramentas de desenvolvimento é possível dimensionar rapidamente, e de forma ágil, um sistema de instrumentação compatível com a norma IEEE1451. Além disto permite, a quem inicia o estudo e o entendimento da norma, fazê-lo de uma forma mais fácil e interativa.

As ferramentas de desenvolvimento facilitam o trabalho de criação de transdutores e servem como ferramentas de ensino, pois são uma mais-valia. As ferramentas de desenvolvimento apresentadas neste estudo contribuem para o esforço de divulgação do padrão IEEE1451. Estas encontram-se online de acesso e livre, após o registo. A plataforma sistematiza a construção de uma TIM, auxilia na codificação das TEDS e cria código C/C++ para programação da TIM. O projeto retornado pela ferramenta é compatível com o *Code Composer Studio*. Dois kits de *hardware* são suportados pela plataforma e caracterizam-se pelo baixo consumo de energia. Durante os testes de validação da TIM, a plataforma começa por detetá-la, baixa as TEDS e armazena-as na base de dados. O utilizador pode validar os comandos enviados à TIM através da NCAP. Todo o processo anteriormente descrito pode ser observado e validado na ferramenta de validação.

Esta plataforma de desenvolvimento conta ainda com outras secções igualmente importantes para o entendimento da norma, como um tutorial da norma e um tutorial da utilização da plataforma.

Com esta plataforma é possível construir um sistema de Aplicação, NCAP e TIMs Wireless permitindo o estudo e a aplicabilidade desta norma em sensores sem fios com baixo recurso energético.

Embora esta norma, seja complexa, possui a flexibilidade necessária para que a aplicação de sensores sem fios baseados em energia *harvest* funcione.

Ao longo do desenvolvimento deste trabalho surgiram algumas dificuldades, mais concretamente em encontrar ferramentas de desenvolvimento para a norma IEEE1451. Dificuldades que foram superadas na construção de uma plataforma de desenvolvimento que pode agora ser utilizada de forma generalizada no site “<http://iml.ubi.pt/ieee1451>”. Com este trabalho acrescido permitiu adquirir um conhecimento aprofundado da norma, critério necessário para a formulação de uma rede de sensores compatível com a norma IEEE1451.

No decorrer dos testes, de sensores sem fios, verificou-se muito interferência no canal de rádio frequência utilizado, interferência criada pela rede Wi-Fi (*Wireless Fidelity*) da universidade. Para minimizar esta interferência foi alterada a frequência portadora de Transmissão das TIM para a frequência suportada pelo rádio CC2500, mas o mais afastada possível dos canais Wi-Fi.

Para validar o sistema de Aplicação/NCAP/TIM foi desenvolvido uma Montagem Experimental constituída por três TIM, uma NCAP, uma aplicação e uma TIM alimentada com energia *harvest*. Ficou demonstrado que é possível desenvolver um sistema de sensores inteligentes, que seja compatível com a norma IEEE1451, utilizando uma pequena quantidade de energia.

Com este estudo ficaram bases sólidas para o desenvolvimento de novas abordagens utilizando a norma, mas também para o estudo de outras fontes de energia *harvest* que possam ser utilizadas. Deixa um legado de uma panóplia de ferramentas, especialmente escritas para a norma, que facilitará em muito o desenvolvimento de sistemas que utilizem a norma IEEE1451. Embora estas ferramentas estejam descritas nesta tese, a sua exploração e utilização são uma mais-valia como forma de as entender plenamente.

Com este estudo foi deixado, para trabalhos futuros, um legado de um conjunto de ferramentas com milhares de linhas de código escrito, estudo de fontes e formas de aquisição de energia *harvest* aplicada à norma IEEE1451.

Com o estudo e o cálculo teórico da energia requerida para a TIM *harvest*, foi possível dimensioná-la, escolhendo como fonte de alimentação um painel fotovoltaico em ambiente interior, um *booster* de energia e a forma de armazenar essa energia. Com o

estudo da forma de armazenamento foi possível o cálculo teórico do tempo e da energia disponíveis no pior dos casos. Com a construção física desta TIM foi possível concluir que o valor calculado teoricamente teve um valor bastante aproximado do valor medido na prática. Para rentabilizar a energia disponível foi realizada a conceptualização e implementação de um algoritmo de gestão de energia que permitiu gerir a energia respeitando a norma IEEE1451 na integra.

Ficou demonstrado no Capítulo 5, que o objetivo inicial deste estudo foi concretizado, ficando provado que, é possível utilizar a norma IEEE1451 numa rede de sensores sem fios com e sem restrições de energia. De salientar que a TIM com restrições energéticas consegue responder 24 vezes com uma carga completa, pelo que com a configuração adequada consegue enviar informações 1 vez a cada hora durante um dia.

## 6.2. Propostas de adições à norma

A criação das “ferramentas de desenvolvimento” e a implementação realizada permitiu o conhecimento da norma IEE1451. Permitiu ainda concluir que é possível construir um sistema, que respeite a norma, com e sem fios, com e sem limitações de energia e ainda sugerir adições à norma que a tornariam mais compatível com sistemas alimentados por fontes de energias *harvest*, tais como:

- Todos os comandos enviados da NCAP para uma TIM devem obter sempre uma resposta. A existência de comandos que não necessitam de resposta deixa sempre a NCAP na dúvida da sua execução por parte da TIM.
- A resposta deveria conter o número de identificação da TIM e do canal transdutor. Libertando a NCAP para executar outros processos, não havendo a necessidade da NCAP esperar por uma possível resposta, que ao não ser concretizada terá de esperar até ao *Operational Time-out*, descrito na *Meta-TEDS* da TIM.
- A TIM poderia ter a possibilidade de enviar dados para a NCAP sem a necessidade de ter de esperar por um comando da NCAP. Esta impossibilidade obriga à TIM a despendar energia para a escuta do comando, comando este que nem sempre existirá. Isto obriga a TIM a ligar o rádio e despendar energia. Se a norma desse permissão à TIM de solicitar a comunicação com a NCAP, seria vantajoso em termos de energia.
- A adição de TEDS de energia permitiria caracterizar uma TIM que possua restrições energéticas, descrito em 4.4. tornando a norma energeticamente mais eficiente com este tipo de TIM.
- A existência de um comando *Read Available TEDS*, antes do comando *Read Query TEDS* permitiria saber de imediato todas as TEDS suportadas pela TIM. O comando teria a seguinte estrutura:

Estrutura de Mensagem do Comando <i>Read Available TEDS</i>							
ID de Destino		ID Canal Transdutor		Classe de Comando	Função de Comando	Tamanho	Tamanho
0x00	0x01	0x00	0x01	0x01	0x0D	0x00	0x00

Tabela 6.1 - Estrutura de mensagem do comando *Read Available TEDS*

A resposta de uma TIM que possui apenas as TEDS obrigatórias, teria a seguinte estrutura:

Estrutura da Mensagem da Resposta <i>Read Available TEDS</i>				
Sucesso/Falha	Tamanho	Tamanho	Valor da Resposta	
			Disponível1	Disponível2
0x01	0x00	0x02	0xA2	0x18

Tabela 6.2 - Estrutura da mensagem da resposta *Read Available TEDS*

Os dois octetos dependentes, configurados bit a bit, indicariam as TEDS existentes na TIM de acordo com a seguinte estrutura:

Available1	lsb	0	Meta-TEDS	Se verdadeiro a Meta-TEDS está presente.	1
		1	MetaIdTEDS	Se verdadeiro a Meta-identification TEDS está presente.	0
		2	ChanTEDS	Se verdadeiro a TransducerChannel está presente.	1
		3	ChanIdTEDS	Se verdadeiro a TransducerChannel Id está presente.	0
		4	CalTEDS	Se verdadeiro a Calibration TEDS está presente.	0
		5	CalIdTEDS	Se verdadeiro a Calibration Id TEDS está presente.	0
		6	EUASTEDS	Se verdadeiro a End users TEDS está presente.	1
	msb	7	FreqRespTEDS	Se verdadeiro a Frequency response TEDS está presente.	0
					0xA2
Available2	lsb	0	TransferTEDS	Se verdadeiro a Transfer Function TEDS está presente.	
		1	CommandTEDS	Se verdadeiro a Commands TEDS está presente.	0
		2	TitleTEDS	Se verdadeiro a Location & Title TEDS está presente.	0
		3	XdcrName	Se verdadeiro a Userss Transducer Name TEDS está presente.	1
		4	PHYTEDS	Se verdadeiro a PHY TEDS está presente.	1
		5	GeoLocTEDS	Se verdadeiro a Geographic Location TEDS está presente.	0
		6	UnitsExtention	Se verdadeiro a Units Extention TEDS está presente.	0
	msb	7	EnergyTEDS	Se verdadeiro a EnergyTEDS está presente.	0
					0x18

Tabela 6.3 - Estrutura do comando *Read Available TEDS*

Com esta adição a NCAP iria perder menos tempo na procura das TEDS existentes e conseqüentemente a TIM iria “consumir” menos energia na altura do *handshake*.

- Na secção da Meta-TEDS, mais propriamente no *Manufacture Globally Unique Identifier* é codificada num total de 80 bits.

Para indicar os pontos cardeais estão reservados dois bits, um bit para indicar o ponto cardinal Norte ou Sul e outro bit para indicar o ponto cardinal Este ou Oeste, mas não há referência a qual deles corresponde o valor de zero ou de um. Propunha a atribuição do valor de “1” para o ponto cardinal Norte e do valor de “0” para o ponto cardinal Sul. O valor de “1” para o ponto cardinal “Este” e o valor de “0” para o ponto cardinal Oeste.

Para o *Manufacturing Time* estão reservados 22 bits, quando se calcula pela equação:

$$\text{Manufacturing Time} = \text{Day of the Year} \times 1000 + \text{Sequence Number} \quad (\text{eq. 6.1})$$

o número de dispositivos fica limitado a 1000 por dia, pois se for superior, embora ainda tenha espaço (nos bits reservados) irá corromper o *Day of The Year*. Propunha a alteração da fórmula para:

$$\text{Manufacturing Time} = (\text{Day of the Year} \times 1000) \& \text{Sequence Number} \quad (\text{eq. 6.2})$$

Na eq. 6.2 o *Sequence Number* é colocado ao lado em vez de ser somado, ficando 8 bits para o dia do ano e 14 bits para o *Sequence Number*. Com esta pequena alteração eleva-se para 16384 o número de dispositivos que podem ser fabricados por dia.

