

Universidade da Beira Interior

Departamento de Engenharia Electromecânica



Covilhã | Portugal

Desenvolvimento de Equipamentos Didáticos para o Ensino de Sistemas Embutidos

Tiago Filipe de Pinho Carrilho Gonçalves

Licenciado em Engenharia Electrotécnica

Dissertação para obtenção do Grau de Mestre
do 2º Ciclo em Engenharia Electrotécnica

Covilhã e UBI, Agosto de 2009

Agradecimentos

Gostaria de expressar o meu especial agradecimento ao meu orientador Prof. António Espírito Santo pelo apoio incondicional, esforço e dedicação na formulação desta dissertação de mestrado. Gostaria também de agradecer especialmente o Prof. Bruno Ribeiro por cooperar com a partilha da sua vasta experiência para o desenvolvimento prático da dissertação. Gostaria também de expressar a minha gratidão à família pelo apoio integral e aos amigos, em particular, ao Ahmad Rostami, ao Nuno Pais e ao Gerardo Osório, pelo apoio dispensado, aos colegas e a todos os que directa e indirectamente contribuíram para a concretização desta dissertação. A todos, o meu sentido agradecimento.

Resumo

O desenvolvimento das tecnologias na indústria de semicondutores permite actualmente um conceito que cada vez mais se integra no quotidiano das pessoas. O aparecimento de sistemas VLSI e MEMS vem comprovar a necessidade de sistemas embutidos dedicados a uma função especial, em contraste ao computador pessoal de propósito geral, com um baixo custo de produção.

No mundo actual, a preparação de um profissional em engenharia passa por diversos estágios de aprendizagem. É fundamental obter um balanço entre os conhecimentos teóricos adquiridos ao longo da formação, bem como uma sólida experiência prática, permitindo uma visão alargada dos desafios reais que qualquer trabalho de engenharia enfrenta.

Neste trabalho, desenvolveu-se uma plataforma para o estudo e desenvolvimento de sistemas embutidos, com fins didácticos, para vários níveis de aprendizagem. A plataforma permite implementação de ferramentas conceptuais de criação e desenvolvimento de sistemas embutidos, com uma vasta gama de tecnologias co-existent para uma aprendizagem e aplicação sólida de conceitos fundamentais para uma qualificação consistente.

Abstract

The technological advances in VLSI, MEMS, and wireless communications enable today a concept that enables the design of systems with one or more dedicated functions.

Over the past decade, advances brought by the semiconductor industry enabled computers smaller than a pin-head, powerful enough to carry all the processing required and affordable enough to be deployed in large quantities or even to be not reused. This tendency is expected to continue over the next decade, with computers being even smaller and cheaper year-by-year. The integration of these computers in a complex system results in the concept of Embedded Systems.

In life, learning is a permanent process, especially in engineering, where the balance between theoretical concepts and the practical experience allows a broader view of the real challenges faced in any engineering design, providing the know-how to overcome even the most challenging problems.

In this work, several frameworks were developed for the research and development of embedded systems, with a wide range of merged technologies, enabling a solid application of the fundamental concepts gathered during the whole learning process.

Conteúdos

1.	Introdução	1
2.	Conceito	3
2.1.	Kit Nível 0	4
2.1.1.	Descrição Geral	4
2.1.2.	Dispositivos no Kit.....	7
2.2.	Kit Nível 1	12
2.2.1.	Descrição Geral	12
2.2.2.	Dispositivos no Kit.....	15
2.3.	Kit Nível 2	21
2.3.1.	Descrição Geral	21
2.3.2.	Cadeia de instrumentação analógica	22
2.3.3.	Funcionalidades integradas no microcontrolador	23
2.3.4.	Dispositivos no kit.....	35
2.4.	Kit Nível 3	41
2.4.1.	Descrição Geral	41
2.4.2.	Descrição do módulo USCI.....	42
2.4.3.	Dispositivos no kit.....	48
3.	Software	56
3.1.	I ² C.....	56
3.1.1.	Comunicação:.....	58
3.2.	SPI	60
3.2.1.	Funcionamento	61
3.3.	Desenvolvimento	62
3.3.1.	Protocolo de transporte.....	63

3.4. Fluxogramas	65
3.4.1. Master.....	65
3.4.2. Slave.....	68
4. Bibliografia	71
Anexo A.....	72
Anexo B.....	75
Anexo C.....	78
Anexo D.....	80
Anexo E.....	83

Índice de Figuras

Fig. 1 - Desenho 3D do kit nível 0.....	4
Fig. 2 - Organização interna do MSP430F2112.....	7
Fig. 3 - Conversor do tipo DC/DC de topologia boost.....	8
Fig. 4 - Display HDSP-523E fabricado pela Agilent Technologies	9
Fig. 5 - Interface do display com o microprocessador	10
Fig. 6 - <i>Encoder</i> BCD modelo P36S101 da APEM.....	11
Fig. 7 - Desenho 3D do kit nível 1	12
Fig. 8 - Organização interna do MSP430F2112.....	15
Fig. 9 - Display HDSP-523E fabricado pela Agilent Technologies	16
Fig. 10 - Interface do <i>display</i> com o microprocessador	17
Fig. 11 - Diagrama temporal do descodificador BCD	18
Fig. 12 - <i>Buffer/Driver</i> de oito saídas modelo SN74LVC244A da Texas Instruments	19
Fig. 13 - Aspecto do interruptor de pressão utilizado	20
Fig. 14 - Desenho 3D do kit nível 2.....	21
Fig. 15 - Ilustração da cadeia de instrumentação	22
Fig. 16 - Estrutura interna do módulo Amp-Op	24
Fig. 17 - Amp-Op de propósito geral.....	25
Fig. 18 - Buffer de ganho unitário	26
Fig. 19 - Comparador	27
Fig. 20 - Amplificador não inversor de ganho programável.....	28
Fig. 21 - Amplificador inversor de ganho programável	29
Fig. 22 - Ligações para uma possível configuração utilizando os Amp-Op OA0, OA1 e OA2.....	30
Fig. 23 - ADC do tipo SAR de 4 bits	31
Fig. 24 - Sensor de temperatura interno ao módulo ADC - relação de transformação	33
Fig. 25 - Estrutura do LCD. a) vista geral e b) detalhe de um carácter.....	36
Fig. 26 - Sequência de sinal típica gerada pelo encoder	38
Fig. 27 - Encoder rotativo	38
Fig. 28 - <i>joystick</i> de navegação	39
Fig. 29 - Ligação do altifalante	39

Fig. 30 – Ligação do microfone	40
Fig. 31 - Desenho 3D do kit nível 3.....	41
Fig. 32 – Estrutura dos dados enviados pelo porto série.....	43
Fig. 33 – Comunicação em modo <i>idle-line</i>	44
Fig. 34 – Comunicação em modo <i>address-bit</i>	44
Fig. 35 – Comunicação em modo UART com detecção automática do baud rate .	45
Fig. 36 – Ilustração de uma comunicação em modo IrDa	45
Fig. 37 – Esquema equivalente do acelerómetro	48
Fig. 38 – Estrutura construtiva do <i>Touchscreen</i>	49
Fig. 39 – Controlador do touchscreen modelo TSC2007	50
Fig. 40 – Conversão dos valores de tensão	51
Fig. 41 – Interligação de diferentes dispositivos através de um porto USB	51
Fig. 42 – Operação de leitura	52
Fig. 43 – Operação de Escrita	53
Fig. 44 – Arquitectura interna de um cartão SD	53
Fig. 45 – <i>Display</i> OLED modelo DD-160128FC	54
Fig. 46 – Módulo RF da Chipcon	55
Fig. 47 - Módulo RF-EZ430	55
Fig. 48 – Esquema simplificado com um master e n slaves	58
Fig. 49 – Diagrama temporal do protocolo I ² C.....	58
Fig. 50 – Esquema de ligação de um barramento SPI	61
Fig. 51 - Gestão de interrupções no vector USCIAB0TX (<i>master</i>).....	65
Fig. 52 - Gestão de interrupções no vector USCIAB0RX (<i>master</i>)	66
Fig. 53 – Diagrama de fluxo do <i>firmware</i> do <i>master</i>	67
Fig. 54 - Gestão de interrupções do vector USCIAB0RX (<i>slave</i>)	68
Fig. 55 - Gestão de interrupções no vector USCIAB0TX (<i>slave</i>)	69
Fig. 56 - Diagrama de fluxo do <i>firmware</i> do <i>slave</i>	70

Índice de tabelas

Tabela 1 – Organização dos recursos de I/O do Kit 0.	6
Tabela 2 – Tabela de estado do funcionamento do <i>display</i>	11
Tabela 3 – Organização dos recursos de I/O do Kit 1	14
Tabela 4 – Tabela de estado do funcionamento do <i>display</i>	18
Tabela 5 – Configuração dos modos de operação dos Amp-Op.....	23
Tabela 6 – Modos de conversão.....	32
Tabela 7 – Organização dos recursos de I/O do Kit 2	34
Tabela 8 – Interface do controlador LCD presente no microcontrolador	35
Tabela 9 – organização da memória do controlador LCD	37
Tabela 10 – Recursos do processador utilizados	47
Tabela 11 – Endereços I ² C Reservados	59
Tabela 12 – Organização do cabeçalho do protocolo de transporte	63
Tabela 13 – Comandos utilizados no protocolo de transporte	64

1. Introdução

A educação é um pilar fundamental da sociedade. Qualquer ser humano é submetido a diversos estágios de aprendizagem importantes ao longo da sua vida. Durante esses estágios, o cérebro humano é estimulado, desafiando uma evolução constante. A recepção dos estímulos muitas vezes é inconsciente mas vital no percurso natural da vida.

Parte dessa aprendizagem realiza-se através do sistema de ensino. Este último é responsável pela formação dos profissionais de amanhã através de um processo complexo, constituído por uma componente teórica bem como prática. Especialmente em engenharia, o balanço entre estas duas componentes é um factor decisivo na formação de um profissional qualificado.

É com essa motivação que este trabalho foi desenvolvido, pensando nos vários graus de aprendizagem, focado na aplicação prática dos conhecimentos teóricos.

Nas últimas décadas, os avanços tecnológicos evidenciados na indústria de semicondutores permitiu uma forte integração de sistemas embutidos no quotidiano, que passam muitas vezes despercebidos à maioria dos seus utilizadores. No núcleo de um sistema embutido está a programação do microcontrolador e, consoante esta programação, o sistema tanto pode ter bom desempenho, ser fiável, e ter baixo consumo energético como ser completamente o oposto. Deste problema surgiu a necessidade dos alunos aprenderem a programar microcontroladores e desenvolverem sistemas embutidos, aprendizagem esta que se torna tanto mais complexa quanto maior o número de aplicações possíveis para o sistema. Sendo esta uma área em grande expansão, este trabalho vem ajudar ao desenvolvimento da capacidade de conceptualizar, desenvolver e concretizar um sistema embebido.

O objectivo deste trabalho é a criação de várias plataformas de desenvolvimento que reúnem um grupo de tecnologias, na sua maioria, actualmente presentes em sistemas embutidos. Para tal, foram concebidas quatro plataformas interactivas com diferentes escalões de complexidade, abrangendo assim desde o principiante ao utilizador avançado. As plataformas estão dirigidas principalmente para as áreas de engenharia de componente electrónica, sendo necessário um *background* em programação bem como noções preliminares de electrónica.

A nível de interface gráfico, as plataformas estão equipadas desde o simples LED ao *display* de tecnologia OLED passando pelos bastante conhecidos *displays* de sete segmentos. As plataformas possuem a possibilidade de comunicar e interagir através de protocolos de comunicação digitais presentes na camada física. O utilizador pode interagir com as plataformas utilizando periféricos integrados para o efeito, destes destacam-se: interruptores de pressão, um *touchscreen* e um acelerómetro.

Estas plataformas destacam-se das soluções já existentes por integrarem diversos componentes e aplicações dirigidas ao ensino, por utilizarem grande parte dos recursos de cada microcontrolador e por permitirem ser interligadas num sistema modular onde a versatilidade é um ponto forte. Nos dias que correm, o consumo energético é um factor cada vez mais importante não tendo sido deixado ao acaso no desenvolvimento destas plataformas.

Os microcontroladores que habilitam o controlo das interfaces e que coordenam os fluxos de sinais foram escolhidos considerando diversos factores determinantes, entre os quais a velocidade de processamento, a memória disponível, o baixo consumo energético, o custo reduzido e uma curva de aprendizagem para a sua programação variável consoante a plataforma alvo. Por estas razões, os microcontroladores que melhor se enquadram nas exigências impostas pertencem à família MSP430 de microcontroladores disponibilizados pela Texas Instruments. Esta família é universalmente conhecida pelo ínfimo consumo energético aliado à elevada capacidade processamento de sinais analógicos e digitais.

2. Conceito

A estratégia adoptada para o *design* destas plataformas define as seguintes características gerais:

- As plataformas podem ser ligadas num sistema modular;
- Em cada plataforma existe um microcontrolador que apenas gere as comunicações entre plataformas e outro para o utilizador programar;
- Dentro da mesma plataforma os microcontroladores estão interligados;
- Os microcontroladores são programados utilizando uma ligação JTAG;
- São integrados dispositivos que permitem a leitura de consumos energéticos em cada plataforma;
- Todas as plataformas têm uma dimensão em comum (altura);
- A ligação entre plataformas é feita através de fichas DB9.

Depois de um estudo aprofundado dos protocolos de comunicação existentes, tendo em conta a versatilidade, modularidade e o reduzido número de ligações foi possível optar pelo I²C como protocolo de comunicação externo entre plataformas e pelo SPI como protocolo de comunicação interno entre microcontroladores da mesma plataforma.

Em cada plataforma existe um módulo de comunicações projectado para gerir as comunicações entre plataformas e fazer a leitura, filtragem e cálculo dos consumos energéticos.

Às plataformas projectadas foi dado o nome de Kit, ao qual foi acrescentado o nível de complexidade da plataforma correspondente, ficando a plataforma com menor complexidade denominada Kit Nível 0 e a de maior complexidade Kit Nível 3.

Nos capítulos seguintes descreve-se cada uma das plataformas projectadas.

2.1. Kit Nível 0

2.1.1. Descrição Geral

O Kit de Nível 0 está direccionado para o aluno que se inicia no estudo dos sistemas embutidos, e que por isso possui poucos conhecimentos no domínio dos microcontroladores. Tem como base um Microcontrolador MSP430F2112 da Texas Instruments (TI), dois *encoders* BCD de dez posições e um *display* de quatro dígitos de sete segmentos. Para além destes periféricos inclui ainda um módulo desenvolvido para fornecer uma tensão regulada a este kit e a todos os outros que lhe sejam ligados através do barramento.

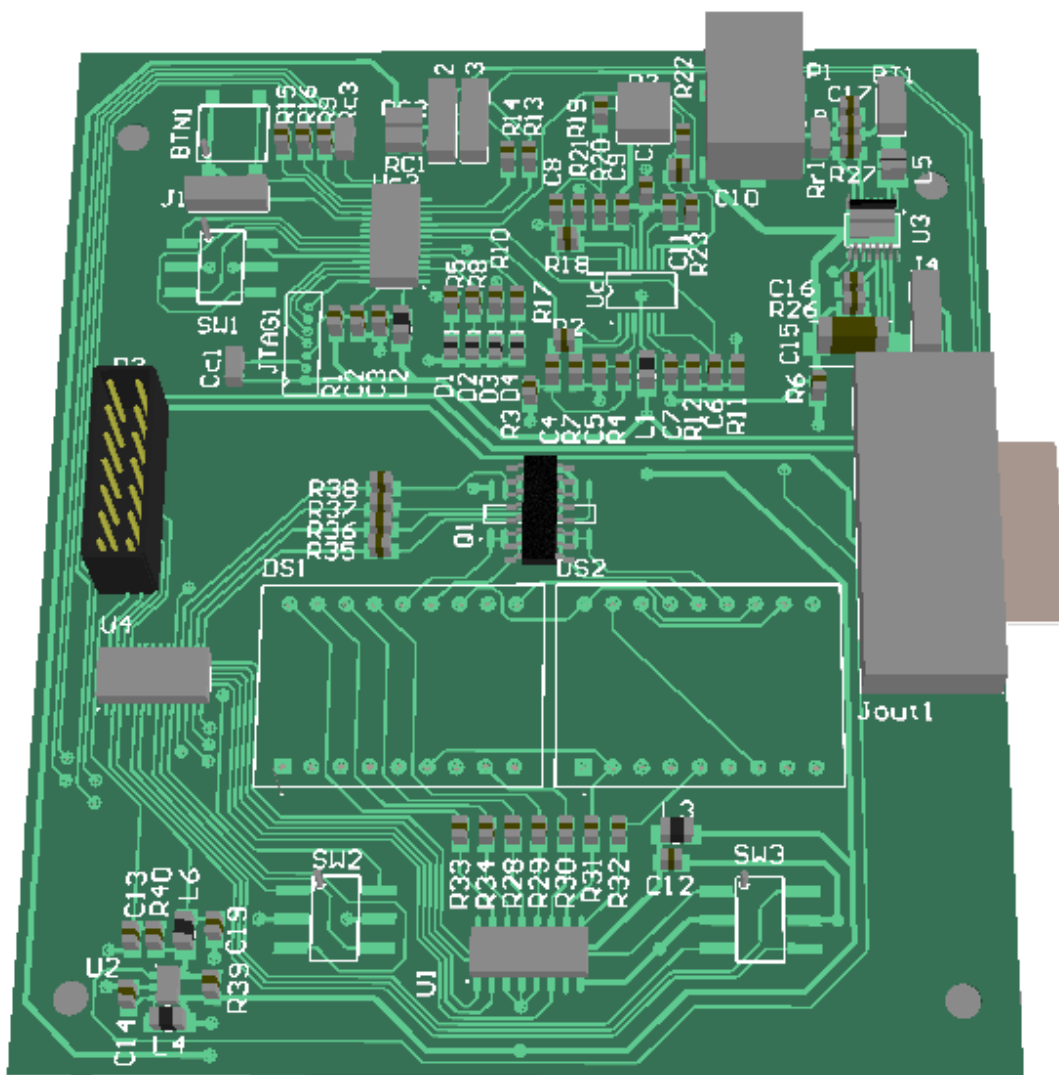


Fig. 1 - Desenho 3D do kit nível 0

Este Kit está desenhado de maneira a que as interfaces com os periféricos sejam tão simples quanto possível. Assim, podemos observar que cada função no Kit usa um *nibble* ou um *byte* de um porto, não existindo a mesma função dispersa em vários portos. Para tornar ainda mais simples a sua utilização por parte do utilizador, as funções que usam portos como saída, partilham o porto com funções dedicadas do microcontrolador, podendo assim ser alterado o valor do porto na sua totalidade sem ser afectado o funcionamento normal dessas funções.

Apesar do kit de nível 0 poder ser utilizado para qualquer funcionalidade, a sua aplicação predefinida é a visualização do consumo energético de cada um dos kits ligados ao sistema. O consumo energético apresentado é recolhido em cada kit pelo módulo de comunicações e é seleccionado através dos dois *encoders* existentes no kit.

A inicialização dos portos é feita numa função incluída que pode ser alterada pelo utilizador, apesar de que não seja necessário para o uso normal. Uma nota importante, que interessa deixar clara, e que diz respeito ao porto onde ligam os *encoders*, é que este tem de ser inicializado com *pull-up* nas entradas.

Para facilitar a utilização do *display* são já incluídas funções que fazem a gestão dos valores do *display* e a reposição dos mesmos.

Exemplifica-se de seguida como interagir com os periféricos usando código em C.

Leitura dos *Encoders* BCD:

```
EncBCD = ~P2IN;
```

Apresentar um dígito no display:

```
P3OUT = DIG0; //dígito seleccionado
```

```
P1OUT = 5; //valor a apresentar no display
```

Utilização do display com as funções fornecidas:

```
Digit(DIG0,5) //valor 5 no dígito 0
```

```
Digit(DIG1,8) //valor 8 no dígito 0
```

```
Display(2009) //valor 2009 no display
```

A Tabela 1 resume atribuição dos recursos disponibilizados para realizar acções de I/O.

Tabela 1 – Organização dos recursos de I/O do Kit 0.

Pino	Função Geral	Nome	Função no Kit
21	P1.0	A	Saídas BCD para displays
22	P1.1	B	
23	P1.2	C	
24	P1.3	D	
25	P1.4	TCK	JTAG
26	P1.5	TMS	
27	P1.6	TDI	
28	P1.7	TDO	
8	P2.0	Add0	Entradas do encoder BCD para Endereço
9	P2.1	Add1	
10	P2.2	Add2	
19	P2.3	Add3	
20	P2.4	Sel0	Entradas do encoder BCD para Função
3	P2.5	Sel1	
6	P2.6	Sel2	
5	P2.7	Sel3	
11	P3.0	STE	Barramento SPI
12	P3.1	SIMO	
13	P3.2	SOMI	
14	P3.3	CLK	
15	P3.4	EnDISP0	Habilitação de displays
16	P3.5	EnDISP1	
17	P3.6	EnDISP2	
18	P3.7	EnDISP3	
7	RST	RST	Pinos Reservados
1	TEST	TEST	
2	Vcc	Vcc	
4	Vss	Vss	

2.1.2. Dispositivos no Kit

2.1.2.1. Microcontrolador

O microcontrolador escolhido para integrar este kit foi o MSP430F2112 que possui as seguintes principais características: 2kB de memória flash, 256B de RAM, ADC de 10bit de 8 canais, 2 temporizadores/contadores com um total de 5 registos de comparação, uma interface de comunicação série e frequência de relógio máxima de 16MHz. Optou-se por este microcontrolador porque cumpre os requisitos necessários para uma iniciação aos sistemas embutidos, e o número de saídas de I/O é suficiente para a aplicação desejada e pelo seu reduzido tamanho e custo. Uma esquematização da estrutura interna do dispositivo é feita na Fig. 2.

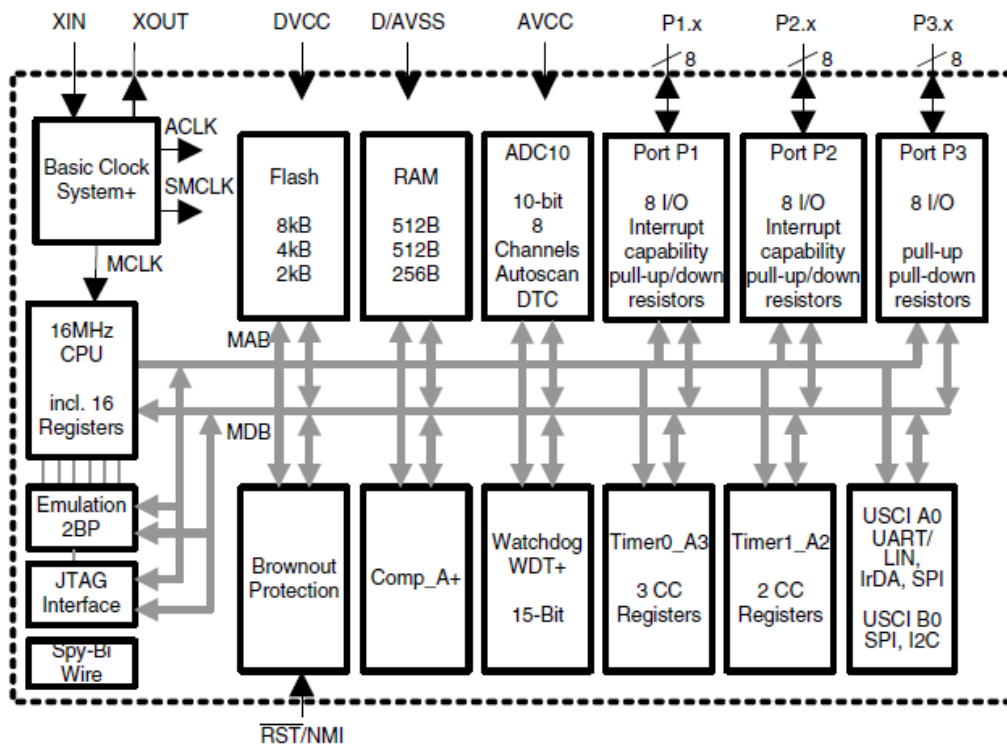


Fig. 2 – Organização interna do MSP430F2112

2.1.2.2. Módulo de alimentação

O módulo de alimentação projectado permite alimentar o kit de nível 0, em que está integrado, e todos os kits ao qual este é ligado, fornecendo uma tensão regulada de 3,3V. Este módulo tem a possibilidade de ser alimentado através de baterias ou fonte de alimentação externa desde que a tensão de entrada seja superior a 1,8V e não ultrapasse 5,5V. O conversor DC-DC utilizado é o TPS61031 disponibilizado pela Texas Instruments (TI), e que é construído usando uma estrutura do tipo *boost* tal como se representa na Fig. 3, com as seguintes características principais: tensão regulada de 3,3V, eficiência óptima de 96%, corrente de saída até 1A, consumo típico de 20 μ A e protecção contra sobreaquecimento. O princípio de funcionamento deste tipo de conversor tem como base a tendência que existe para a corrente que atravessa uma bobina se manter constante, sabendo isto, ligando uma das extremidades da bobina a uma tensão de alimentação, e alternando a ligação da outra extremidade entre o comum e o circuito que se quer alimentar, consegue-se obter uma tensão superior à inicial. Quando a ligação é feita directamente ao comum forma-se uma malha de muito baixa resistência e a corrente que atravessa a bobina atinge valores elevados. Ligando de seguida ao circuito que se pretende alimentar, a corrente na bobina mantém-se quase constante e, como esta malha tem maior resistência a tensão à saída da bobina torna-se superior à da sua alimentação. Para manter a tensão de saída regulada para uma tensão fixa, independentemente da carga aplicada, é necessário ajustar a duração de cada uma destas ligações e/ou a frequência com que as mesmas se repetem. O conversor utilizado tem uma frequência de repetição fixa, sendo apenas ajustado o *duty cycle*. A indutância da bobina de entrada e a capacidade do condensador de saída foram definidas seguindo tanto as indicações como as especificações do fabricante do conversor DC-DC, para a gama de tensões de entrada pretendida e corrente máxima necessária na saída.

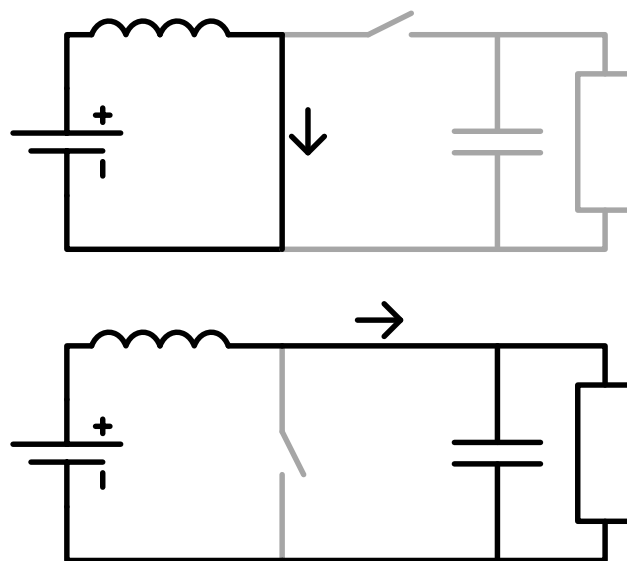


Fig. 3 – Conversor do tipo DC/DC de topologia boost

2.1.2.3. *Displays* de sete segmentos

O modelo de *display* seleccionado para este kit foi o HDSP-523E oferecido pela Agilent Technologies composto por dois dígitos de sete segmentos, de cor vermelha, com cátodo comum e com uma dimensão de 14,2mm. São utilizados dois componentes deste tipo instalados lado a lado, ficando o kit com a capacidade de representar quatro dígitos.

