

**Sistema automatizado do toque, iluminação e
relógios de torres sineiras
Aplicação à Igreja do Sagrado Coração de Jesus
(Covilhã)**

Jorge Manuel Baptista

Dissertação para obtenção do Grau de Mestre em
Engenharia Eletrotécnica e de Computadores
(2^o ciclo de estudos)

Orientador: Prof. Doutor Pedro Miguel de Figueiredo Dinis Oliveira Gaspar

junho de 2021

Agradecimentos

Este espaço é dedicado a todos aqueles que deram a sua contribuição para que esta dissertação fosse realizada.

Para começar, quero agradecer à comunidade e párocos e ex-párocos da Paróquia de S. Pedro da Cidade da Covilhã, por ajudarem a formar-me como pessoa e por permitirem aumentar o meu conhecimento nas mais variadas áreas. Sem o grande contributo e apoio de toda a comunidade e dirigentes não seria possível a realização desta dissertação.

Quero agradecer também ao Professor Doutor Pedro Dinis Gaspar pela sua orientação desta dissertação, estando sempre disponível para qualquer problema ou dúvida, dando-me total liberdade para o desenvolvimento desta.

Agradeço também a toda a minha família por todo o carinho, apoio e dedicação em todos os momentos da minha vida.

A todos os meus amigos, por todo o apoio, tolerância compreensão e amizade e incentivo a esta dissertação. Em especial à Mafalda Proença, ao Tiago Fonseca, ao Ricardo Mendes ao Gabriel Andrade, à Maria Teresa Santos, ao Pedro Gomes e ao Pedro Coelho pelo incentivo à escolha desta dissertação, auxiliando-me não só em todos os momentos desta dissertação, mas também na minha vida.

Por fim, quero agradecer também à minha querida Ana Beatriz Antunes, não só por todo o apoio e incentivo à instalação de novas tecnologias em património, atendendo sempre à sua preservação, mas também por todo o apoio e carinho que me tem dado nesta fase da minha vida. À sua família também deixo uma nota especial.

Resumo

A arte de tocar o sino é uma arte que conta com milhares de anos e tem como objetivo sinalizar as horas e também marcar eventos importantes, desde nascimentos a funerais ou até mesmo eventos litúrgicos.

Existem várias torres sineiras, normalmente associadas a espaços religiosos, sendo uma delas a torre sineira da Igreja do Sagrado Coração de Jesus na Cidade da Covilhã. Sendo um marco bastante importante nesta cidade, concluída em 1888, esta torre tem cerca de 46,5m de altura e conta com um carrilhão de 9 grandes sinos, perfazendo um peso total de cerca 3,6 toneladas. A torre também conta com um sistema de iluminação, no momento, inoperacional.

Esta dissertação tem como objetivo o desenvolvimento de um dispositivo de automação de toques para o carrilhão da torre da Igreja do Sagrado Coração de Jesus e controlo de iluminação para exterior. O controlo deverá permitir ao utilizador programar estas funcionalidades com base na semana, ano e imediato, dando total liberdade ao utilizador. Para cumprir os objetivos, é utilizado um microcomputador Raspberry Pi Zero W que, complementado com a ferramenta Django e Apache, permitem ao utilizador controlar o dispositivo sineiro através da rede sem fios já existentes na igreja, dispensando assim também a necessidade de o utilizador se encontrar junto deste para executar as alterações. Para albergar todo o dispositivo foi desenhada uma PCB, sendo este dispositivo conectado ao sistema já existente que não permite ao utilizador a fácil alteração das tarefas, sendo apenas necessário a substituição deste.

Palavras-chave

Carrilhão, Sinos, Torre Sineira, Dispositivo de Controlo, Automação

Abstract

The art of ringing a bell have thousands of years and have as aim not only indicate the time but also important events, like a birth, a dead or liturgical event.

Around the globe, exists various tower bells, associated normally to a religious space. One of the many tower bells existing in Covilhã city is the tower bell of Sagrado Coração de Jesus Church. One of the most iconic structure of the city and finished in 1888, this tower has a Carillon with 9 bells, with a total weight of 3,6 tons. The tower also has a light system, currently not working.

This dissertation has as aim a development of an automation device to control the carillon of the tower bell church and a control of outdoor illumination system. This control must allow current user to modify all settings of the system, according to week, year or immediately, giving all the freedom to user. To fulfil this aim, it used a microcomputer Raspberry Pi Zero W, complemented by using frameworks Django and Apache, allow user to control the system with existing Wi-Fi, allowing user also control the system anywhere. To integrate all in one small device and connectable to existing system, a PCB was designed.

Keywords

Carrilion, Bells, Tower Bells, Control Device, Automation

Índice

Agradecimentos	iii
Resumo	v
Abstract	vii
Índice	ix
Lista de Figuras	xi
Lista de Tabelas	xv
Nomenclatura	xvii
1. Introdução	1
1.1. Enquadramento	1
1.2. O problema em estudo e a sua relevância	3
1.3. Objetivos e contribuição da dissertação	6
1.4. Visão geral e organização da dissertação.....	7
2. Estado da Arte	9
2.1. Sistemas sineiros	9
2.1.1. Mecanismos mecânicos de torres sineiras	9
2.1.2. Sistemas eletromecânicos ou eletrônicos.....	12
2.2. Dispositivos de Controlo de Iluminação	25
2.3. Nota conclusiva.....	27
3. Materiais e Métodos	29
3.1. Dispositivos	29
3.1.1. Escolha do controlador	29
3.1.2. Instalação do Sistema Operativo e das aplicações e configuração do Raspberry	30
3.2. Interface de Controlo do Dispositivo	34
3.2.1. Controlo de Horas	34
3.2.2. Toque Imediatos de Melodias.....	43
3.2.3. Controlo de Iluminação.....	46
3.3. Desenho do Hardware do Dispositivo	50
3.3.1. Placa de circuito impresso (PCB)	51
3.3.2. Caixa.....	55
4. Testes de funcionamento	59
4.1. Montagem do dispositivo	59
4.2. Testes.....	60

4.3. Nota conclusiva.....	66
5. Conclusões	67
5.1. Conclusões gerais	67
5.2. Sugestões de trabalhos futuros	68
6. Referências Bibliográficas	69

Lista de Figuras

FIGURA 1- EXTERIOR DA IGREJA E TORRE SINEIRA, NA ATUALIDADE.....	2
FIGURA 2 - INTERIOR DA IGREJA ANTES DA REVOLUÇÃO DE 1910 (É POSSÍVEL OBSERVAR UM DOS ALTARES OFERECIDOS À IGREJA DE SANTA MARIA MAIOR, NA COVILHÃ, APÓS A REVOLUÇÃO DE 1910)	2
FIGURA 3- MECANISMO DA TORRE SINEIRA.....	4
FIGURA 4- SISTEMA ELETROMECAÂNICO ATUALMENTE EM FUNCIONAMENTO.....	5
FIGURA 5- PRINCIPIO DE FUNCIONAMENTO DO CARRILHÃO A) MANUAL B) AUTOMÁTICO.	10
FIGURA 6 – MECANISMO INSTALADO EM TILLMAN HALL [4].....	11
FIGURA 7 – MECANISMO DE RELÓGIO INSTALADO EM TILLMAN HALL [4].....	11
FIGURA 8 - MECANISMO DE TOQUE DE HORAS [4]	12
FIGURA 9 – EXACTUS [5].....	13
FIGURA 10 - FORMATOS SIGMA MASTER CLOCK [6].....	14
FIGURA 11 - ESQUEMA DE FUNCIONAMENTO DO SIGMA H MASTER CLOCK [7].....	14
FIGURA 12 - ESQUEMA DE FUNCIONAMENTO DO SIGMA P MASTER CLOCK [8].....	15
FIGURA 13 - ESQUEMA DE FUNCIONAMENTO DO SIGMA C MASTER CLOCK [9].....	15
FIGURA 14 - ESQUEMA DE FUNCIONAMENTO DO SIGMA MOD MASTER CLOCK [10].....	16
FIGURA 15 - SOFTWARE SIGMA MASTER CLOCK [11].....	17
FIGURA 16 – MICROQUARTZ S [12].	17
FIGURA 17 – ESQUEMA DE ESTRUTURA DE HARDWARE DO DISPOSITIVO [13]	18
FIGURA 18 – AT89S52 [13]	19
FIGURA 19 - DS12C887 [13].....	19
FIGURA 20 – CIRCUITO DE TRANSMISSÃO, À ESQUERDA, E CIRCUITO DE RECEÇÃO, À DIREITA [13].....	20
FIGURA 21 – DIAGRAMA DE FUNCIONAMENTO DO DISPOSITIVO [14].....	21
FIGURA 22- ESQUEMA DE CIRCUITO DO SISTEMA [14].	21
FIGURA 23- ESQUEMA DO SISTEMA [15].....	22
FIGURA 24 – INTERFACE DO DISPOSITIVO.	23
FIGURA 25 – ESQUEMA DO DISPOSITIVO.....	24
FIGURA 26 – INTERRUPTOR HORÁRIO DE 1 CANAL DA EFAPEL.....	25
FIGURA 27 – INTERRUPTOR CREPUSCULAR EE200 E SENSORES.	26
FIGURA 28 - PHILIPS DYNALITE.....	27
FIGURA 29- RASPBERRY PI IMAGER.....	30
FIGURA 30 – INTRODUÇÃO DO NOME FICHEIRO “HORAS” E “ILUMINAÇÃO” NA SECÇÃO “INSTALLED_APPS”	32
FIGURA 31- ENTRADAS A ADICIONAR NO FICHEIRO “000-DEFAULT.CONF”	33