- Os sistemas embutidos estão vinte anos à frente do início da formulação desta norma. Estes possuem atualmente poder computacional suficiente para englobarem diretamente a função de NCAP e de TIM, pelo que propunha a junção da mesma num só *hardware*.
- No Capítulo 2.8.7 - TEDS, estão descritas de uma forma completa as TEDS obrigatórias de acordo com a norma. As restantes não se encontram descritas dificultando a sua consulta. Sendo as TEDS uma parte fundamental da norma, propunha a escrita de um único documento que descrevesse todas as TEDS.

- Considero que seria uma mais-valia colocar na PHY TEDS um campo que informasse que a TIM não suporta comunicações sem fios e outro campo que indicasse qual o protocolo utilizado numa rede com fios.

### **6.3. Linhas Orientadoras para Futuras Contribuições**

Com este estudo ficaram instaladas infraestruturas físicas para que quem continuar esta pesquisa, possa utilizar a norma IEEE1451, sem ter a preocupação da instalação e configuração de *hardware* e *software*.

Abriram-se assim portas para a implementação, simplificada, de sensores e atuadores para a norma IEEE1451, permitindo agora aprofundar estudos para a sua aplicabilidade na indústria. Com as ferramentas diminui-se o tempo de desenvolvimento, permitindo assim dedicar o estudo a outros temas.

Poderá ser possível completar a implementação integral da TIM e da NCAP, expandir para novas placas com processadores e periféricos diferentes.

Devido à falta de material e de equipamento de medição no laboratório foram descurados alguns pormenores que poderão ser melhorados, tais como:

- A medição individual da corrente nas secções da TIM *harvest*: rádio e microcontrolador;
- A comparação de diferentes tipos de painéis fotovoltaicos;
- A comparação de diferentes formas de armazenamento de energia;
- A comparação de diferentes conversores de energia *harvest*.

Abriu ainda portas para o estudo da utilização de outras fontes de energia *harvest* neste tipo de sensores. Bem como a conjugação de mais do que uma fonte, criando sistemas híbridos de recolha de energia.

### **6.4. Contribuições Resultantes da Pesquisa deste Estudo**

A primeira contribuição foi a implementação de uma TIM que respeita a norma IEEE1451, onde foram implementados grande número de comandos, TEDS e canais transdutores com a função de sensor, sensor de eventos e atuadores.

A segunda contribuição foi a criação de uma plataforma de desenvolvimento para a norma IEEE1451, que contém um conjunto de ferramentas necessárias para desenvolver uma TIM. Todas estas ferramentas possuem tutoriais de utilização para facilitar a sua utilização.

No decorrer deste estudo, construí o sítio da internet do Laboratório de Instrumentação e Medida e o sítio para CoE (*Center of Expertise Consortium*) onde é possível visualizar os parceiros constituintes deste projeto.

#### 6.4.1. Lista de Publicações

- R. Pinto, J. Pereira, H. da Rocha, R. I. Martin and A. Espírito-Santo, "A Discussion about the Implementation of a WSN to Industry 4.0 based on the IEEE 1451 Standard," 2019 IEEE 17th International Conference on Industrial Informatics (INDIN), 2019, pp. 1573-1578, doi: 10.1109/INDIN41052.2019.8972222.
- J. L. D. Pereira, H. da Rocha and A. E. Santo, "A Platform for IEEE 1451 Standard's Education, Development and Validation for Industry 4.0," 2020 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), 2020, pp. 1-6, doi: 10.1109/I2MTC43012.2020.9129114.
- J. L. Dâmaso Pereira and A. Espírito-Santo, "Energetic Evaluation and Management of Low-Power Wireless Smart Transducers," IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society, 2020, pp. 4572-4577, doi: 10.1109/IECON43393.2020.9254414.
- H. da Rocha, R. Abrishambaf, J. L. Dâmaso Pereira e A. Espírito Santo, "Integrating the IEEE 1451 and IEC 61499 Standards with the Industrial Internet Reference Architecture", MDPI Sensors, Published: 15 fevereiro 2022, doi: 10.3390/s22041495.
- H. da Rocha, J. L. D. Pereira, T. A. G. N. Rodrigues, J. A. Salvado and A. Espirito-Santo, "An Energy-Efficient Process for Optimal Communication Synchronization in Low Power Wireless Smart Sensors," 2022 IEEE International Symposium on Measurements & Networking (M&N), 2022, pp. 1-6, doi: 10.1109/MN55117.2022.9887722.
- J. D. Pereira and A. Espírito-Santo, "Open Access Online Platform for Transducers Development Conforming the IEEE 1451," 2022 IEEE International Symposium on Measurements & Networking (M&N), 2022, pp. 1-6, doi: 10.1109/MN55117.2022.9887739.
- M. Moreira, H. da Rocha, J. L. Dâmaso Pereira, J.A. Salvado e A. Espírito Santo, "Smart Transducers Promoting Smart Cities Interoperability", IECON2022, doi: 10.1109/IECON49645.2022.9968431

- H. da Rocha; J. Pereira; R. Abrishambaf; A. Espírito Santo, “*An Interoperable Digital Twin with the IEEE 1451 Standards*”, MDPI Sensors, 2022, doi: 10.3390/s22197590.

#### **6.4.2. Participação em Conferências Científicas, no País e no Estrangeiro**

- INDITECH 4.0 – Novas tecnologias para a fabricação inteligente, Workshop, FEUP, Porto, 12 de julho.
- INTEROP – Standards Interoperability and Plugfest - IEEE 45th Annual Conference of the IEEE Industrial Electronics Society – IECON 2019, Lisboa, 17 a 19 de outubro.
- INTEROP – Standards Interoperability and Plugfest - IEEE 46th Annual Conference of the IEEE Industrial Electronics Society – IECON 2020, Singapore, from October 18 to 21. (online).
- IEEE 46th Annual Conference of the IEEE Industrial Electronics Society – IECON 2020, Singapore from October 18 to 21.
- INTEROP – Standards Interoperability and Plug fest – The 30th International Symposium on Industrial Electronics – ISIE 2021, Kyoto, Japan from June 20 to 23 (online).
- IEEE 47th Annual Conference of the IEEE Industrial Electronics Society – IECON 2021, Canada from October 13 to 16 (online).
- IEEE International Symposium on Measurements & Networking, M&N 2022, Padua, Italy from July 17.

# Apêndice A - COMANDOS NCAP → TIM

Destination TIM ID		Destination TransducerChannel Number		Class Command	Function Command	Lenght		Command Dependent
1 Octet	1 Octet	1 Octet	1 Octet	1 Octet	1 Octet	1 Octet	1 Octet	N Octets

cmdClassId	Attribute name	Category
0	Reserved	Reserved
1	CommonCmd	Commands common to the TIM and TransducerChannel
2	XdcrIdle	Transducer idle state

**Table 16—Common Commands (page 75)**

cmdFunctionId	Command	State		Reply expected	Required/optional
		TransducerChannel	TIM		
0	Reserved	—	—	—	—
1	Query TEDS	Any	Active	Yes	Required
2	Read TEDS segment	Any	Active	Yes	Required
3	Write TEDS segment	Any	Active	No	Required
4	Update TEDS	Any	Active	Yes	Required
5	Run self-test	Idle	Active	No	Required
6	Write service request mask	Any	Active	No	Required
7	Read service request mask	Any	Active	Yes	Required
8	Read status-event register	Any	Active	Yes	Required
9	Read status-condition register	Any	Active	Yes	Required
10	Clear status-event register	Any	Active	No	Required
11	Write status-event protocol state	Idle	Active	No	Required
12	Read status-event protocol state	Any	Active	Yes	Required
13–127	Reserved	—	—	—	—
128–255	Open for manufacturers	—	—	—	—

**Table 25—Common Commands (page 81)**

cmdFunctionId	Command	Address class			Reply expected	Required/optional
		TransducerChannel	Proxy	Group		
0	Reserved	—	—	—	—	—
1	Set TransducerChannel data repetition count	Yes	No	No	No	Optional
2	Set TransducerChannel pre-trigger count	Yes	No	No	No	Optional
3	AddressGroup definition	Yes	Yes	No	No	Required
4	Sampling mode	Yes	Yes	No	No	Optional
5	Data Transmission mode	Yes	Yes	Yes	No	Optional
6	Buffered state	Yes	Yes	No	No	Optional
7	End-of-data-set operation	Yes	Yes	No	No	Optional
8	Actuator-halt mode	Yes	Yes	No	No	Optional
9	Edge-to-report	Yes	Yes	No	No	Optional
10	Calibrate TransducerChannel	Yes	Yes	Yes	Yes	Optional
11	Zero TransducerChannel	Yes	Yes	Yes	Yes	Optional
12	Write corrections state	Yes	Yes	No	No	See 7.1.2.12
13	Read corrections state	Yes	Yes	No	No	See 7.1.2.13
14	Write TransducerChannel initiate trigger state	Yes	No	No	No	Optional
15	Write TransducerChannel initiate trigger configuration	Yes	No	No	No	Optional
16–127	Reserved	—	—	—	—	—
128–255	Open for manufacturers	—	—	—	—	—

NOTE—Optional commands become required if the features they support are programmable.

**Table 17—TEDS access codes (page 76)**

TEDS access code	TEDS name attribute	TEDS	Required/optional
0	—	Reserved	—
1	MetaTEDS	Meta-TEDS1	Required
2	MetaIdTEDS	Meta-identification TEDS2	Optional
3	ChanTEDS	TransducerChannel TEDS1	Required
4	ChanIdTEDS	TransducerChannel Identification TEDS2	Optional
5	CalTEDS	Calibration TEDS1	Optional
6	CalIdTEDS	Calibration identification TEDS2	Optional
7	EUASTEDS	End users' application-specific TEDS3	Required
8	FreqRespTEDS	Frequency response TEDS1	Optional
9	TransferTEDS	Transfer function TEDS1	Optional
10	CommandTEDS	Commands TEDS2	Optional
11	TitleTEDS	Location and title TEDS2	Optional
12	XdcrName	User's transducer name TEDS3	Required
13	PHYTEDS	PHY TEDS1	Required
14	GeoLocTEDS	Geographic location TEDS2	Optional
15	UnitsExtention	Units extention TEDS2	Optional
16–127	—	Reserved	—
128–255	—	Manufacturer-defined TEDS	Optional

NOTES  
 1—A binary TEDS.  
 2—A text-based TEDS.  
 3—User-defined information content.

**Table 21—Data field for a read TEDS segment command (page 77)**

Field	Data type	Field attribute name	Function
1	UInt8	TEDSAccessCode	TEDS access code, as defined in Table 17.
2	UInt32	TEDSOffset	TEDS Segment offset (0 to [current size - 1])—This is the address relative to the beginning of the TEDS at which the block of data shall be read.