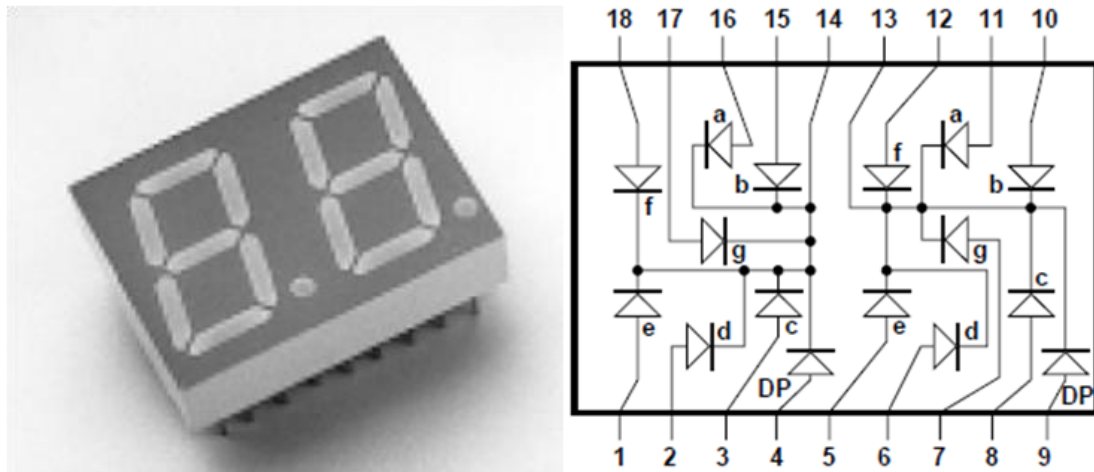


Fig. 4 - Display HDSP-523E fabricado pela Agilent Technologies

Cada dígito de um *display* de sete segmentos tem nove ligações eléctricas, uma para cada ânodo dos LED e mais uma que interliga todos os cátodos. Neste kit, o ponto presente à direita de cada dígito não foi ligado, restando assim 8 ligações por dígito. Para diminuir o número de portas utilizadas para controlar o *display*, e simultaneamente simplificar a sua utilização, foi ligado ao microcontrolador um descodificador BCD para sete segmentos e a este os ânodos dos LED de todos os dígitos. Esta estratégia é ilustrada na Fig. 5. Foram introduzidas resistências de 68Ω , em série entre o descodificador e o *display*, de modo a limitar a corrente que atravessa cada segmento a um valor máximo de 20mA. Cada cátodo comum de cada dígito está ligado a um transístor, permitindo deste modo ligar ou desligar a sua ligação ao comum electrónico, controlando o nível lógico presente num porto do microcontrolador. Com este esquema de ligações, para controlar os quatro dígitos, apenas é necessário utilizar oito portas de IO do microcontrolador, ficando os dígitos do *display* multiplexados, i.e., só pode ser ligado um dígito em cada instante, assim, para que à vista desarmada pareça que todos os dígitos estão ligados em qualquer instante, é necessário repor o valor de cada dígito com uma frequência mínima de 15Hz. Para apresentar o valor num dígito é necessário que o cátodo comum do mesmo se encontre ligado ao comum electrónico, esta ligação é feita forçando um nível lógico alto na saída EnDisp# correspondente ao dígito, que por sua vez satura o transístor permitindo a passagem de corrente do colector para o emissor. Controlando a quantidade de tempo que se mantém activa cada uma das saídas

EnDisp e a frequência de reposição de cada dígito é possível controlar facilmente a intensidade luminosa do *display* e o seu consumo energético.

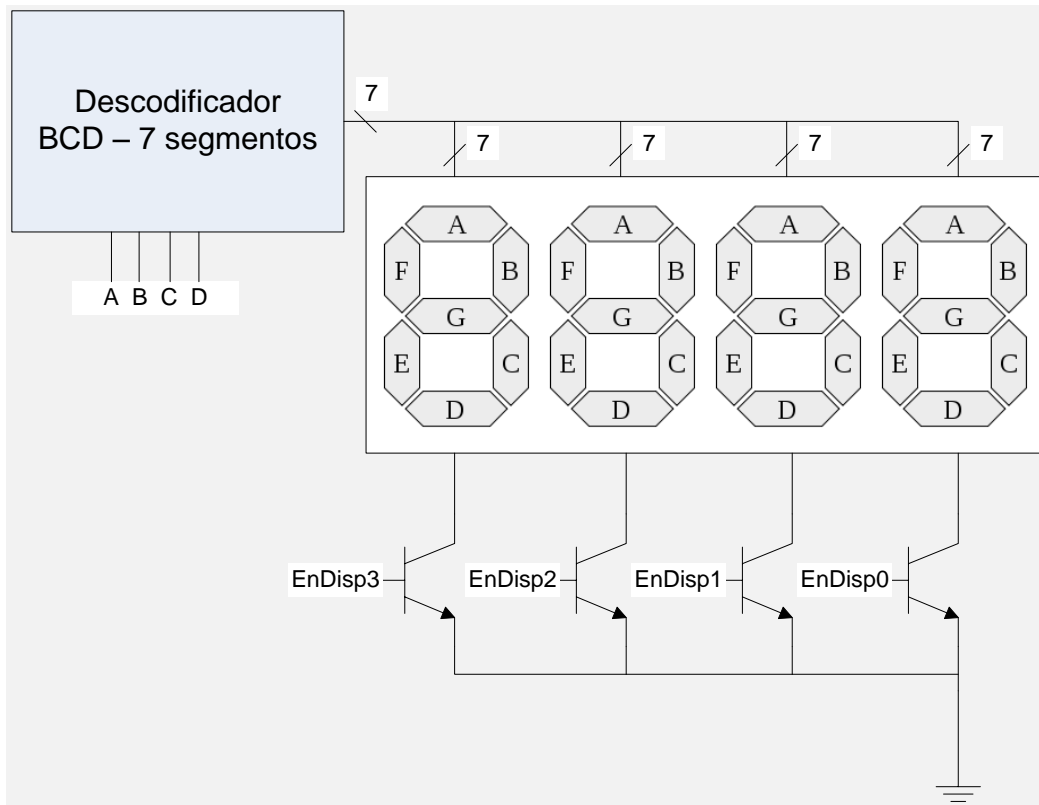


Fig. 5 – Interface do display com o microprocessador

2.1.2.4. Descodificador BCD para sete segmentos

O decodificador utilizado é um MC1451B fabricado pela On Semiconductor, que além de decodificador é também *latch* e *driver*. Este dispositivo tem quatro entradas de dados, três entradas de controlo e sete saídas para ligar ao *display* (Tabela 2). As entradas de controlo *Lamp Test* (LT), *Blanking* (Bl) e *Latch Enable* (LE) são utilizadas para testar o *display*, desligar o *display* ou regular a sua intensidade luminosa através da modulação por pulso da tensão aplicada e armazenar um código BCD respectivamente. Neste kit as entradas de controlo LT e Bl foram ligadas permanentemente a nível lógico alto e a entrada LE a nível lógico baixo, o que faz com que o valor que está nas saídas para o *display* seja sempre correspondente ao que está nas entradas BCD, não podendo ser efectuado armazenamento de dados.

Tabela 2 – Tabela de estado do funcionamento do *display*

Inputs							Outputs							
LE	BI	LT	D	C	B	A	a	b	c	d	e	f	g	Display
X	X	0	X	X	X	X	1	1	1	1	1	1	1	8
X	0	1	X	X	X	X	0	0	0	0	0	0	0	Blank
0	1	1	0	0	0	0	1	1	1	1	1	1	0	0
0	1	1	0	0	0	1	0	1	1	0	0	0	0	1
0	1	1	0	0	1	0	1	1	0	1	1	0	1	2
0	1	1	0	0	1	1	1	1	1	1	0	0	1	3
0	1	1	0	1	0	0	0	1	1	0	0	1	1	4
0	1	1	0	1	0	1	1	0	1	1	0	1	1	5
0	1	1	0	1	1	0	0	0	1	1	1	1	1	6
0	1	1	0	1	1	1	1	1	1	0	0	0	0	7
0	1	1	1	0	0	0	1	1	1	1	1	1	1	8
0	1	1	1	0	0	1	1	1	1	0	0	1	1	9
0	1	1	1	0	1	0	0	0	0	0	0	0	0	Blank
0	1	1	1	0	1	1	0	0	0	0	0	0	0	Blank
0	1	1	1	1	0	0	0	0	0	0	0	0	0	Blank
0	1	1	1	1	1	0	0	0	0	0	0	0	0	Blank
0	1	1	1	1	1	1	0	0	0	0	0	0	0	Blank
1	1	1	X	X	X	X				*				*

X = Don't Care

*Depends upon the BCD code previously applied when LE = 0

2.1.2.5. Encoders BCD

Os *encoders* BCD utilizados neste kit são o modelo P36S101 da APEM com dez posições (Fig. 6). Estes *encoders* têm um funcionamento totalmente mecânico e internamente são constituídos por quatro interruptores, com uma ligação comum, que estão abertos ou fechados consoante a posição da seta central. A ligação comum dos *encoders* foi ligada ao comum electrónico e as outras quatro directamente ao microcontrolador. Para um funcionamento correcto é necessário que cada uma das quatro saídas do *encoder* tenha uma ligação à tensão de nível lógico alto através de uma resistência elevando assim a tensão da saída para este nível sempre que o interruptor correspondente é aberto. Neste kit utilizam-se as resistências internas presentes nos portos do microcontrolador, configuráveis por software.

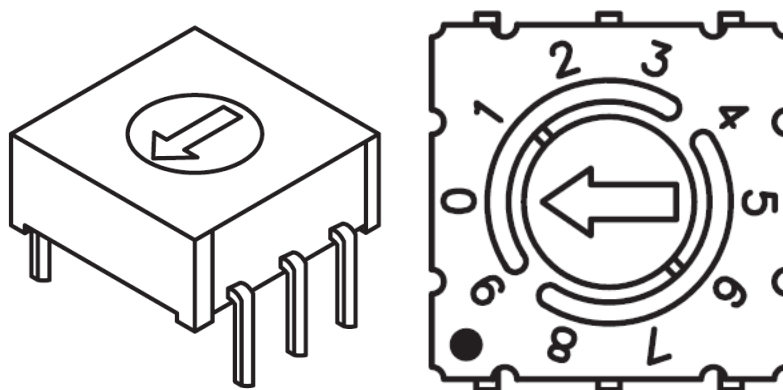


Fig. 6 - *Encoder* BCD modelo P36S101 da APEM

2.2. Kit Nível 1

2.2.1. Descrição Geral

O Kit de Nível 1 está direccionado para o aluno que possuam já alguns conhecimentos no domínio dos sistemas embutidos. Tem como base um Microcontrolador MSP430F2112 da TI, 8 botões de pressão, 8 LED e um *display* de dois dígitos de sete segmentos.

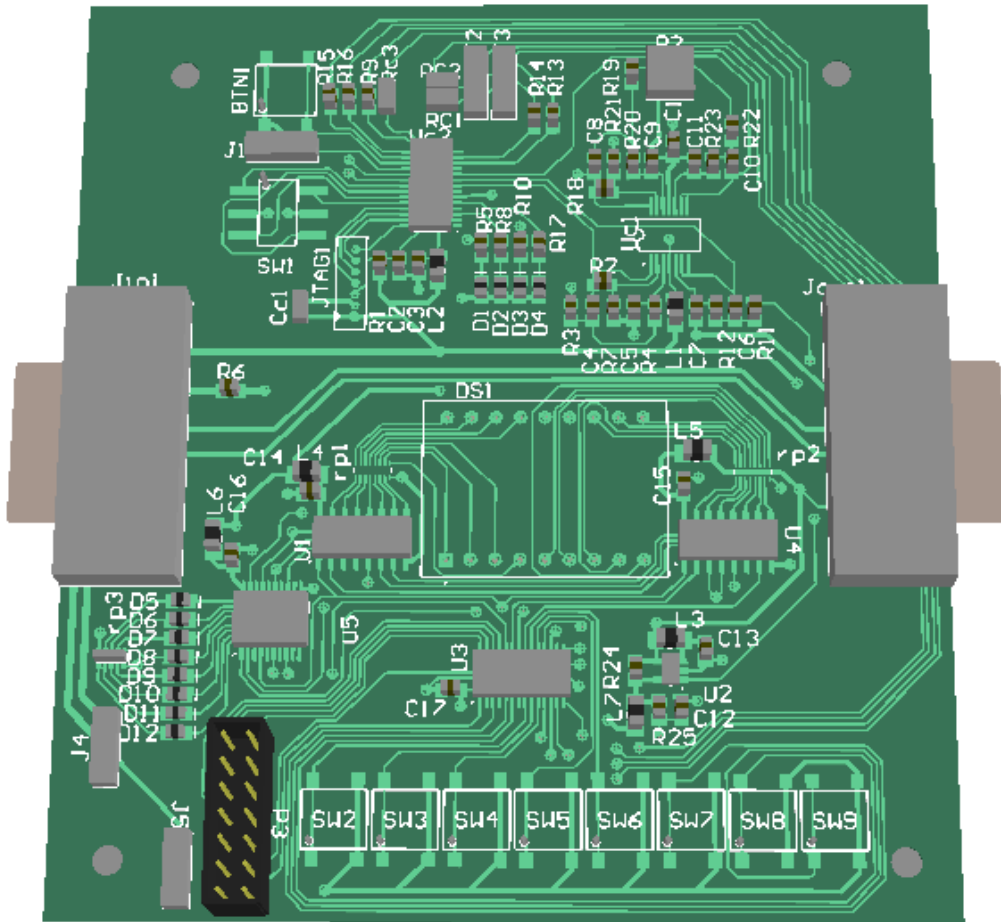


Fig. 7 – Desenho 3D do kit nível 1

A iniciação ao uso de interrupções e temporizadores/contadores é um ponto forte deste Kit, assim como a possibilidade de explorar as técnicas para aquisição de valor de um porto de IO, por *pooling* ou baseada em interrupção, analisando para cada uma o consumo de energia, e desenvolvendo código que minimize os erros de leitura originados pela flutuação dos sinais no porto, quando um botão é pressionado. Permite também a utilização de temporizadores/contadores ligados a interrupção para multiplexação dos *displays* e dos LED. Com este kit, o utilizador pode ter um primeiro contacto com a simples ligação do microcontrolador a outro dispositivo, obrigando-o a aprender por si próprio a necessidade de respeitar tempos de acesso e de atraso do dispositivo ligado.

O barramento SPI é utilizado para comunicar com os outros Kits existindo para esse efeito funções predefinidas.

Mostra-se a seguir como interagir com os periféricos usando código em C.

Leitura do valor dos botões de pressão sem *debounce*:

```
Buttons = P2IN;
```

Apresentar um dígito no display:

```
P3OUT = LE0; //latch/dígito seleccionado
```

```
P1OUT = 5; //valor a apresentar no display
```

```
P3OUT = NOSEL; //remover selecção (valor mantém-se no display)
```

Acender grupo de LEDs:

```
P3OUT = OE0; //Driver seleccionado
```

```
P1OUT = BIT1+BIT0; //Acende os dois primeiros LEDs
```

```
P3OUT = NOSEL; //remover selecção (LEDs apagam)
```

A Tabela 1 resume atribuição dos recursos disponibilizados para realizar acções de I/O.

Tabela 3 – Organização dos recursos de I/O do Kit 1

Pino	Função Geral	Nome	Função no Kit
21	P1.0	LED0/LED4/A0/A1	Saídas para <i>drivers</i> de LED e BCD para <i>displays</i>
22	P1.1	LED1/LED5/B0/B1	
23	P1.2	LED2/LED6/C0/C1	
24	P1.3	LED3/LED7/D0/D1	
25	P1.4	TCK	JTAG
26	P1.5	TMS	
27	P1.6	TDI	
28	P1.7	TDO	
8	P2.0	BT0	Entradas Botões
9	P2.1	BT1	
10	P2.2	BT2	
19	P2.3	BT3	
20	P2.4	BT4	
3	P2.5	BT5	
6	P2.6	BT6	
5	P2.7	BT7	
11	P3.0	STE	Barramento SPI
12	P3.1	SIMO	
13	P3.2	SOMI	
14	P3.3	CLK	
15	P3.4	LE0	Habilitação dos LED e <i>displays</i>
16	P3.5	LE1	
17	P3.6	OE0	
18	P3.7	OE1	
7	RST	RST	Pinos Reservados
1	TEST	TEST	
2	Vcc	Vcc	
4	Vss	Vss	

2.2.2. Dispositivos no Kit

2.2.2.1. Microcontrolador

O microcontrolador escolhido para integrar este kit foi o MSP430F2112 que tem as seguintes características principais: 2kB de memória flash, 256B de RAM, ADC de 10bit de 8 canais, 2 temporizadores/contadores com um total de 5 registos de comparação, uma interface de comunicação série e frequência de relógio máxima de 16MHz. Optou-se por este microcontrolador porque cumpre os requisitos necessários para uma iniciação aos sistemas embutidos, o número de saídas de I/O é suficiente para a aplicação desejada e pelo seu reduzido tamanho e custo. Uma esquematização da estrutura interna do dispositivo é feita na Fig. 2.

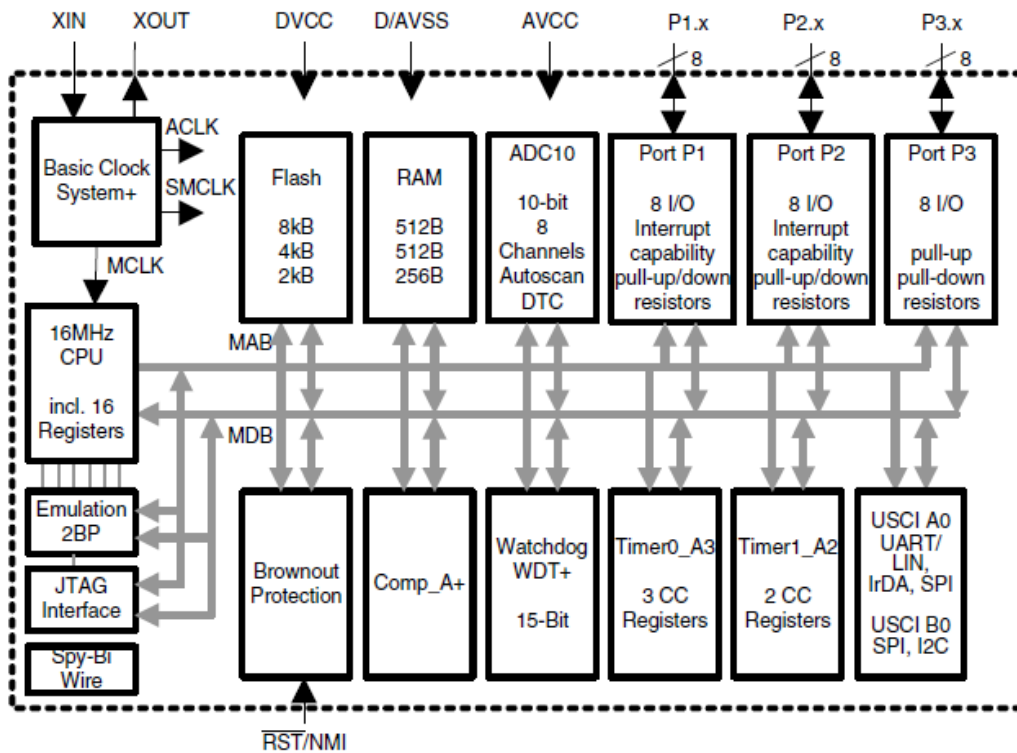


Fig. 8 – Organização interna do MSP430F2112

2.2.2.2. *Displays* de sete segmentos

O modelo de *display* seleccionado para este kit foi o HDSP-523E da Agilent Technologies composto por dois dígitos de sete segmentos de cor vermelha com cátodo comum e com uma dimensão de 14,2mm.

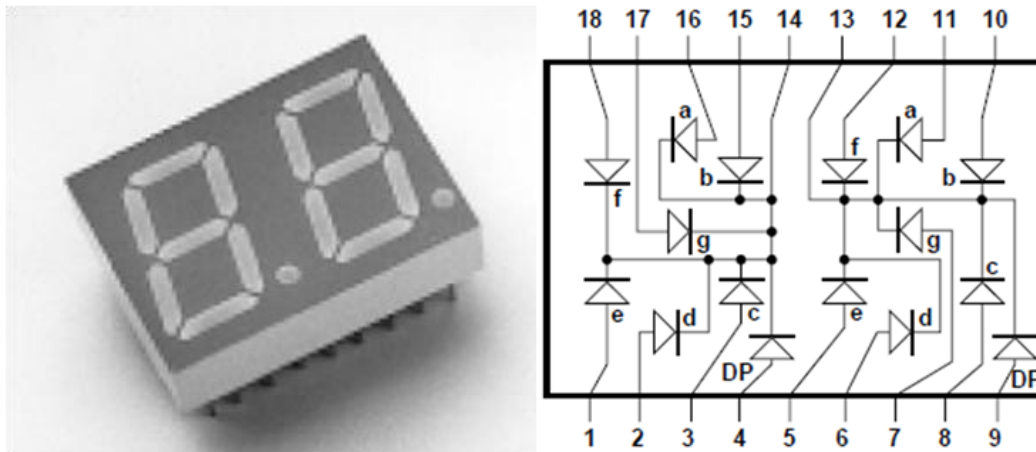


Fig. 9 - Display HDSP-523E fabricado pela Agilent Technologies

Cada dígito de um *display* de sete segmentos tem nove ligações eléctricas, uma para cada ânodo dos LED e mais uma que interliga todos os cátodos. Esta estratégia é ilustrada na Fig. 5. Neste kit, o ponto presente à direita de cada dígito não foi ligado, restando assim 8 ligações por dígito. Para diminuir o número de portas utilizados para controlar o *display* e simplificar a sua utilização, foram ligados ao microcontrolador dois descodificadores BCD para sete segmentos e a cada um destes os ânodos dos LED de cada dígito. Foram introduzidas resistências de 68Ω em série entre os descodificadores e o *display* de modo a limitar a corrente que atravessa cada segmento a 20mA. Os cátodos comuns do *display* foram ligados permanentemente ao comum electrónico e as entradas de código BCD dos descodificadores são partilhadas com oito LED utilizando um barramento paralelo de quatro fios. Para apresentar um valor num dígito do *display* é necessário activar o descodificador correspondente e desactivar todos os outros dispositivos ligados ao barramento utilizando as saídas LE e OE do microcontrolador.

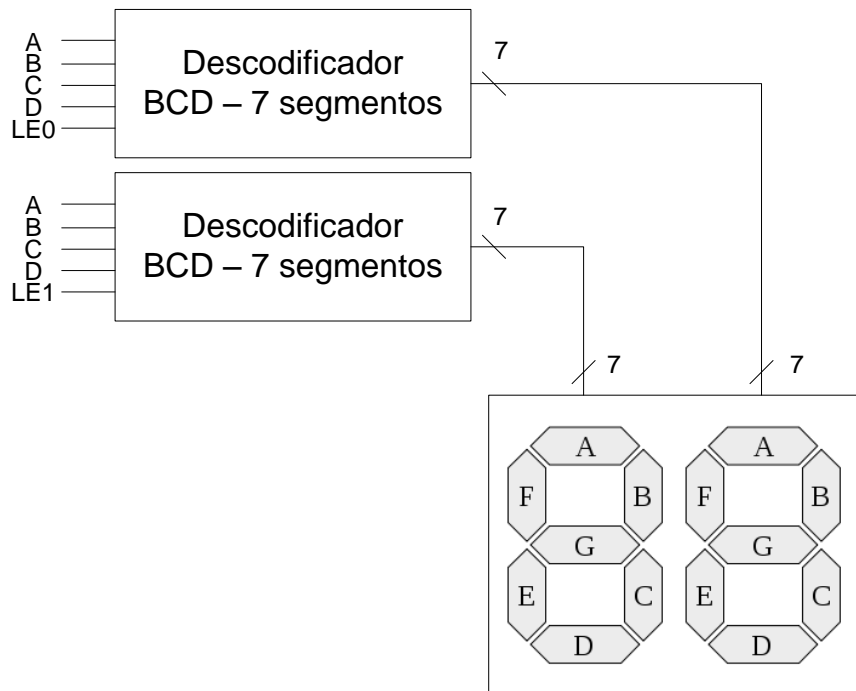


Fig. 10 – Interface do *display* com o microprocessador

2.2.2.3. Descodificadores BCD para sete segmentos

Os descodificadores utilizados são os MC1451B da On Semiconductor que além de descodificadores são também *latches* e *drivers*. Este dispositivo tem quatro entradas de dados, três entradas de controlo e sete saídas para ligar ao *display* tal como se representa na Tabela 2. As entradas de controlo *Lamp Test* (LT), *Blanking* (Bl) e *Latch Enable* (LE) são utilizadas para testar o *display*, desligar o *display* ou regular a sua intensidade através de modulação por pulsos e armazenar um código BCD respectivamente. Neste kit as entradas de controlo LT e Bl foram ligadas permanentemente a nível lógico alto e a entrada LE foi ligada ao microcontrolador. Quando a entrada LE se encontra a nível lógico baixo a informação presente nas entradas BCD é armazenada nos *latches* e, ao mesmo tempo, é descodificada para as saídas. Quando se passa a entrada LE para nível alto o valor descodificado para as saídas é o correspondente ao código BCD armazenado anteriormente nos *latches*.

Para modificar o valor apresentado por um dígito do *display* é necessário introduzir o valor nas saídas A a D do microcontrolador, habilitar os *latches* do descodificador correspondente e desabilitá-los novamente respeitando as especificações do fabricante apresentadas por um diagrama temporal da Fig. 11, onde t_{WL} corresponde ao intervalo de tempo durante o qual é necessário manter activa a entrada LE para que o código BCD seja armazenado correctamente nos *latches*. Este intervalo de tempo tem um valor típico de 260ns e um valor máximo de 520ns.