FIGURA 32- PÁGINA INICIAL.....	33
FIGURA 33 – ESQUEMA DE FUNCIONAMENTO DO PROGRAMA PRINCIPAL.....	35
FIGURA 34- CLASSE DEFINIDA PARA SEMANA/DIÁRIO.....	36
FIGURA 35 - CLASSE DEFINIDA PARA ANUAL.....	37
FIGURA 36 - ÁREA DE ADMINISTRAÇÃO COM AS TABELAS INSERIDAS.....	37
FIGURA 37 – ROTINA PARA LEITURA DE UMA TABELA PROVENIENTE DA BASE DE DADOS.....	38
FIGURA 38 - MÓDULOS A IMPORTAR.....	39
FIGURA 39 – PROGRAMA PRINCIPAL – LEITURA DE DADOS.....	40
FIGURA 40 - PROGRAMA PRINCIPAL – ATRIBUIÇÃO DA MELODIA.....	41
FIGURA 41 - PROGRAMA PRINCIPAL – ATRIBUIÇÃO DO NÚMERO DE TOQUES.....	41
FIGURA 42 – ROTINA “DIRETORIOS”, QUE DIRECIONA O PROGRAMA PRINCIPAL ÀS ROTINAS DE TOQUE.....	42
FIGURA 43 – EXEMPLO DE MELODIA DE TOQUE.....	42
FIGURA 44 - ESQUEMA DE FUNCIONAMENTO DO SISTEMA COM TOQUES IMEDIATOS E HORÁRIOS.....	43
FIGURA 45 – ROTINA A ADICIONAR PARA A CRIAÇÃO DA TABELA DE TOQUE IMEDIATOS.....	44
FIGURA 46 - REGISTO DA TABELA NA ÁREA DE ADMINISTRAÇÃO.....	44
FIGURA 47 – LEITURA DA TABELA EXISTENTE NA BASE DE DADOS.....	45
FIGURA 48 - ROTINA DE TOQUES IMEDIATOS NA ROTINA PRINCIPAL.....	45
FIGURA 49 - ROTINA DE REMOÇÃO DA ENTRADA EXISTENTE NA TABELA DE TOQUES IMEDIATOS.....	46
FIGURA 50 – ATIVAÇÃO DO PORTO DE SAÍDA DO DISPOSITIVO.....	46
FIGURA 51 – PROGRAMA PRINCIPAL COM A INTEGRAÇÃO DE CONTROLO DE ILUMINAÇÃO.....	47
FIGURA 52 – CRIAÇÃO DA TABELA NO FICHEIRO “MODELS . PY”.....	48
FIGURA 53 – REGISTO DA TABELA NA ÀREA DE ADMINISTRAÇÃO.....	48
FIGURA 54 - LEITURA DA TABELA “ILUMINAÇÃO_SEMANAIS”.....	49
FIGURA 55 – ROTINA DE ILUMINAÇÃO.....	50
FIGURA 56 – QUADRO ELÉTRICO, ABERTO, EXISTENTE NO CIMO DA TORRE, JUNTO AO CONJUNTO SINEIRO.....	51
FIGURA 57 – ESQUEMA DE CIRCUITO PARA UMA SAÍDA DO DISPOSITIVO DE CONTROLO.....	52
FIGURA 58 – REGULADOR DE TENSÃO.....	53
FIGURA 59 – LAYOUT DA PCB DA PLACA PRINCIPAL.....	54
FIGURA 60 – MODELO 3D DA PLACA DE CONTROLO.....	54
FIGURA 61 - OBTENÇÃO DAS DIMENSÕES DA PCB E DA RESPECTIVA FURAÇÃO.....	55
FIGURA 62 - PARTE DE BAIXO DA CAIXA.....	56
FIGURA 63 - DISPOSITIVO FINAL.....	57
FIGURA 64 - PCB DESENVOLVIDA.....	59
FIGURA 65 - PLACA FINAL.....	60
FIGURA 66 – MARTELO NA POSIÇÃO ORIGINAL.....	61
FIGURA 67 - MECANISMO DOS MOSTRADORES APÓS MANUTENÇÃO.....	61
FIGURA 68 – MONTAGEM DE TESTE DO DISPOSITIVO DE CONTROLO.....	62
FIGURA 69 - CÓDIGO DE ALTERAÇÃO DE REDE WIFI A CONECTAR.....	63

FIGURA 70 – CONFIGURAÇÃO DE UM TOQUE ANUAL.....	63
FIGURA 71 - DENIÇÃO DE TOQUE SEMANAL.....	64
FIGURA 72 - DEFINIÇÃO DE TOQUE IMEDIATO.....	64
FIGURA 73 - DEFINIÇÃO DE TEMPO DE FUNCIONAMENTO DE ILUMINAÇÃO.....	65
FIGURA 74 - EXEMPLO DE LISTA DE TOQUES ANUAIS.	65

Lista de Tabelas

Tabela 1 – Especificações técnicas de 3 controladores considerados.29

Nomenclatura

Geral:

<i>A</i>	Corrente elétrica, [A];
<i>V</i>	Tensão elétrica, [V];

Acrónimos:

<i>ALS</i>	Ambient Light Sensor
<i>CAD</i>	Computer Aided Design
<i>DCF</i>	Deutsche langwelliges Signal Frankfurt
<i>GPIO</i>	General Purpose Input/Output
<i>GPS</i>	Global Positioning System
<i>HTML</i>	HyperText Markup Language
<i>HTTP</i>	Hypertext Transfer Protocol
<i>I²C</i>	Inter-Integrated Circuit
<i>IDE</i>	Integrated Development Environment
<i>IP</i>	Internet Protocol
<i>MIDI</i>	Musical Instrument Digital Interface
<i>NTP</i>	Network Time Protocol
<i>PCB</i>	Printed Circuit Board
<i>RAM</i>	Randdom Acess Memory
<i>SMD</i>	Surface Mounted Components
<i>SSH</i>	Secure Shell
<i>STL</i>	Standard Triangle Language

1. Introdução

1.1. Enquadramento

Desde o início dos tempos, o ser humano tem a necessidade de ter uma noção de tempo, criando assim uma das suas maiores invenções: o relógio. O relógio pode assumir muitas formas e está presente nos mais variados lugares, desde o dispositivo que regularmente o ser humano utiliza no pulso, a satélites. Um dos locais onde regularmente se encontra esta invenção, ainda que por vezes disfarçado, é em torres sineiras, podendo também conter os mostradores.

Dependendo da data de construção, uma certa torre sineira tem um mecanismo próprio de funcionamento. Em construções recentes, o princípio de funcionamento deste mecanismo é controlado eletronicamente, apenas contendo como partes móveis os motores para girar os ponteiros do relógio e martelos elétricos para picar o sino ou motores para bamboar o sino. Já em construções mais antigas, os mecanismos de funcionamento seguem um funcionamento mecânico, com uma série de rodas dentadas e roldanas e um conjunto de pesos. Estes mecanismos mecânicos também têm a possibilidade de tocar uma melodia ou toques simples na passagem de horas. Estes mecanismos mecânicos também têm a particularidade de ser necessário, em certos períodos, “dar à corda” ao relógio, para este não parar.

Com o tempo, as manutenções descuidadas ou inexistentes dos mecanismos antigos podem levar a que estes acabem por avariar. Como os custos de reparação são elevados, em grande parte das vezes, ou porque, simplesmente, cada vez existem menos pessoas disponíveis para subir à torre e tocar os sinos ou dar corda ao relógio, é realizada uma alteração para um sistema eletrónico automatizado.

Existem por todo o mundo milhares de torres sineiras, e a Covilhã não é exceção. Grande parte das torres sineiras nesta zona têm poucos sinos instalados, sendo comum a utilização de 3 sinos. Das torres sineiras existentes, apenas se destaca a torre da Igreja do Sagrado Coração de Jesus com 46,5 metros de altura, sendo a mais alta da cidade e com 9 sinos fabricados no Porto, afinados em lá, e com peso total de 3600 kg, conforme exposto na Figura 1 [1].



Figura 1- Exterior da igreja e torre sineira, na atualidade.

A sua história começa em 1888, 11 anos após o início do culto nesta igreja. A igreja continha 5 altares, em estilo gótico, sendo possível verificar dois destes altares na Igreja de Santa Maria Maior, juntamente com o órgão de tubos, expostos na Figura 2.



Figura 2 - Interior da Igreja antes da Revolução de 1910 (é possível observar um dos altares oferecidos à Igreja de Santa Maria Maior, na Covilhã, após a revolução de 1910)

Em 1910, a igreja e anexos foram tomados pela República e os Jesuítas foram expulsos. Este complexo passou então a funcionar os Paços do Concelho, celeiro municipal e, mais tarde, Tribunal Judicial da Comarca da Covilhã [1]

Em 1948, após um grande incêndio, o edifício foi adquirido pelos Jesuítas novamente, sendo requalificada a igreja e inaugurada em 1952. Em 1957 foram adicionados mais 4 sinos, produzidos pela Fundação de Braga – Serafim da Silva Jerónimo & Filhos, LDA. Estes 4 sinos não se encontram instalados na torre, desconhecendo-se o motivo.