**Table 23—Data field for a write TEDS command (page 78)**

Field	Data type	Field attribute name	Function
1	UInt8	TEDSAccessCode	TEDS access code, as defined in Table 17.
2	UInt32	TEDSOffset	TEDS Segment offset (0 to [current size - 1])
3	OctetArray	RawTEDSBlock	TEDS Contents

3	XdcrOperate	Transducer operating state
4	XdcrEither	Transducer either idle or operating state
5	TIMsleep	Sleep state
6	TIMActive	Tim active state commands
7	AnyState	Any state
8–127	ReservedClass	Reserved
128–255	ClassN	Open for manufacturers – N = class number

**Table 31—Transducer operating state commands (page 86)**

cmdFunctionId	Command	Address class			Reply expected	Required/optional
		TransducerChannel	Proxy	Group/global		
0	Reserved	—	—	—	—	
1	Read TransducerChannel data-set segment	Yes	Yes	No	Yes	See NOTE
2	Write TransducerChannel data set segment	Yes	Yes	No	No	See NOTE
3	Trigger command	Yes	Yes	Yes	No	Required
4	Abort Trigger	Yes	Yes	Yes	No	Optional
5–127	Reserved	—	—	—	—	—
128–255	Open for manufacturers	—	—	—	—	—

NOTE—A Read TransducerChannel data set segment command is required for sensors. A Write TransducerChannel data set segment command is required for an actuator.

**Table 35—Sleep state Command (page 92)**

cmdFunctionId	Command	Address class		Reply expected	Required/optional
		TIM	Global		
0	Reserved	—	—	—	—
1	Wake-up	Yes	No	Yes	Optional
2–127	Reserved	—	—	—	—
128–255	Open for manufacturers	—	—	—	—

**Table 36—TIM active state commands (page 93)**

cmdFunctionId	Command	Address class		Reply expected	Required/optional
		TIM	Global		
0	Reserved	—	—	—	—
1	Read TIM version	Yes	No	Yes	Required
2	TIM Sleep	Yes	No	No	Optional
3	Store operational setup	Yes	No	No	Required
4	Recall operational setup	Yes	No	No	Required
5	Read IEEE 1451.0 Version	Yes	No	Yes	Required
6–127	Reserved	—	—	—	—
128–255	Open for manufacturers	—	—	—	—

**Table 37—Enumeration of IEEE 1451.0 version numbers (page 94)**

Standard version	IEEE 1451.0 standard version
0	Reserved for prototype or other nonstandard IEEE 1451.0 versions
1	This will correspond to the original release of this standard
2–127	Reserved for future releases of the standard
128–255	Open to manufacturers

# Apêndice B - Estrutura DATA BLOCK da Meta-TEDS

Table 17—TEDS access codes (page 76)

TEDS access code	TEDS name attribute	TEDS	Required/optional
0	—	Reserved	—
1	MetaTEDS	Meta-TEDS1	Required
2	MetaIdTEDS	Meta-identification TEDS2	Optional
3	ChanTEDS	TransducerChannel TEDS1	Required
4	ChanIdTEDS	TransducerChannel Identification TEDS2	Optional
5	CalTEDS	Calibration TEDS1	Optional
6	CalIdTEDS	Calibration identification TEDS2	Optional
7	EUASTEDS	End users' application-specific TEDS3	Required
8	FreqRespTEDS	Frequency response TEDS1	Optional
9	TransferTEDS	Transfer function TEDS1	Optional
10	CommandTEDS	Commands TEDS2	Optional
11	TitleTEDS	Location and title TEDS2	Optional
12	XdcrName	User's transducer name TEDS3	Required
13	PHYTEDS	PHY TEDS1	Required
14	GeoLocTEDS	Geographic location TEDS2	Optional
15	UnitsExtention	Units extention TEDS2	Optional
16-127	—	Reserved	—
128-255	—	Manufacturer-defined TEDS	Optional

NOTES  
 1—A binary TEDS.  
 2—A text-based TEDS.  
 3—User-defined information content.

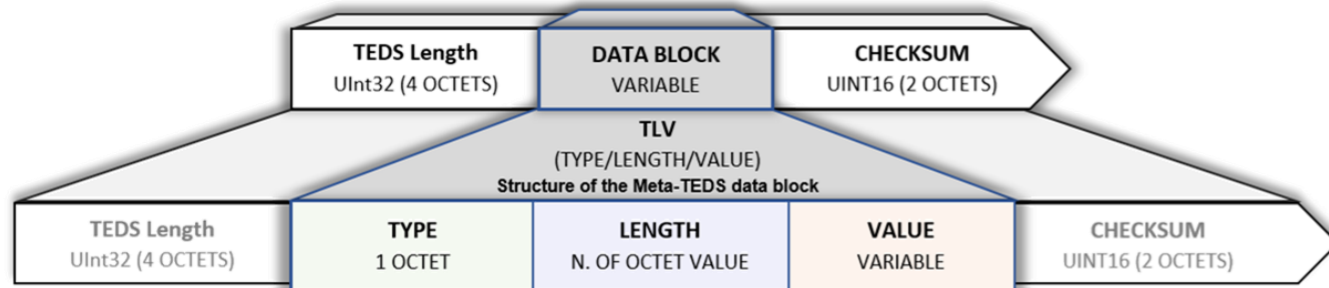


Table 43—Structure of the Meta-TEDS data block (page 99)

Field type	Field name	Description	Data type	# octets
—	—	Length	UInt32	4
0-2	—	Reserved	—	—
3	TEDSID	TEDS Identification Header	UInt8	4
4	UUID	Globally Unique Identifier	UUID	10
5-9	—	Reserved	—	—

Table 41—TEDS identifier structure (page 98)

Field	Contents	Function
Type	03	Type field for the TEDS Identifier.
Length	04	This field is always set to 04, indicating that the value field contains 4 octets.
Family	00	This field identifies the member of the IEEE 1451 family of standards that defines this TEDS.
Class	See Table 17	This field identifies the TEDS being accessed. The value is the TEDS access code found in Table 17.
Version	See Table 42	This field identifies the TEDS version. The value is the version number identified in the standard. A value of zero in this field indicates that the TEDS do not conform to any released standard. Table 42 lists the allowable values for this field.
Tuple Length	Number of octets	This field gives the number of octets in the length field of all tuples in the TEDS except this tuple.  NOTE—For most TEDS, the number of octets in the length field of the tuples is one, meaning that there are 255 or less octets in the value field. However, some cases may require more than 8 bits for the number of octets in the value field, so this field specifies the number of octets in the length field of a tuple. All tuples within a TEDS, except the TEDS Identifier, shall have the same number of octets in the length field.

TEDS version	IEEE 1451.0 standard version
0	Reserved for prototype or other nonstandard TEDS versions.
1	This will correspond to the original release of this standard.
2-255	Reserved for future releases of the standard.

Table 42—Enumeration of TEDS version numbers (page 98)

Number of implemented TransducerChannels				
13	MaxChan	Number of implemented TransducerChannels	UInt16	2
14	CGroup	<b>ControlGroup information sub-block</b>	—	—
Types 20, and 21 define one ControlGroup.				
20	GrpType	ControlGroup type	UInt8	1
21	MemList	ControlGroup member list	array of UInt16	NTc
15	VGroup	<b>VectorGroup information sub-block</b>	—	—
Types 20 and 21 define one VectorGroup.				
20	GrpType	VectorGroup type	UInt8	1
21	MemList	VectorGroup member list	array of UInt16	NTv

Enumeration	Member list order	Function
0	—	Reserved
1	1	Analog event sensor TransducerChannel
	2	Analog input sensor TransducerChannel that measures the input value for the same input as the member 1 event sensor provides the state
	3	Upper threshold embedded actuator TransducerChannel
	4	Hysteresis embedded actuator TransducerChannel
2	1	Sensor TransducerChannel (any type)
	2	High-pass filter embedded actuator TransducerChannel
	3	Low-pass filter embedded actuator TransducerChannel
	4	Scale factor embedded actuator TransducerChannel
3	1	TransducerChannel (any type)
	2	Sample interval embedded actuator TransducerChannel
4	1	Digital event sensor TransducerChannel
	2	Digital input sensor TransducerChannel that measures the input value for the same input as the member 1 event sensor provides the state
	3	Event pattern embedded actuator TransducerChannel
5	1	Time interval sensor TransducerChannel
	2	TransducerChannel number of the transducer that causes the output of the time interval sensor to be latched
6	1	TimeInstance sensor TransducerChannel
	2	TransducerChannel number of the transducer that causes the output of the TimeInstance sensor to be latched
7	1	TransducerChannel number of an event sensor used to trigger other transducers
	2 - N	The remaining N entries give a list of TransducerChannels triggered by the event
8	1	TransducerChannel (any type)
	2	Embedded time delay actuator TransducerChannel
9	1	Location sensor TransducerChannel
	2	TransducerChannel number of the transducer that causes the output of the location sensor to be latched
10-127	—	Reserved for future expansion
128-255	—	Open to manufacturers

Table 44—Enumeration of ControlGroup types (page 103)

Enumeration	Member list order	Function
0	—	An arbitrary relation
1	1	x component of a right-hand rectangular spatial vector
	2	y component of a right-hand rectangular spatial vector
	3	z component of a right-hand rectangular spatial vector
2	1	$\rho$ component of a right-hand cylindrical spatial vector
	2	$\phi$ component of a right-hand cylindrical spatial vector
	3	z component of a right-hand cylindrical spatial vector
3	1	r component of a right-hand spherical spatial vector
	2	$\theta$ component of a right-hand spherical spatial vector
	3	$\phi$ component of a right-hand spherical spatial vector
4	1	Latitude component of a planetary coordinate system
	2	Longitude component of a planetary coordinate system
	3	Altitude component of a planetary coordinate system
5	1	In-phase component of a two-dimensional vector
	2	Quadrature component of a two-dimensional vector
6	1	Red component in a color vector
	2	Green component in a color vector
	3	Blue component in a color vector
7	1	Real component of a complex number
	2	Imaginary component of a complex number
8-127	—	Reserved for future expansion
128-255	—	Open to manufacturers

Table 45—Enumeration of VectorGroup types (page 105)

Table 43—Structure of the Meta-TEDS data block (page 103)

16	GeoLoc	Specialized VectorGroup for geographic location	—	—
Types 24, 20, and 21 define one set of geographic location information.				
24	LocEnum	An enumeration defining how location information is provided	UInt8	1
20	GrpType	VectorGroup type	UInt8	1
21	MemList	VectorGroup member list	array of UInt16	NTv
17	Proxies	TransducerChannel proxy definition sub-block	—	—
Types 22, 23, and 21 define one TransducerChannel proxy.				
22	ChanNum	TransducerChannel number of the TransducerChannel proxy	UInt16	1
23	Organiz	TransducerChannel proxy data-set organization	UInt8	1
21	MemList	TransducerChannel proxy member list	array of UInt16	NTp
18–19	—	Reserved	—	—
25–127	—	Reserved	—	—
128–255	—	Open to manufacturers	—	—
—	—	Checksum	UInt16	2

**Table 43—Structure of the Meta-TEDS data block (page 99)**

Value	Meaning
0	No geographic location information is provided by this TIM
1	Static geographic location information is provided via the geographic location TEDS
2	Dynamic geographic location information is provided
3	Dynamic geographic location information is provided that is relative to the location specified in the geographic location TEDS
4–255	Reserved for future expansion

**Table 46—Enumeration of geographic location types (page 106)**

Enumeration	Meaning
0	Block method.
1	Interleave method 1.
2	Interleave method 2: This is the same as Interleave method 1 except that the list of interleaved TransducerChannels shall be preceded by a one-sample data set from another TransducerChannel or proxy. This is typically used for a time stamp identifying the time of the first sample in the combined data set.
3–255	Reserved.