Tabela 4 – Tabela de estado do funcionamento do *display*

Inputs							Outputs							
LE	BI	LT	D	C	B	A	a	b	c	d	e	f	g	Display
X	X	0	X	X	X	X	1	1	1	1	1	1	1	8
X	0	1	X	X	X	X	0	0	0	0	0	0	0	Blank
0	1	1	0	0	0	0	1	1	1	1	1	1	0	0
0	1	1	0	0	0	1	0	1	1	0	0	0	0	1
0	1	1	0	0	1	0	1	1	0	1	1	0	1	2
0	1	1	0	0	1	1	1	1	1	1	0	0	1	3
0	1	1	0	1	0	0	0	1	1	0	0	1	1	4
0	1	1	0	1	0	1	1	0	1	1	0	1	1	5
0	1	1	0	1	1	0	0	0	1	1	1	1	1	6
0	1	1	0	1	1	1	1	1	1	0	0	0	0	7
0	1	1	1	0	0	0	1	1	1	1	1	1	1	8
0	1	1	1	0	0	1	1	1	1	0	0	1	1	9
0	1	1	1	0	1	0	0	0	0	0	0	0	0	Blank
0	1	1	1	0	1	1	0	0	0	0	0	0	0	Blank
0	1	1	1	1	0	0	0	0	0	0	0	0	0	Blank
0	1	1	1	1	1	0	0	0	0	0	0	0	0	Blank
0	1	1	1	1	1	1	0	0	0	0	0	0	0	Blank
0	1	1	1	1	1	1	0	0	0	0	0	0	0	Blank
1	1	1	X	X	X	X	*							*

X = Don't Care

*Depends upon the BCD code previously applied when LE = 0

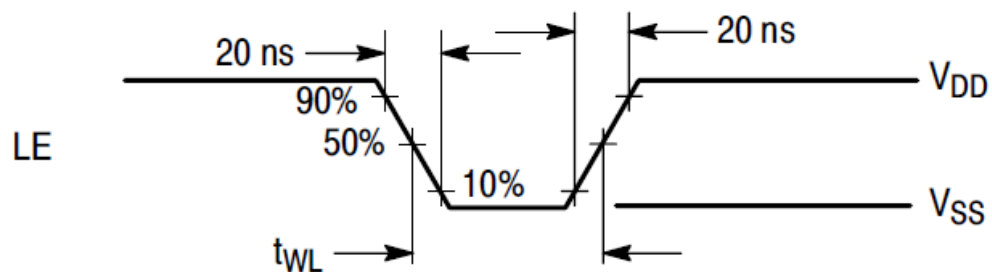


Fig. 11 – Diagrama temporal do decodificador BCD

2.2.2.4. *Buffer/Driver* de oito saídas

Integrado no kit está também um *Buffer/Driver* de oito saídas modelo SN74LVC244A da Texas Instruments (TI). Este dispositivo está organizado internamente como dois *drivers* de quatro bits com entradas de habilitação de saídas independentes (OE0 e OE1) tal como se representa na Fig. 12. Enquanto estiver presente o nível lógico baixo numa das entradas OE, o valor das linhas ABCD é repetido nas saídas correspondentes. Este dispositivo alimenta directamente as saídas não sobrecarregando os portos do microcontrolador com a corrente necessária para acender cada LED.

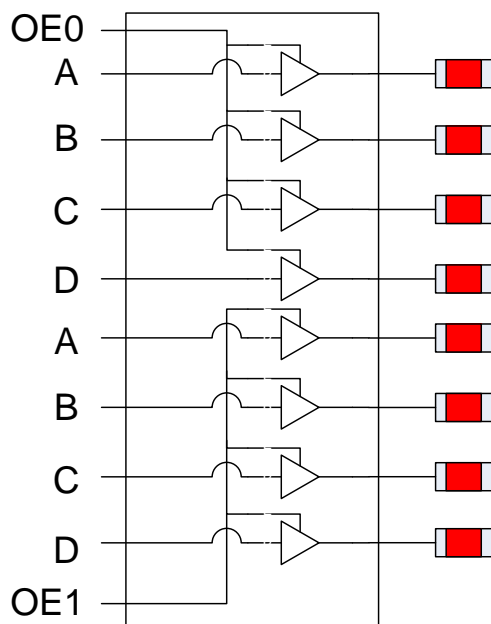


Fig. 12 - *Buffer/Driver* de oito saídas modelo SN74LVC244A da Texas Instruments

2.2.2.5. LED

Os LED seleccionados para este kit são os SML-210VTT86 da ROHM. As características principais destes LED são: cor vermelha, maior dimensão não ultrapassa 2mm, utilizam tecnologia SMD e têm uma intensidade luminosa típica de 4mcd. Neste kit, os LED foram ligados ao *driver* de oito saídas utilizando resistências de 68 Ω para limitar a corrente que atravessa cada LED a um valor máximo de 20mA.

2.2.2.6. Interruptores de pressão

Os interruptores de pressão aplicados neste kit são os B3S 1000 da Omron do tipo normalmente aberto, de forma quadrada, com 6mm de lado e tempo máximo de oscilação de 5ms que possui o aspecto apresentado na Fig. 13. Os oito interruptores de pressão incluídos no kit têm uma ligação comum que pode ser ligada a nível lógico alto ou baixo, utilizando um *jumper* instalado para esse efeito, consoante se queira que o pressionar do interruptor corresponda a nível lógico baixo ou a alto, respectivamente. A inicialização do porto onde ligam os interruptores tem obrigatoriamente de ser alterada de acordo com a posição do *jumper* exterior, se a ligação comum estiver a nível baixo é necessário uma resistência de *pull-up*, se estiver a nível alto é necessário uma resistência de *pull-down*.

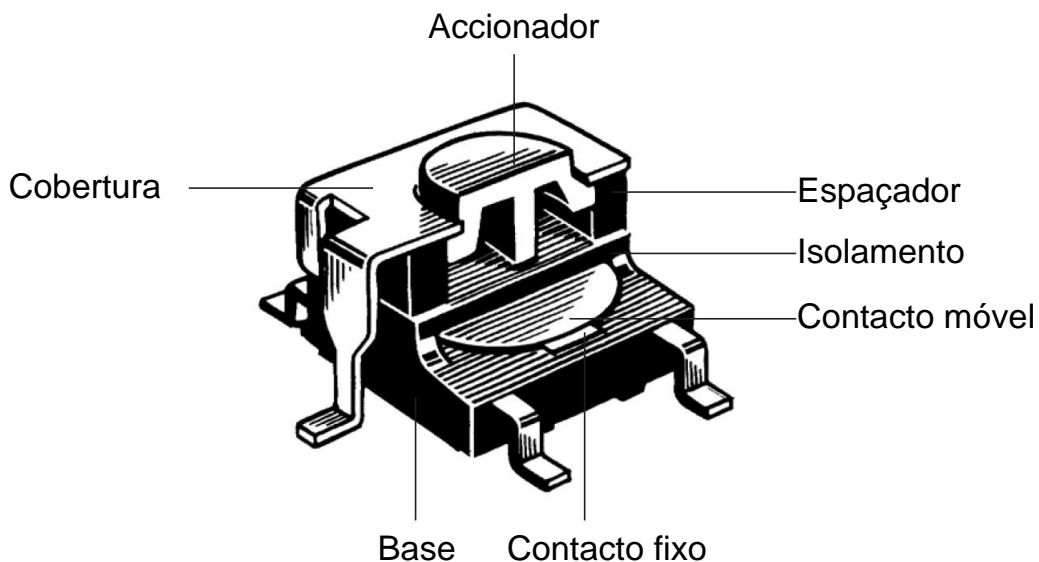


Fig. 13 – Aspecto do interruptor de pressão utilizado

2.3. Kit Nível 2

2.3.1. Descrição Geral

O Kit de Nível 2 está direccionado para o aluno com amplo conhecimento em sistemas embutidos, experiência e à-vontade na utilização das funções internas dos microcontroladores da Texas Instruments (TI). Tem por base um microcontrolador MSP430FG4618 da TI, um *joystick* de navegação, um encoder rotativo multivolta, altifalante, microfone e respectivas saídas em *jack* de 3,5mm e um LCD alfanumérico de 8 caracteres.

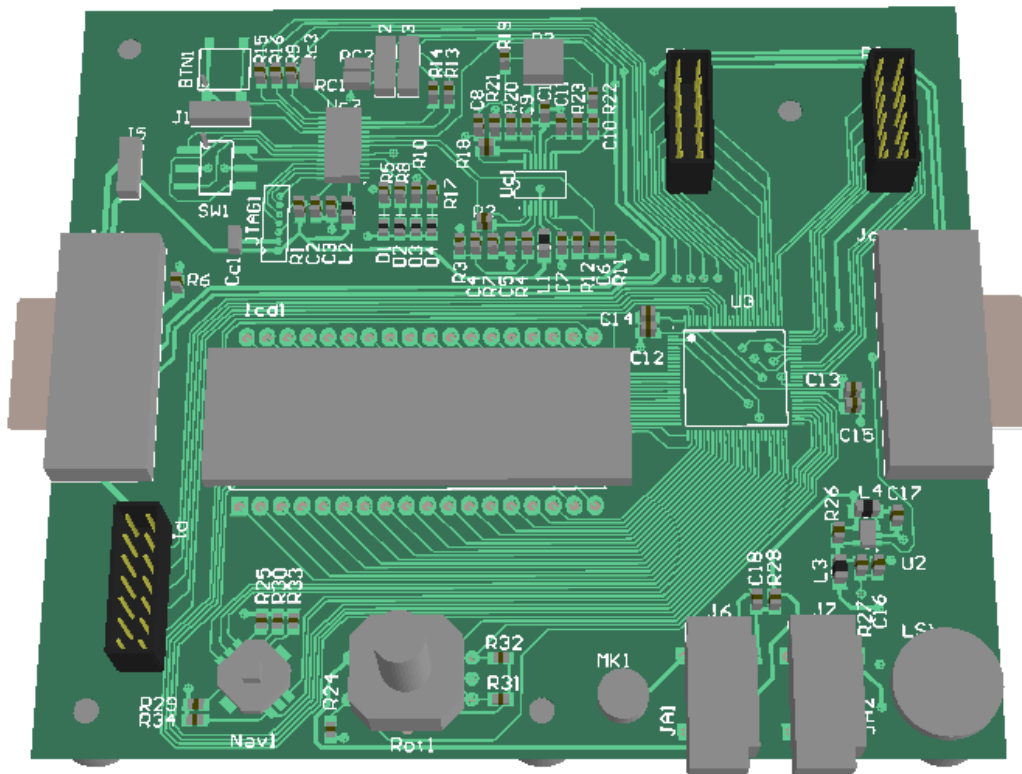


Fig. 14 - Desenho 3D do kit nível 2

Com este Kit pretende-se explorar a área de conhecimento relacionada com o domínio analógico utilizando o máximo de funcionalidades do MSP430FG4618, desde o reconhecimento de comandos de voz à construção de filtros digitais, para além destes existem muitos trabalhos que podem ser desenvolvidos. O LCD permite uma boa introdução ao funcionamento dos LCD em geral, uma vez que o controlador é integrado no microcontrolador, sendo necessário fazer a programação do mesmo. Tanto o *joystick* de navegação como *encoder* rotativo são duas interfaces de grande versatilidade quando associadas ao LCD permitindo o desenvolvimento de aplicações com elevado grau de complexidade.

O MSP430FG4618 está orientado para as áreas de sinais analógicos e controlo de LCD. De todas as funcionalidades incluídas destacam-se: um ADC de 12bit com 16 canais multiplexados, dois DACs de 12bit, três Amp-Ops, um controlador DMA de três canais e um controlador de LCD de 160 segmentos multiplexados quatro a quatro.

O *Direct-Memory Access* (DMA), utilizado actualmente em todos os computadores, permite que outros dispositivos, que não o processador, acessem directamente à memória sem interferência do processador, deixando-o livre para desempenhar outras tarefas. Esta característica permite reduzir em larga escala o *overhead* do processador quando são realizadas transferências de dados de periféricos para a memória. No microcontrolador utilizado, o controlador DMA não permite acessos directos à memória por periféricos ligados externamente ao microcontrolador. Este controlador é totalmente configurável via *software* e permite, por exemplo, que os valores convertidos pelo ADC sejam transferidos directamente para a memória ou, por exemplo, realizar uma transferência de vários bytes entre diferentes posições de memória. Para qualquer uma destas aplicações obtêm-se grandes melhorias na performance geral do dispositivo em relação ao uso do processador para realizar a mesma operação sem que os canais de DMA sejam utilizados.

2.3.2. Cadeia de instrumentação analógica

Na parte dedicada a sinais analógicos deste Kit pretende-se que os alunos aprendam a configurar os vários módulos internos do microcontrolador, para isso foi criada uma cadeia de instrumentação analógica (Fig. 15). O sinal inicial é adquirido a partir do microfone ou da entrada por *jack*, e está ligado directamente à entrada 0 do Amp-Op1 e ao canal 4 do ADC. O valor do sinal pode ser lido imediatamente com o ADC, ou ser tratado previamente pelo Amp-Op1. A saída do Amp-Op1 está ligada internamente ao canal 13 do ADC e o seu valor pode ser lido a partir daqui. Nesta fase da cadeia, o sinal analógico foi convertido em digital e pode ser utilizado para a aplicação pretendida. A fase seguinte é a conversão do sinal digital para analógico. Para isso é utilizado o DAC0 que tem a sua saída ligada à entrada 0 do Amp-Op2 e ao canal 6 do ADC. Para finalizar, a cadeia de instrumentação encontram-se ligados à saída do Amp-Op2, o altifalante e respectiva saída em *jack* e o canal 5 do ADC.

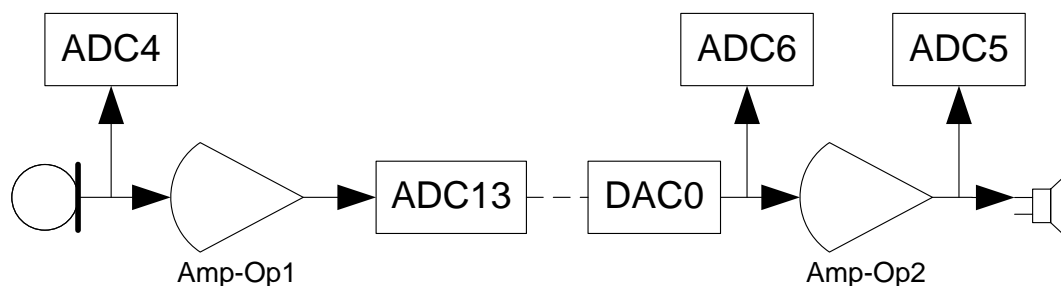


Fig. 15 – Ilustração da cadeia de instrumentação

2.3.3. Funcionalidades integradas no microcontrolador

2.3.3.1. Amplificadores Operacionais

As características principais dos Amp-Op integrados no microcontrolador são: baixo consumo energético, saídas e entradas *rail-to-rail*, entradas seleccionáveis por *software* e uma malha de resistências de *feedback* seleccionadas por *software* para amplificadores de ganho programável (PGA). As entradas inversora e não-inversora destes Amp-Op são configuradas utilizando os bits OANx e OAPx onde se podem seleccionar tanto sinais exteriores, a partir de um ou mais entradas do microcontrolador, como um sinal interno gerado pelo DAC do microcontrolador. As saídas são também configuráveis e podem ser ligadas internamente ao ADC, utilizando o bit OAADC0, ou externamente a através de uma entrada do microcontrolador. Para configurar cada Amp-Op utilizam-se dois registos do microcontrolador. O registo OAxCTL0 que inclui os bits OANx, OAPx, OAPMx, OAADC1 e OAADC0 e o registo OAxCTL1 que inclui os bits OAFBRx, OAFcx, OARRIP.

Os bits OAPMx definem a *slew rate*, correspondente ao tempo de resposta de uma alteração no sinal de entrada do Amp-Op até à sua verificação na saída, e permitem quatro posições. Para quanto maior for configurada a velocidade de resposta do amplificador uma maior quantidade de energia será consumida pelo mesmo, ficando ao critério do utilizador qual a configuração mais adequada de acordo com a sua aplicação. A ligação da saída de cada amplificador configura-se utilizando os bits OAADC1 e OAADC0.

Utilizando os bits OAFcx é possível configurar o modo de operação dos Amp-Op segundo a Tabela 5.

Tabela 5 – Configuração dos modos de operação dos Amp-Op

OAFcx	Modos de funcionamento
000	Amp-Op de propósito geral
001	<i>Buffer</i> de ganho unitário
010	Reservado
011	Comparador
100	Amplificador não inversor de ganho programável
101	Reservado
110	Amplificador inversor de ganho programável
111	Amplificador diferencial

Descrição dos modos de funcionamento

Amp-Op de propósito geral

Neste modo, tal como se esquematiza na Fig. 17, a malha de resistências de feedback é desligada do amplificador, os bits OANx e OAPx definem as entradas do amplificador e a sua saída é ligada internamente ao ADC12 segundo o que for definido no registo OAxCTL0.

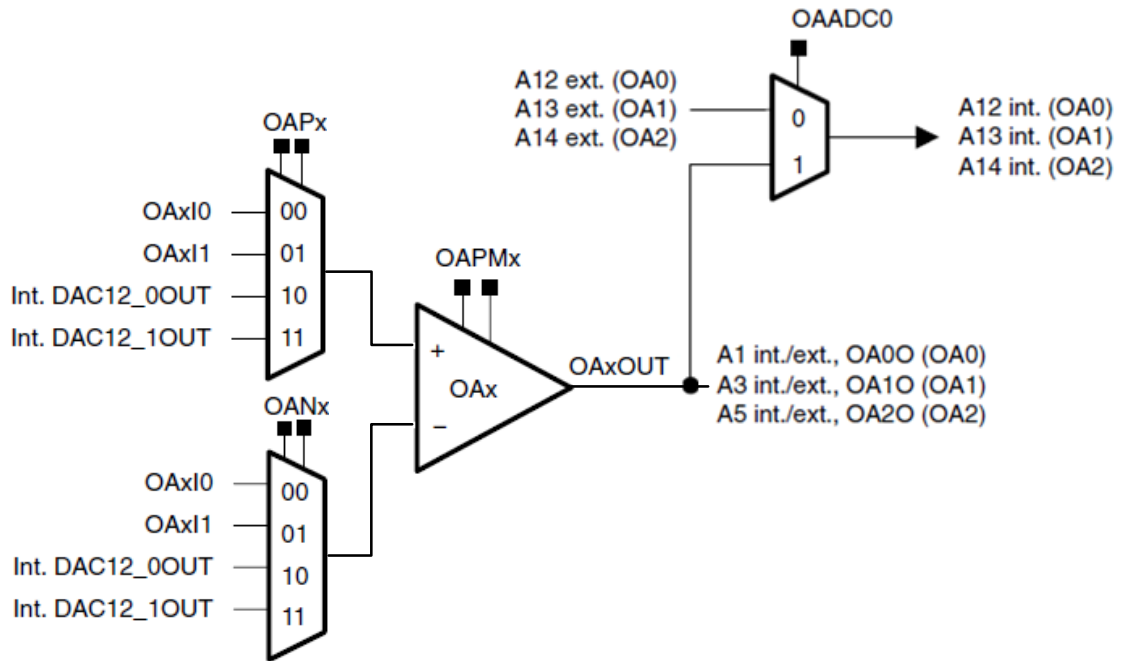


Fig. 17 - Amp-Op de propósito geral

Buffer de ganho unitário

Neste modo, tal como se esquematiza na Fig. 18, a saída do amplificador é ligada a uma das extremidades da malha de resistências (R_{bottom}) e à entrada inversora do amplificador. A entrada não inversora é seleccionada a partir dos bits OAPx e a ligação externa da entrada inversora é desligada ficando os bits OANx sem qualquer função. A saída do amplificador é ligada internamente ao ADC12 segundo o que for definido pelo programador no registo OAxCTL0.

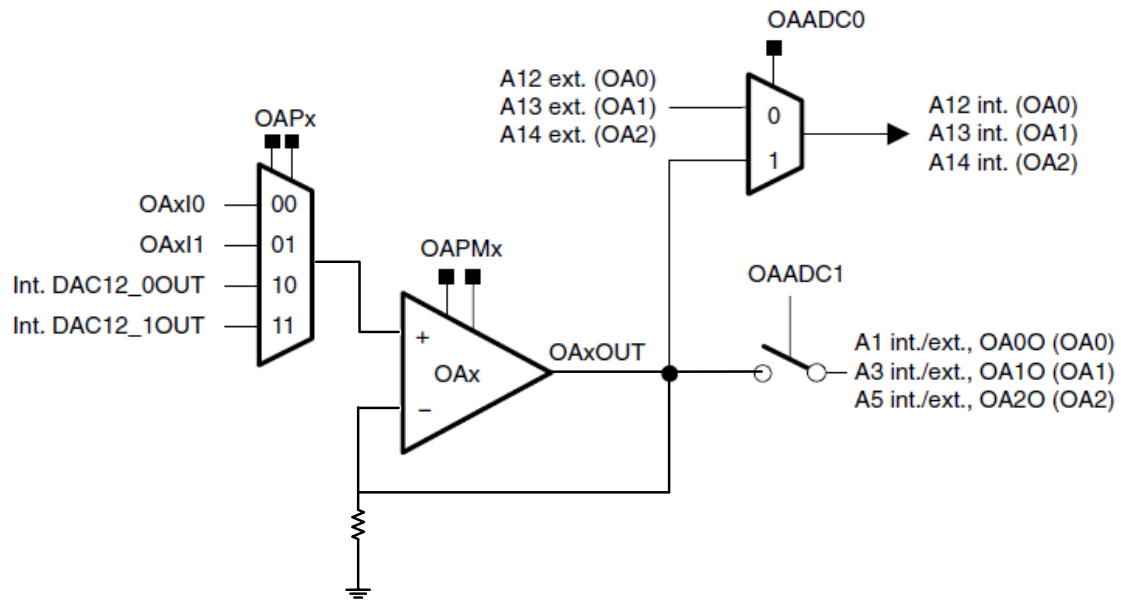


Fig. 18 - Buffer de ganho unitário

Comparador

Neste modo, tal como se esquematiza na Fig. 19, a saída do amplificador é desligada da malha de resistências. Ao topo da malha de resistências é ligado Vss e à base é ligado Vcc. O sinal de saída do selector de resistências é ligado à entrada inversora do amplificador fornecendo assim uma tensão regulável a partir dos bits OAFBRx. A entrada não inversora é seleccionada a partir dos bits OAPx e a ligação externa da entrada inversora é desligada ficando os bits OANx sem qualquer função. A saída do amplificador é ligada internamente ao ADC12 segundo o que for definido no registo OAxCTL0.

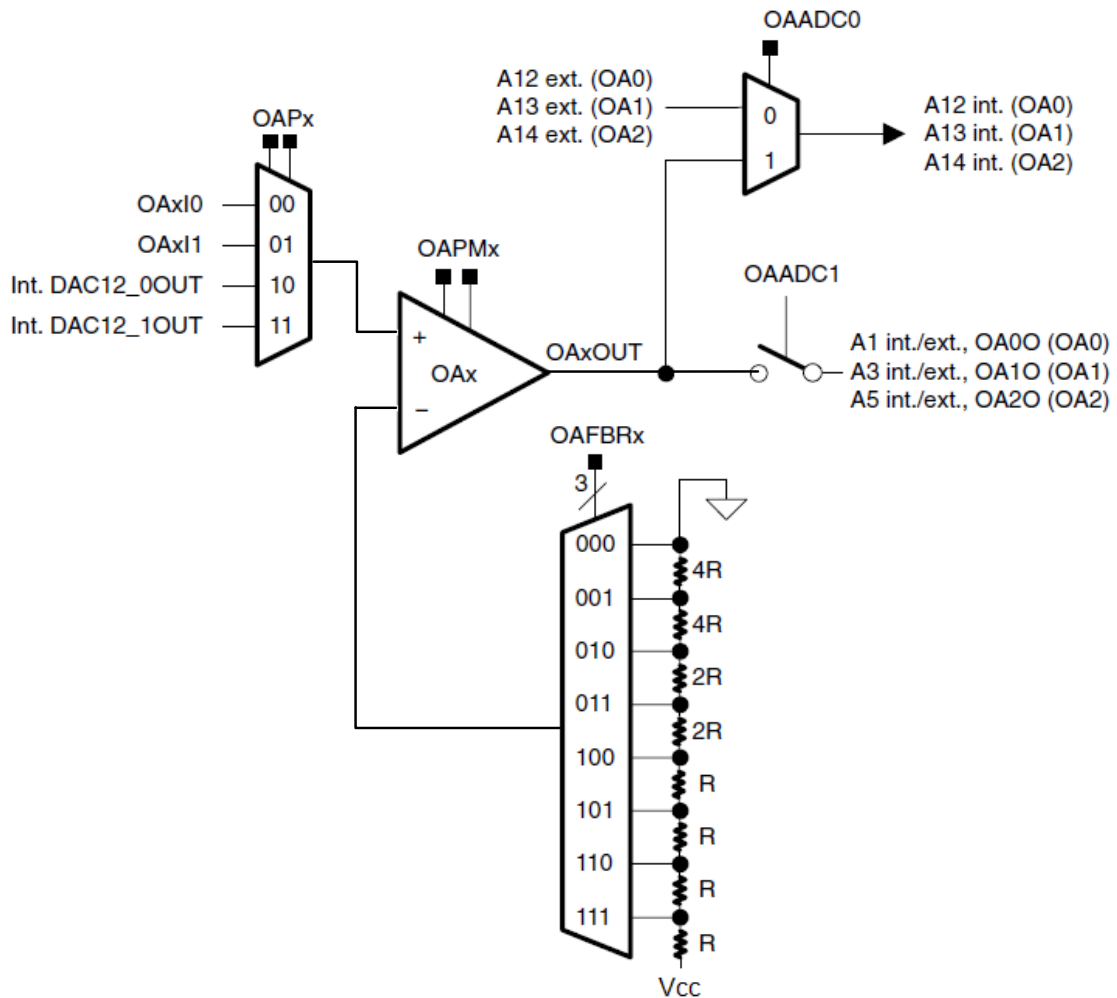


Fig. 19 - Comparador

Amplificador não inversor de ganho programável

Neste modo, tal como se esquematiza na Fig. 20, a saída do amplificador está ligada ao topo da malha de resistências e à base desta é ligado Vss. O sinal de saída do selector de resistências é ligado à entrada inversora do amplificador, realizando assim um amplificador não inversor de ganho ($1 + \text{rácio de resistências}$). Este rácio é estabelecido utilizando os bits OAFBRx e fica configurado para ganho unitário quando OAFBRx=0. A entrada não inversora é seleccionada a partir dos bits OAPx e a ligação externa da entrada inversora é desligada ficando os bits OANx sem qualquer função. A saída do amplificador é ligada internamente ao ADC12 segundo o que for definido no registo OAxCTL0.