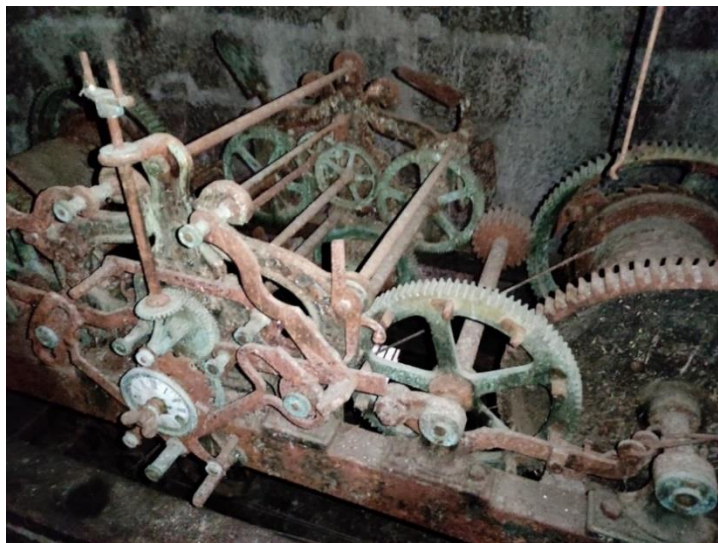
Em monumentos de interesse histórico é comum a utilização de iluminação pois permite a requalificação e valorização do património e de centros históricos, pelo que a Câmara da Covilhã instalou na torre e na área circundante, iluminação para que esta pudesse ser observada à noite, tal como todo o património da cidade. O sistema é bastante simples. A torre dispõe um conjunto de 4 projetores na zona do conjunto de sinos, 4 na zona dos mostradores e um conjunto de projetores em torno da torre sineira. Após vários anos sem qualquer tipo de manutenção e face às exigências de poupança de energia, esta iluminação foi desligada e assim ficou estes últimos anos.

1.2. O problema em estudo e a sua relevância

Após mais 50 anos em funcionamento, o mecanismo original da torre sineira desta igreja acabou por avariar, sendo a sua reparação bastante dispendiosa.

O mecanismo original é bastante similar a outros mecanismos existentes noutros locais. Este mecanismo é composto por três tambores de peso, que desciam numa conduta com o comprimento da torre, uma série de engrenagens e duas ventoinhas. Contém também um mostrador das horas (Figura 3).

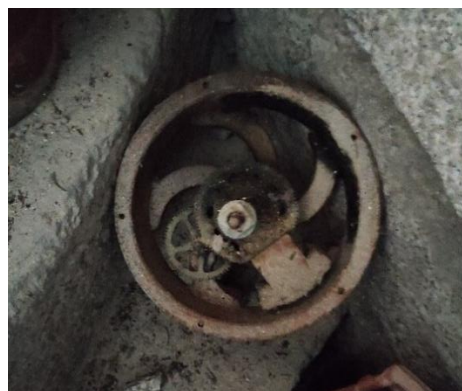
É possível observar, na Figura 3, o avançado estado de degradação do sistema, que se encontra exposto a bastante humidade. Durante muitos anos esteve também exposto a pombas, sendo este problema parcialmente resolvido na década de 2000. No entanto, o carrilhão continua exposto a aves, devido à má colocação das redes que impedem a entrada destas, e a condições meteorológicas adversas. Estes dois fatores fazem com que seja necessária uma manutenção preventiva, especialmente nas madeiras colocadas nos sinos do carrilhão. Encontra-se em falta os componentes utilizado para conectar este mecanismo aos mostradores exteriores, bem como os componentes necessários para tocar os sinos. Suspeita-se que seriam utilizados 3 sinos para este toque de horas.



(a) Mecanismo anterior à instalação do mecanismo eletromecânico.



(b) Mecanismo mecânico anterior à instalação do mecanismo eletromecânico.



(c) Componente do mecanismo do relógio, provavelmente atuaria num dos mostradores exteriores.



(d) Componente do mecanismo mecânico do relógio, responsável por toque do mecanismo mecânico do relógio.



(e) Teclado do Carrilhão da Torre, em estado avançado de deterioração.

Figura 3- Mecanismo da torre sineira.

Na década de 2000, foi efetuada a instalação de um sistema de automação para apenas 3 sinos da torre sineira, efetuada pela empresa Fundição de Sinos de Braga – Serafim da Silva Jerónimo & Filhos, LDA. Este mecanismo é composto por um sistema computadorizado para toques de sinos (Figura 4.a) e 3 martelos elétricos de picar sino (Figura 4.b), um deles avariado, e um acionamento elétrico para os 4 mostradores (Figura 4.c e 4.d), também avariado.



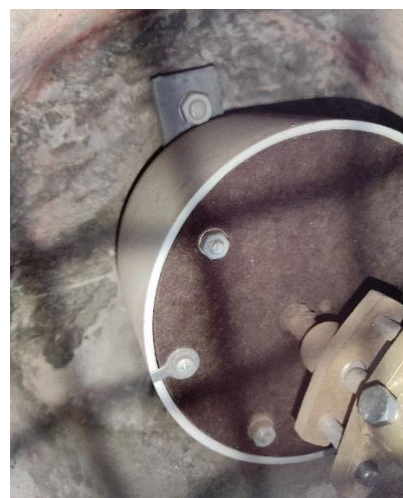
a) Dispositivo atualmente em utilização, Opus 10.



b) Martelo de Picar.



c) Mecanismo de controlo de mostradores.



d) Interior de mostrador do relógio

Figura 4- Sistema eletromecânico atualmente em funcionamento

Como é possível verificar na Figura 4, este dispositivo tem um ecrã LCD, onde é possível acionar os vários comandos, e saídas suficientes para poder tocar os 9 sinos colocados na torre. Este sistema tem um relógio interno e realiza o toque de um sino de hora a hora e à hora da missa. Este sistema também possui um comando para execução de melodias e um sensor de *Global Positioning System* (GPS) para sincronização automática da hora. Este sistema apresenta como principal problema, a ausência de martelos de picar para todos os sinos e o não funcionamento do acionamento para os 4 mostradores de relógio existentes na torre. O sistema computadorizado também tem uma interface pouco apelativa e bastante complexa, não permitindo ao utilizador a fácil marcação de toques para os sinos.

Já a iluminação atualmente instalada, para além de não permitir qualquer tipo de controlo por parte do utilizador, encontra-se completamente obsoleta, sendo utilizadas lâmpadas de iodetos metálicos e halogénio, com um consumo energético bastante elevado, para além de, devido à falta de manutenção, se encontrarem fundidas várias lâmpadas.

Esta torre é um marco importante na história da cidade da Covilhã, sendo a primeira construída após elevação a cidade. Tendo características únicas na cidade, é importante preservar as características originais desta torre sineira: dar a conhecer à população a hora e, neste caso específico, de toques religiosos, como por exemplo o chamamento para a missa.

1.3. Objetivos e contribuição da dissertação

Não sendo possível a reparação do mecanismo original do sistema, o objetivo da dissertação prende-se em projetar um dispositivo eletrónico para automatizar o toque dos 9 sinos instalados e um acionamento para os 4 mostradores da torre. Este dispositivo terá de ser facilmente programável pelo utilizador, podendo este determinar os dias e a hora em que os sinos tocam e escolher a melodia que o conjunto de sinos poderá executar.

Para além da automatização dos sinos, pretende-se também que o sistema de iluminação da torre seja acoplado ao dispositivo eletrónico para que este seja totalmente automatizado e controlável pelo utilizador, para que o sistema funcione em intervalo de horas e dias específicos.

Esta instalação irá permitir, do ponto de vista religioso, que a torre sineira, em funcionamento parcial há vários anos, volte a cumprir o seu papel na totalidade, nomeadamente a de sinalizar e mostrar horas por quem ali passa, seja de avisos sonoros ou visuais, de anunciar notícias de alegria ou tristeza à população local ou de simplesmente o chamamento para as missas nesta igreja. Do ponto de vista cultural e de património, a recuperação da iluminação que se encontra fora de serviço é uma forma de relembrar a história, não só dos Jesuítas na cidade da Covilhã,

mas também os quase 40 anos que esta torre esteve ao serviço do estado, já para não falar do seu valor arquitetónico.

1.4. Visão geral e organização da dissertação

A organização geral da dissertação decorre do geral para o específico, de modo a facilitar a compreensão dos assuntos expostos, estendendo-se por 5 capítulos conforme o assunto abordado.

O capítulo 1, a Introdução, apresenta uma primeira abordagem do tema da dissertação, nomeadamente o seu enquadramento, a apresentação do problema e a sua relevância, a descrição dos objetivos pretendidos e contribuição destes e a estrutura da dissertação.

Já no capítulo 2, apresenta-se o estado de arte, onde são analisadas algumas tecnologias sobre sistemas de automatização de torres sineiras e controlo de iluminação.

No capítulo 3 é apresentado o projeto do dispositivo eletrónico proposto para o problema em causa.

O capítulo 4 é exposto o procedimento de montagem da placa e o teste do dispositivo na torre sineira.

No capítulo 5 são avaliados os objetivos para esta dissertação, deixando algumas propostas para eventuais melhorias no futuro.

2. Estado da Arte

A arte de bater o sino é bastante antiga, remontando aos primórdios da civilização. Durante muitos séculos, os sinos foram tocados à mão pelo sineiro ou através de meios mecânicos. A partir do século XX foram introduzidos elementos tecnológicos, de modo que estes continuassem a tanger, à medida que as torres sineiras começaram a cair em esquecimento [2].

Na secção 2.1 deste capítulo, são abordadas algumas tecnologias já existentes na área das torres sineiras para o toque de sinos, onde são evidenciadas as suas principais características, vantagens e desvantagens. Já na secção 2.2 são descritas algumas tecnologias utilizadas para controlo de iluminação, apresentando, uma vez mais, os prós e contras da utilização de cada uma da tecnologia. Por fim, na secção 2.3 deste capítulo é realizada uma nota conclusiva.

2.1. Sistemas sineiros Como se sabe, existem várias formas de tocar um sino, utilizando meios mecânicos ou eletromecânicos, fazendo com que o sino seja balançado ou que este seja batido por um martelo elétrico. Existem também formas de simular um conjunto de sinos, utilizando meios eletrónicos. Neste capítulo encontram-se expostos alguns sistemas existentes nesta arte milenar.

2.1.1. Mecanismos mecânicos de torres sineiras

Antes de ser inventada a eletricidade, já havia a necessidade de automatizar as torres sineiras. Grande parte dos conjuntos sineiros hoje existentes têm como princípio original, meios mecânicos. Um desses exemplos são os carrilhões do Palácio Nacional de Mafra. Através do livro “Carrilhões de Mafra” [3], onde são retratadas as intervenções de conservação e restauro do conjunto sineiro levadas até ao ano 2020, é possível verificar o grau de complexidade e a dimensão que estes conjuntos poderiam vir a ter.