**Table 47—Enumerations for combined data sets (page 108)**

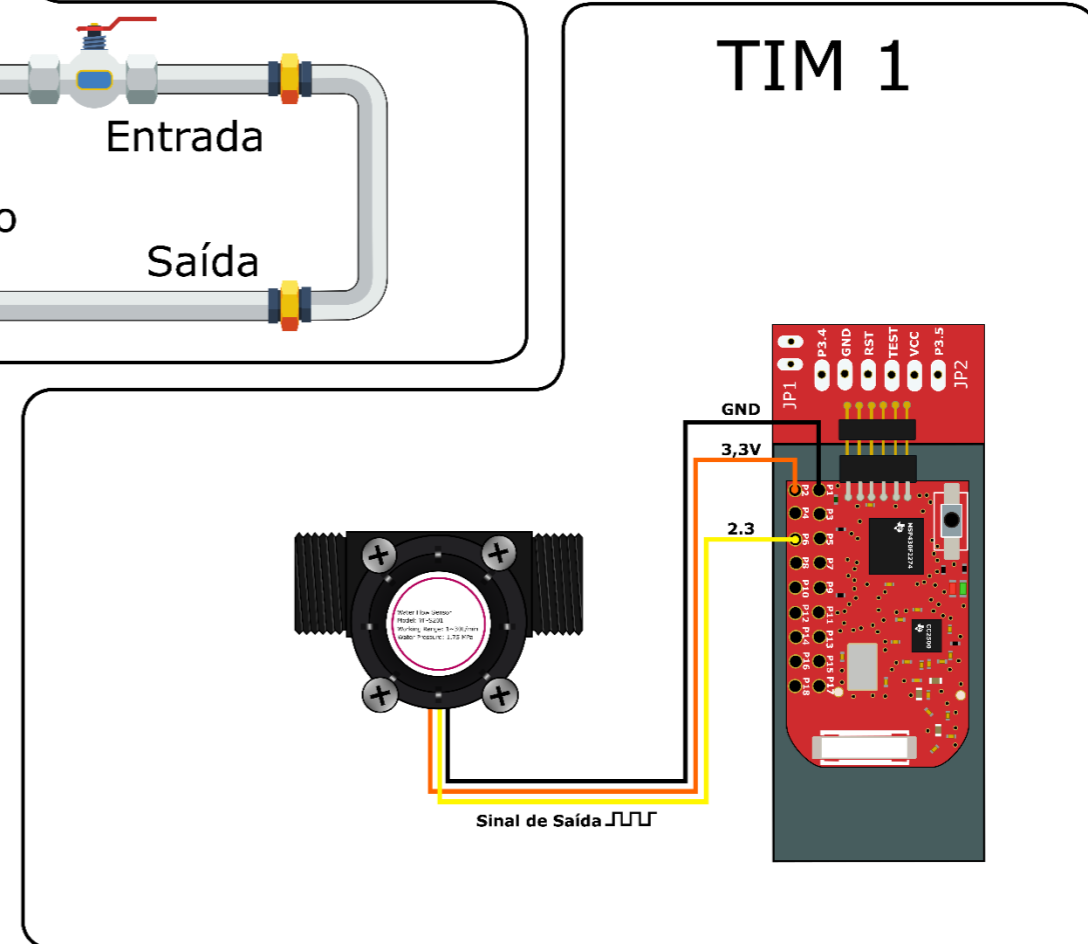
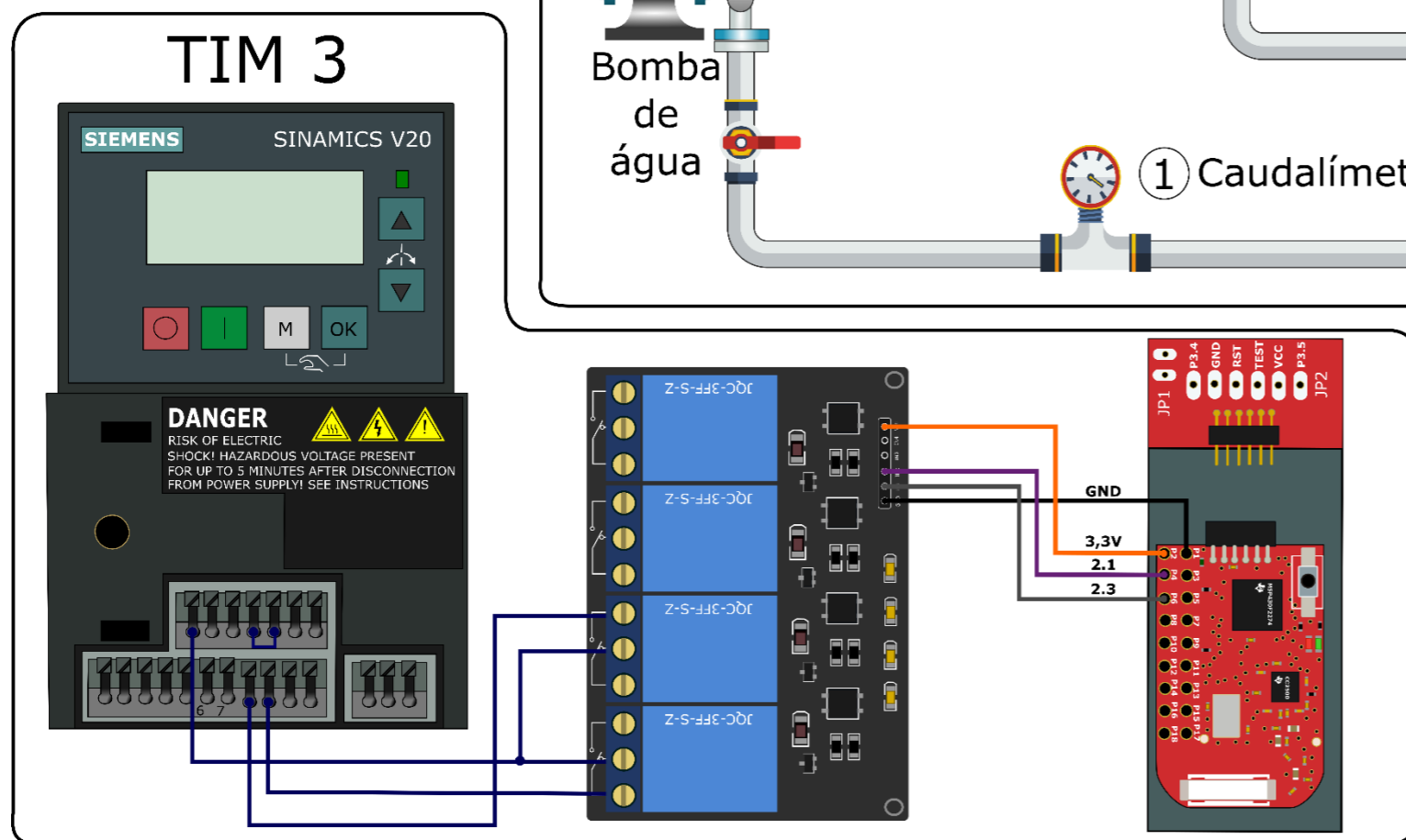
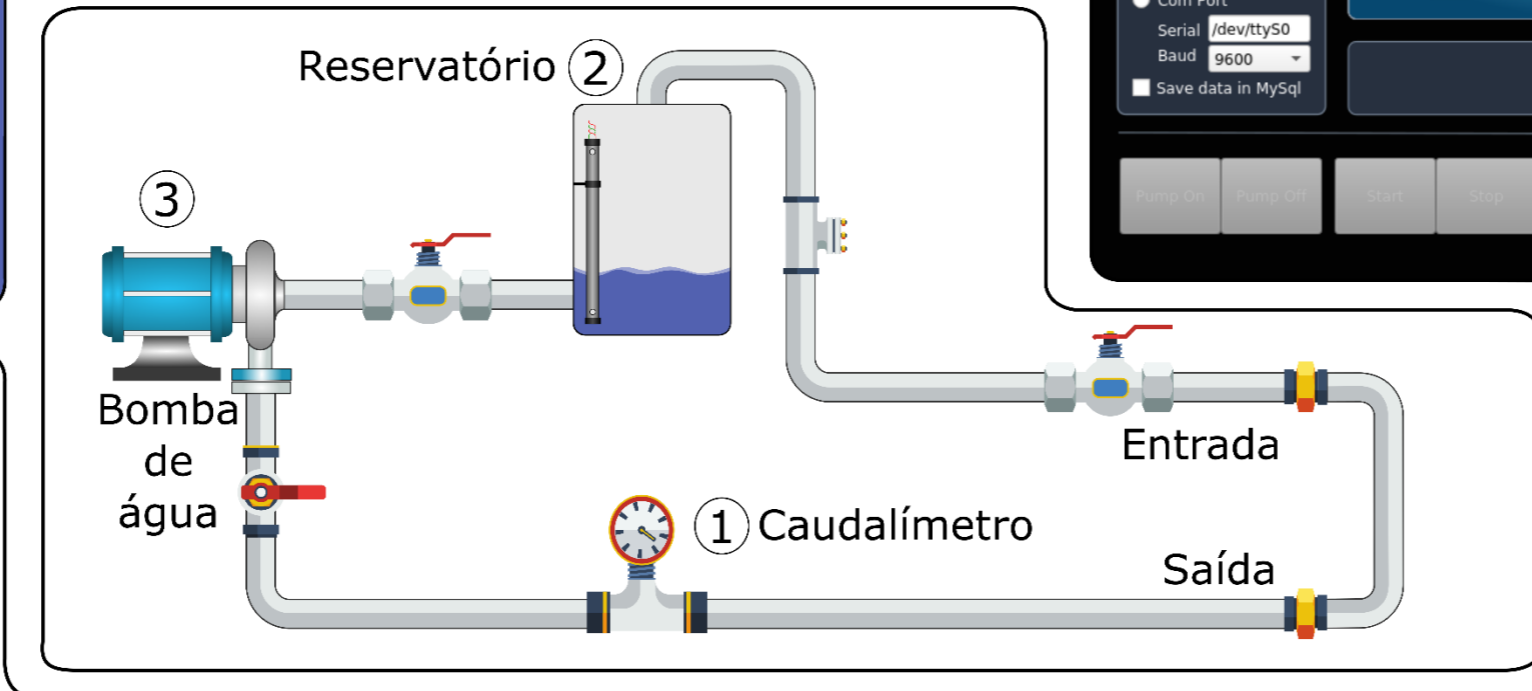
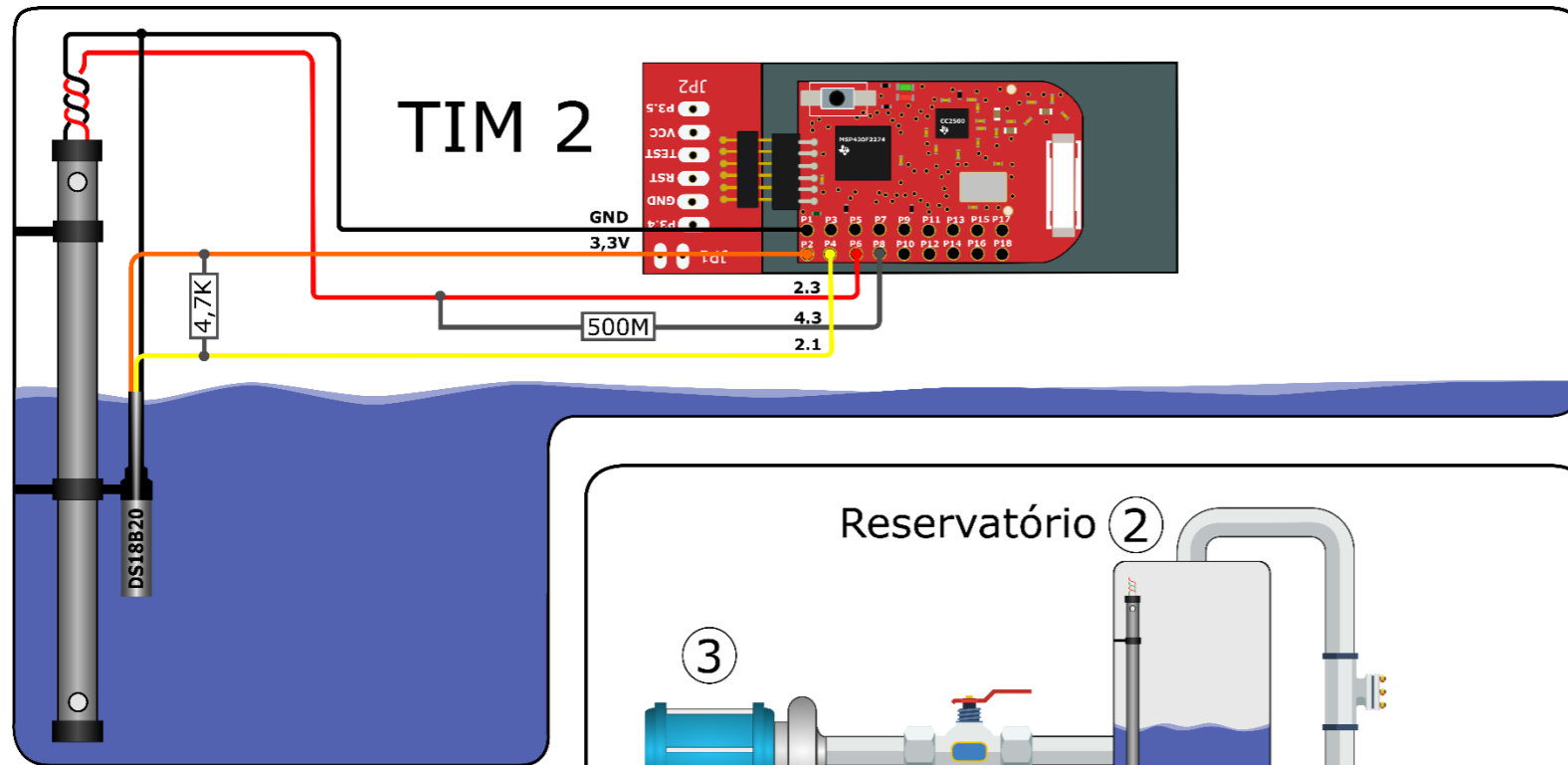
## Apêndice C - Comandos NCAP → Respostas TIM para a “Montagem Experimental”