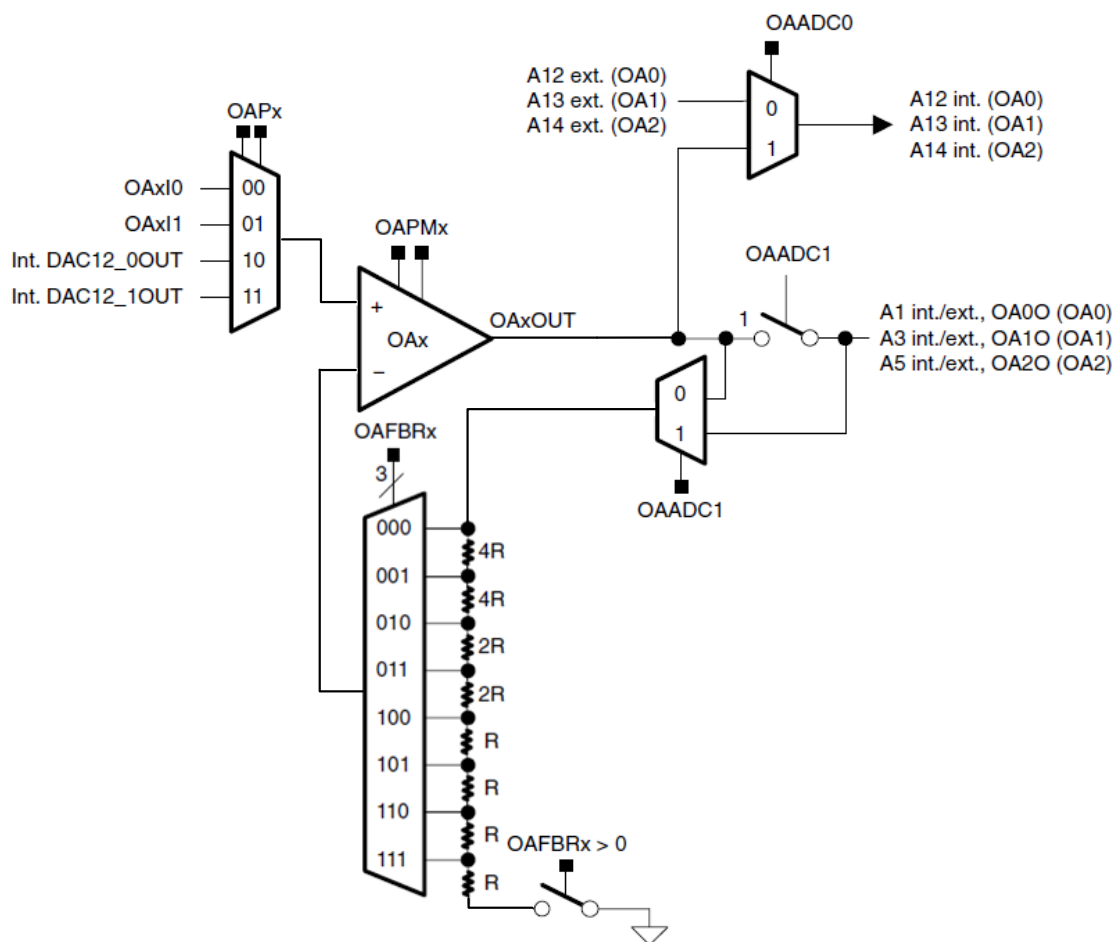


Fig. 20 - Amplificador não inversor de ganho programável

Amplificador inversor de ganho programável

Neste modo, tal como se esquematiza na Fig. 21, a saída do amplificador é ligada ao topo da malha de resistências. A ligação da base da malha de resistências é seleccionada utilizando os bits OANx. O sinal de saída do selector de resistências é ligado à entrada inversora do amplificador realizando assim um amplificador inversor de ganho (– rácio de resistências). Este rácio é estabelecido utilizando os bits OAFBRx. A entrada não inversora é seleccionada a partir dos bits OAPx. A saída do amplificador é ligada internamente ao ADC12 segundo o que for definido no registo OAxCTL0.

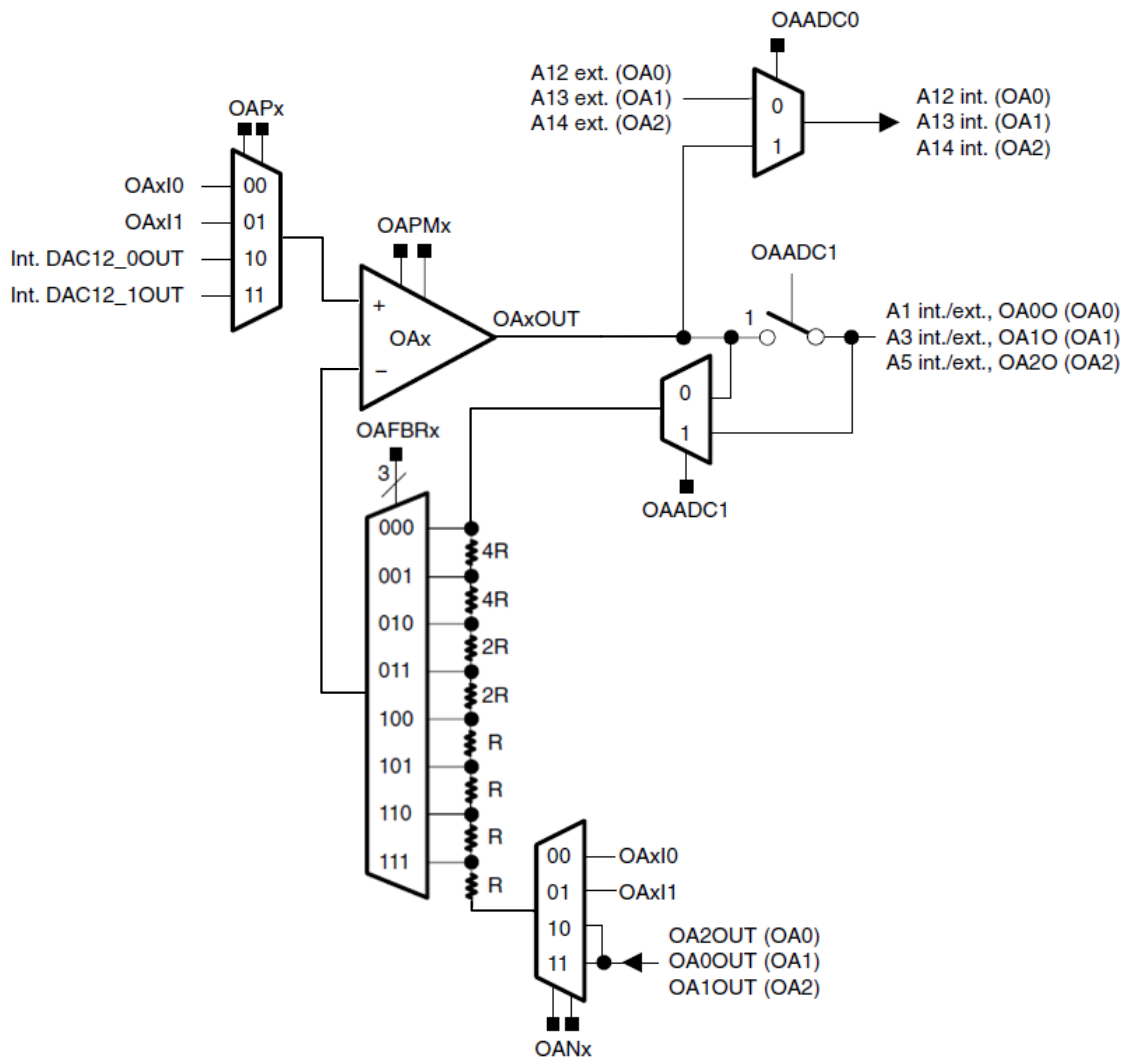


Fig. 21 - Amplificador inversor de ganho programável

Amplificador diferencial

Este modo permite realizar ligações internas dos sinais dos Amp-Op de forma a construir um amplificador de instrumentação de dois ou três Amp-Op. Neste modo, a saída de um amplificador é ligada ao topo da malha de resistências através da ligação a outro amplificador configurado no modo amplificador inversor de ganho programável. A base da malha de resistências não é ligada ficando com um ganho unitário. Este *buffer* é então combinado com um ou dois dos restantes amplificadores formando um amplificador de instrumentação. A saída do amplificador é ligada internamente ao ADC12 segundo o que for definido no registo OAxCTL0.

A Fig. 22 representa o esquema de ligações para duas possíveis configurações utilizando os Amp-Op OA0, OA1 e OA2.

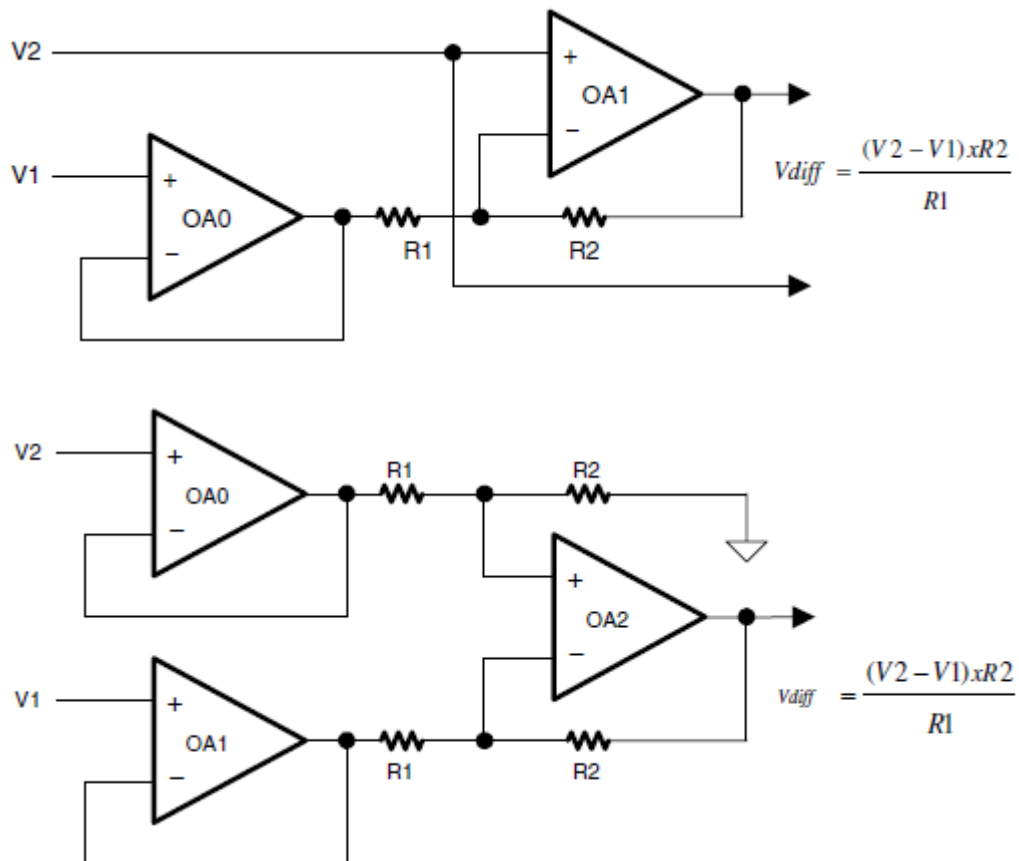


Fig. 22 - Ligações para uma possível configuração utilizando os Amp-Op OA0, OA1 e OA2.

2.3.3.2. Conversor analógico para digital

O microcontrolador utilizado neste kit integra um ADC do tipo SAR de 12 bit com 16 canais multiplexados, 12 deles com ligação externa, que tem como características principais: taxa de conversão superior a 200ksps, módulo de *sample-and-hold* com períodos de amostragem controlados por software ou *timers*, tensão de referência gerada internamente e seleccionada por software (1,5V ou 2,5V) e um *buffer* de 16 palavras que permite ao módulo realizar e armazenar até 16 conversões sem intervenção do processador.

Um ADC do tipo SAR tem como princípio de funcionamento a comparação do valor de tensão a converter (V_{in}) com valores de tensão fornecidos por um DAC (V_{dac}) através de um processo iterativo de aproximações sucessivas. A entrada digital do DAC (registo de 12bit) é sucessivamente alterada para possibilitar as comparações necessárias. No final da conversão, o valor do registo do DAC corresponde a V_{in} convertido. O processo de conversão é iniciado capturando uma amostragem de V_{in} . De seguida, activa-se o bit mais significativo do registo do DAC, o que força $V_{dac} = V_{REF}/2$. V_{in} é comparado com V_{dac} e, no caso de V_{in} ser maior que V_{dac} mantém-se o bit mais significativo a '1', caso seja menor, o bit mais significativo é alterado para '0'. A conversão continua forçando o bit seguinte a '1' e realizando outra comparação e só termina depois de ser comparado o bit menos significativo.

Os ADC deste tipo destacam-se dos outros pelo baixo consumo energético, alta resolução e precisão e tamanho reduzido.

A Fig. 23 representa as sucessivas comparações/aproximações de um ADC do tipo SAR de 4 bits.

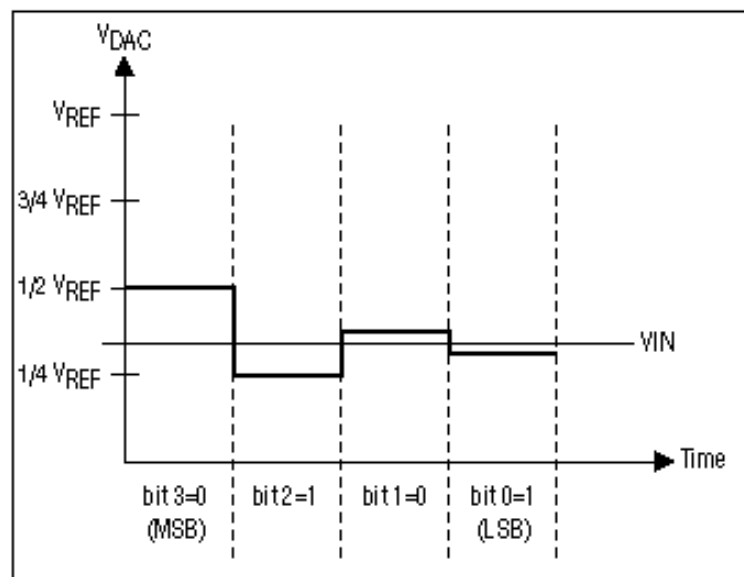


Fig. 23 - ADC do tipo SAR de 4 bits

Funcionamento do módulo ADC

O ADC integrado no microcontrolador é completamente configurável por *software* e antes de poder ser utilizado é necessário definir alguns parâmetros. O núcleo do ADC é configurado utilizando os registos ADC12CTL0 e ADC12CTL1.

A frequência de relógio de funcionamento do ADC é configurada seleccionando um *clock* do microcontrolador através dos bits ADC12SSELx, sendo possível aplicar-lhe um divisor através dos bits ADC12DIVx, ambos no registo ADC12CTL1. Podem ser seleccionadas como fonte de relógio o SMCLK, MCLK, ACLK ou o ADC12OSC, oscilador gerado internamente no módulo do ADC com uma frequência de 5MHz.

O módulo do ADC integra um gerador de tensões de referência de 2 níveis (1,5V e 2,5V). Definindo o bit REFON do registo ADC12CTL0 activa-se o gerador e a tensão de 2,5V pode ser seleccionada definindo REF2_5V=1. No caso de REF2_5=0 a tensão de referência é de 1,5V.

O registo ADCCTL0 permite configurar ainda os intervalos de amostragem para os registos 8 a 15 e para os registos 0 a 7 utilizando os bits SHT1x e SHT0x respectivamente, assim como interrupções associadas à conversão e modo de amostragem e conversão. O bit Enc activa o ADC e o bit ADC12SC dá início à conversão.

A memória de conversão está dividida em duas partes, uma para armazenar os valores convertidos (ADC12MEMx) e outra para armazenar configurações (ADC12MCTLx). A cada registo ADC12MEMx corresponde um registo ADC12MCTLx que guarda as configurações para a conversão. Os registos ADC12MCTLx definem a tensão de referência e o canal do ADC a converter e têm ainda mais um bit (EOS) que indica o fim da sequência, utilizado nalguns modos de conversão.

Existem quatro modos de conversão disponíveis tal como se descreve na Tabela 6, configuráveis através dos bits CONSEQx do registo ADC12CTL1.

Tabela 6 – Modos de conversão

CONSEQx	Modo de conversão
00	Um canal convertido uma vez
01	Sequência de canais convertidos uma vez
10	Um canal convertido repetidamente
11	Sequência de canais convertidos repetidamente

Os bits CSTARTADDx definem o registo em que irá ser iniciada a conversão, ou seja, o primeiro ADC12MCTLx utilizado para qualquer conversão que se inicie. Se o modo de conversão escolhido for sequencial, a conversão é iniciada neste registo e termina no primeiro registo que tiver o bit EOS activo.

O ADC integrado no microcontrolador tem também um sensor de temperatura e a sua tensão pode ser lida utilizando o canal 10 do ADC. A função de transferência do sensor é apresentada na Fig. 24.

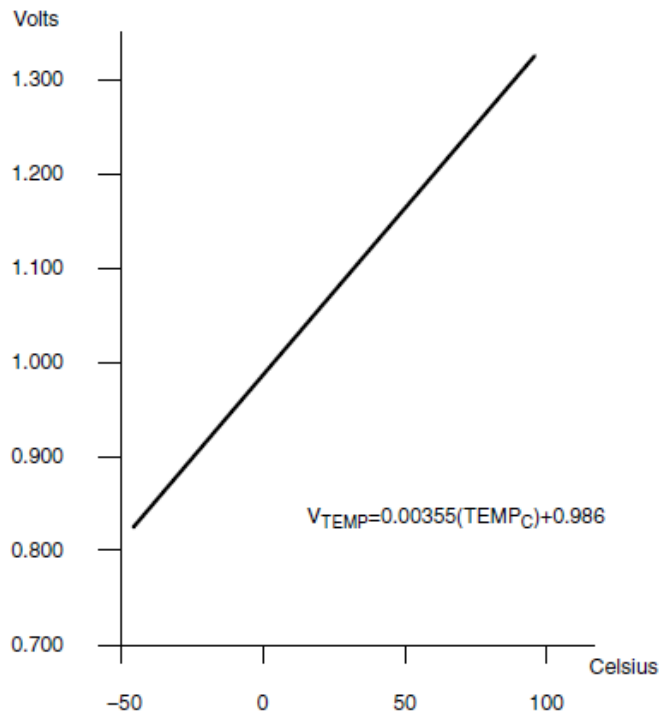


Fig. 24 – Sensor de temperatura interno ao módulo ADC – relação de transformação

2.3.3.3. Conversor digital para analógico

O microcontrolador deste kit também possui integrados dois módulos DAC de 12 bits configuráveis via *software*. O módulo DAC pode funcionar com uma resolução de 8 ou 12 bits, definida utilizando o bit DAC12RES. Para configurar o módulo DAC utiliza-se o registo DAC12_xCTL onde se define a tensão de referência, a resolução do DAC e o *trigger* para iniciar uma nova conversão. Depois de configurado, o seu funcionamento de base é simples e para realizar uma conversão apenas é necessário escrever num registo o valor a converter. A Tabela 7 descreve a função atribuída a cada um dos pinos do microcontrolador.

Tabela 7 – Organização dos recursos de I/O do Kit 2

Pinos	Função Geral	Nome	Função no Kit
1	Vcc	Vcc	Alimentação
2	P6.3	A3	Cadeia Analógica
3	P6.4	OA1in0/A4	
4	P6.5	OA2out	
5	P6.6		Não Utilizados
6	P6.7		
7	Vref+		
8	Xin		
9	Xout		
10	Ve.ref+		
11	Vref-		
12	P5.1	SEG0	LCD
13	P5.0	SEG1	
14	P10.7	SEG2	
15	P10.6	SEG3	
16	P10.5	SEG4	
17	P10.4	SEG5	
18	P10.3	SEG6	
19	P10.2	SEG7	
20	P10.1	SEG8	
21	P10.0	SEG9	
22	P9.7	SEG10	
23	P9.6	SEG11	
24	P9.5	SEG12	
25	P9.4	SEG13	
26	P9.3	SEG14	
27	P9.2	SEG15	
28	P9.1	SEG16	
29	P9.0	SEG17	
30	P8.7	SEG18	
31	P8.6	SEG19	
32	P8.5	SEG20	
33	P8.4	SEG21	
34	P8.3	SEG22	
35	P8.2	SEG23	
36	P8.1	SEG24	
37	P8.0	SEG25	
38	P7.7	SEG26	
39	P7.6	SEG27	
40	P7.5	SEG28	
41	P7.4	SEG29	
42	P7.3	SEG30	
43	P7.2	SEG31	
44	P7.1		
45	P7.0		
46	P4.7	NavD	Joystick de Navegação
47	P4.6	NavB	
48	P4.5	NavA	
49	P4.4	NavE	
50	P4.3	NavC	

Pinos	Função Geral	Nome	Função no Kit
51	P4.2		Não Utilizado
52	COM0	COM0	LCD
53	P5.2	COM1	
54	P5.3	COM2	
55	P5.4	COM3	
56	P5.5	RotA	Encoder Rotativo
57	P5.6	RotB	
58	P5.7	RotS	
59	LDCAP	LCDCA P	Reservado
60	Vcc	Vcc	Alimentação
61	Vss	Vss	
62	P4.1		Não Utilizados
63	P4.0		
64	P3.7	P3.7	Porto de Expansão
65	P3.6	P3.6	
66	P3.5	P3.5	
67	P3.4	P3.4	
68	P3.3	CLK	Barramento SPI
69	P3.2	SOMI	
70	P3.1	SIMO	
71	P3.0	STE	
72	P2.7	P2.7	Porto de Expansão
73	P2.6	P2.6	
74	P2.5	P2.5	
75	P2.4	P2.4	
76	P2.3	P2.3	
77	P2.2	P2.2	
78	P2.1	P2.1	
79	P2.0	P2.0	
80	P1.7	P1.7	
81	P1.6	P1.6	
82	P1.5	P1.5	
83	P1.4	P1.4	
84	P1.3	P1.3	
85	P1.2	P1.2	
86	P1.1	P1.1	
87	P1.0	P1.0	
88	XT2out		Não Utilizados
89	XT2in		
90	TDO	TDO	JTAG
91	TDI	TDI	
92	TMS	TMS	
93	TCK	TCK	
94	RST	RST	Reset
95	P6.0		Não Utilizados
96	P6.1		
97	P6.2		
98	Avss	Avss	Alimentação
99	Vss	Vss	
100	Avcc	Avcc	

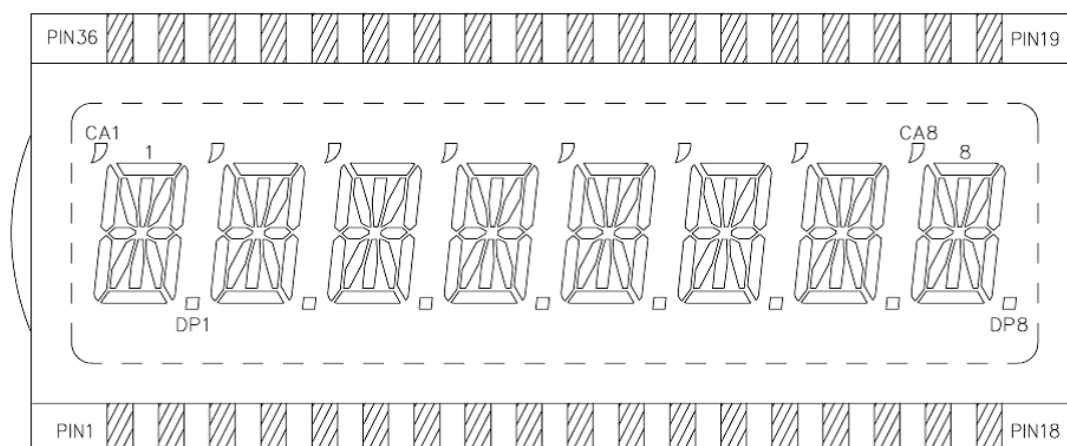
2.3.4. Dispositivos no kit

2.3.4.1. LCD alfanumérico

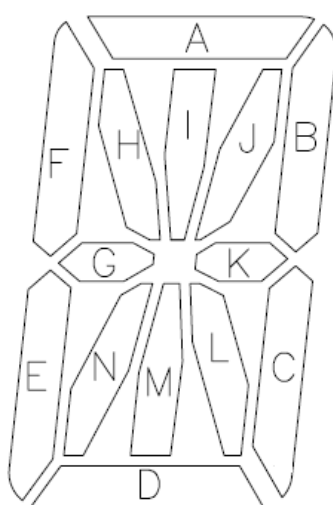
O LCD integrado no Kit tem oito caracteres com 16 segmentos cada, incluindo um ponto e uma plica em cada um. Estes segmentos são multiplexados através de quatro linhas comuns e encontram-se ligados às saídas do controlador LCD do microcontrolador segundo a Tabela 8

Tabela 8 – Interface do controlador LCD presente no microcontrolador

Pino uC	Pino LCD	COM0	COM1	COM2	COM3
COM0	17	COM0	-	-	-
COM1	18	-	COM1	-	-
COM2	19	-	-	COM2	-
COM3	20	-	-	-	COM3
SEG00	36	1M	1N	1G	1H
SEG01	35	DP1	1C	1B	1A
SEG02	34	2M	2N	2G	2H
SEG03	33	DP2	2C	2B	2A
SEG04	32	3M	3N	3G	3H
SEG05	31	DP3	3C	3B	3A
SEG06	30	4M	4N	4G	4H
SEG07	29	DP4	4C	4B	4A
SEG08	28	5M	5N	5G	5H
SEG09	27	DP5	5C	5B	5A
SEG10	26	6M	6N	6G	6H
SEG11	25	DP6	6C	6B	6A
SEG12	24	7M	7N	7G	7H
SEG13	23	DP7	7C	7B	7A
SEG14	22	8M	8N	8G	8H
SEG15	21	DP8	8C	8B	8A
SEG16	16	8L	8K	8J	8I
SEG17	15	8D	8E	8F	CA8
SEG18	14	7L	7K	7J	7I
SEG19	13	7D	7E	7F	CA7
SEG20	12	6L	6K	6J	6I
SEG21	11	6D	6E	6F	CA6
SEG22	10	5L	5K	5J	5I
SEG23	9	5D	5E	5F	CA5
SEG24	8	4L	4K	4J	4I
SEG25	7	4D	4E	4F	CA4
SEG26	6	3L	3K	3J	3I
SEG27	5	3D	3E	3F	CA3
SEG28	4	2L	2K	2J	2I
SEG29	3	2D	2E	2F	CA2
SEG30	2	1L	1K	1J	1I
SEG31	1	1D	1E	1F	CA1



a)



b)

Fig. 25 – Estrutura do LCD. a) vista geral e b) detalhe de um carácter

O controlador de LCD integrado no MSP430FG4618 tem capacidade para controlar até 160 segmentos multiplexados quatro a quatro, possui memória dedicada para o *display*, inclui bomba de carga regulada e permite controlar o contraste do LCD via *software*. Para alimentar o LCD através da bomba de carga é necessário definir $VLCD_{EXT} = 0$, $VLCD_{PEN} = 1$ e em $VLCD_x$ seleccionar a tensão a aplicar ao LCD. Através dos bits $VLCD_x$ é possível seleccionar uma tensão de 2,6V a 3,44V para alimentar o LCD.

Neste Kit foram ligadas 32 das 40 saídas disponíveis para segmentos, pelo que se usam os endereços de memória 091h a 0A0h em que cada bit corresponde a um segmento, como se observa na Tabela 9. Uma vez que os segmentos de cada dígito não se encontram ordenados ou agrupados na memória, torna-se necessário criar uma matriz de possíveis caracteres e funções de escrita e leitura no LCD para facilitar a utilização do LCD.