O complexo sineiro do Palácio de Mafra é formado por dois carrilhões, dos maiores do mundo, fazendo um conjunto total de 119 sinos. Cada carrilhão pode ser tocado de forma manual, através de um teclado, existente em cada uma das torres, ou de forma automática, através do mecanismo automático [3].

O mecanismo automático encontra-se ligado a dois enormes cilindros do relógio, usando o mesmo princípio de funcionamento de uma caixa de música, isto é, através do ajuste de cavilhas/pinos nos tambores, é possível acionar automaticamente uma melodia, após cada toque de horas. Esta “caixa de música” é ligada aos sinos, através de cabos de aço que atravessam todos o complexo, ligados aos martelos dos sinos (figura 5.b). É importante referir que o mecanismo automático é acionado pela máquina do relógio. A máquina do relógio é constituída por um conjunto de roldanas e pesos. Os pesos são das peças fundamentais para que o relógio funcione, visto que é por este meio que é possível provocar o movimento das roldanas, precisamente calibradas, para que o relógio funcione. De tempos a tempos, neste tipo de mecanismo, é necessário “dar à corda” para o relógio funcionar, pelo que por vezes são colocados motores para que, quando o peso maior chega ao fim do curso, o motor seja acionado e coloque o peso no início de curso. No palácio Nacional de Mafra também é possível verificar que cada mecanismo de hora é diferente, isto é, a torre norte tem adotada uma hora litúrgica, de 6 horas, e a torre sul tem hora adotada legalmente, a de 24 horas. É importante referir que o mecanismo de hora também se encontra ligado a um mostrador exterior [3].

O mecanismo manual, no Palácio Nacional de Mafra, é constituído por um teclado e um sistema de transmissão entre o teclado e os badalos dos sinos. O sistema de transmissão também é feito por via de cabos até ao badalo (figura 5.a). Por vezes, nestes mecanismos manuais, é considerada a substituição dos badalos de maiores dimensões e pesados por uns mais leves, pois estes oferecem uma maior resistência ao toque que os equivalentes atuais.

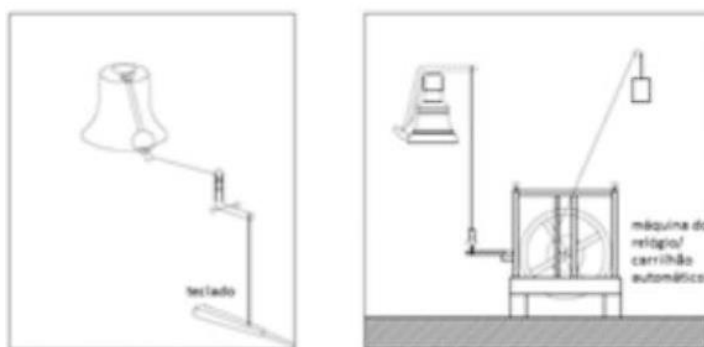


Figura 5- Princípio de Funcionamento do Carrilhão a) Manual b) Automático.

O Palácio Nacional de Mafra também dispõe de um conjunto de sinos de bamboar, num total de 11, sendo necessário serem puxados por duas ou três pessoas cada sino.

Outro mecanismo de relógio, onde é possível verificar a complexidade destes mecanismos é o que está instalado em Tillman Hall. Este relógio encontra-se ligado a 4 mostradores com cerca de 2,5

m de diâmetro e tem dois pesos de 45,4 kg e 124,7 kg. O mecanismo encontra-se dividido em 2 sistemas: o mecanismo de configuração e o mecanismo de toque [4]. Na Figura 6 é apresentado este mecanismo.



Figura 6 – Mecanismo instalado em Tillman Hall [4].

Através da análise do mecanismo de configuração apresentado na Figura 7, verifica-se que este é composto por uma engrenagem (Figura 7, posição A1) de grandes dimensões, que se encontra ligada ao sistema de pesos e é a engrenagem responsável pelo movimento do mecanismo do relógio. Esta engrenagem A1 está conectada a uma engrenagem mais pequena no eixo B. O movimento do eixo B é responsável também pelo movimento do pêndulo V. É também neste eixo onde é feita a transmissão para o movimento dos mostradores. A engrenagem 5 do eixo B é responsável por acionar o mecanismo de toque. O eixo C, D e E são responsáveis pelo movimento do pêndulo V [4].

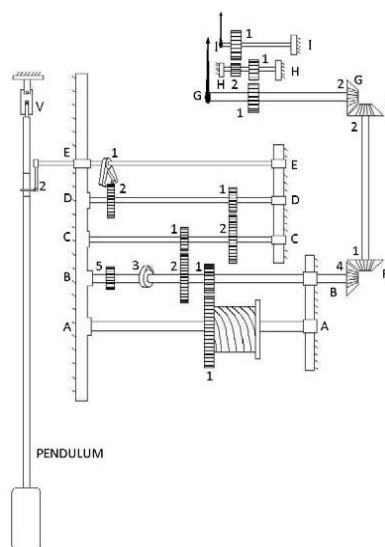


Figura 7 – Mecanismo de Relógio instalado em Tillman Hall [4].

Já na Figura 8, apresenta-se o mecanismo de toque. Neste mecanismo também se encontra um segundo tambor (N1), com o segundo peso, que faz rodar o eixo N. É no eixo que também é acionado o martelo Q. O eixo O é utilizado para “dar corda” ao mecanismo, movimentando as engrenagens O1 e N2. Os eixos M, L, K e J são responsáveis pela transmissão de quando o eixo N deve rodar. Esta transmissão é feita através do eixo B. É importante referir que o eixo J é responsável pela transmissão do número de toques, dependendo das horas [4].

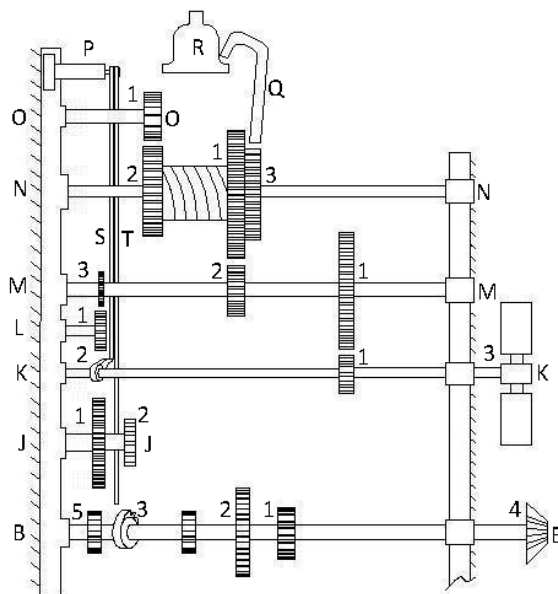


Figura 8 - Mecanismo de toque de horas [4]

Os Carrilhões de Mafra são um exemplo perfeito de como os mecanismos sineiros, antes do aparecimento de acionamentos elétricos, poderiam ser de grandes dimensões e com um grau de complexidade bastante elevada. Outro grande problema existente nestes grandes mecanismos é precisamente a necessidade de “dar à corda” ao mecanismo do relógio, embora no caso do Palácio Nacional de Mafra tenha sido resolvido. Existe uma outra desvantagem no mecanismo automático deste tipo de carrilhões, isto é, a necessidade de alteração das cavilhas, se possível, nos grandes cilindros deste mecanismo, sempre que se deseja alterar uma melodia, impedindo assim também o toque de mais de que uma melodia.

2.1.2. Sistemas eletromecânicos ou eletrónicos

Os sistemas eletromecânicos são os dispositivos mais utilizados nas torres sineiras, pois permitem a substituição do mecanismo de relógio, permitindo assim a redução dos volumes necessários para a colocação deste. Estes sistemas são compostos por um controlador eletrónico que aciona

os mecanismos eletromecânicos, como por exemplo, os martelos de picar ou bamboar para o sino ou os motores para os mostradores do relógio.

Os sistemas eletrônicos, tal como os sistemas eletromecânicos, também permitem a redução do volume necessário ao mecanismo do relógio. São sistemas mais baratos porque, em vez de utilizar os sinos reais, utilizam colunas para simular o toque dos sinos, podendo reproduzir um maior conjunto de sinos que os dispositivos eletromecânicos ou mecânicos.

Exactus

O Exactus [5] é um sistema computadorizado de toque de sinos desenvolvido pela Fundação de Sinos de Braga - Serafim da Silva Jerónimo & filhos, LDA, a única empresa deste ramo em ainda em funcionamento em Portugal, e a empresa Lincis.

De acordo com o fabricante, a sua interface é amigável, através da utilização de um display e um teclado. Este sistema permite o controlo de sinos, mostradores, iluminação ou até mesmo da ventilação e permite a utilização de simulação de sinos, através da utilização de amplificadores e colunas externos. Aceita vários protocolos, tais como RS485, Can, I²C, MIDI e Wireless. Apresenta também a possibilidade de programação de toque para cada registo, seja este entre datas, tempo do ano ou festividades. A sincronização do fuso horário é automática, podendo ser feita por um dispositivo GPS. A comunicação com o dispositivo é feita através de um comando ou através de um envio de SMS, fazendo a execução de uma melodia. Outra característica bastante interessante consiste na imitação dos sinos de bamboar, que é bastante útil para quem não tem sinos capazes de bamboar.

Apesar de estas características bastante interessantes, o dispositivo apresenta como desvantagem a falta de uma interface que permita ao utilizador a programação do sistema sem a necessidade de estar junto do dispositivo. Na Figura 9 é apresentada a interface do sistema Exatus.



Figura 9 – Exactus [5].