### Comando NCAP

### Resposta TIM

	TIM ID	Canal Trans. ID	Classe	Função	Tamanho	Dep. Comando		Sucesso	Tamanho	Valor								
<b>TIM 1 – Caudalímetro</b>																		
Canal Transdutor 1 à Temperatura sensor da placa																		
Lê Temp.	0x0	0x01	0x00	0x01	0x03	0x01	0x00	0x00				0x01	0x00	0x04	0xXX	0xXX	0xXX	0xXX
Canal Transdutor 2 à Nível de Bateria																		
Lê Bateria	0x0	0x01	0x00	0x02	0x03	0x01	0x00	0x00				0x01	0x00	0x04	0xXX	0xXX	0xXX	0xXX
Canal Transdutor 3 à RSSI																		
Lê RSSI	0x0	0x01	0x00	0x03	0x03	0x01	0x00	0x00				0x01	0x00	0x04	0xXX	0xXX	0xXX	0xXX
Canal Transdutor 4 à Caudal																		
Lê Valor	0x0	0x01	0x00	0x04	0x03	0x01	0x00	0x00				0x01	0x00	0x04	0xXX	0xXX	0xXX	0xXX
<b>TIM 2 – Nível de água (Temperatura da água)</b>																		
Canal Transdutor 1 à Temperatura sensor da placa																		
Lê Temp.	0x0	0x02	0x00	0x01	0x03	0x01	0x00	0x00				0x01	0x00	0x04	0xXX	0xXX	0xXX	0xXX
Canal Transdutor 2 à Nível de Bateria																		
Lê Bateria	0x0	0x02	0x00	0x02	0x03	0x01	0x00	0x00				0x01	0x00	0x04	0xXX	0xXX	0xXX	0xXX
Canal Transdutor 3 à RSSI																		
Lê RSSI	0x0	0x02	0x00	0x03	0x03	0x01	0x00	0x00				0x01	0x00	0x04	0xXX	0xXX	0xXX	0xXX
Canal Transdutor 4 à Nível de água																		
Lê nível água	0x0	0x02	0x00	0x04	0x03	0x01	0x00	0x00				0x01	0x00	0x04	0xXX	0xXX	0xXX	0xXX
Canal Transdutor 5 à Temperatura da água ds18b20																		
Lê Temp.	0x0	0x02	0x00	0x05	0x03	0x01	0x00	0x00				0x01	0x00	0x04	0xXX	0xXX	0xXX	0xXX
Canal Transdutor 6 ao alarme de máximo de nível																		
Lê Alarme	0x0	0x02	0x00	0x06	0x03	0x01	0x00	0x00				0x01	0x00	0x01	alarme			
Canal Transdutor 7 ao alarme de mínimo de nível																		
Lê Alarme	0x0	0x02	0x00	0x07	0x03	0x01	0x00	0x00				0x01	0x00	0x01	alarme			
<b>TIM 3 – Bomba via macro Cno06 - Relé (SIEMENS SINAMICS V20)</b>																		
Canal Transdutor 1 à Temperatura sensor da placa																		
Lê Temp.	0x0	0x03	0x00	0x01	0x03	0x01	0x00	0x00				0x01	0x00	0x04	0xXX	0xXX	0xXX	0xXX
Canal Transdutor 2 à Nível de Bateria																		
Lê Bateria	0x0	0x03	0x00	0x02	0x03	0x01	0x00	0x00				0x01	0x00	0x04	0xXX	0xXX	0xXX	0xXX
Canal Transdutor 3 à RSSI																		
Lê RSSI	0x0	0x03	0x00	0x03	0x03	0x01	0x00	0x00				0x01	0x00	0x04	0xXX	0xXX	0xXX	0xXX
Canal Transdutor 4 à Liga/Desliga/Lê Estado																		
Liga	0x0	0x03	0x00	0x04	0x03	0x02	0x00	0x01	0x00			0x01	0x00	0x01	estado			
Desliga	0x0	0x03	0x00	0x04	0x03	0x02	0x00	0x01	0x01			0x01	0x00	0x01	estado			
Lê estado	0x0	0x03	0x00	0x04	0x03	0x02	0x00	0x01	0x02			0x01	0x00	0x01	estado			