Tabela 9 – organização da memória do controlador LCD

Associated Common Pins	3	2	1	0	3	2	1	0	Address	n	Associated Segment Pins
	7							0			
0A4h	--	--	--	--	--	--	--	--	38	39, 38	
0A3h	--	--	--	--	--	--	--	--	36	37, 36	
0A2h	--	--	--	--	--	--	--	--	34	35, 34	
0A1h	--	--	--	--	--	--	--	--	32	33, 32	
0A0h	--	--	--	--	--	--	--	--	30	31, 30	
09Fh	--	--	--	--	--	--	--	--	28	29, 28	
09Eh	--	--	--	--	--	--	--	--	26	27, 26	
09Dh	--	--	--	--	--	--	--	--	24	25, 24	
09Ch	--	--	--	--	--	--	--	--	22	23, 22	
09Bh	--	--	--	--	--	--	--	--	20	21, 20	
09Ah	--	--	--	--	--	--	--	--	18	19, 18	
099h	--	--	--	--	--	--	--	--	16	17, 16	
098h	--	--	--	--	--	--	--	--	14	15, 14	
097h	--	--	--	--	--	--	--	--	12	13, 12	
096h	--	--	--	--	--	--	--	--	10	11, 10	
095h	--	--	--	--	--	--	--	--	8	9, 8	
094h	--	--	--	--	--	--	--	--	6	7, 6	
093h	--	--	--	--	--	--	--	--	4	5, 4	
092h	--	--	--	--	--	--	--	--	2	3, 2	
091h	--	--	--	--	--	--	--	--	0	1, 0	

Sn+1
Sn

2.3.4.2. Encoder rotativo

O *encoder* rotativo tem 24 posições estacionárias em cada volta. Nas posições estacionárias as saídas A e B encontram-se a nível baixo. Quando se passa de uma posição estacionária, linhas verticais a vermelho Fig. 26, para outra observa-se uma transição dos sinais A e B conforme a Fig. 26. Para determinar o sentido de rotação liga-se o sinal A a uma interrupção do microcontrolador e, aquando da transição deste de nível baixo para nível alto, lê-se o valor do sinal B. Caso o sinal B esteja a nível baixo está a decorrer uma rotação à direita, caso esteja a nível alto está a decorrer uma rotação à esquerda.

No diagrama podemos observar também que as transições de nível dos sinais apresentam algum ruído. Este ruído deve-se ao uso de interruptores mecânicos para efectuar as transições. Por esta razão é necessário realizar uma programação cuidada para a aquisição destes sinais de maneira a filtrar o ruído pois este pode provocar falsos impulsos de rotação. Esta filtragem pode ser realizada impondo um tempo de espera entre uma leitura e a seguinte, ou realizando sucessivas leituras até que, por exemplo, as últimas três apresentem todas o mesmo valor lógico. Se o sistema funcionar com interrupções é recomendável desligar a interrupção do sinal A após ser detectada uma transição e só tornar a ligar quando houver estabilidade no sinal, evitando assim sobrecarga no sistema de interrupções do microcontrolador. Este *encoder* inclui também um interruptor de pressão de funcionamento normal.

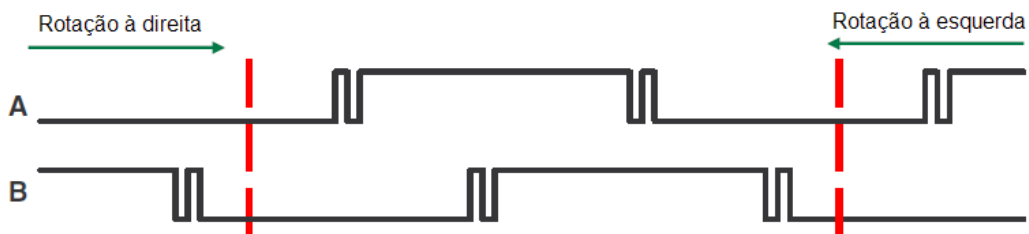


Fig. 26 – Sequência de sinal típica gerada pelo encoder



Fig. 27 – Encoder rotativo

2.3.4.3. Joystick de navegação

O *joystick* de navegação tem quatro posições, norte, sul, este, oeste e um interruptor de pressão (Fig. 28). É necessário ter em atenção que, como em qualquer interruptor, existem ruídos nos instantes de transição e para além destes, como estamos na presença de um *joystick*, por vezes o botão central de pressão é activado acidentalmente quando se movimenta o *joystick* para uma das outras posições, estes factores devem ser tidos em conta na programação.



Fig. 28 – *joystick* de navegação

2.3.4.4. Altifalante

O altifalante escolhido para integrar neste kit foi um KDMG13008L fabricado pela Kingsgate. Este altifalante tem uma potência de 0,2W e uma impedância de 8 Ω . A ligação do altifalante é feita por um contacto de mola, existente dentro do conector de *jack* de 3,5mm. Este contacto é aberto quando um *jack* é introduzido no conector, desligando assim o altifalante (Fig. 29). Existe também um *jumper* que permite isolar a saída do microcontrolador do resto do sistema.

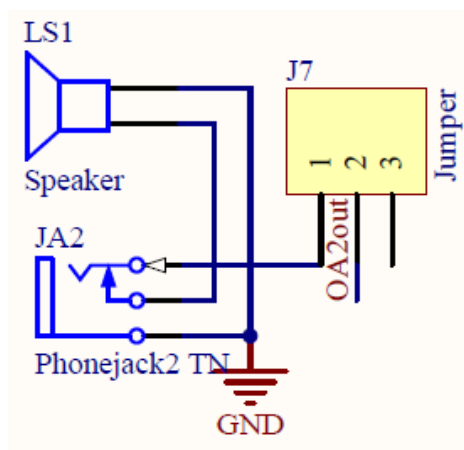


Fig. 29 – Ligação do altifalante

2.3.4.5. Microfone

O microfone escolhido para integrar este kit foi um KECG2742PBL fabricado pela Kingsgate. Este microfone é do tipo *electret* e tem uma tensão de funcionamento típica de 2V. A ligação do microfone é feita por um contacto de mola, existente dentro do conector de *jack* de 3,5mm (Fig. 30Fig.). Este contacto é aberto quando um *jack* é introduzido no conector. Existe também um *jumper* que permite isolar a entrada de sinal do microcontrolador do resto do sistema.

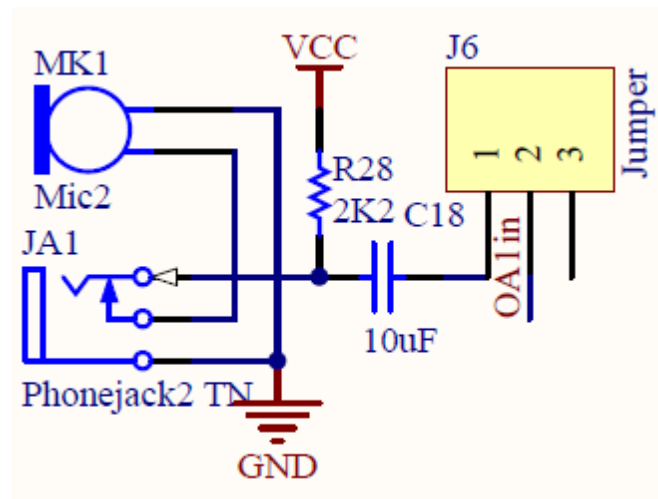


Fig. 30 – Ligação do microfone

2.4. Kit Nível 3

2.4.1. Descrição Geral

O Kit de Nível 3 é direccionado para o aluno experiente e com vasto conhecimento das potencialidades e forma de programação dos microcontroladores da TI. Tem como base um Microcontrolador MSP430F5419 da TI, *display* de tecnologia OLED de 262 mil cores com formato 160x128 pixels e tamanho total de 1,45", acelerómetro de três eixos, leitor de cartões SD, *touchscreen*, porta USB com controlador série, duas portas PS/2, ligações para módulos Chipcon-RF e RF-EZ430.

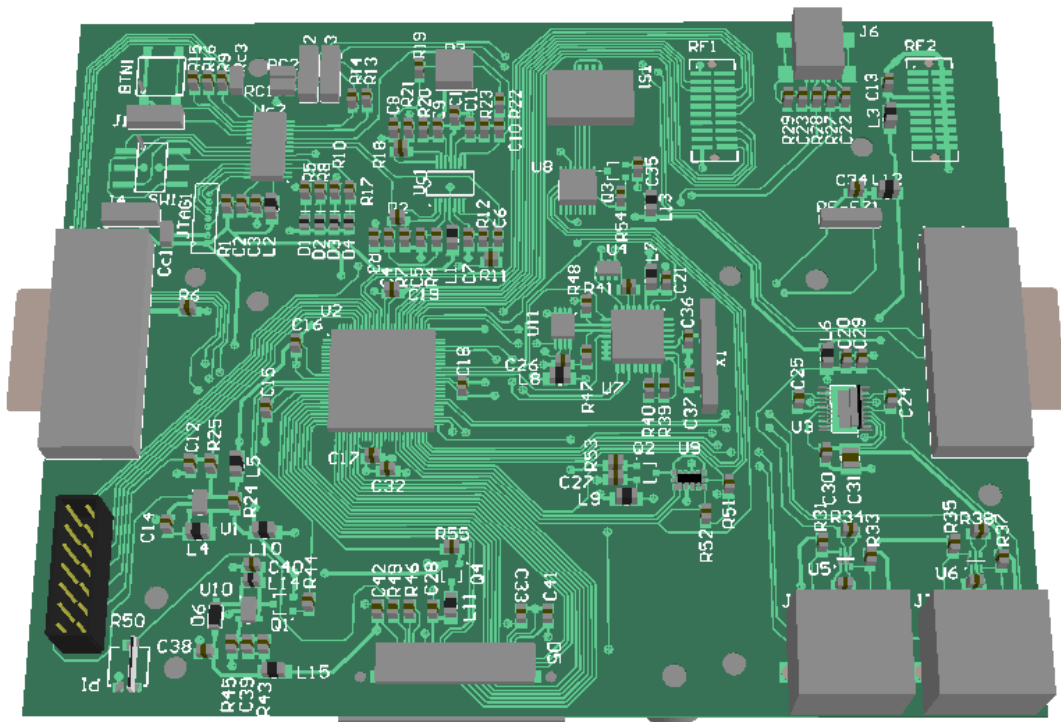


Fig. 31 - Desenho 3D do kit nível 3

Com este Kit é dada ao aluno a oportunidade de explorar a área do conhecimento relacionado com as comunicações sem fios, comunicar com um computador através da porta USB, realizar uma interface dinâmica utilizando o *display*, o *touchscreen* e o acelerómetro. Podem ser também ligados periféricos que utilizamos diariamente como o teclado ou o rato utilizando as portas PS/2.

O MSP430F5419 foi concebido para a área das comunicações, inclui por isso quatro módulos de comunicação que suportam até 8 barramentos de diferentes protocolos de comunicação, podendo ser usados: comunicação série, infra-vermelhos, SPI e I²C. Inclui também três contadores/temporizadores com um total de 15 registos de comparação, um controlador DMA de três canais e suporte para multiplicações de 32bit através de hardware dedicado.

O *Direct-Memory Access* (DMA), utilizado actualmente em todos os computadores, permite que outros dispositivos, que não o processador, acessem directamente à memória sem interferência do processador, deixando-o livre para desempenhar outras tarefas. Esta característica permite reduzir em larga escala o *overhead* do processador quando são realizadas transferências de dados de periféricos para a memória. No microcontrolador utilizado, o controlador DMA não permite acessos directos à memória por periféricos ligados externamente ao microcontrolador. Este controlador é totalmente configurável via *software* e permite, por exemplo, que os valores convertidos pelo ADC sejam transferidos directamente para a memória ou, por exemplo, realizar uma transferência de vários bytes entre diferentes posições de memória. Para qualquer uma destas aplicações obtêm-se grandes melhorias na performance geral do dispositivo em relação ao uso do processador para realizar a mesma operação sem que os canais de DMA sejam utilizados.

2.4.2. Descrição do módulo USCI

Cada módulo de comunicação série do microcontrolador (USCI) está dividido em dois sub-módulos (USCI_Ax e USCI_Bx) e permite criar dois barramentos independentes de comunicação. Estes módulos de comunicação têm os protocolos de cada barramento que podem ser implementados internamente em hardware e a possibilidade de os configurar via *software*. Isto permite libertar o processador das tarefas de gestão das comunicações série, sempre dispendiosas em tempo de processamento. Os módulos USCI têm dois vectores de interrupção configuráveis que se activam quando algum dos módulos precisa de intervenção do processador.

2.4.2.1. Descrição do módulo USCI_A

O módulo USCI_A suporta as seguintes funcionalidades: configuração para modo UART, gerador de impulsos para comunicação por infravermelhos (IrDA), detecção automática de *baud rate* para comunicações LIN e configuração para modo SPI.

Modos assíncronos

Nos modos assíncronos, o módulo USCI_A permite ligar o microcontrolador a dispositivos externo de interface série utilizando as portas UCAXRXD e UCAXTXD. Estes modos são seleccionados quando UCSYNC=0.

A inicialização do USCI_Ax faz-se começando por activar o bit WCSWRST para passar o módulo USCI ao modo de *reset* e limpar todos os registos associados ao USCI_Ax. Estes modos configuram-se utilizando os registos UCAxCTL0 e UCAxCTL1. No registo UCAxCTL0 definem-se os modos assíncronos limpando o bit UCSYNC e define-se o modo de comunicação utilizando os bits UCMODEx. Os modos de comunicação assíncrona possíveis são: modo UART, modo *idle-line*, modo *address-bit* e modo UART com detecção automática de *baud rate*. O formato do carácter UART é configurável utilizando o registo UCAxCTL0 e é composto por um *start bit*, 7 ou 8 bits de dados, um bit de endereço, um bit de paridade e um ou dois *stop bits*. O bit de endereço é enviado apenas no modo multiprocessador *address-bit* e o bit de paridade não é enviado no caso de na configuração ser definido que não existe bit de paridade. Os bits de configuração correspondentes estão representados na Fig. .

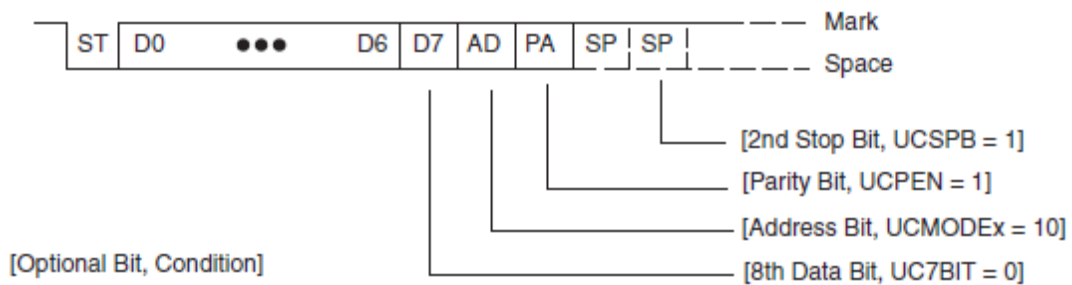


Fig. 32 – Estrutura dos dados enviados pelo porto série

Modo UART

O modo UART é utilizado para a comunicação entre dois dispositivos onde a origem e destino dos dados é constante. Neste modo o bit de endereço nunca é transmitido.

Modo *idle-line*

O modo multiprocessador *idle-line* permite interligar mais de dois dispositivos ao barramento (Fig.). Neste modo, os blocos de dados são separados por períodos de 10 ciclos de relógio em que a linha RX se mantém a nível alto. Após este tempo de espera, o primeiro carácter a ser transmitido é o endereço do dispositivo com que se quer comunicar, seguido dos caracteres de dados. Sempre que se pretende iniciar uma comunicação com qualquer dispositivo é necessário enviar o seu endereço no primeiro carácter do bloco.

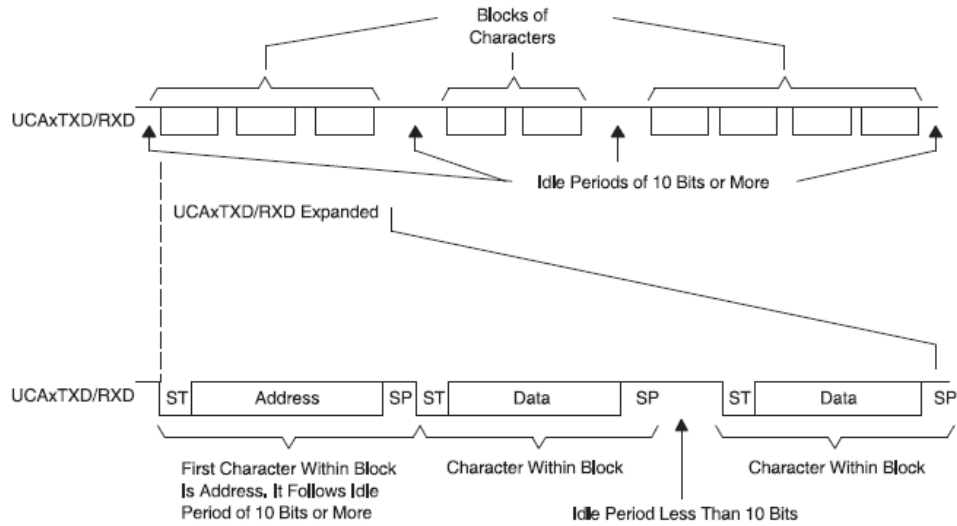


Fig. 33 – Comunicação em modo *idle-line*

Modo *address bit*

O modo multiprocessador *address-bit* permite interligar mais do que dois dispositivos ao barramento (Fig.). O dispositivo com o qual se quer comunicar é seleccionado enviando um carácter de endereço. Neste modo não são necessários tempos de espera entre duas transmissões de blocos pois o envio de endereço é sinalizado activando o bit de endereço.

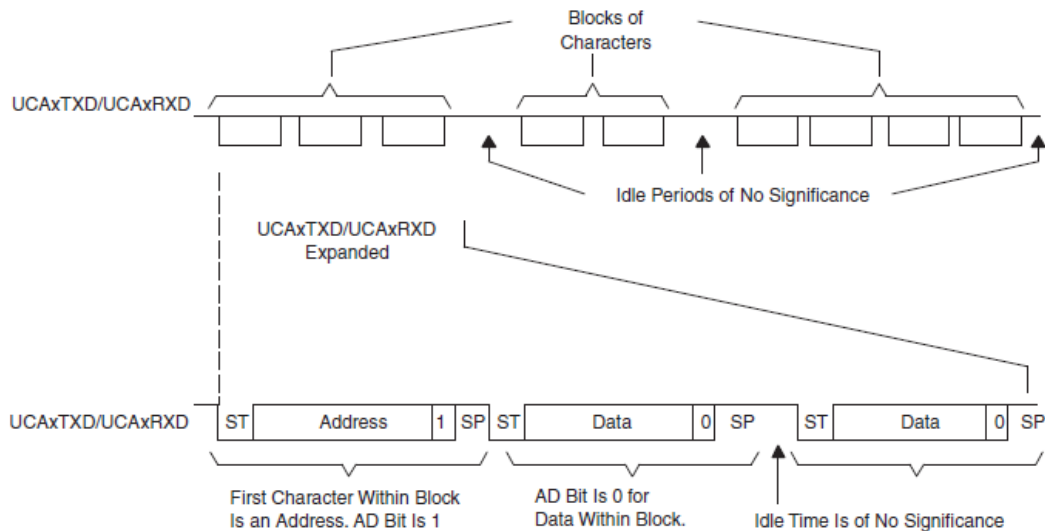


Fig. 34 – Comunicação em modo *address-bit*

Modo UART com detecção automática de baud rate

Neste modo é necessário enviar uma sequência de sincronização antes de se poder iniciar a transmissão de dados (Fig. 35). A sequência de sincronização consiste numa paragem (nível 0 na linha) de 11 ou mais zeros e da transmissão de 8 bit de sincronização.

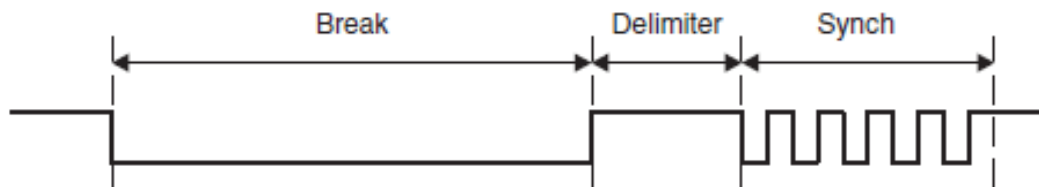


Fig. 35 – Comunicação em modo UART com detecção automática do baud rate

Modo de comunicação por infravermelhos (IrDA)

Este modo de comunicação é uma variante do modo UART e todas as configurações definidas no modo UART são aplicadas (Fig. 36). Seleccionando o bit UCIREN do registo UCAXIRTCTL activam-se o codificador e o decodificador IrDA e os dados enviados/recebidos nas linhas UART são convertidos para este formato. A codificação IrDA é simples e consiste apenas no envio de um impulso, de cada vez que é enviado o nível zero na UART.

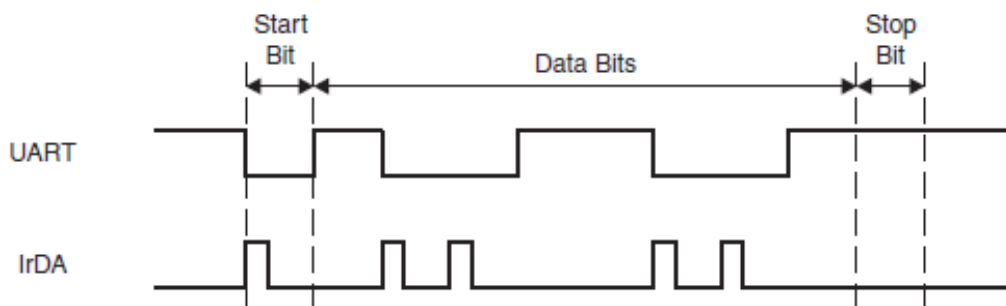


Fig. 36 – Ilustração de uma comunicação em modo IrDa

Modo síncrono

No modo síncrono o módulo USCI_A permite ligar o microcontrolador a dispositivos externos com interface SPI utilizando as portas UCxSIMO, UCxSOMI, UCxCLK e UCxSTE. O modo síncrono é seleccionado activando o bit UCSYNC e o modo SPI (3 ou 4 fios) é definido utilizando os bits UCMODEx. A interface SPI não tem um protocolo bem estabelecido e dependendo do fabricante do dispositivo externo pode ser necessário alterar as configurações do barramento. Antes de mudar qualquer configuração é necessário colocar o módulo em modo *reset* activando o bit UCSWRST. O protocolo SPI suporta transferência de caracteres de 7 ou 8 bit seleccionada utilizando o bit UC7BIT. No modo de 7 bit, o bit mais significativo é sempre zero. Outras configurações importantes são a polaridade e a fase do sinal de relógio definidas utilizando os bits UCCKPL e UCCKPH.

Em qualquer um dos modos descritos anteriormente, para transmitir dados através da interface é necessário armazenar a informação no registo UCxTXBUF, para o módulo a enviar segundo as especificações do protocolo e configurações do utilizador. Quando é recebida a informação, esta fica armazenada no registo UCxRXBUF, sendo posteriormente recolhida pelo processador.

2.4.2.2. Descrição do módulo USCI_B

O módulo USCI_B suporta as seguintes funcionalidades: configuração para modo SPI e configuração para modo I²C. O funcionamento do modo SPI é semelhante ao do módulo USCI_A descrito anteriormente.

O modo I²C segue as especificações da versão 2.1 da Philips para o protocolo I²C suportando endereçamento de 7 e 10 bit, *General Call*, modos *master*, *multi-master* e *slave*, suporte para a velocidade padrão de 100kbps e modo rápido de 400kbps. No modo *slave*, depois de configurar o módulo apenas é necessária a intervenção do processador quando são recebidos dados ou quando é iniciada uma comunicação para envio de dados. Quando o módulo necessita da intervenção do processador, gera uma interrupção num dos vectores de interrupção existentes para o efeito, isto permite ao utilizador desenhar um sistema em que o processador se mantém num nível de baixo consumo de energia e apenas acorda para realizar pequenas operações. No modo *master*, as comunicações são iniciadas pelo utilizador seleccionando o endereço de destino no registo UCBxI2CSA e activando o bit UCTXSTT de modo a criar uma condição de *start*. Durante a transferência de dados, o funcionamento é semelhante ao do *slave*, tendo como base as interrupções geradas pelo módulo. Para finalizar a transferência, o utilizador activa o bit UCTXSTP gerando uma condição de *stop* do protocolo I²C. A informação a enviar é introduzida no registo UCBxTXBUF e a informação recebida pode ser recolhida no registo UCBxRXBUF.

Na Tabela 10 é apresentada uma listagem dos recursos utilizados pelo processador.

Tabela 10 – Recursos do processador utilizados

Pinos	Função	Nome	Função no Kit
1	P6.4		Não Utilizados
2	P6.5		
3	P6.6		
4	P6.7		
5	P7.4		
6	P7.5		
7	P7.6		
8	P7.7		
9	P5.0		
10	P5.1		
11	AVcc	AVcc	Alimentação
12	AVss	AVss	
13	P7.0	VI3en	Alimentação do Display
14	P7.1	VoledEN	
15	Vss	Vss	Alimentação
16	Vcc	Vcc	
17	P1.0	CPU	Controlo do Display
18	P1.1	PS	
19	P1.2	RS	
20	P1.3	CSB	
21	P1.4	RDB	
22	P1.5	WRB	
23	P1.6	RESETB	
24	P1.7	D9	
25	P2.0	D10	
26	P2.1	D11	
27	P2.2	D12	Barramento Paralelo do Display
28	P2.3	D13	
29	P2.4	D14	
30	P2.5	D15	
31	P2.6	D16	
32	P2.7	D17	Barramento SPI do Display
33	P3.0	STE	
34	P3.1	SIMO	
35	P3.2	SOMI	
36	P3.3	CLK	
37	Vss	Vss	Alimentação
38	Vcc	Vcc	
39	P3.4	TX	Ligação Série RF-EZ
40	P3.5	RX	
41	P3.6	CLK	SPI do cartão SD
42	P3.7	SDA	I ² C acel. e Saídas auxiliares do acelerómetro
43	P4.0	INT2acc	
44	P4.1	INT1acc	
45	P4.2	VaccEN	Alimentação acel. e touchscr.
46	P4.3	VtscEN	
47	P4.4	PenIRQ	S. aux. touchscr.
48	P4.5	KB_CLK	Sinais de relógio portas PS/2
49	P4.6	M_CLK	
50	P4.7	V5en	5V portas PS/2

Pinos	Função	Nome	Função no Kit
51	P5.4	SCL	I ² C acel. e touch.
52	P5.5	STE	SPI do cartão SD
53	P5.6	SIMO	
54	P5.7	SOMI	
55	P7.2	CDsd	Ligações aux. cartão SD
56	P7.3	WPsd	
57	P8.0	KB_DATA	Sinais de dados portas PS/2
58	P8.1	M_DATA	
59	P8.2		Não Utilizados
60	P8.3		
61	P8.4		
62	Vcore	Vcore	Alimentação
63	Vss	Vss	
64	Vcc	Vcc	
65	P8.5	WAKEusb	Sinais auxiliares do controlador USB
66	P8.6	RESusb	
67	P8.7	SUSPusb	
68	P9.0	STE	Barramento SPI para módulo de comunicações
69	P9.1	SIMO	
70	P9.2	SOMI	
71	P9.3	CLK	
72	P9.4	TX	Ligação Série para USB
73	P9.5	RX	
74	P9.6		Não Utilizados
75	P9.7		
76	P10.0	CLK	SPI para Chipcon
77	P10.1	SDA	I ² C controlador USB
78	P10.2	SCL	
79	P10.3	STE	SPI para Chipcon RF
80	P10.4	SIMO	
81	P10.5	SOMI	
82	P10.6	RESETCC	Sinais auxiliares Chipcon RF
83	P10.7	VREG_EN	
84	P11.0	FIFOP	
85	P11.1	FIFO	
86	P11.2		Não Utilizado
87	Vcc	Vcc	Alimentação
88	Vss	Vss	
89	P5.2	RFGDO0	Sinais auxiliares Chipcon RF
90	P5.3	RFGDO2	
91	TEST	TEST	Reservado
92	TDO	TDO	JTAG
93	TDI	TDI	
94	TMS	TMS	
95	TCK	TCK	
96	RST	RST	Reset
97	P6.0		Não Utilizados
98	P6.1		
99	P6.2		
100	P6.3		

2.4.3. Dispositivos no kit

2.4.3.1. Acelerómetro

O acelerómetro integrado no kit de nível 3 foi um MMA7455L fornecido pela Freescale Semiconductor com sensibilidade ajustável e leitura em 3-eixos diferentes. Este acelerómetro tem como princípio de funcionamento a variação da capacidade de dois condensadores de duas placas fixas e uma placa central que se movimenta consoante a aceleração tal como se ilustra na Fig. 37. A ligação de saída do acelerómetro é digital e permite utilizar tanto uma interface SPI como I²C. A interface escolhida para ligação ao kit foi a I²C e para comunicar com este dispositivo é necessário utilizar o endereço 1Dh comunicando através do módulo USCIB1. A sensibilidade do acelerómetro é ajustável entre três valores predefinidos: 16LSB/g, 32LSB/g e 64LSB/g. A estes valores de sensibilidade estão associados os valores máximos de aceleração de 8, 4 e 2g. O acelerómetro tem também duas saídas configuráveis por software que podem estar activas quando é detectada uma situação de queda livre, quando é detectado movimento ou quando existem dados a enviar pelo acelerómetro. Para reduzir o consumo energético, existe um porto no microcontrolador que controla a alimentação deste dispositivo (VaccEN).