Sigma Master Clocks

O dispositivo Sigma, é desenvolvido pelo fabricante Bodet. Este dispositivo está disponível em 4 versões, onde a versão mais básica apenas transmite a hora e a versão mais completa permite, adicionalmente, o controlo de relés, conjuntos de sinos e microfones e a sincronização do computador na rede. Este sistema pode assumir a formato de Rack ou pode ser colocado na parede [6], conforme apresentado na Figura 10.



Figura 10 - Formatos Sigma Master Clock [6].

O Sigma H Master Clock é a versão mais simples deste dispositivo. Esta versão apenas possui como características um master clock, um oscilador, que pode ser sincronizado por Ambient Light Sensor (ALS), Deutsche langwelliges Signal Frankfurt (DCF) ou GPS, um relé para alarme, uma saída para mostrador de relógio e comutação automática de horário de verão e inverno e proteção para curto circuitos. O dispositivo, tanto o formato de parede ou de rack, possui também indicadores LED e um ecrã de LCD onde são apresentadas as horas, minutos, segundos e datas. É importante referir que este sistema não tem possibilidade de programação por computador [7]. O esquema de funcionamento do Sigma H Master Clock é apresentado na Figura 11.

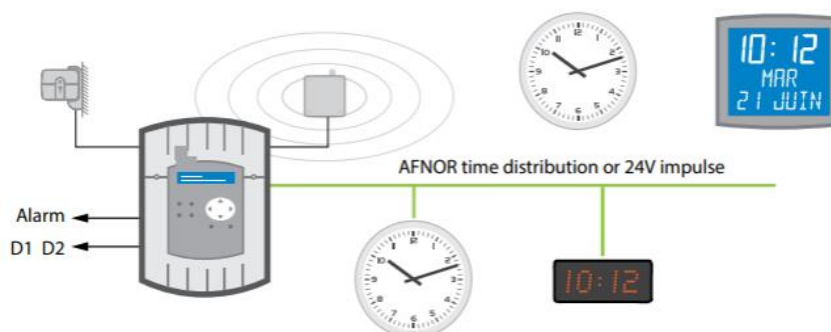


Figura 11 - Esquema de Funcionamento do Sigma H Master Clock [7].

Já o Sigma P Master Clock, para além das características apresentadas do Sigma H, possui 3 relés AC para ativação de sinos ou outros aparelhos que necessitem de controlo. Estas saídas também podem ser programadas para ativar em períodos específicos e uma USB [8]. O esquema de funcionamento do Sigma P Master Clock é apresentado na Figura 12.

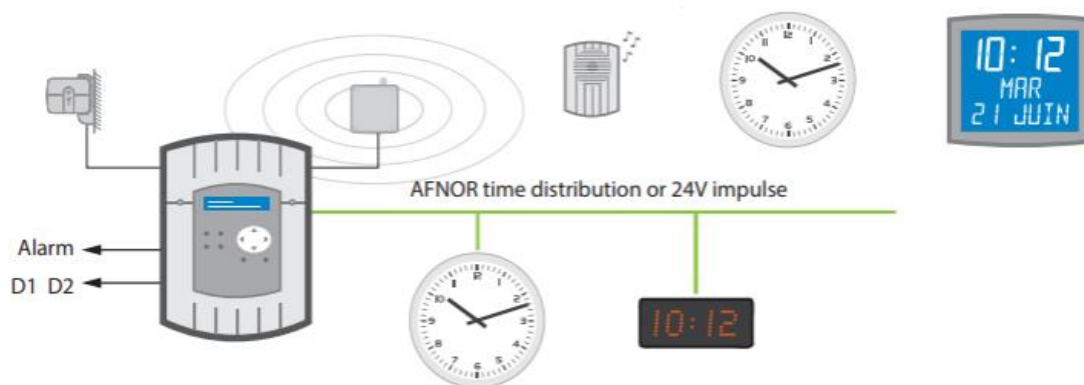


Figura 12 - Esquema de Funcionamento do Sigma P Master Clock [8].

O Sigma C Master Clock, para além das características já mencionadas, embora não possuindo uma saída direta para mostradores exteriores, como nas versões anteriores desta série, permite a adição de uma rede de relógios, relés e sirenes, através da ligação de um distribuidor de protocolo NTP a uma porta RJ45. Este dispositivo possui uma interface que pode ser controlada por um computador [9]. O esquema de funcionamento do Sigma H Master Clock apresenta-se na Figura 13.

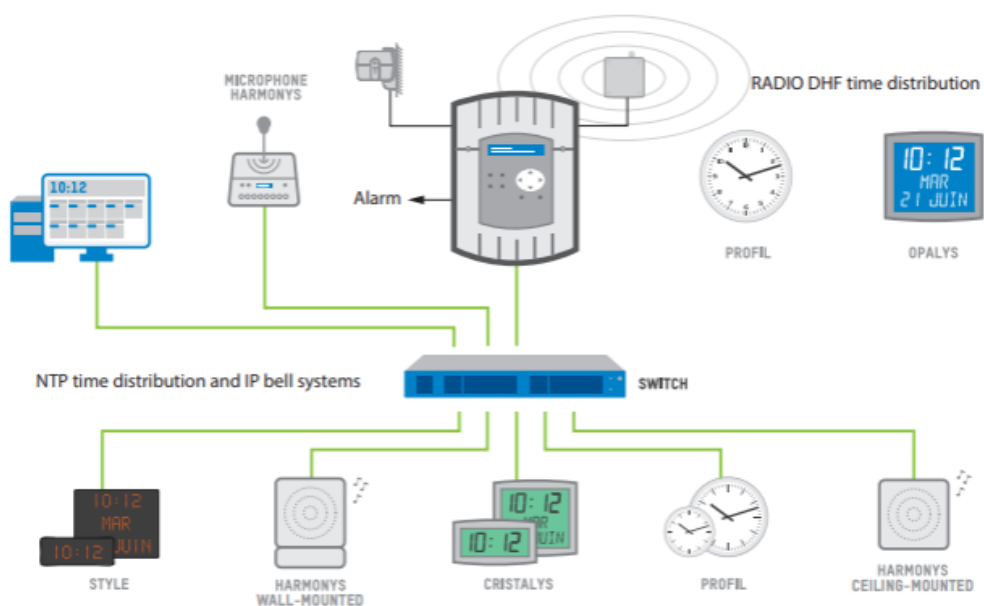


Figura 13 - Esquema de Funcionamento do Sigma C Master Clock [9].

Por fim, o Sigma Mod Master Clock é bastante semelhante ao Sigma C Master Clock, no entanto possui uma saída para mostradores exteriores, conforme indicado na Figura 14 [10].

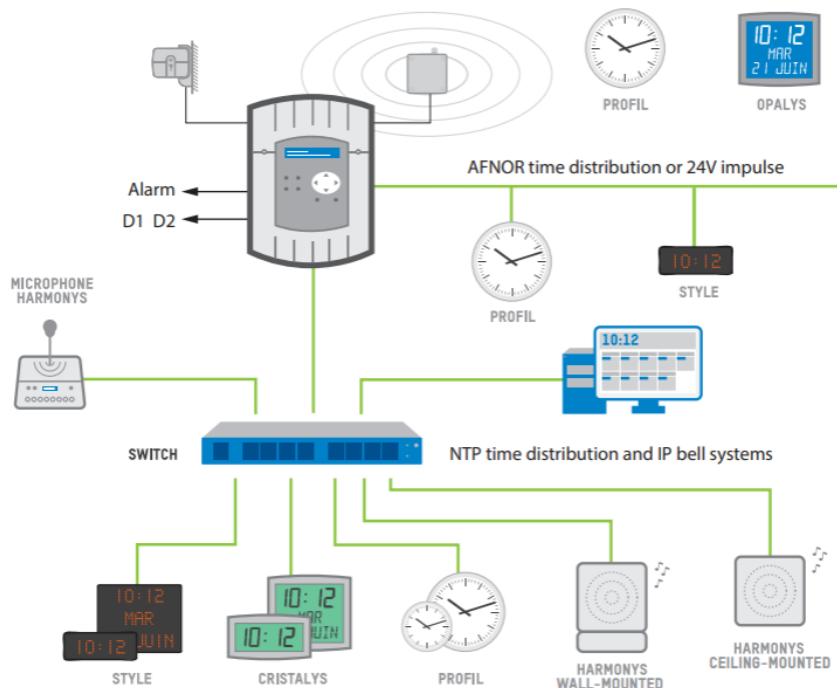


Figura 14 - Esquema de Funcionamento do Sigma Mod Master Clock [10].

O software, representado na Figura 15, dependendo da versão, permite a configuração e controlo dos vários dispositivos conectados via IP, a programação melodias ou toques, via USB ou IP, ou a transmissão direta de áudio diretamente do computador para os altifalantes instalados [11]. A interface é bastante simples para o utilizador, fazendo com que este tenha o controlo total do sistema. Outra característica bastante interessante deste dispositivo é a capacidade de dispositivos que este consegue controlar. Apesar do Sigma H ser um dispositivo bastante rudimentar, permitindo apenas uma saída de relógio, o Sigma Mod é bastante completo, podendo controlar uma quantidade bastante elevada de dispositivos, o que torna um dispositivo interessante de utilizar se for necessário o controlo de um carrilhão como existe no complexo sineiro do Palácio de Mafra.



Figura 15 - Software Sigma Master Clock [11].

MicroQuartz S

O MicroQuartz [12] é um dispositivo descontinuado da Bodet. Este permite o controlo de 4 saídas de acordo com o modo semanal, anual ou período e a conexão de um ou mais mostradores. Este sistema também permite a conexão com um computador através da porta ASCII RS 232. A hora do dispositivo é alterada automaticamente para a hora de Verão ou Inverno, através de programação ou via rádio. A interface do dispositivo é bastante simples, tendo apenas um ecrã de duas linhas de 16 caracteres, um teclado numérico e de funções, LEDs de indicação de utilização de relés e de alimentação e conexão via rádio, conforme apresentado na Figura 16.