## Apêndice D - Montagem Experimental



## Apêndice E - Montagem Experimental



Figura 1 - Montagem experimental ☐

## Apêndice F - Hardware Utilizado

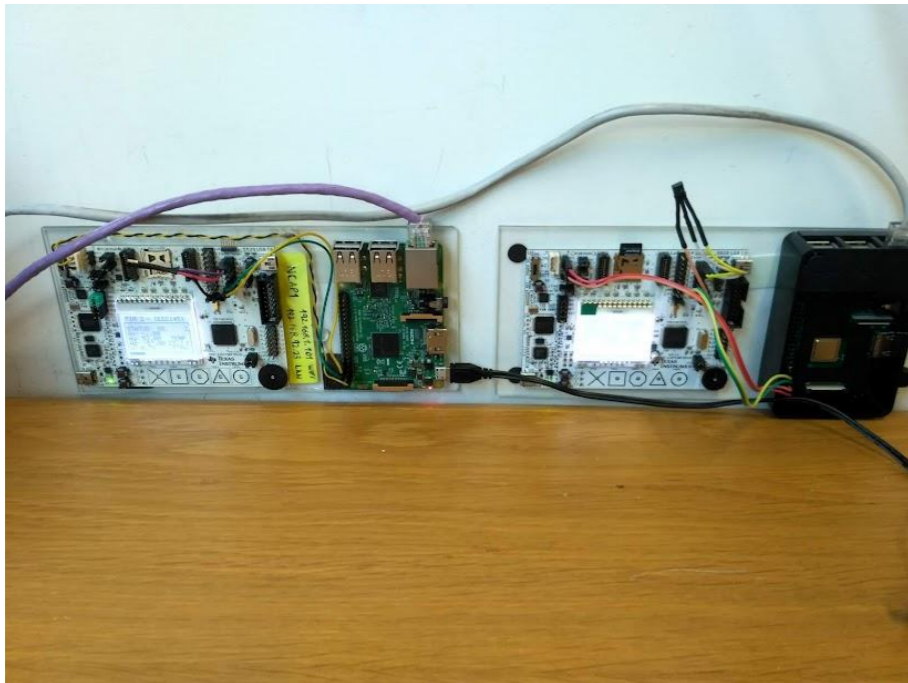


Figura 1 - TIM e NCAP

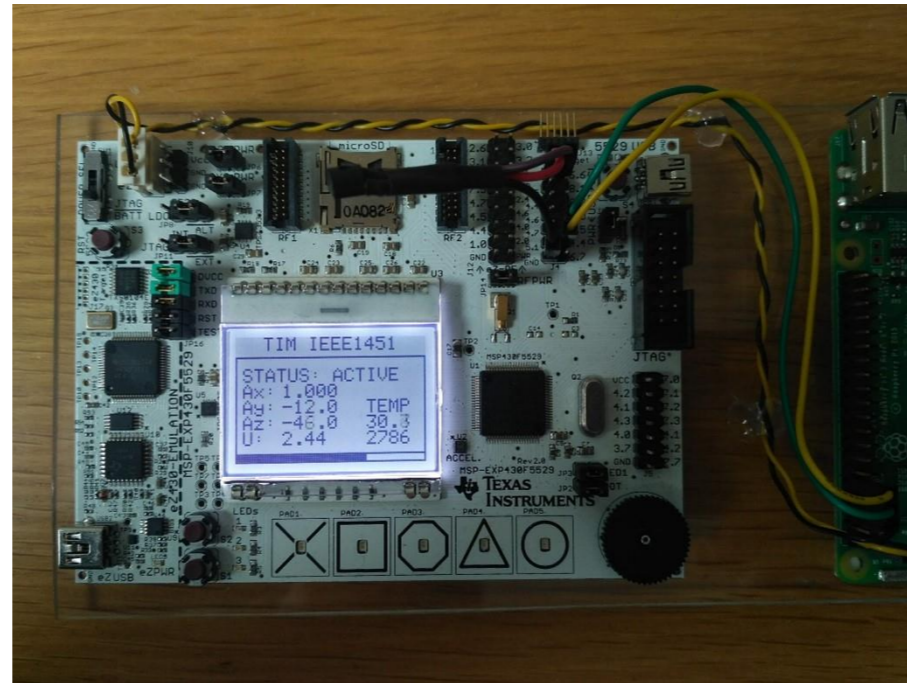


Figura 2 - TIM com interface série com fios

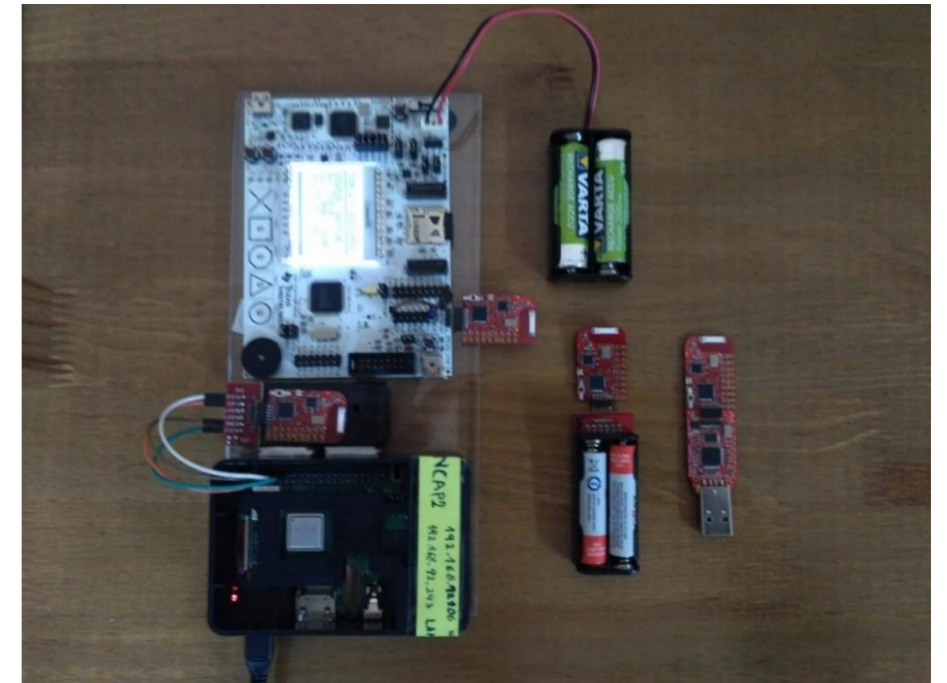


Figura 3 - Ponte sem fios

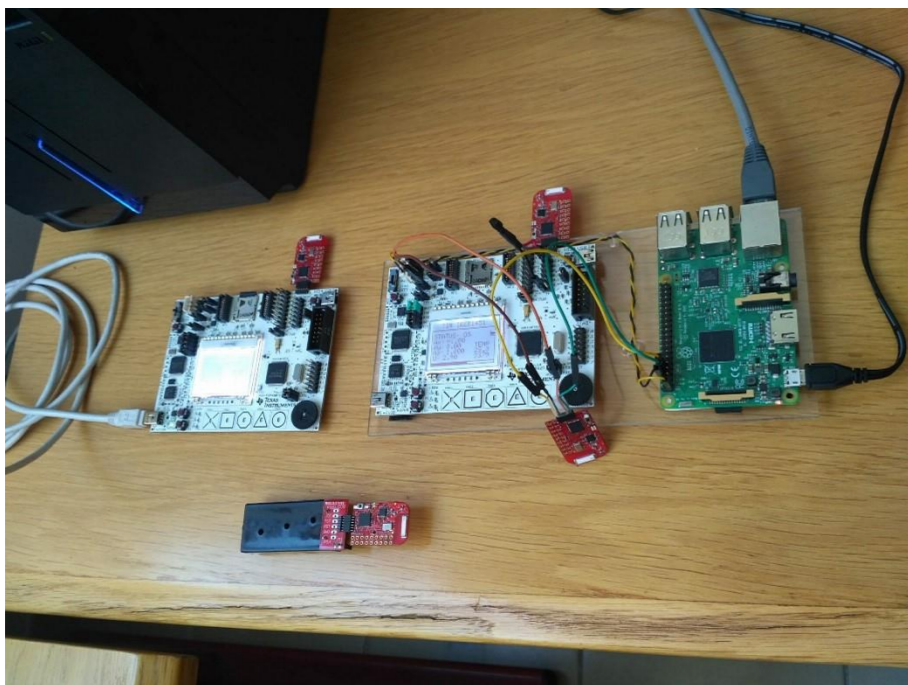


Figura 4 - Rede NCAP ↔ TIM sem fios



Figura 5 - Raspberry Pi com a aplicação

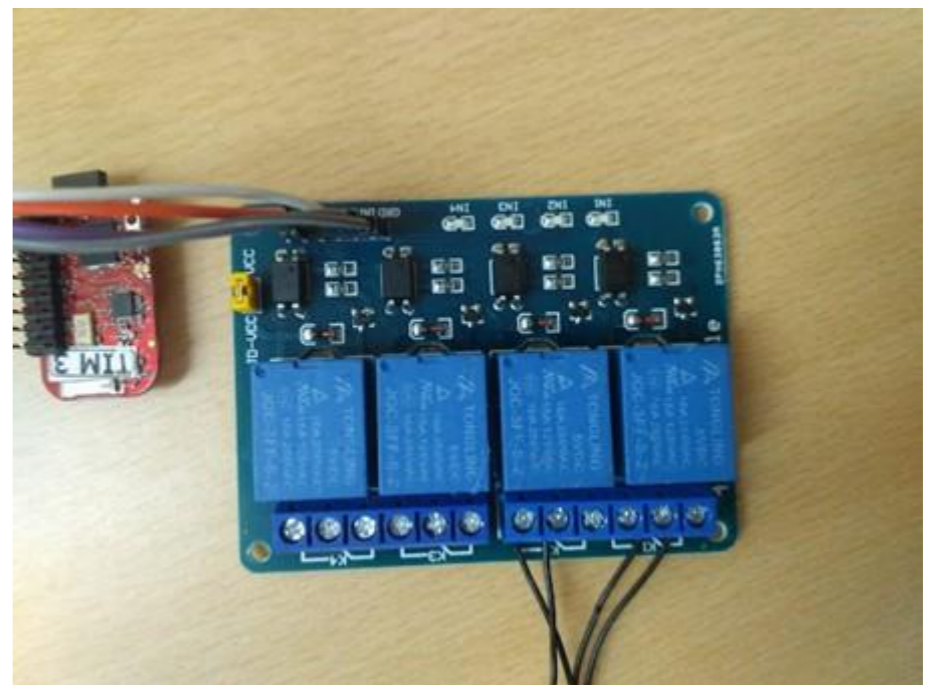


Figura 6 - Interface TIM ↔ Inversor SIEMENS

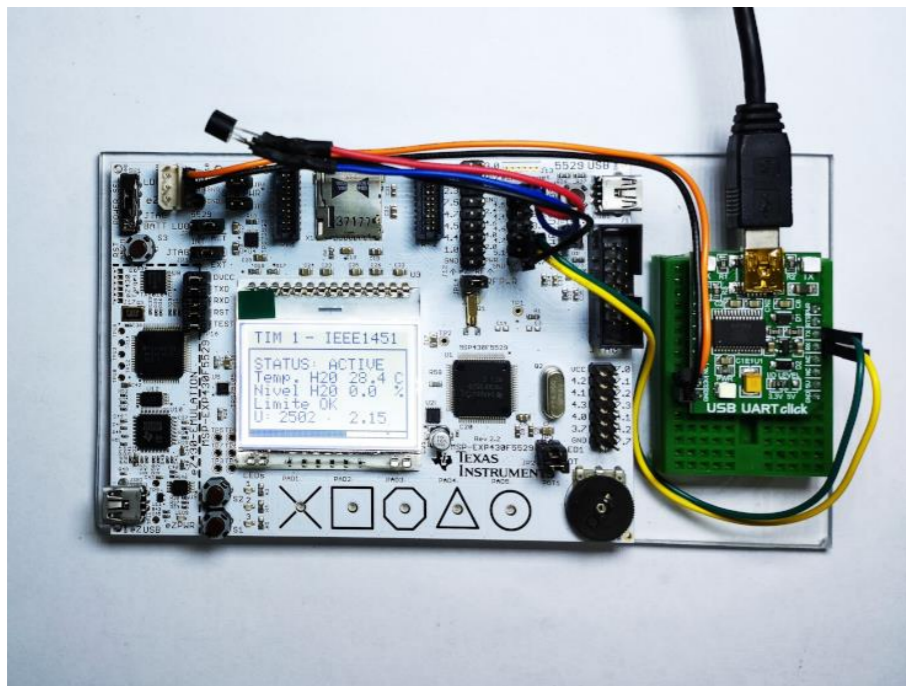