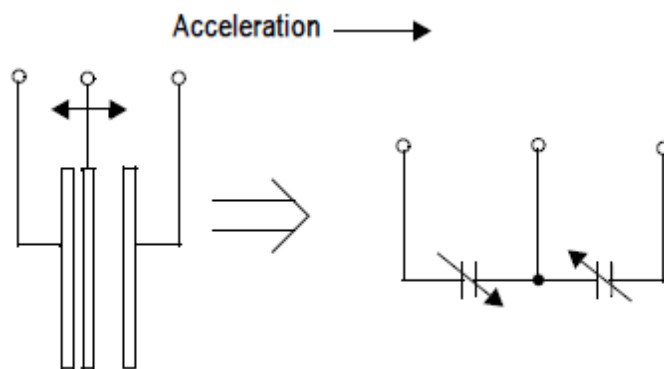


Fig. 37 – Esquema equivalente do acelerómetro

2.4.3.2. Touchscreen

No kit de nível 3 foi incluído um *touchscreen* resistivo de 4 fios, modelo N010-0554-T703 fabricado pela Fujitsu de dimensões 6x8cm. Os *touchscreen* deste tipo são constituídos por duas camadas finas de acrílico, com um revestimento condutor, separadas por espaçadores invisíveis. A estrutura é ilustrada na Fig. 38.

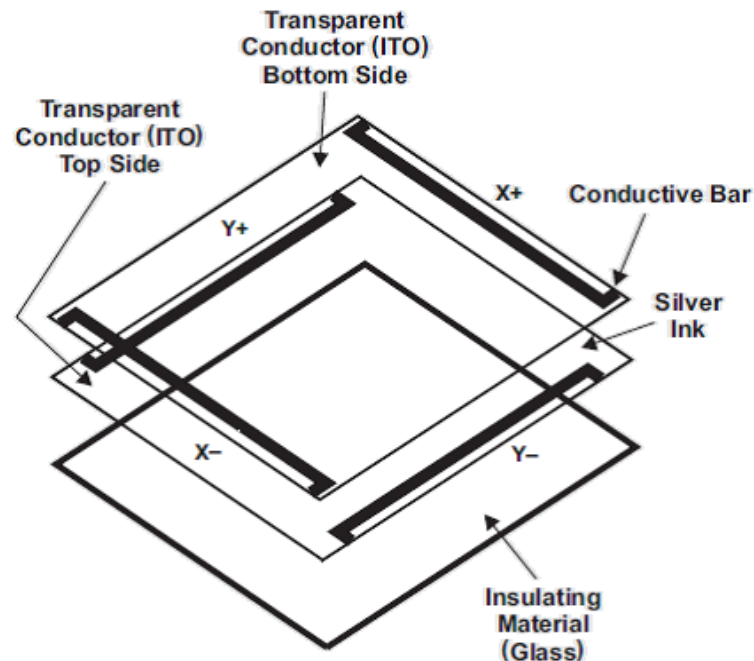


Fig. 38 – Estrutura construtiva do *Touchscreen*.

Para calcular a posição de um objecto que pressione o *touchscreen* é necessário aplicar uma tensão entre dois pontos, Y+ e Y-, e ler o valor de tensão num dos pólos de X. Entre os pólos de Y é gerado um divisor de tensão ao longo da camada correspondente do *touchscreen* e, no caso de existir um objecto a pressionar o *touchscreen*, a tensão presente na camada Y passa para a camada X. Como as resistências são de valor baixo, a queda de tensão entre o ponto de contacto e uma das extremidades X pode ser desprezada. Com esta leitura é determinada a posição do objecto em Y. Para determinar a posição do objecto na dimensão X repete-se o processo aplicando uma tensão entre X+ e X- e fazendo a leitura num dos pólos de Y.

Este processo de captura da posição necessita de uma constante gestão dos níveis de tensão em cada camada assim como das leituras sucessivas dos valores de tensão presentes nas linhas. Para libertar o processador de toda esta gestão, e diminuir o número de componentes utilizados, foi decidido utilizar um controlador de *touchscreen* modelo TSC2007 fabricado pela Texas Instruments (Fig. 39).

Este controlador encarrega-se de todo o processo de captura e fornece um valor lógico equivalente à posição do objecto que pressiona o *touchscreen* através de uma interface I²C. Neste kit, o controlador de *touchscreen* foi ligado ao módulo de comunicação USCIB1, partilhando o barramento com o acelerómetro, e a comunicação é feita utilizando o endereço 90h. Este controlador tem ainda uma saída activa quando o *touchscreen* é pressionado. Para reduzir o consumo energético, existe um porto no microcontrolador que controla a alimentação deste dispositivo (VtscEN).

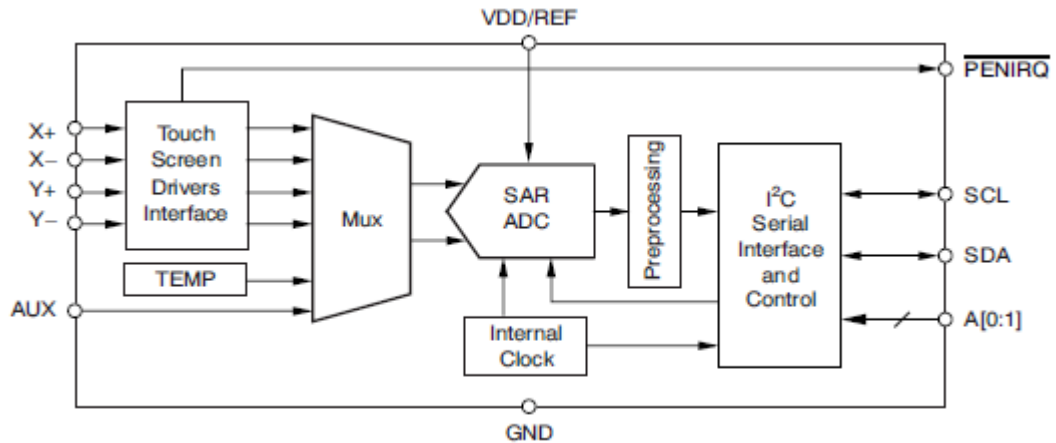


Fig. 39 – Controlador do touchscreen modelo TSC2007

2.4.3.3. Portas PS/2

Foram instaladas no kit duas portas PS/2 que permitem ao utilizador ligar teclados, ratos ou outros dispositivos PS/2. As ligações de dados e relógio das portas PS/2 estão ligadas ao microcontrolador através de *level shifters* bidireccionais que realizam a necessária conversão dos 3,3V utilizados pelo microcontrolador para os 5V utilizados pelos periféricos e vice-versa como se esquematiza na Fig. 40. Os sinais de relógio ligam no microcontrolador a duas saídas de comparação do Timer_B existindo assim a possibilidade de criar um sinal de relógio sem intervenção do processador. Existe também uma ligação ao microcontrolador para ligar/desligar a alimentação de 5V das portas PS/2 (V5en). Os *level shifters* bidireccionais utilizados foram construídos utilizando dois MOSFET do tipo N para cada porta.

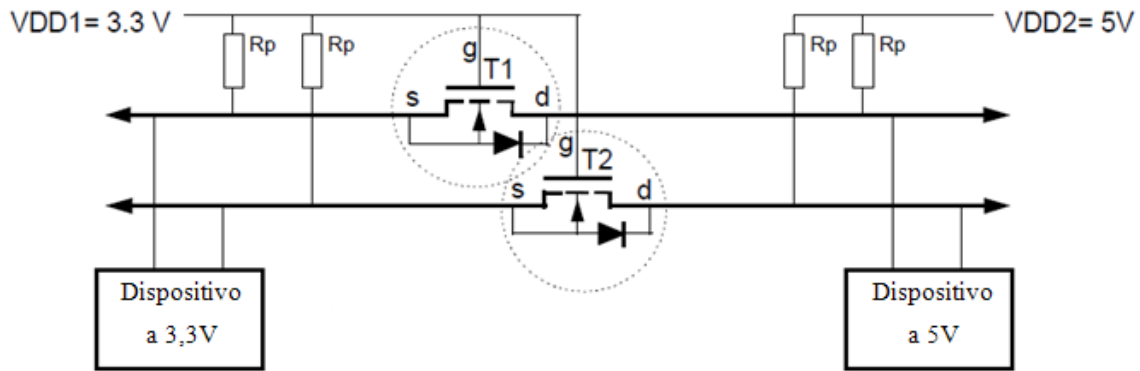


Fig. 40 – Conversão dos valores de tensão

2.4.3.4. Controlador USB

O controlador USB integrado no kit é o modelo TUSB4310 fornecido pela Texas Instruments. Este controlador tem uma ligação I²C para configuração e uma ligação UART para dados. A ligação I²C foi ligada ao microcontrolador no módulo de comunicações USCIB3 e partilha o mesmo barramento com uma EEPROM de 128KB utilizada para armazenar código de arranque do controlador USB. Após configuração do controlador, feita utilizando o barramento I²C ou da porta USB a partir do computador, a comunicação faz-se através da ligação UART do módulo de comunicações USCIA2. Existem ainda mais três ligações ao microcontrolador utilizadas para suspender, acordar e reiniciar o controlador USB. A interligação dos diferentes dispositivos pode ser observada na Fig. 41.

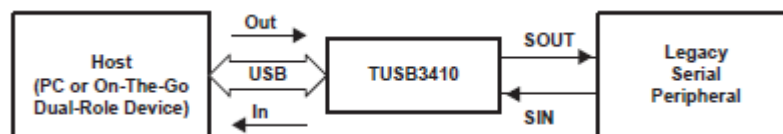


Fig. 41 – Interligação de diferentes dispositivos através de um porto USB

O controlador USB integra um microcontrolador 8052 com 256B de RAM, 10kB de ROM e 16kB de memória de programa. Inclui também um controlador I²C e funciona com uma frequência de relógio de 24MHz. No arranque, o controlador USB, carrega o código de programa através do barramento I²C a partir da EEPROM ou através da porta USB a partir do computador.

2.4.3.5. EEPROM

A EEPROM incluída neste kit é o modelo 24LC128 fornecido pela Microchip. Este dispositivo tem 16kB de memória e permite tanto leituras aleatórias como sequenciais em toda a extensão da memória. A comunicação com este dispositivo é feita utilizando uma interface I²C, que neste kit foi ligada ao módulo de comunicação USCIB3. A memória pode ser acessada pelo microcontrolador e pelo controlador USB através do barramento I²C.

2.4.3.6. Leitor de cartões SD

O leitor de cartões SD deste kit, como dispositivo físico, é na verdade apenas um encaixe para um cartão SD, com uma ligação directa para o módulo de comunicação USCIA1. A comunicação com um cartão de memória do tipo SD pode ser feita utilizando uma interface própria (SD) ou utilizando uma interface SPI, tal como se representa nas Fig. 42, Fig. 43 e Fig. 44. Neste kit projectaram-se as ligações de modo a ligar o cartão utilizando o protocolo SPI que, apesar de apenas especificar a interface física, permite ao utilizador concentrar a sua atenção no protocolo de acesso e transferência de dados. Este protocolo pode ser dividido em mensagens de comando e de resposta e blocos de dados. Em todas as transferências de dados é introduzido no final do bloco de dados um CRC de 16 bit calculado utilizando o polinómio definido pelo CCITT.

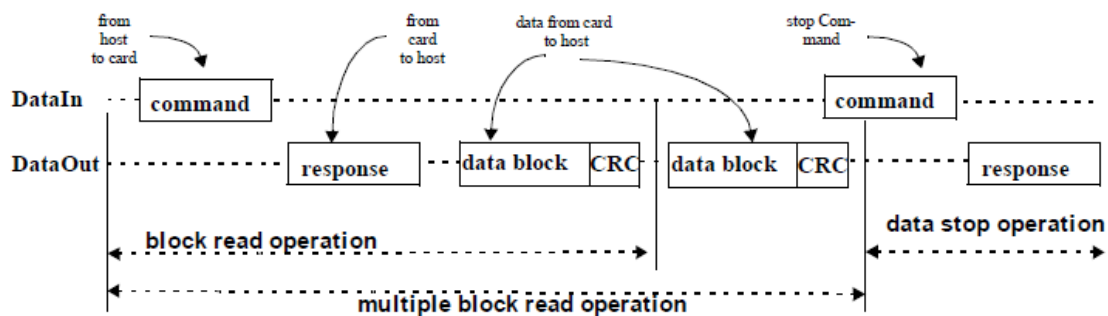


Fig. 42 – Operação de leitura

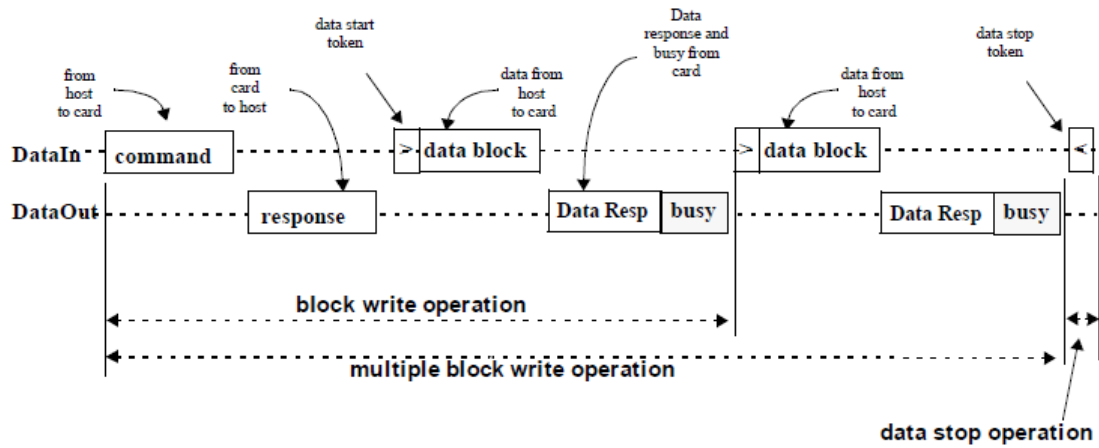


Fig. 43 – Operação de Escrita

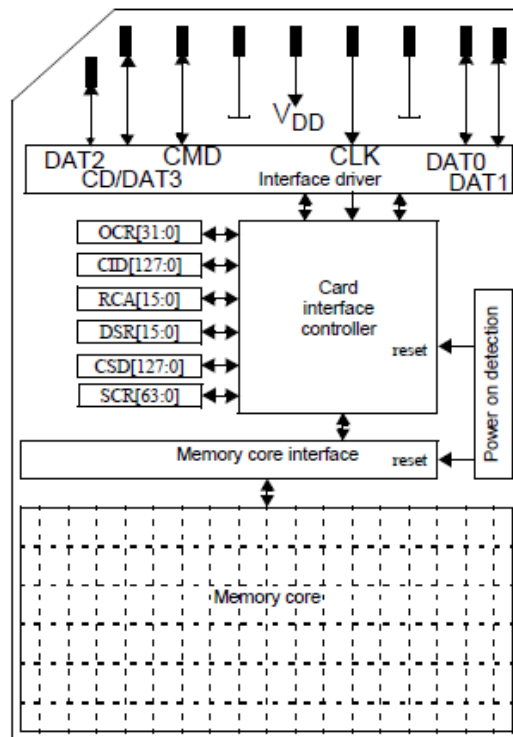


Fig. 44 – Arquitectura interna de um cartão SD

Existem ainda mais duas ligações do leitor de cartões para o microcontrolador, uma activa quando é detectado um cartão e outra quando a protecção de escrita está ligada.

2.4.3.7. Display OLED

O *display* escolhido para integrar este kit foi o modelo DD-160128FC fabricado pela Densitron com características principais: resolução de 160x128 pixel, 262k cores e dimensão total de 1,45". Este *display* possui integrado um controlador SEPS525, produzido pela Syncoam, e o produto final permite utilizar uma interface paralela de 8 ou 9 bit ou uma interface SPI. A ligação ao microcontrolador foi projectada de modo a permitir as duas interfaces de ligação, e a interface SPI foi ligada ao módulo de comunicação USCIB0.



Fig. 45 – *Display* OLED modelo DD-160128FC

A tecnologia OLED permite fabricar *displays* de baixo custo com resolução, contraste e brilho muito superior aos que utilizam a tecnologia LCD. As principais vantagens da tecnologia OLED são: a possibilidade de impressão de uma matriz de OLED utilizando uma impressora do tipo jacto de tinta, o que diminui em larga escala os custos de fabrico dos *displays*, a possibilidade de impressão em materiais flexíveis, a superior quantidade de cores, brilho e contraste que se consegue atingir e muito menores consumos energéticos. Comparando com um LCD, que para apresentar uma imagem precisa duma iluminação de fundo que posteriormente é filtrada e polarizada e com um consumo constante elevado, um *display* com tecnologia OLED apenas consome o necessário para mostrar cada imagem. Para apresentar, por exemplo, um quadrado preto num LCD, os pixéis correspondentes filtram toda a luz fornecida pela iluminação do *display*, num *display* de OLED apenas é necessário desligar os OLED correspondentes.

O controlador do *display* disponibiliza duas interfaces de comunicação paralela direccionadas para a utilização com microprocessadores Motorola da família 6800 e Intel da família 8080. Estas interfaces não se encontram disponíveis no microcontrolador utilizado, como tal, para ser utilizada qualquer uma das interfaces de comunicação em paralelo é necessário criar em software o protocolo de comunicação. Para facilitar a utilização do *display* recomenda-se a utilização da interface série SPI.

2.4.3.8. Módulo RF da Chipcon

Neste kit, integrou-se como porta de expansão um encaixe para um módulo RF fabricado pela Chipcon. Este dispositivo comunica com o microcontrolador através duma interface SPI e foi ligado ao módulo de comunicação USCIA3.

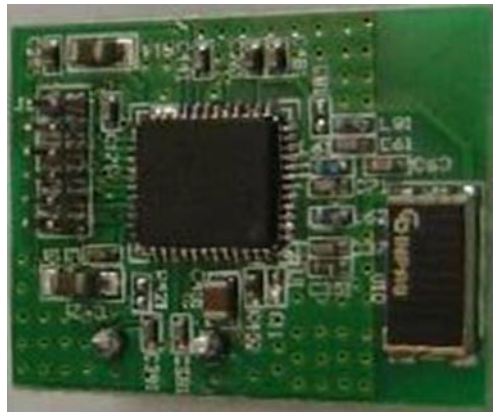


Fig. 46 – Módulo RF da Chipcon

2.4.3.9. Módulo RF-EZ430

Neste kit, integrou-se como porta de expansão um encaixe para um módulo RF-EZ430. Este dispositivo comunica com o microcontrolador através duma interface série e foi ligado do módulo de comunicações USCIA0.

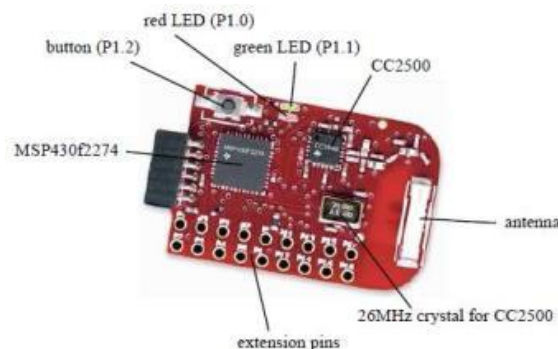


Fig. 47 - Módulo RF-EZ430

3. Software

A comunicação entre Kits oferece grande versatilidade ao sistema no seu todo, permitindo que, quando necessário, sejam usadas funcionalidades de dois ou mais Kits em conjunto, aumentando em larga escala o número de aplicações possíveis para o sistema. Em cada Kit é incluído um módulo que faz a gestão do barramento de comunicações, libertando o microcontrolador principal do Kit para o processamento necessário à função pretendida pelo utilizador/programador. O sistema é modular e permite a ligação simultânea de um máximo de dez Kits ao barramento de comunicações. O barramento utilizado é assente numa camada física que usa o protocolo I²C, na camada de dados e transporte é usado um checksum CRC de 8 bit e para a camada de aplicação foram criadas funções que tornam transparente ao utilizador todo o processo de envio e recepção de dados entre Kits. Dentro de cada Kit existem dois microcontroladores ligados por um barramento SPI seguindo regras mais simplificadas que o principal, uma vez que a origem e destino dos dados são constantes.

3.1. I²C

O barramento I²C é uma interface série de dois fios originalmente desenvolvida pela Philips para interligar dispositivos digitais internos de uma televisão. O barramento é composto por uma ligação de dados (SDA), uma ligação de relógio (SCL) e uma ligação ao comum electrónico. A taxa de transferência no modo normal é de 100 kbit/s e no modo mais rápido chega a atingir, em condições óptimas, 3,4 Mbit/s. Este protocolo está amplamente difundido por todo o tipo de dispositivos electrónicos, entre eles, drivers LCD, portas de I/O, memórias RAM e EEPROM, conversores de dados e em muitos microcontroladores.

Algumas vantagens podem ser atribuídas ao protocolo I²C. Destacam-se entre elas:

- Organização funcional em blocos, dando origem a um simples diagrama esquemático final.
- Não há necessidade de desenvolver novas interfaces já que todos os dispositivos integram as interfaces *on-chip*, aumentando a agilidade no desenvolvimento.
- Possibilidade de inclusão ou exclusão de dispositivos no barramento sem afectar os dispositivos ligados ao mesmo.
- Diagnóstico de falhas extremamente simples. O mau funcionamento é imediatamente detectado.
- Facilidade no desenvolvimento de placas de circuito impresso, devido à reduzida quantidade de ligações.

Características Gerais do Barramento I2C:

- Duas vias de comunicação: *serial data* (SDA) e *serial clock* (SCL), ambas bidireccionais, ligadas a nível lógico alto através de uma resistência de *pull-up*. Enquanto o barramento está livre as linhas mantêm-se a nível lógico alto.
- A taxa de transferência máxima é de 100kbps no modo padrão (*standard*), 400kbps no modo rápido (*fast-mode*) e 3,4Mbps no modo de alta velocidade (*HS Mode*).
- Cada dispositivo possui um endereço único no barramento.
- Qualquer dispositivo ligado pode funcionar como transmissor ou receptor, dependendo da sua natureza, por exemplo, um LCD pode não funcionar como transmissor, assim como um teclado pode não funcionar como receptor. Independente disto, a qualquer dispositivo endereçado dá-se o nome de *slave*.
- O número de interfaces ligadas está dependente apenas da capacitância máxima do barramento (400pF).

Definições:

- Transmissor: dispositivo que envia dados através do barramento.
- Receptor: dispositivo que recebe dados através do barramento.
- *Master*: dispositivo que inicia e termina a comunicação e que gera o sinal de *clock*.
- *Multi-master*: vários dispositivos podem controlar o barramento. Quando isto ocorre temos vários dispositivos a funcionar em modo *master*.
- Arbitrariedade: processo para decidir qual o *master* que vai comunicar quando é detectada colisão de dados no barramento em modo *Multi-Master*.
- Sincronização: procedimento para sincronizar o *clock* de um ou mais dispositivos.

O barramento I²C é constituído por duas ligações activas e uma ligação ao comum electrónico. As 2 ligações, SDA (*Serial Data*) e SCL (*Serial Clock*) correspondem às linhas de dados e relógio respectivamente e são bidireccionais, caso exista apenas um *master* no barramento a linha SCL é sempre controlada pelo mesmo. Qualquer dispositivo ligado ao barramento tem um endereço único e pode receber e/ou transmitir informação dependendo das suas funcionalidades. Por outro lado o I²C é um barramento *Multi-Master*, i.e., podem ser-lhe ligados vários dispositivos com capacidade para iniciar uma transferência de dados. Desta forma o protocolo I²C especifica que um dispositivo que inicie uma transmissão através do barramento é considerado *master* e, conseqüentemente, todos os outros dispositivos se comportam como *slaves* ate ao fim da transmissão.

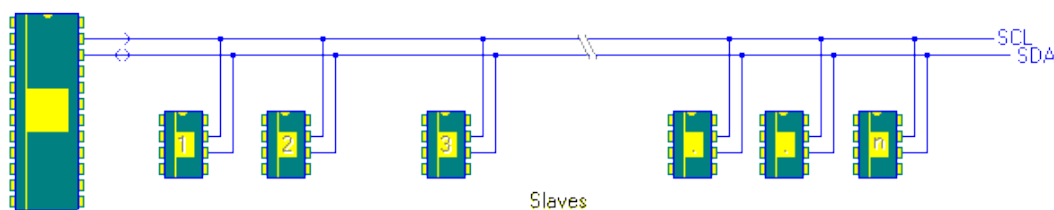


Fig. 48 – Esquema simplificado com um master e n slaves

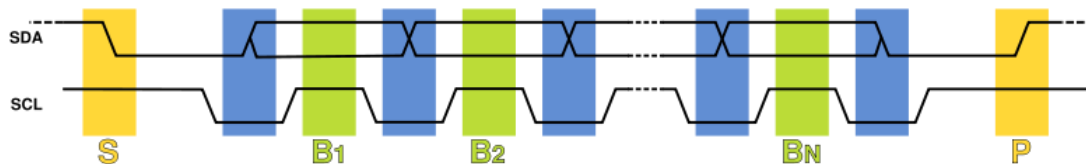


Fig. 49 – Diagrama temporal do protocolo I²C

3.1.1. Comunicação:

Para se iniciar a comunicação utilizando o protocolo I²C é necessário à partida enviar um *startbit*, que funciona como um alerta para que todos os dispositivos ligados recebam o byte que é enviado a seguir, mesmo que estes se encontrem num nível de energia de baixo consumo. A seguir, o *master* envia o endereço do dispositivo com que pretende comunicar ao qual concatena um bit que informa se a operação é de leitura ou de escrita. Quando um dispositivo recebe este endereço compara-o com o seu e, no caso de ser igual, envia um *acknowledge*.

Se o endereço não for igual, os dispositivos ficam à espera dum *stopbit* ou doutro *startbit*.