Figura 16 – MicroQuartz S [12].

Este dispositivo é bastante simples, sendo programado na totalidade no próprio dispositivo, podendo a programação ser bastante difícil devido a uma interface bastante limitado e ecrã de reduzidas dimensões. A questão da necessidade de haver fácil acesso deste equipamento para programação e acionamento manual dos relés é outro contra deste sistema.

Wireless Bell Control System Based on Microcontroller

Um dos sistemas que pode ser utilizado para controlo de sinos é o dispositivo desenvolvido por Zou et al. [13]. Este sistema sem fios destinou-se ao sistema de toque da Campus da Universidade, de modo a que seja evitada a utilização de cabos entre o dispositivo final e a campainha ou conjunto de sinos. Embora não seja idealizado para torres sineiras, o dispositivo de controlo pode ser facilmente adaptado para estas.

O dispositivo de controlo é formado por um microcontrolador AT89S52, um circuito de relógio, um circuito de armazenamento, um teclado e um circuito de transmissão de dados, que transmite para um circuito recetor de dados sem fios, que envia estes dados para o circuito de toque. As informações, tais como as horas, a data, o dia da semana e o modo de toque, encontram-se no ecrã LCD. Na Figura 17 encontra-se o esquema funcional do hardware do dispositivo.

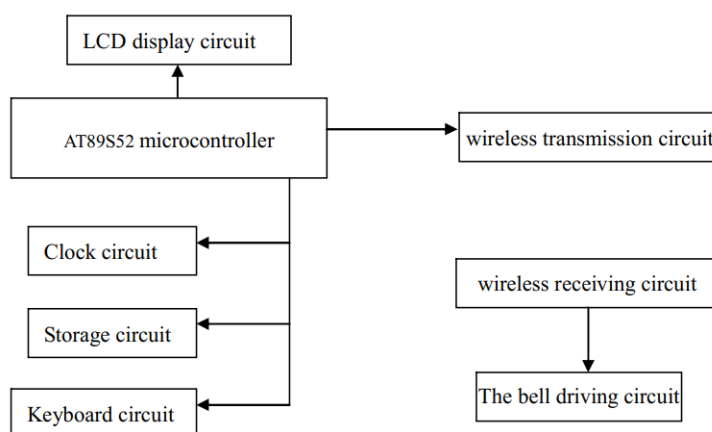


Figura 17 – Esquema de Estrutura de Hardware do Dispositivo [13].

O AT89S52 é um controlador de baixo consumo com 8 kB de memória flash e 256 x 8 bit de RAM. É responsável por todo o controlo do sistema. Na Figura 18 é apresentado o pinout do microcontrolador.

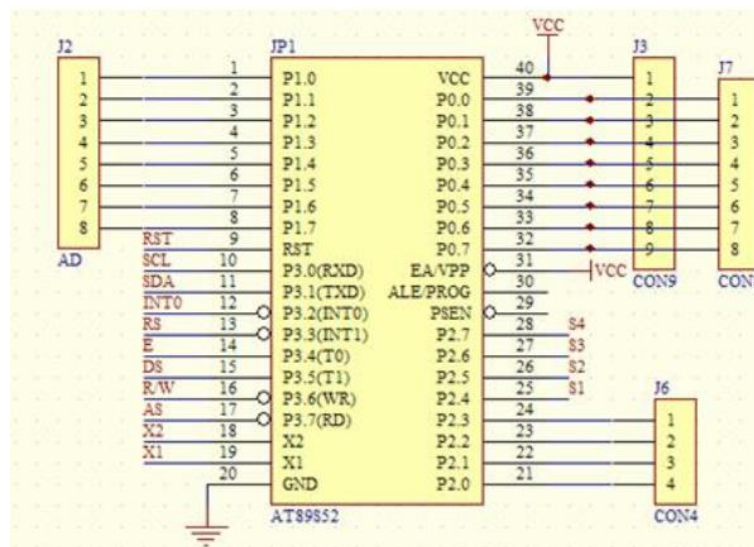


Figura 18 – AT89S52 [13].

O circuito de relógio, ocupando as portas 1 a 8 do microcontrolador, tem como objetivo fornecer ao sistema uma base de tempo, produzindo a hora exata, os dias, as semanas, os meses, os anos e dias da semana. Para isto, é utilizado um circuito integrado DS12C887 (ver Figura 19), de acordo com o circuito da Figura 28. Este circuito também tem uma bateria de lítio integrada, permitindo que o sistema possa ficar sem energia durante 10 anos sem perder os dados.

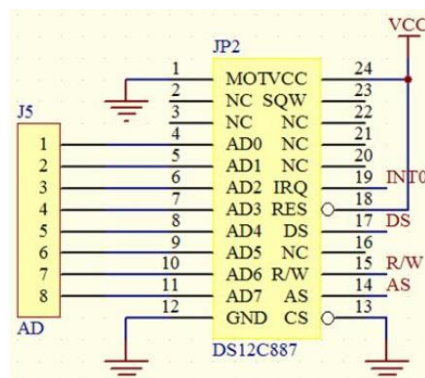


Figura 19 - DS12C887 [13].

Já o circuito de teclado é bastante simples, utilizando apenas 4 botões, tendo como objetivo a marcação das horas e horário.

O circuito de transmissão apresentado na Figura 20 é composto por um circuito integrado de usos gerais para codificação e descodificação de baixo consumo, PT2262. Este circuito envia um sinal,

via wireless, onde contém o endereçamento, as informações e sincronização. Esta informação é enviada para o para o recetor, que verifica o endereçamento duas vezes.

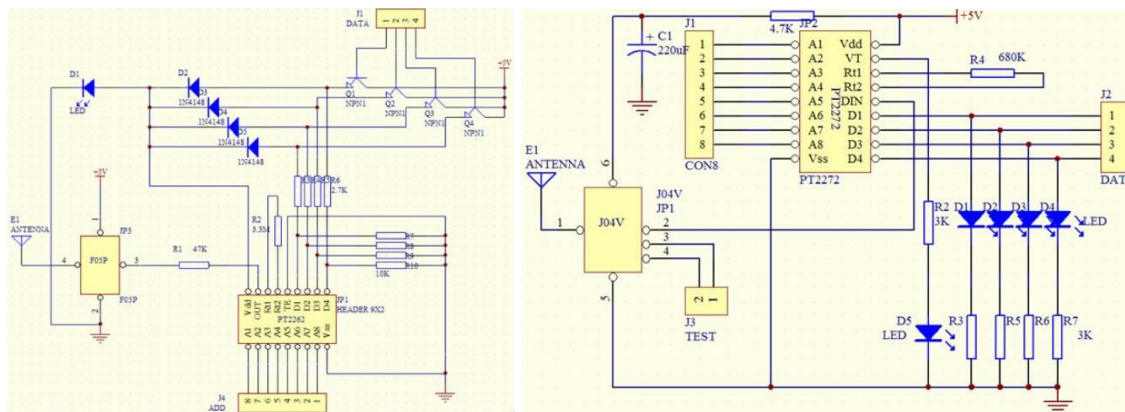


Figura 20 – Circuito de Transmissão, à esquerda, e Circuito de Recepção, à Direita [13].

Por fim, o circuito de memória é composto por uma EEPROM CMOS AT24Co4. Esta EEPROM é responsável por armazenar o horário para as diferentes estações e para os diversos dias.

Apesar do sistema ser bastante simples e de baixo custo, apresenta como problema conectividade, isto é, para o utilizador poder alterar as definições como seja alterar o horário, é necessário que o utilizador se encontre junto deste. Outro problema existente é relativo à interface utilizada. Com possíveis atualizações ao controlador, com a adição de novas funções e menus, a interface ficará obsoleta, comprometendo assim a facilidade com que o utilizador consegue programar as várias funções.

College Electric Bell System with Time Display

Outro dispositivo que não tem como finalidade inicial a sua instalação em torres sineiras, mas facilmente pode ser adaptado é o dispositivo desenvolvido por Megalingam et al. [14]. Este dispositivo tem como finalidade automatizar o sistema de toques de escolas. O seu esquema de funcionamento encontra-se exposto na Figura 21.

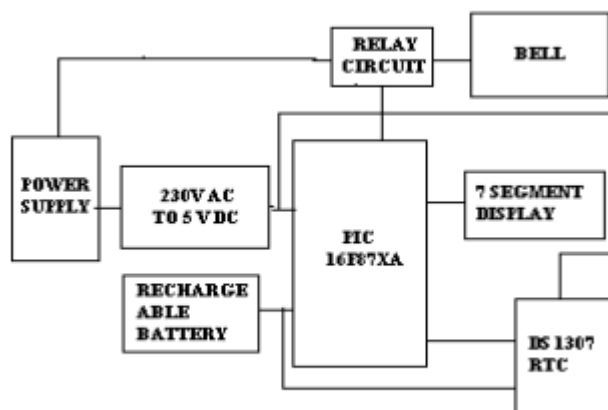


Figura 21 – Diagrama de funcionamento do dispositivo [14].

Como é possível observar na Figura 21, o dispositivo é constituído por um microcontrolador PIC16F877A, um ecrã, uma fonte de alimentação, uma bateria, um circuito para relés, onde é conectada a campainha, e um circuito de relógio RTC DS1307. O microcontrolador é responsável pelo controlo de toque da campainha, contendo os horários na sua programação. Quando a hora que se encontra no RTC DS1307 é igual ao que se encontra na programação do microcontrolador, é acionado o circuito do relé, tocando a campainha. As horas também podem ser visualizadas no ecrã. Na Figura 22 encontra-se o esquema de circuito do dispositivo.