Figura 7 - Hardware de validação TIM IEEE1451

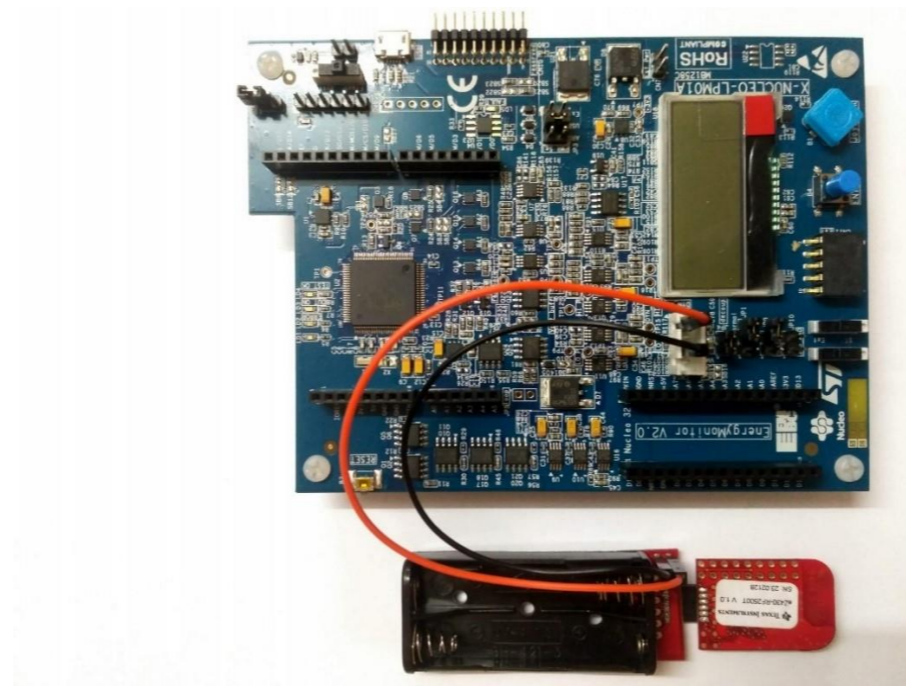


Figura 8 - Medição de corrente TIM sem fios

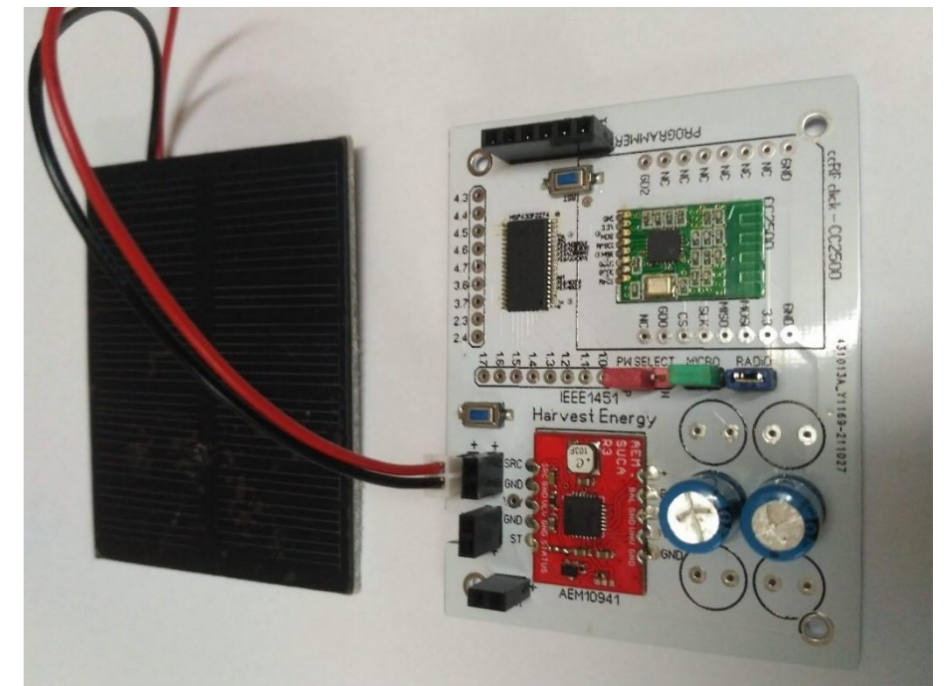


Figura 9 - TIM energia "Harvest"

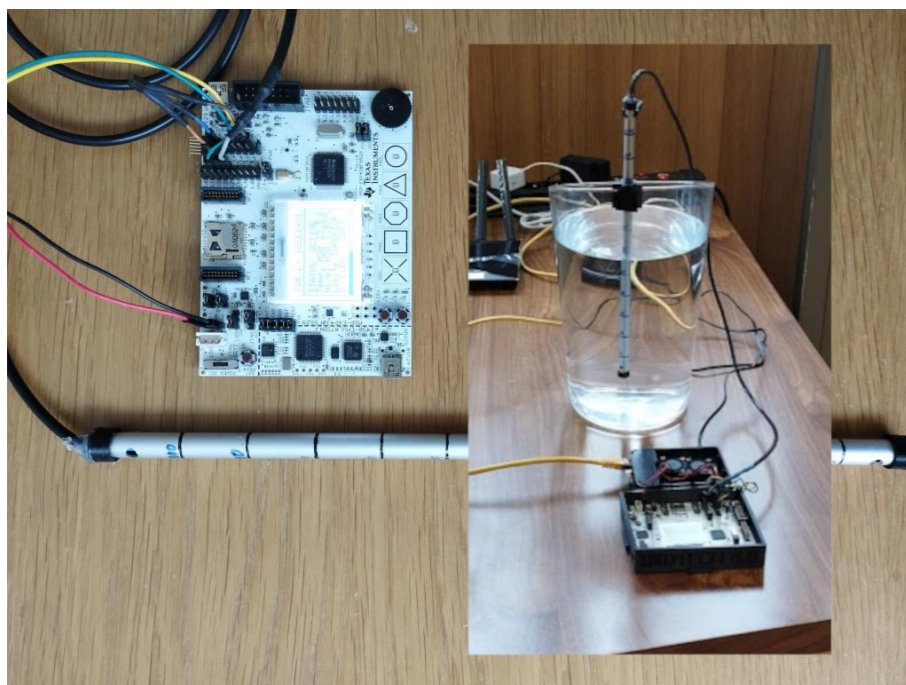


Figura 10 - Sensor nível de água

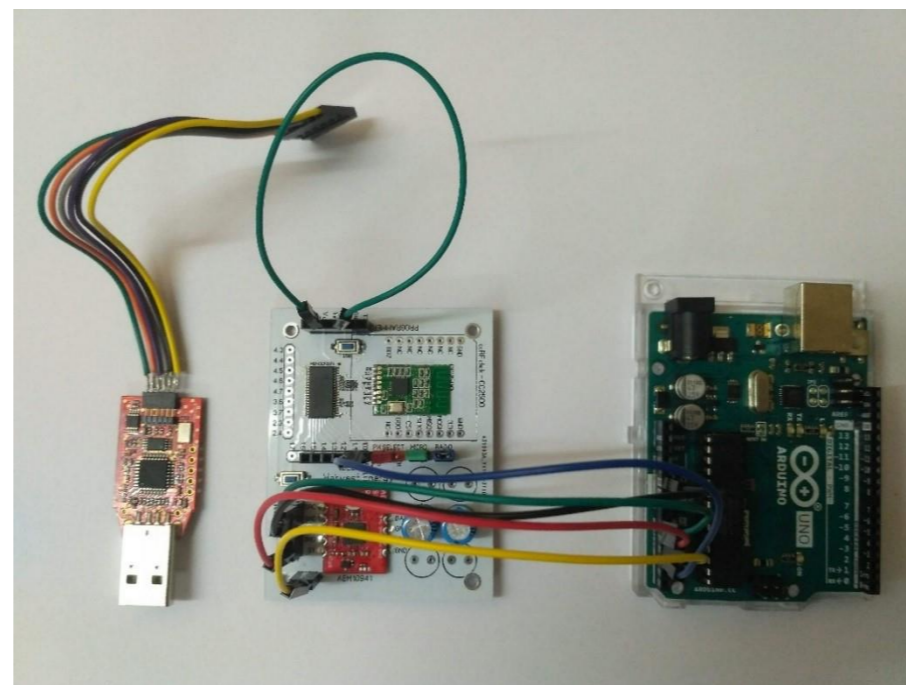


Figura 11 - TIM energia "Harvest" + Programador

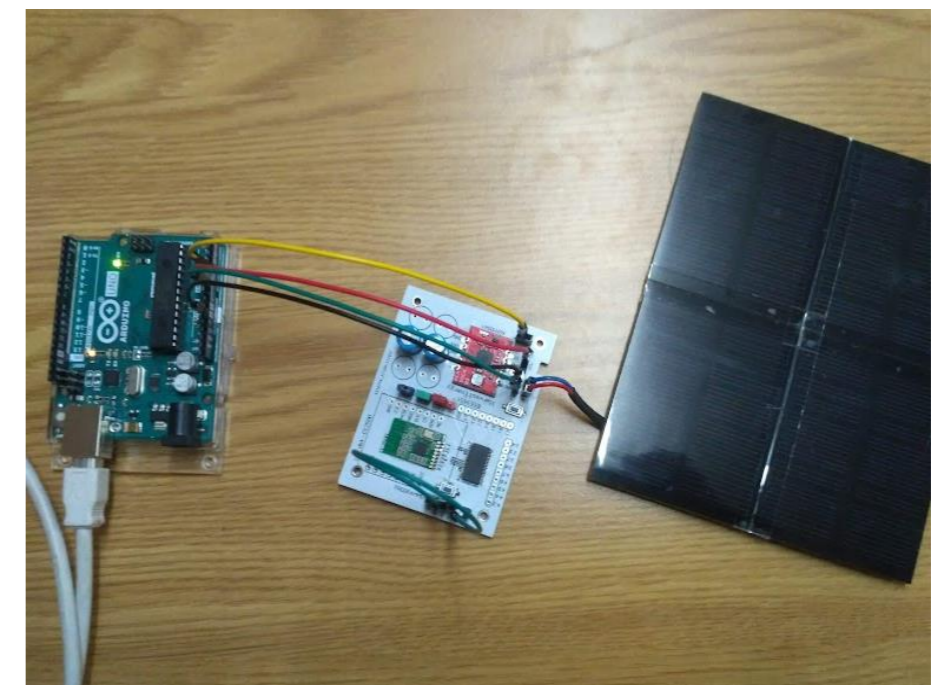


Figura 12 - TIM "Harvest" + Sistema de aquisição de tensões

# Apêndice G - Energy TEDS

## 1. Energy TEDS

The Energy TEDS is an experimental TEDS for Harvest Energy Wireless TIM. Energy TEDS octets are constant and read-only.