De todos os endereços que se podem utilizar existem alguns reservados para propósitos especiais:

Tabela 11 – Endereços I²C Reservados

Endereços	Função dedicada
0000-0000	Endereço para <i>General Call</i>
0000-0001	<i>Start byte</i>
0000-001x	Reservado para o formato CBus
0000-010x	Reservado para um protocolo diferente
0000-011x	Reservado
0000-1xxx	Reservado
1111-1xxx	Reservado
1111-0xxx	Modo de endereçamento de 10 bit

Endereço para *General Call*

Este endereço é usado para aceder a todos os dispositivos presentes no barramento que suportem esta função. Dispositivos que não precisem desta função não respondem à mesma. Todos os bytes transferidos depois deste endereço podem ou não ser recebidos pelos *slaves* que responderam ao mesmo. Se nenhum *slave* enviar um *acknowledge* após transmissão de um byte a operação é cancelada com o envio de um *stopbit*. A função do *General Call* vem especificada no primeiro byte de dados transmitido e pode conter a seguinte informação:

0000-0110 -> Todos os dispositivos que responderem a este endereço fazem *reset* e actualizam a parte programável do seu endereço. Esta operação é realizada lendo o valor dos portos de selecção de endereço do dispositivo (quando presentes). Este comando pode ser usado para fazer *reset* a todos os dispositivos num barramento I²C.

0000-0100 -> Todos os dispositivos que responderem a este endereço actualizam a parte programável do seu endereço.

xxxx-xxx1 -> *Hardware Call*. Se um dispositivo necessita de atenção com urgência por parte de um *master* e não sabe com qual deve comunicar, pode usar esta função. Seguido deste endereço o dispositivo envia o seu próprio endereço. Todos os *masters* recebem esta mensagem e o *master* que souber como comunicar com o dispositivo que fez o pedido irá comunicar posteriormente com o *slave*.

Start byte

Esta função pode ser usada entre *masters*. Um *master* que não possua uma interface I²C por hardware necessita de monitorizar continuamente o barramento. O *Start Address* foi introduzido devido ao elevado tempo de processamento que esta operação necessita. Após transmissão deste endereço os *masters* diminuem a velocidade de transmissão. Quando detectam que a linha SDA está a nível baixo (fica a nível baixo aproximadamente durante *7 clocks*) podem novamente retomar a frequência normal. A comunicação deste endereço não tem um *Stopbit* no final mas sim um reenvio do *Startbit*.

CBus e outros protocolos

Estes endereços são usados quando se pretende utilizar o barramento para transmitir dados utilizando outro protocolo que não o I²C. Quando os dispositivos recebem este endereço, ignoram a informação transmitida no barramento até que seja enviado um *stopbit* ou um *startbit*.

Modo de endereçamento de 10 bit

Permite a utilização de endereços de 10 bit adicionando mais 1024 endereços disponíveis. Esta função não é suportada pelos dispositivos I²C mais antigos.

3.2. SPI

O barramento SPI é uma interface de comunicação série síncrona de quatro fios definida pela Motorola e usada actualmente por muitos microcontroladores para ligação aos seus periféricos. Funciona em modo Full-duplex com frequências de relógio até 70MHz permitindo assim uma taxa de transferência teórica máxima de 70Mbit/s simultaneamente nos dois sentidos. Esta interface utiliza três ligações partilhadas por todos os periféricos (SCLK, MOSI, MISO) e uma ligação adicional (SS) por cada periférico ligado ao barramento utilizada para activar a comunicação no dispositivo, como se pode observar na Fig. 50.

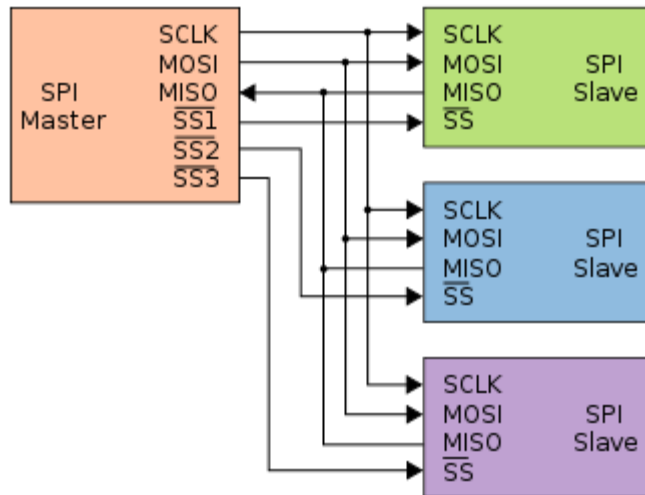


Fig. 50 – Esquema de ligação de um barramento SPI

3.2.1. Funcionamento

O barramento SPI funciona com um *master*, que controla o relógio e inicia as transferências, e com um ou mais *slaves*. Quando é ligado apenas um *slave* ao barramento este não é partilhado, como tal, a existência da ligação SS (*Slave Select*) torna-se obsoleta e pode ser ligada directamente a zero lógico em vez de ser ligada ao microcontrolador. Nalguns dispositivos a comunicação é iniciada na transição do sinal SS de nível alto para baixo sendo necessária a ligação destes ao microcontrolador independentemente do número de *slaves* no barramento. Outra limitação que pode existir é de o *slave* não permitir alta-impedância nas saídas, limitando assim o barramento a um dispositivo.

3.2.1.1. Transferência de dados

Para iniciar uma transferência de dados é necessário configurar o sinal de relógio do *master* para uma frequência suportada pelo *slave*, assim como a sua polaridade e fase. Nos microcontroladores podemos encontrar estas duas variáveis com o nome CPOL e CPHA respectivamente. CPOL define se a base do sinal de relógio é nível lógico alto ou baixo enquanto CPHA define se a leitura dos dados é realizada na primeira ou segunda transição de nível do sinal de relógio. Seguidamente, selecciona-se o dispositivo com o qual se pretende comunicar activando a saída SS correspondente e inicia-se a transmissão de ciclos de relógio. Em cada ciclo de relógio é transmitido um bit do *master* para o *slave* e um bit do *slave* para o *master*. Após a transferência de oito bits os dois dispositivos recolhem a informação e usam-na de acordo com a sua finalidade podendo ser iniciada uma nova transferência.

Vantagens:

- Comunicação *full duplex*
- Taxas de transferência maiores que no I²C
- Grande flexibilidade no protocolo
- Interface extremamente simples
 - Não existe necessidade do uso de *pull-up*
 - Não há possibilidade de colisão de dados
 - Não é necessário o uso de endereçamento (I²C)

Desvantagens:

- Necessita de um maior número de pinos que o I²C
- Necessita de uma ligação extra por cada *slave*
- Não existe acknowledge do *slave* (o *master* não consegue detectar se o *slave* está a receber a informação)
- Apenas é suportado um *master* no barramento
- O protocolo não tem um standard bem definido

3.3. Desenvolvimento

O software e *firmware* para os Kits foram desenvolvidos tendo como objectivo a simplicidade e transparência na sua utilização, libertando o utilizador/programador da criação de interfaces ou protocolos não necessários para os objectivos do seu trabalho.

O módulo de comunicações dos Kits tem um funcionamento independente e autónomo, não sendo necessária intervenção do utilizador. Para isso foi criado *firmware*, que está constantemente a ser executado no microcontrolador, encarregue da gestão do barramento I²C sendo apenas necessário ao utilizador a escolha do *master* do barramento e do endereço de cada Kit, através de um jumper e de um *encoder* BCD. A programação do microcontrolador foi orientada para estabilidade do sistema e minimização de erros nas transferências de dados ficando para segundo plano a taxa de transferência no barramento. O *master* do barramento tem como funções principais a recolha de dados de todos os *slaves* e o reenvio da mesma para o seu destino final. Os *slaves* apenas armazenam a informação que o microcontrolador principal do Kit pretende enviar e a informação recebida que tem como destino o microcontrolador principal do Kit. O armazenamento de dados temporários é feito utilizando dois buffers de 32 bytes criados na memória RAM do microcontrolador.

3.3.1. Protocolo de transporte

A interface I²C não tem limite de dados enviados/recebidos em cada transmissão, assim, numa transmissão de dados do *slave* para o *master*, o *master* não sabe quando se ainda existem dados a transmitir por parte do *slave* ou não. No caso geral, os dispositivos *slave* ligados a um barramento I²C têm protocolos definidos que incluem alguns comandos e transmissão de dados com tamanho fixo. Nesta aplicação não poderia ter sido utilizado um protocolo deste género porque os dados transmitidos entre os Kits são da responsabilidade do utilizador tornando-se assim de tamanho variável e informação aleatória. Para evitar desnecessárias transferências de dados e uma melhor optimização do barramento criou-se um protocolo de transporte composto por um cabeçalho de três bytes e uma frame de dados com tamanho máximo de treze bytes. Este protocolo permite em qualquer altura saber a origem e destino do pacote, enviar comandos directos ao microcontrolador do módulo de comunicações e detectar de erros de transmissão. O protocolo foi definido fazendo um balanço entre *overhead* de informação e benefícios para o sistema.

O protocolo de transporte está dividido em duas secções:

- Cabeçalho – constituído por cinco campos
- Dados – informação a ser enviada

O cabeçalho apresenta a organização definida pela Tabela 12

Tabela 12 – Organização do cabeçalho do protocolo de transporte

Bit offset	0-3	4-7
0	Endereço de Origem	Endereço de destino
8	Comando	Dimensão do pacote
16	Checksum	
24+	Dados	

Endereço de origem – identifica a origem dos dados

Endereço de destino – identifica o destino dos dados

Comando – os comandos utilizados no protocolo de transporte encontram-se descritos na Tabela 13

Tabela 13 – Comandos utilizados no protocolo de transporte

Valor	Descrição do comando
0	Utilizado para verificar se o slave está ligado e para teste do barramento
1	Checksum correcto
2	Checksum incorrecto
3	Espaço livre nos buffers
4	Envio de endereço do master
5	Nível do Kit
6	Não Utilizado
7	Buffer overflow
8	Soft Reset
9	Envio ou pedido de leitura de consumo do microcontrolador
10	Envio ou pedido de leitura de consumo do Kit
11	Envio ou pedido de leitura da entrada AUX1
12	Envio ou pedido de leitura da entrada AUX2
13-14	Não Utilizados
15	Dados para o microcontrolador principal

Dimensão do pacote – tamanho total em bytes do pacote a transmitir

Checksum – Valor calculado para detecção de erros na transferência de dados

Dados – Dados a transmitir

3.4. Fluxogramas

3.4.1. Master

3.4.1.1. Gestão de interrupções

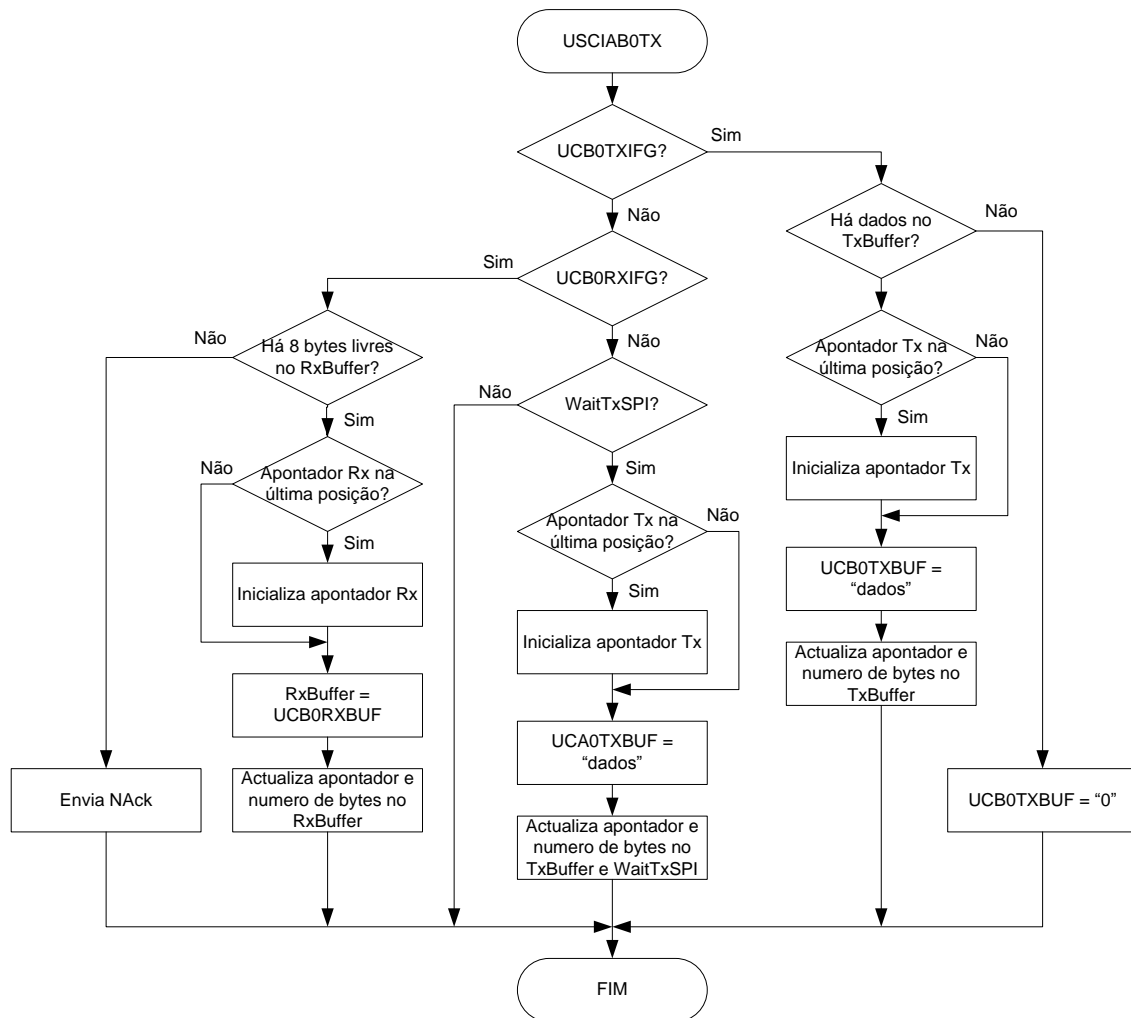


Fig. 51 - Gestão de interrupções no vector USCIAB0TX (master)

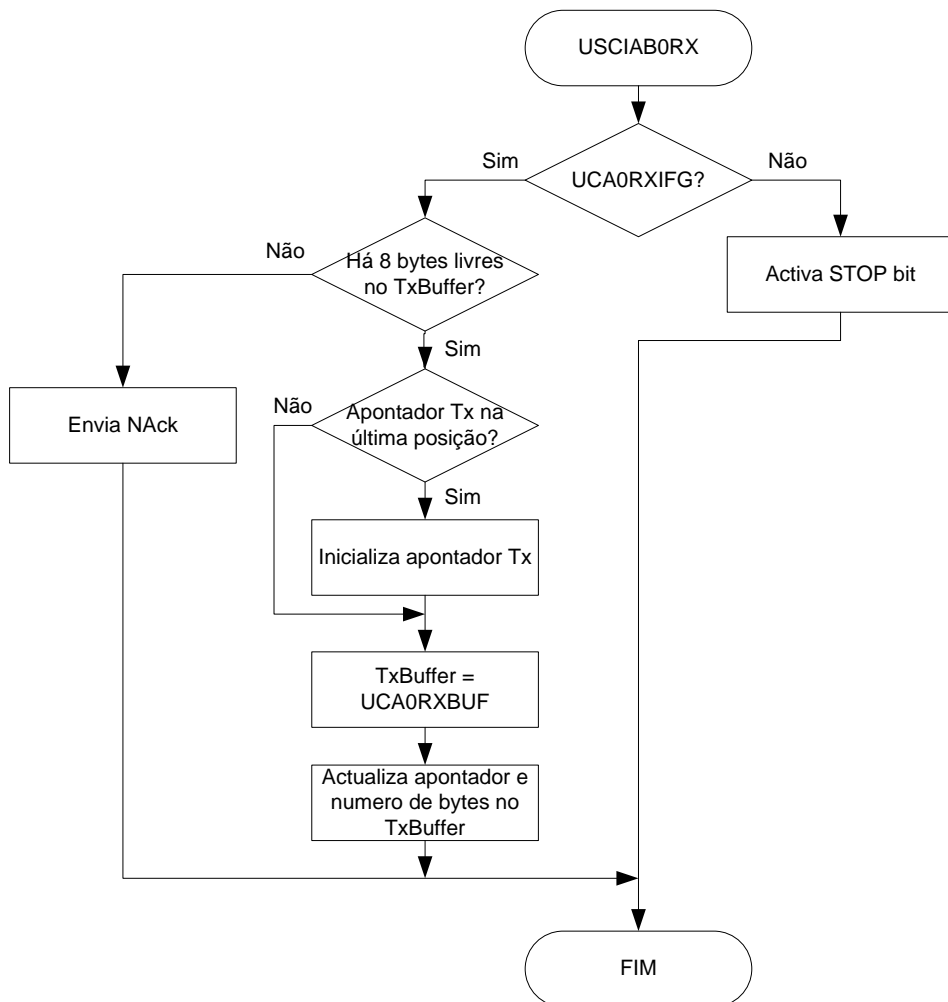


Fig. 52 - Gestão de interrupções no vector USCIAB0RX (*master*)

3.4.1.2. Firmware

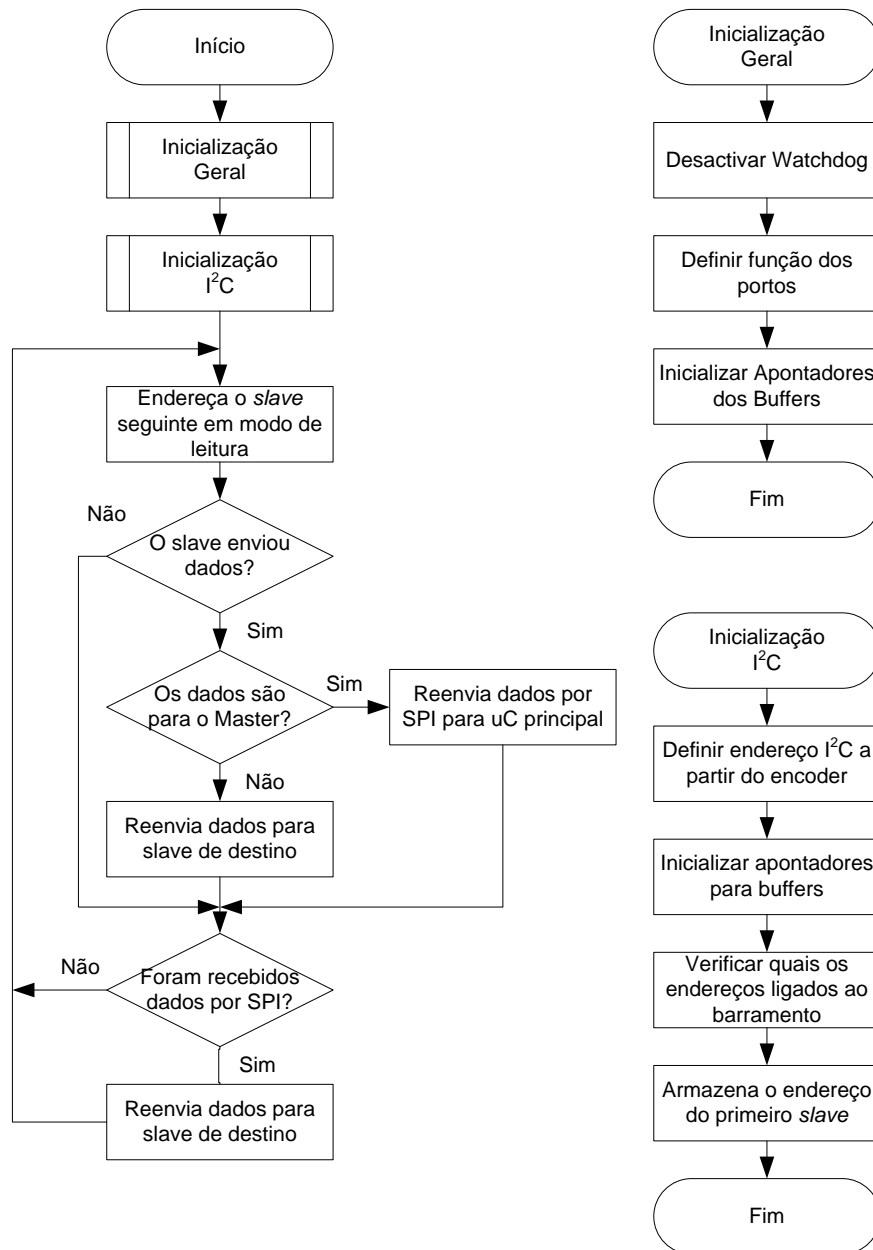


Fig. 53 – Diagrama de fluxo do *firmware* do *master*

3.4.2. Slave

3.4.2.1. Gestão de interrupções

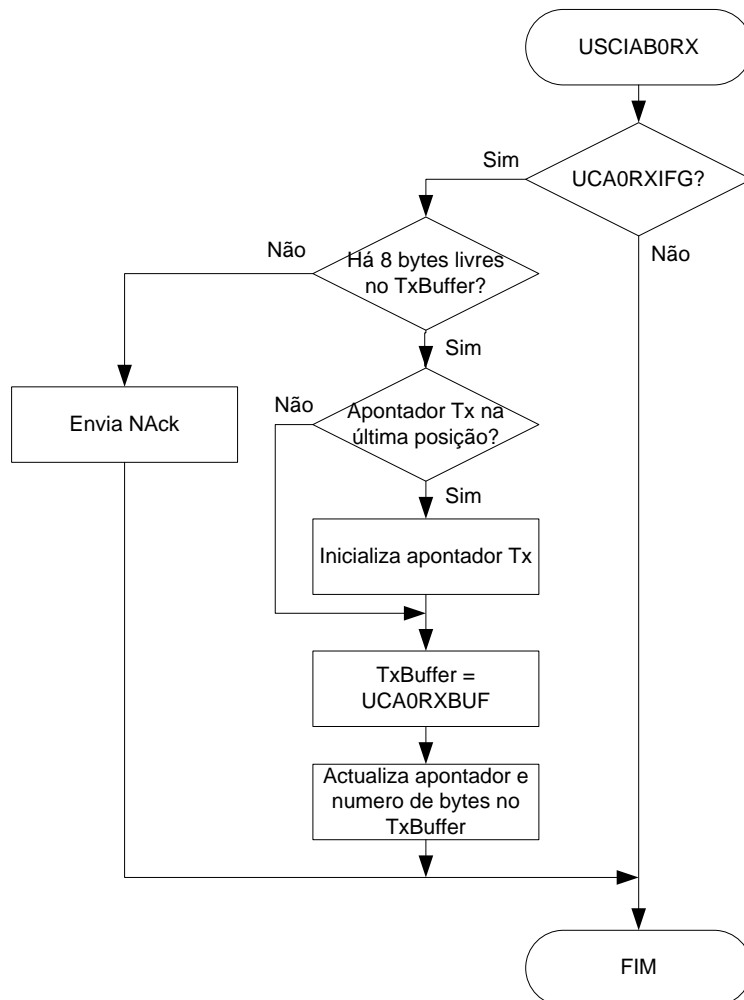


Fig. 54 - Gestão de interrupções do vector USCIAB0RX (slave)

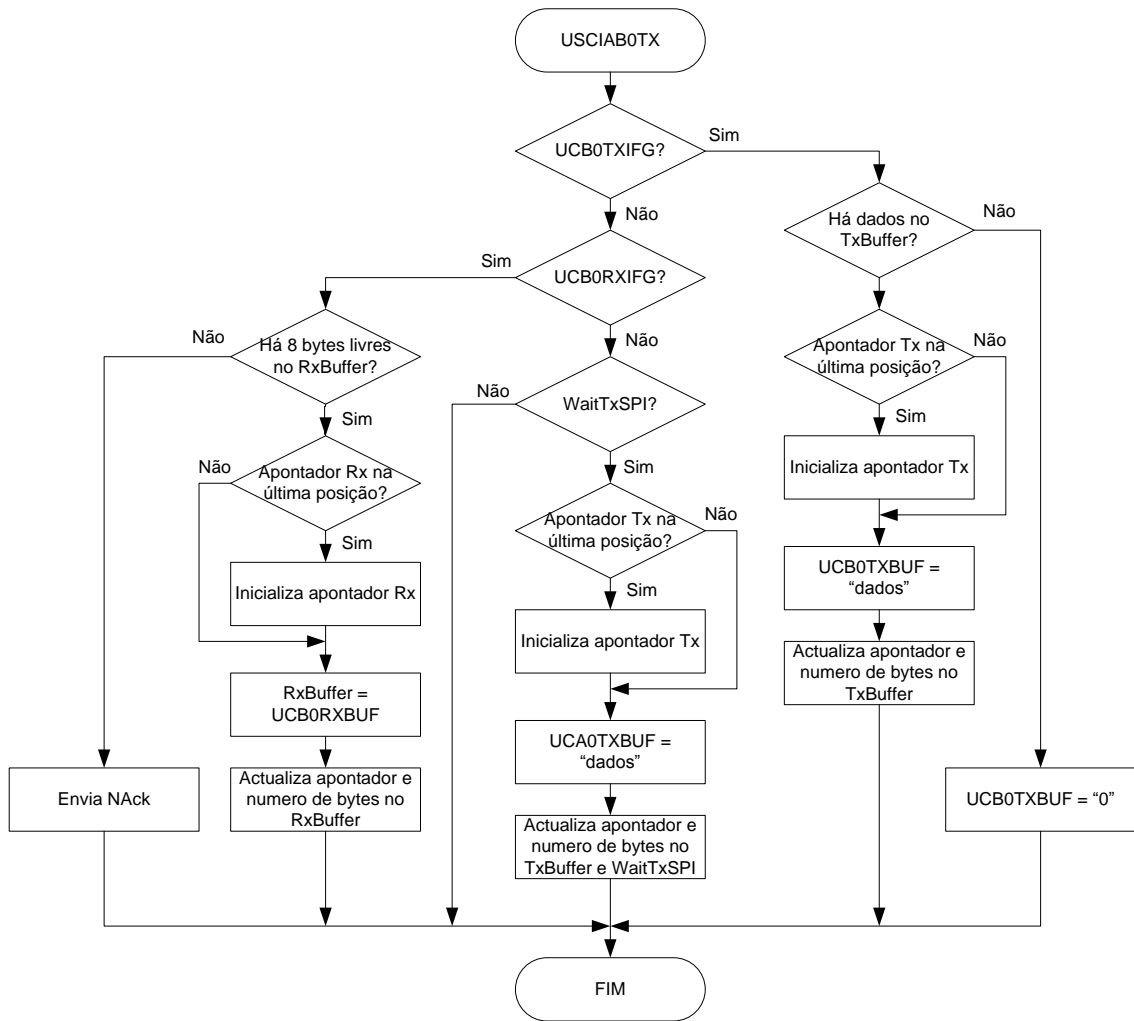


Fig. 55 - Gestão de interrupções no vector USCIAB0TX (slave)