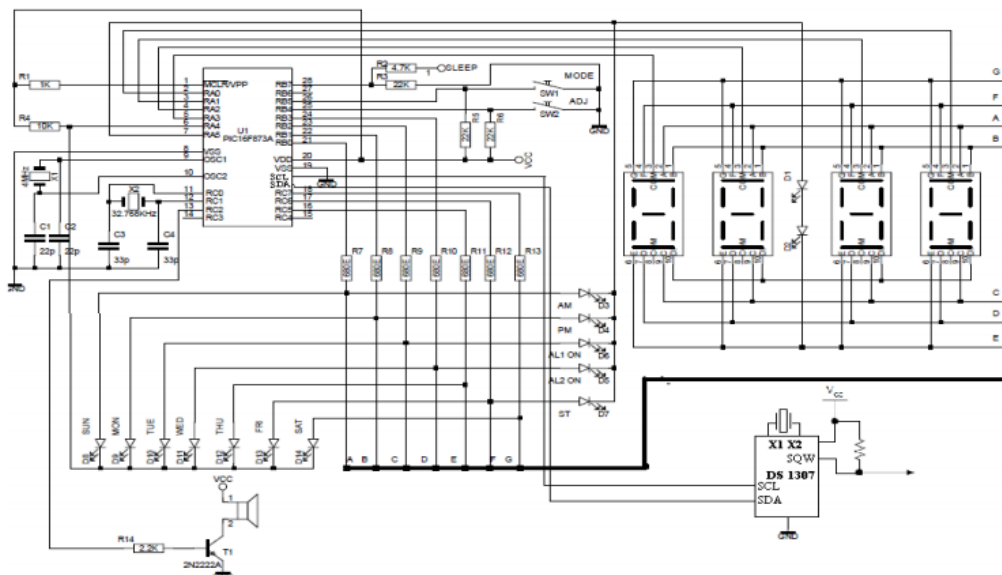


Figura 22- Esquema de circuito do sistema [14].

Este dispositivo, embora rudimentar, é um dispositivo simples e fácil de implementar. Outra vantagem, ao contrário de outros dispositivos, é que este dispõe de um modo de SLEEP. Este modo permite que o dispositivo entre num modo de baixo consumo, fazendo com que a bateria integrada dure mais tempo em caso de falta de energia. Apesar disto, a falta de uma interface que permita ao utilizador alterar os horários predefinidos é um contra deste dispositivo. Outro aspeto a considerar seria a colocação de um botão para permitir o toque manual da campainha.

Transportable Bell Arrangement

Mouakkassa [15] desenvolveu um sistema sineiro com o objetivo de tornar um conjunto de sinos transportáveis para qualquer tipo de eventos, não sendo necessário um conjunto de sinos na torre.

Os sinos, neste sistema, são colocados numa plataforma móvel, prontos a tocar. A plataforma móvel é composta, para além dos sinos, por um controlador, um conjunto de badalos, uma fonte de alimentação, um sistema de comunicações e um cartão. Esta plataforma encontra-se ligada a uma central, onde é feita toda a programação para os cartões que plataforma móvel lê, assim como a faturação desta plataforma móvel. O sistema de comunicações é utilizado para o caso de haver alterações da agenda. O controlador recebe toda a informação das melodias e transmite para o sistema de badalos. O controlador também tem melodias predefinidas. Na Figura 23 é apresentado um esquema de funcionamento do sistema.

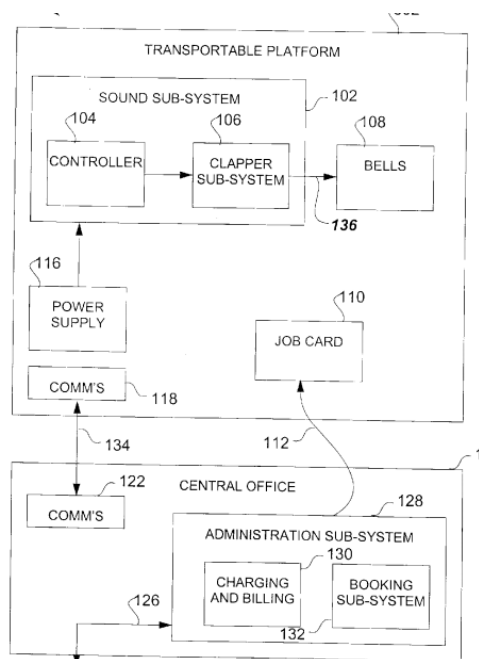


Figura 23- Esquema do sistema [15].

Este sistema é bastante interessante pois permite levar o conjunto sineiro para várias localizações, como por exemplo, um casamento a beira-mar, não necessitando assim de realizar os eventos junto a um complexo sineiro, embora todas as comunicações pudessem ser feitas pelo sistema de comunicações, evitando assim o uso de um cartão.

Electronic Carillon system and Sequencer module Therefor

A partir do século XX começaram a ser introduzidos elementos tecnológicos que, por vezes, acabam por substituir os sinos fisicamente. Um desses sistemas é o dispositivo desenvolvido por Schwartz e Frick [16], que tem como finalidade de simular o toque de um conjunto de sinos. A interface do sistema é apresentada na Figura 24.

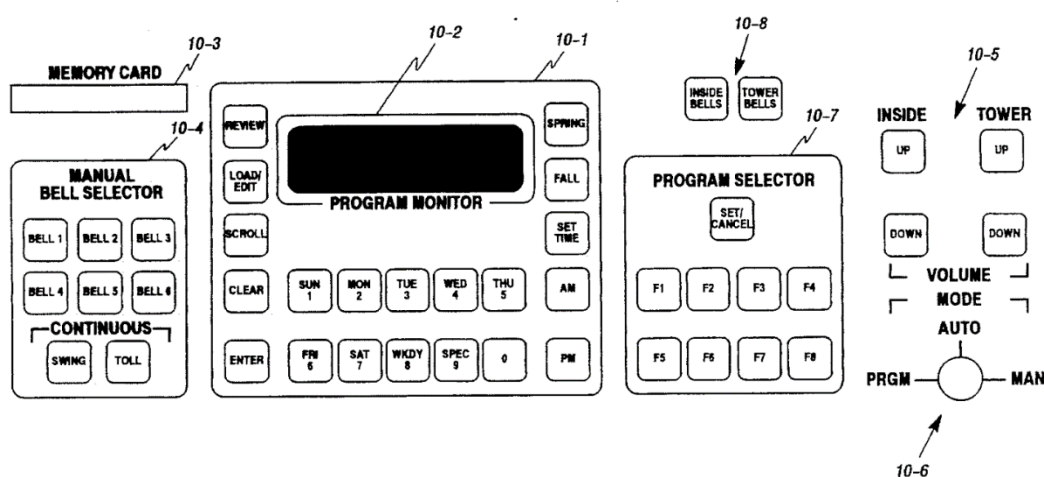


Figura 24 – Interface do dispositivo.

Este sistema é composto por uma unidade de controlo e um altifalante. O altifalante é colocado na zona onde se pretende que o carrilhão seja ouvido. A unidade de controlo é composta por um módulo de relógio/sequência, um módulo para gerar tons, uma placa para distribuição de energia e um conjunto de amplificadores. Na Figura 25 é apresentado um esquema de funcionamento do dispositivo.

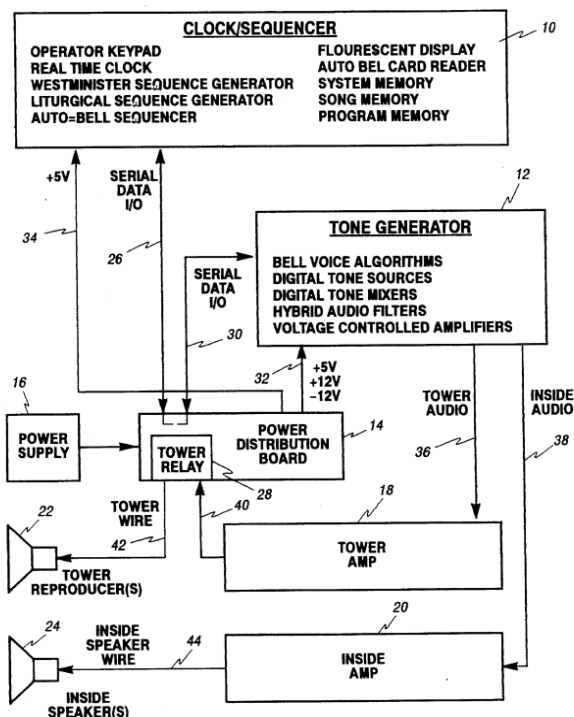


Figura 25 – Esquema do dispositivo.

Como é possível analisar na figura 25, o módulo de Relógio/Sequenciador encontra-se ligado à placa de distribuição de energia através de uma porta série. Este módulo é a unidade central do sistema, encontrando-se aqui todos os dados de sequências de toques e é o relógio de sistema. Este módulo também tem acoplado um ecrã fluorescente, um teclado, um botão de chave, uma memória de músicas e um cartão de leitura. Quando requerido, os dados são enviados pela porta série até ao módulo gerador de tons. Este módulo também se encontra ligado via comunicação série à placa de distribuição de energia. O gerador de tons envia então o sinal gerado para dois amplificadores, um que está ligado a um altifalante que se encontra na torre e outro que se encontra junto da unidade central, para o utilizador poder ouvir o que está a ser reproduzido ou para programar o módulo central. O relé existente na placa de distribuição de energia, para além de permitir o tom no altifalante, tem como finalidade impedir que o tom seja enviado fora do tempo.