Table 1 summarizes the content of the data block. Subordinate subclauses explain each data field in this data block. The length and checksum fields are not technically part of the TEDS data block but are shown in Table 1 to describe the TEDS more completely.

**Table 1 - Structure of the Energy-TEDS data block**

Field Type	Field Name	Description	Data type	# Octets
-		Length	UInt32	4
0-2	-	Reserved	-	-
3	TEDSID	TEDS Identification Header	UInt8	4
4-5	-	Reserved	-	-
Energy Source Type				
6	ESrT	Source type	UInt8	1
7	ESrME	Maximum Energy	Float32	4
Energy Storage Type				
8	ESST	Energy Storage Type	UInt8	1
9	ESMES	Maximum Energy Storage	Float32	4
10	ESMTR	Minimum Time Recharge	Float32	4
TIM Energy Consumption Related				
11	TECP	Processing Energy Consumption	Float32	4
12	TEECRX	Radio Energy Consumption RX Mode	Float32	4
13	TECDRP	Dynamic Radio TX Power	UInt8	1
14	TEECTX	Radio Energy Consumption TX Mode	Float32	4
Timing-related Information				
15	TMTW	Minimum Time Window	Float32	4
16	TITBW	Initial Time Between Windows	Float32	4
17-127		Reserved	-	-
128-255		Open to manufacturers	-	-
-		Checksum	UInt16	2

### 1.1 Identificador TEDS

Field Type: 03

Field Name: TEDSID

Data type: UInt8, 4 octets

This field is required. If this field is omitted, the NCAP shall report a fatal TEDS error.

This field identifies the TEDS, it must have an integer value of 16.

### 1.2 Source type

Field Type: 06

Field Name: ESrT

Data type: UInt8, 1 octet

This field is required. If this field is omitted, the NCAP shall report a fatal TEDS error.

In this field the NCAP is inform about the source type of TIM. The values for Source type key are defined in Table 2.

**Table 2 - Enumeration of Source types**

Value	Meaning
0	Reserved
1	Photovoltaic
2	Thermoelectric
3	Piezoelectric
4	RF antenna
5	Micro wind turbine
6 to 254	Reserved for future expansion

### 1.3 Maximum Energy

Field Type: 07

Field Name: ESrME

Data type: Float32, 4 octets

This field is required. If this field is omitted, the NCAP shall report a fatal TEDS error.

This field indicates the maximum energy, in Joule, that the source (previously selected) provides in the best-case scenario.

### 1.4 Energy Storage Type

Field Type: 08

Field Name: ESST

Data type: UInt8, 1 octet

This field is optional.

This field indicates the type of energy storage that TIM uses. The values for Energy Storage type key are defined in Table 3.

**Table 3 - Enumeration of Energy Storage types**

<b>Value</b>	<b>Meaning</b>
0	Reserved
1	Capacitor
2	Supercapacitor
3	Battery
4 to 254	Reserved for future expansion

### **1.5 Maximum Energy Storage**

Field Type: 09

Field Name: ESMES

Data type: Float32, 4 octets

This field is optional.

This field indicates the maximum energy, in Joule, that the storage element can store.

### **1.6 Minimum Time Recharge**

Field Type: 10

Field Name: ESMTR

Data type: Float32, 4 octets

This field is optional.

This field indicates the minimum charge time, in seconds, of the TIM storage element in the best-case scenario.

### **1.7 Processing Energy Consumption**

Field Type: 11

Field Name: TECP

Data type: Float32, 4 octets

This field is optional.

This field indicates the energy consumption, in Joule, by the TIM processor in active mode.

### **1.8 Radio Energy Consumption RX Mode**

Field Type: 12

Field Name: TECRP

Data type: Float32, 4 octets

This field is optional.

This field indicates the energy consumption, in Joule, by the radio in RX mode.

### 1.9 Dynamic Radio TX Power

Field Type: 13

Field Name: TECDRP

Data type: UInt8, 1 octet

This field is required. If this field is omitted, the NCAP shall report a fatal TEDS error.

If set to Dynamic TIM should calculate the best radio power transmission in relation of RSSI and available energy from source. The values for Energy Storage type key are defined in Table 4.

**Table 4 - Enumeration of Dynamic Radio Power**

Value	Meaning
0	Reserved
1	Fixed
2	Dynamic
3 to 254	Reserved for future expansion

### 1.9 Radio Energy Consumption TX Mode

Field Type: 14

Field Name: TECRP

Data type: Float32, 4 octets

This field is required. If this field is omitted, the NCAP shall report a fatal TEDS error.

This field indicates the energy consumption, in Joule, by the radio in TX mode, if Radio TX Power field is set to Fixed. If set to Dynamic the value should be zero.

### 1.11 Minimum Time Window

Field Type: 15

Field Name: TMTW

Data type: Float32, 4 octets

This field is required. If this field is omitted, the NCAP shall report a fatal TEDS error.

This field indicates the minimum time, in seconds, that the TIM radio can be in receive mode. The time the radio will be able to receive commands from NCAP.

### 1.12 Initial Time Between Windows

Field Type: 16

Field Name: TITBW

Data type: Float32, 4 octets

This field is required. If this field is omitted, the NCAP shall report a fatal TEDS error.

This field indicates the time between time slots, in seconds. This field is dynamic and calculated by TIM, it must keep NCAP informed of when it will be listening again. NCAP may skip window, depending on the number of pending tasks. The default value should be the minimum time when de energy storage is full.

# Apêndice H – Download TEDS

```
*****
* File: teds.txt
* Name:
* Description: Example from standard (Appendix O)
* Created: 2022-08-22 22:24:00
* Autor: iml@ubi.pt
*****
```

```
*****
* META-TEDS
*****
```

```
0x00 0x00 0x00 0x2A
0x03 0x04 0x00 0x00 0x00 0x01
0x04 0x0A 0x91 0xB2 0x60 0x1A 0x9D 0x05 0xF8 0xC4 0x93 0xE1
0x0A 0x04 0x3E 0x99 0x99 0x9A
0x0B 0x04 0x3D 0xCC 0xCC 0xCD
0x0C 0x04 0x00 0x00 0x00 0x00
0x0D 0x02 0x00 0x01
0xF5 0x47
```

```
*****
* TransducerChannel 1 TEDS
*****
```

```
0x00 0x00 0x00 0x4A
0x03 0x04 0x00 0x00 0x00 0x01
0x0A 0x01 0x01
0x0B 0x01 0x01
0x0C 0x09 0x32 0x01 0x0 0x33 0x01 0x82 0x34 0x01 0x82
0x0D 0x04 0x3F 0x80 0x00 0x00
0x0E 0x04 0x3F 0x80 0x00 0x00
0x0F 0x04 0x3F 0x80 0x00 0x00
0x10 0x01 0x01
0x11 0x01 0x01
0x12 0x28 0x01 01 0x29 0x01 0x01 0x2A 0x01 0x01
0x13 0x2B 0x01 0x01 0x2C 0x01 0x01 0x2D 0x01 0x01 0x2E 0x01 0x01 0x2F
0x01 0x01
0x14 0x04 0x3F 0x80 0x00 0x00
0x15 0x04 0x3F 0x80 0x00 0x00
0x16 0x04 0x00 0x00 0x00 0x00
0x17 0x04 0x3F 0x80 0x00 0x00
0x18 0x04 0x3F 0x80 0x00 0x00
0x19 0x04 0x3F 0x80 0x00 0x00
0x1A 0x04 0x3F 0x80 0x00 0x00
0x1B 0x01 0x00
0x1C 0x04 0x3F 0x80 0x00 0x00
0x1D 0x04 0x3F 0x80 0x00 0x00
0x1E 0x04 0x3F 0x80 0x00 0x00
0x1F 0x30 0x01 0x01
0x31 0x01 0x01
0x20 0x01 0x01
0x21 0x01 0x01
0x22 0x01 0x01
0x23 0x01 0x01
0x24 0x01 0x01
```

0x25 0x04 0x00 0x00 0x00 0x01  
0x26 0x08 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x01  
0x27 0x01 0x01  
0xFF 0x94

\*\*\*\*\*  
\* Users TransducerChannel Name TEDS process  
\*\*\*\*\*

0x00 0x00 0x00 0x19  
0x03 0x04 0x00 0x0C 0x01 0x01  
0x04 0x01 0x00  
0x05 0x08 0x49 0x4D 0x4C 0x2D 0x54 0x49 0x4D 0x31  
0xFD 0x95

\*\*\*\*\*  
\* PHY TEDS  
\*\*\*\*\*

0x00 0x00 0x00 0x46  
0x03 0x04 0x00 0x00 0x00 0x0C  
0x0A 0x01 0x04  
0x0B 0x04 0x00 0x03 0xD0 0x90  
0x0C 0x02 0x00 0x01  
0x0D 0x02 0x00 0x01  
0x0E 0x02 0x04 0x64  
0x0F 0x01 0x01  
0x10 0x02 0x00 0x08  
0x11 0x02 0x00 0x10  
0x12 0x02 0x00 0x80  
0x13 0x04 0x00 0x00 0x00 0x01  
0x14 0x04 0x00 0x00 0x00 0x01  
0x15 0x01 0x01  
0x16 0x01 0x01  
0x17 0x02 0x01 0x02  
0x18 0x02 0x00 0x04  
0xFD 0xB5



The End 