3.4.2.2. Firmware

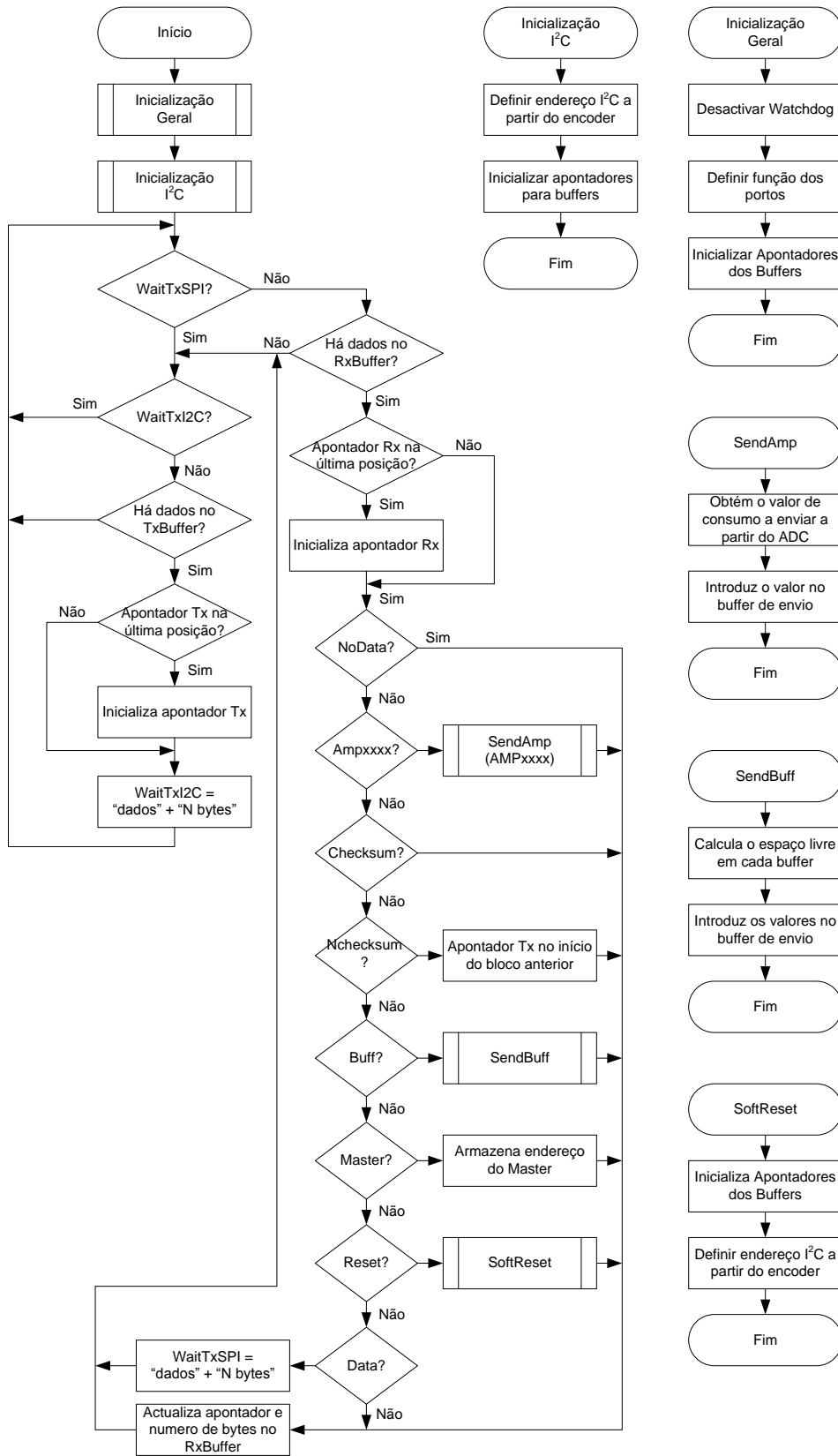


Fig. 56 - Diagrama de fluxo do *firmware* do *slave*

4. Bibliografia

J. Davies, “MSP430 Microcontroller Basics”, 2008

J. Luecke, “Analog and Digital Circuits for Electronic Control System Applications: Using the TI MSP430 Microcontroller”, 2004

C. Nagy, “Embedded Systems Design Using the TI MSP430 Series”, 2003

J. Ganssle, “The Art of Designing Embedded Systems”, 2008

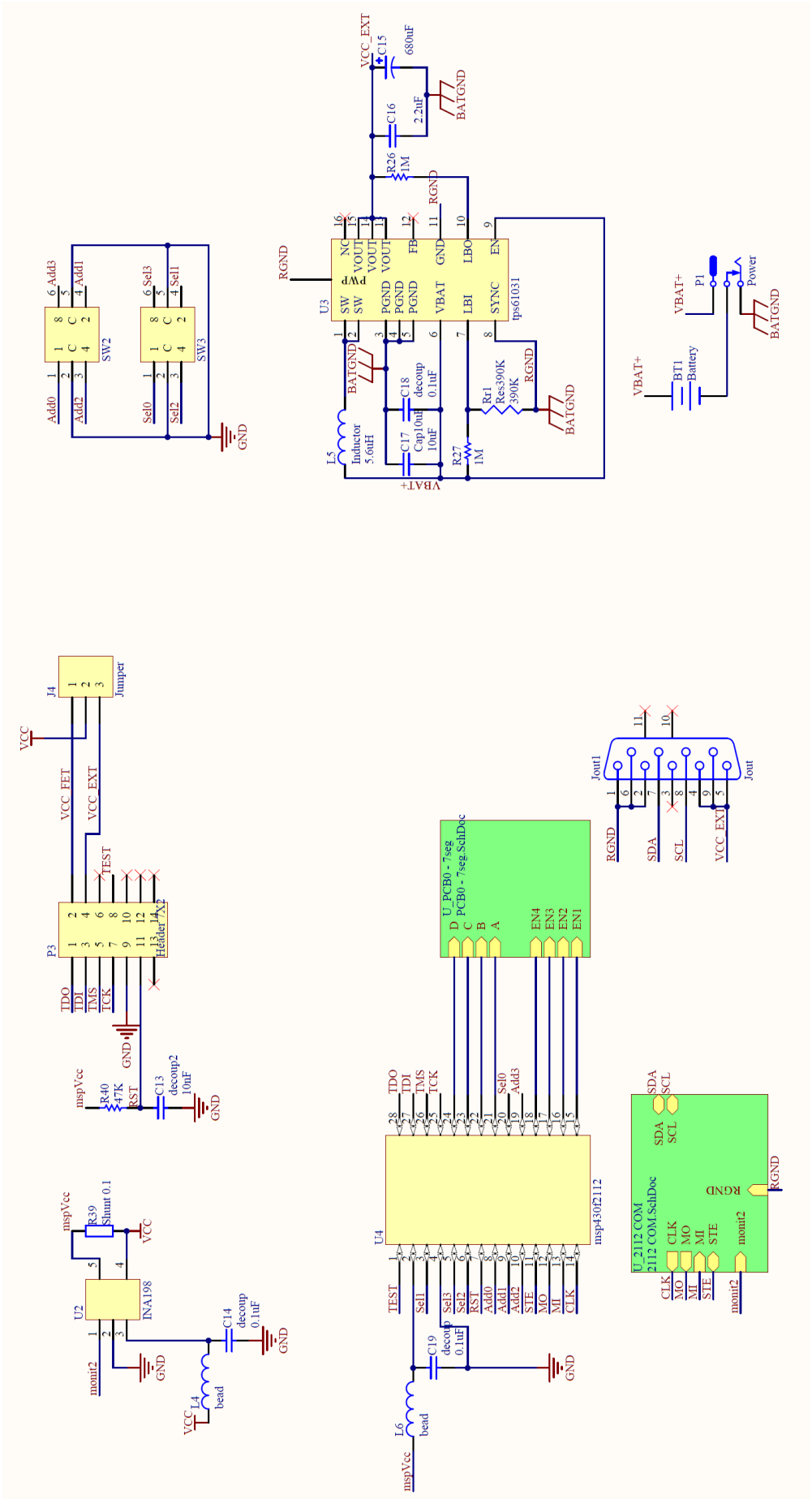
T. Noergaard, “Embedded Systems Architecture: A Comprehensive Guide for Engineers and Programmers (Embedded Technology)”, 2005

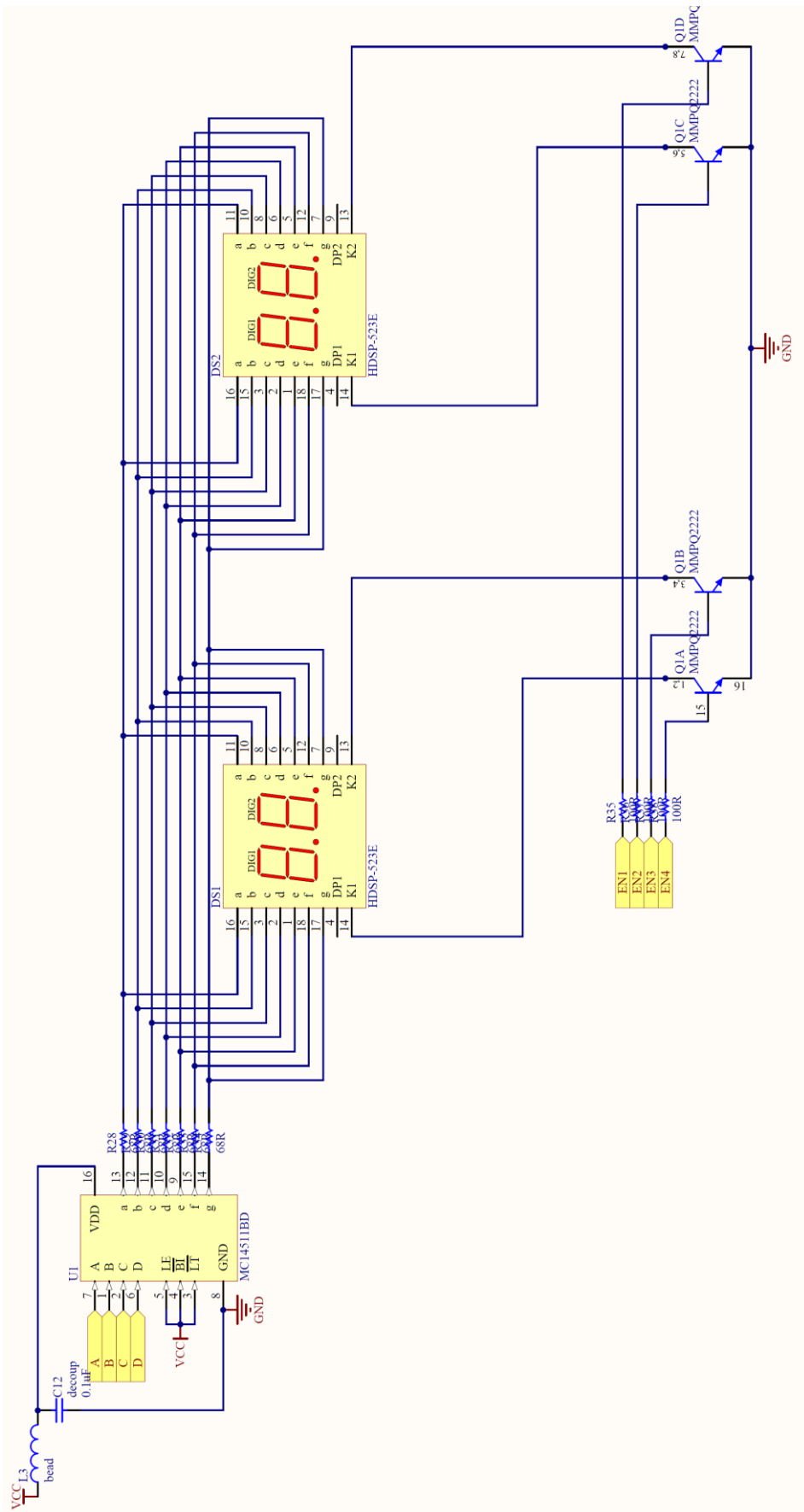
J. Catsoulis, “Designing Embedded Hardware”, 2005

J. Labrosse, “Embedded Systems Building Blocks: Complete and Ready-to-use Modules in C”, 1999

Anexo A

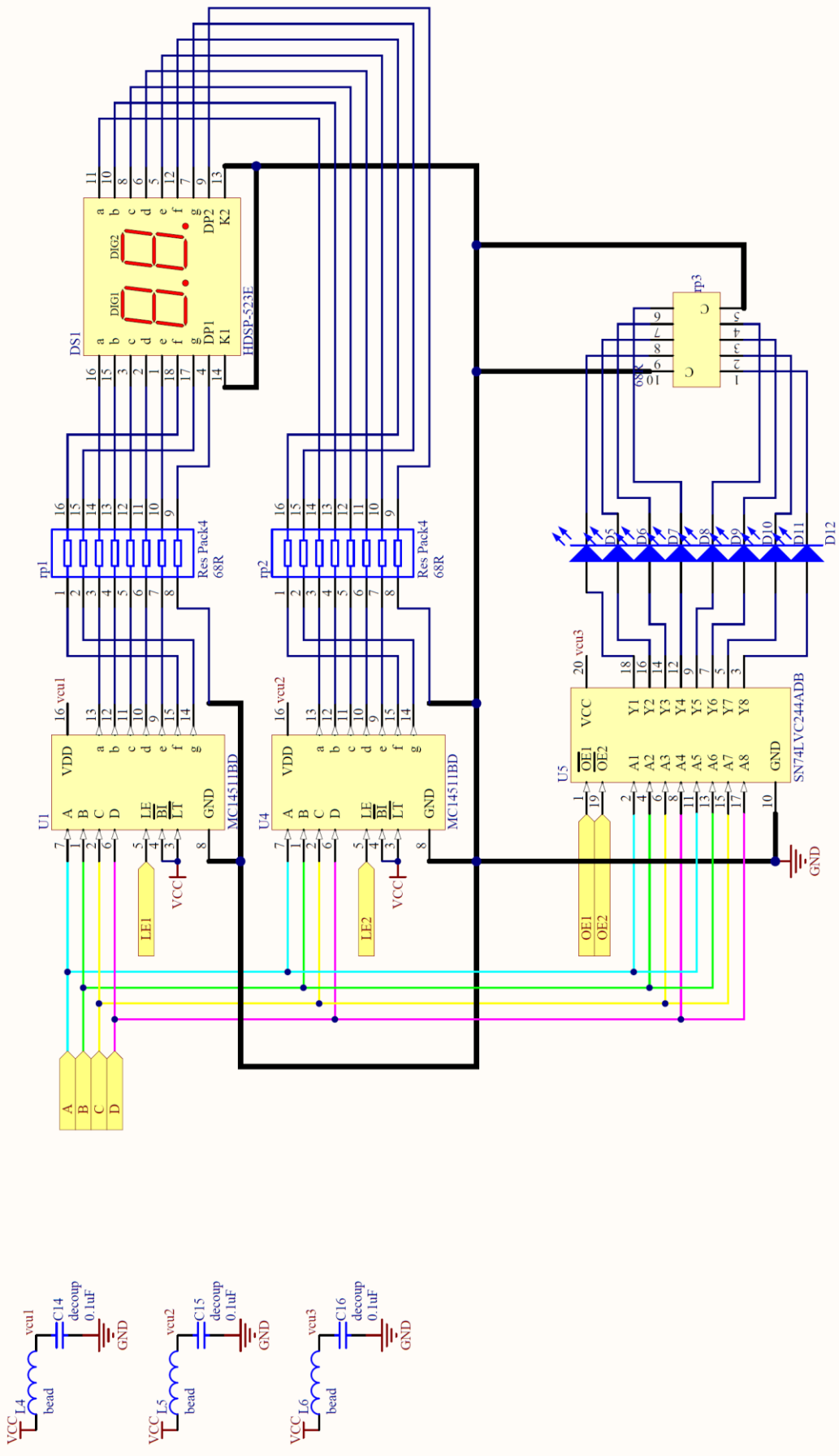
Esquemáticos kit nível 0





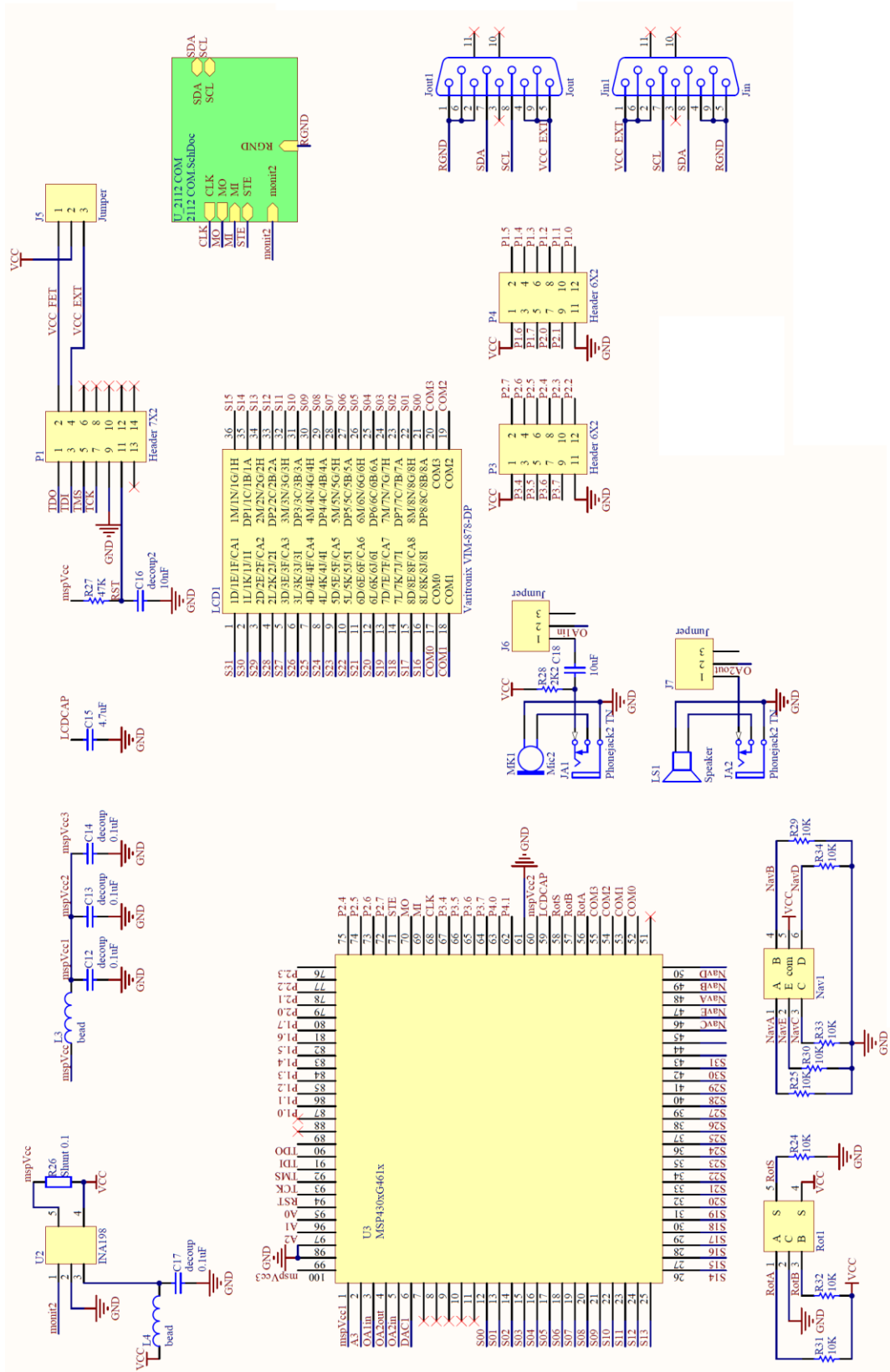
Anexo B

Esquemáticos kit nível 1



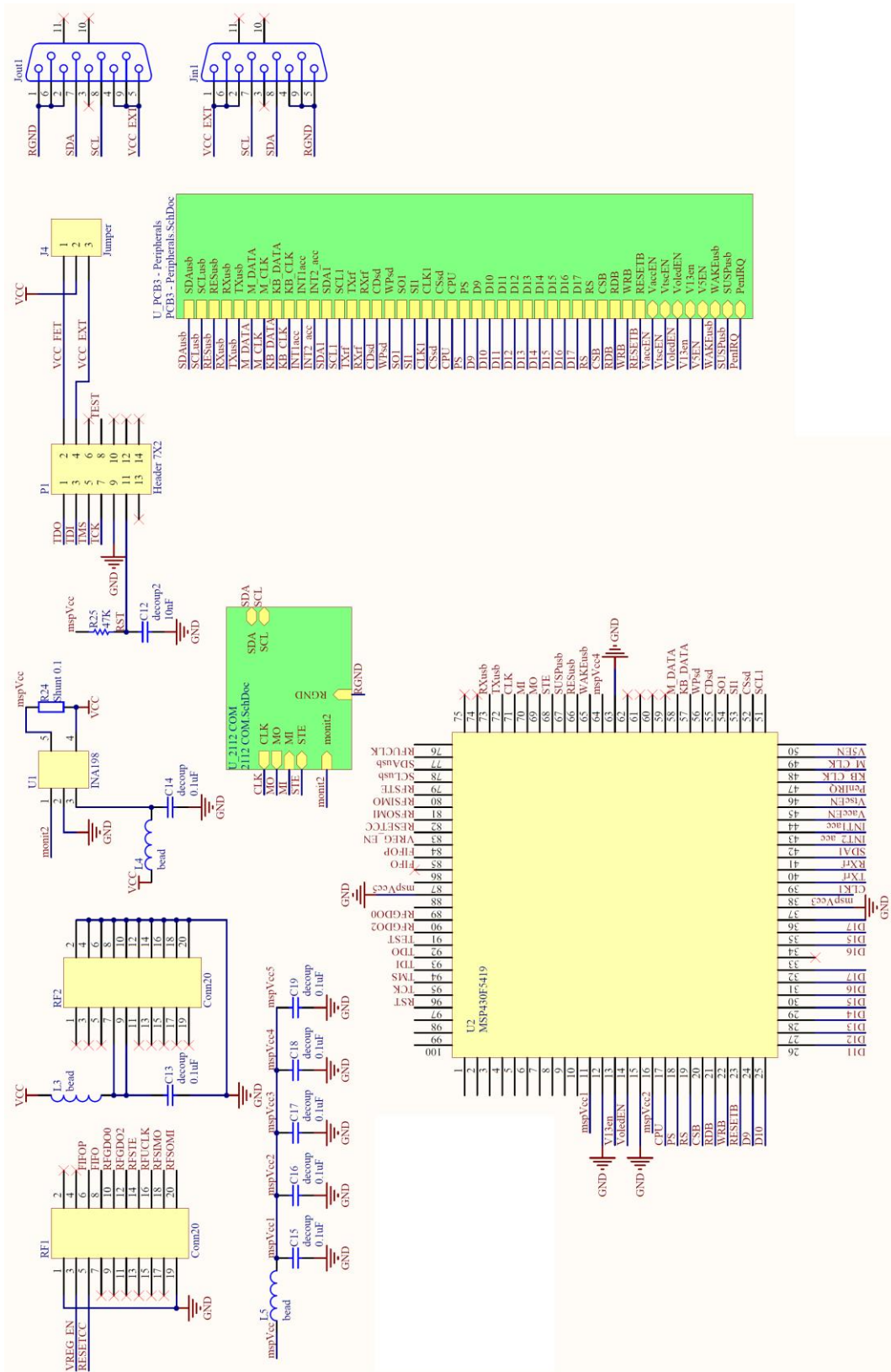
Anexo C

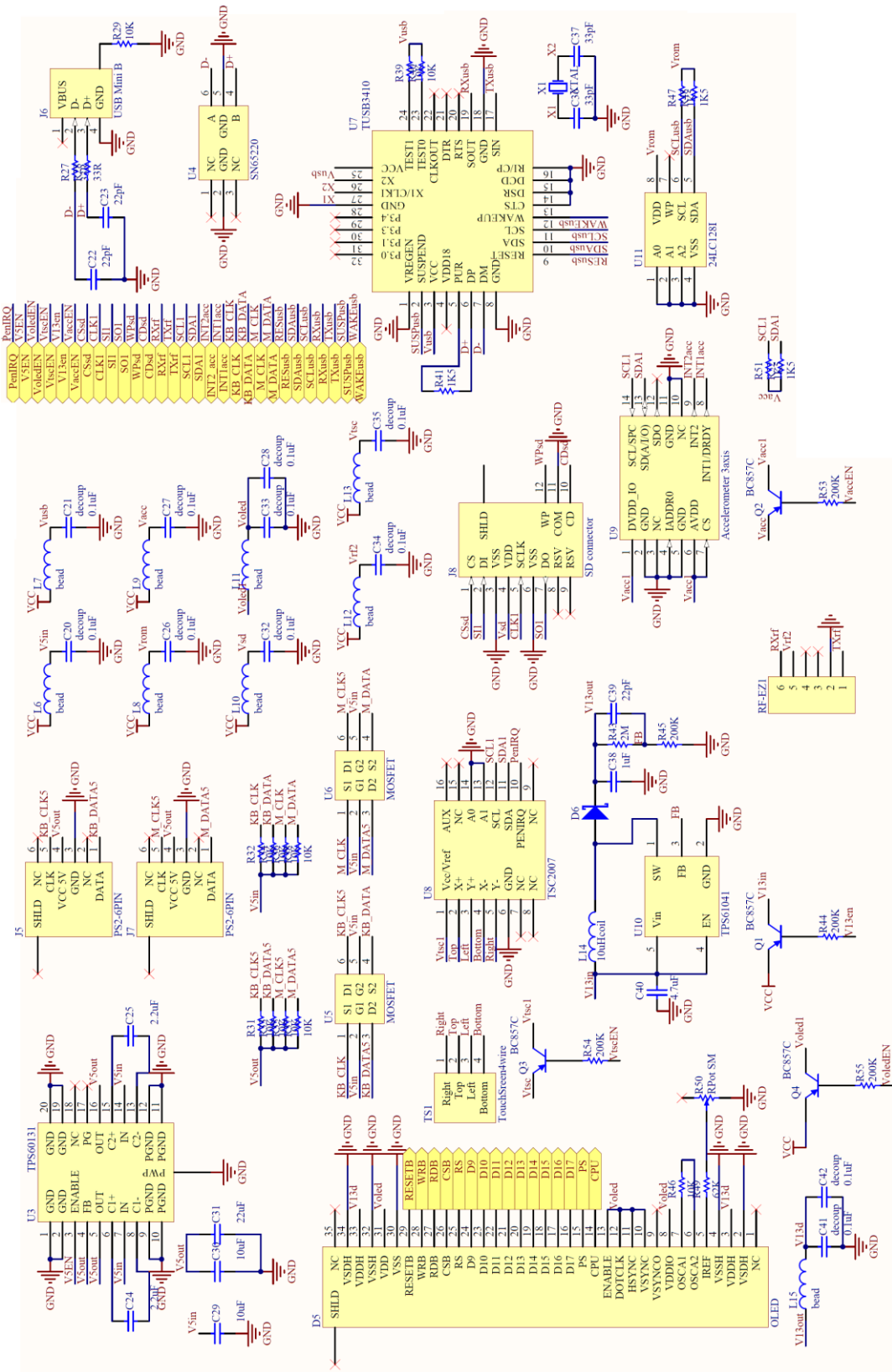
Esquemáticos kit nivel 2



Anexo D

Esquemáticos kit nível 3





Anexo E

Esquemáticos do módulo de comunicações