Este dispositivo tem como vantagens a sua facilidade de programação e a grande variedade de funções integradas (tais como a execução de melodias predefinidas, programação de novas melodias, toque manual) e grande flexibilidade no tipo sino a sintetizar ou a reproduzir. Apesar destas vantagens, em especial a última, devido ao facto deste sistema não possuir o sino fisicamente, não ficando assim limitado ao tom único deste, o sistema apresenta algumas desvantagens. Uma desvantagem é a sua falta de conectividade, isto é, para executar qualquer

tipo de programação ou toque manual da melodia, é necessário que o utilizador se encontre junto do dispositivo, pelo que o dispositivo terá de se encontrar em locais bastante acessíveis. Outra possível desvantagem é fiabilidade do tom gerado pelo altifalante em relação ao tom original de um determinado sino. Este problema poderá ser derivado a vários fatores tais como a forma de como são gerados os tons e a qualidade e tipo de altifalante instalado na torre sineira.

2.2. Dispositivos de Controlo de Iluminação

Existem vários dispositivos para controlo de iluminação, tendo estes como objetivo ligar ou desligar iluminação, seja de que tecnologia for.

Interruptores Horários

Um dos dispositivos mais utilizados na área são os interruptores horários. Estes dispositivos podem ser instalados diretamente nos quadros elétricos ou nas tomadas elétricas e podem ser analógicos ou digitais. Um exemplo de interruptor horário analógico é Interruptor Horário de 1 Canal Diário Analógico fornecido pela EFAPEL, apresentado na Figura 26. Este interruptor permite controlar uma instalação em função de uma programação diária, com um tempo mínimo entre manobras de 15 minutos. [17]



Figura 26 – Interruptor Horário de 1 Canal da EFAPEL.

Apesar de os interruptores horários analógicos permitirem que a iluminação ligue e desligue automaticamente, em períodos facilmente definidos pelo utilizador, este tipo de interruptores são bastante rudimentares, permitindo apenas um número limitado de manobras a cada 15 minutos. Outra desvantagem é a incapacidade da mudança automática de hora, sendo necessário que utilizador altere a hora do dispositivo, sempre que necessário. O facto de serem relógios diários, impossibilita que a iluminação seja configurada para cada dia da semana, impedindo uma maior adaptação às necessidades do utilizador. Já os interruptores horários digitais resolvem, em parte,

os problemas dos interruptores horários analógicos. Um exemplo deste tipo de interruptores o Interruptor Horário de 1 Canal Diário/Semanal digital, produzido pela EFAPEL. Estes permitem o controlo da instalação em função programação diário e/ou semanal, mudam automaticamente a hora. Também possibilitam a programação de período de inatividade [17].

No entanto, estes dispositivos, em comparação a outros sistemas, não permitem um acionamento automático, fazendo com que estes dispositivos possam ativar a iluminação quando não há necessidade quando não há sol, por exemplo.

Interruptores crepusculares

Ao contrário dos interruptores horários, os interruptores crepusculares são acionados apenas quando o ambiente se torna demasiado escuro.

Existem vários modelos de interruptores crepusculares. Um destes modelos é o modelo EE200 produzido pela HAGER, apresentado na Figura 27. O modelo é bastante simples, sendo também instalado em quadros elétricos, tendo apenas um conjunto de interruptores e potenciômetros para o controlo das gamas de luminosidade. Têm também um LED de indicação de funcionamento da saída. Estes dispositivos são ligados a uma sonda crepuscular, que permite obter dados de luminosidade para o sistema [18].



Figura 27 – Interruptor Crepuscular EE200 e sensores.

Tal como nos interruptores horários, este tipo de dispositivos apresenta como principal problema a falta de conectividade para controlar o dispositivo remotamente, sendo necessário que o utilizador esteja junto do dispositivo. No entanto, ao contrário dos interruptores horários, este

dispositivo só ativa a saída quando a luminosidade do espaço se encontra abaixo de um determinado valor.

Philips Dynalite

O dispositivo Philips Dynalite apresentado na Figura 28 é basicamente um autómato lógico programável (PLC) que pode proporcionar corrente elétrica indutiva até 20 A por saída. Este dispositivo permite que cada saída seja controlada através de uma rede cablada ou por DMX e permite a funcionalidade de diagnóstico. Tem como núcleo, controladores lógicos e indicadores LED de diagnóstico para cada saída [19].



Figura 28 - Philips Dynalite.

Embora a interface com utilizador seja bastante simples, podendo utilizar vários softwares de controlo de DMX, a falta de automatismo, com base em sensores é inexistente, sendo necessário a utilização de um dispositivo de controlo externo, utilizando DMX por exemplo, para fazer este automatismo com base em sensores.

2.3. Nota conclusiva

Como é possível observar, neste capítulo, existem vários dispositivos que permitem o controlo do relógio, sinos e iluminação.

Nos dispositivos de controlo de relógios e sinos com recurso a utilização mecânica, ou seja, os dispositivos mecânicos, é importante salientar que, durante muitos séculos, foram os dispositivos predominantes para o controlo de hora. Sendo dispositivos mecânicos, estes são de grandes dimensões e necessitam de uma maior manutenção e cuidados, já para não falar da necessidade

de “dar à corda” com alguma regularidade. Com o aparecimento dos dispositivos eletromecânicos e eletrônicos, os dispositivos mecânicos foram postos de parte, grande parte devido à sua necessidade de manutenção e cuidados, para além das suas dimensões. Os dispositivos eletromecânicos e eletrônicos permitiram que o sistema geral fosse mais compacto e, no caso do dispositivo eletrônico, pudessem simular uma maior quantidade de sinos, produzindo assim mais melodias e permitindo uma maior precisão de horas e toques. É possível verificar o nível de complexidade que estes dispositivos chegaram, sendo o dispositivo “Exactus”, fabricado pela Fundação de Braga, e o dispositivo “Sigma Master Clock”, fabricado pela Bodet, a merecer um lugar de destaque. No entanto, os dois apresentam falhas. No caso do dispositivo “Exactus”, apesar de poder ativar várias funções, tais como iluminação e ventilação, para além de controlar os sinos e mostradores, a interface com o utilizador acaba por ser mais pobre em comparação ao dispositivo “Sigma Master Clock”, dificultando as alterações das definições do aparelho. Já o dispositivo “Sigma Master Clock”, apesar de possuir software que permite a programação do relógio, a configuração das várias melodias, iluminação e ventilação, o dispositivo em si, precisa de estar conectado a um servidor NTP, específico da marca. Assim, para além de muito mais dispendioso que o dispositivo “Exactus”, a infraestrutura necessária acaba por ser muito maior e complexa.

Quanto aos dispositivos de iluminação, os dispositivos mais utilizados para esta finalidade são os interruptores horários ou os interruptores de crepusculares. São dispositivos bastante simples e fiáveis, no entanto, a falta de conectividade para uma ativação manual ou agendada são os contras deste tipo de dispositivos, sendo necessário que o utilizador se desloque junto dos mesmo para executar uma ativação manual. Os dispositivos anteriormente referidos, o “Exactus” e “Sigma Master Clock”, também têm estas funções integradas, pelo que, para o caso em estudo, seria uma mais-valia, apesar dos contras que estes têm já referidos.

3. Materiais e Métodos

Neste capítulo, é apresentado o projeto do dispositivo eletrônico proposto para o problema em causa.

3.1. Dispositivos

3.1.1. Escolha do controlador

Antes do desenho do dispositivo final, foi necessário escolher o controlador a utilizar. Como se pretende que o dispositivo seja possível controlar remotamente e que a sua interface com o utilizador seja a mais fácil possível é necessário que o dispositivo contenha o máximo de memória disponível possível para poder armazenar todos os dados que estas necessidades acarretam. Outro aspeto importante é o número de portas disponíveis do controlador. A conectividade integrada também é também um fator que pesa na escolha. É importante salientar também que todos os controladores considerados são de baixo de custo, pelo que a escolha do controlador com base no custo é pouco relevante. Na Tabela 1 são apresentadas algumas especificações técnicas dos três dispositivos considerados.

Tabela 1 – Especificações técnicas de 3 controladores considerados.

	Tiva TM4C123GH6PM [20]	Esp8266 [21]	Raspberry Pi Zero W [22]
Tipo	Microcontrolador	Microcontrolador	Minicomputador
Número de Portas (Outputs e Inputs)	Até 43	17	26
Placa Wireless Embutida	Não	Sim	Sim
Memória RAM Disponível	32KB	<50KB	512MB
Memória Flash Disponível	256KB	16MB	Cartão SD
Processador	Arm Cortex M4F	Tensilica L106	Broadcom Single-Core
Velocidade do Processador	80MHz	160MHz	1GHz

Apesar da placa de desenvolvimento Tiva TM4C123GH6PM ter uma grande quantidade de portas disponíveis, a falta de rede sem fios embutido, baixa RAM e memória flash e processador de baixa frequência são alguns requisitos contra esta solução. O Esp8266 é um microcontrolador excelente, sendo programado pelo mesmo *Integrated Development Environment* (IDE) que o Arduino. No entanto, o Raspberry Pi Zero W apresenta uma maior memória RAM disponível e número de portas mais que suficientes para acionar os vários periféricos, já para não falar do processador mais rápido. O Raspberry Pi Zero W necessita de um sistema operativo para poder correr programas, nomeadamente o Raspbian, tornando a configuração inicial do dispositivo mais complexa que os outros dispositivos. A utilização de Raspberry Pi Zero W abre a possibilidade de também utilizar as outras placas sem a necessidade de alterar portas no programa entre dispositivos pelo que, se houver esgotamento de capacidade do Raspberry Pi Zero, facilmente se fará a migração para a versão seguinte do Raspberry Pi.

Analisando as vantagens e desvantagens das várias placas optou-se, por fim, na utilização do Raspberry Pi Zero W.

3.1.2. Instalação do Sistema Operativo e das aplicações e configuração do Raspberry

Para instalar o sistema operativo do Raspberry é necessário um cartão de memória Secure Digital (SD), um computador, acesso à internet e a ferramenta fornecida pela Raspberry: Raspberry Pi Imager. Como é possível verificar na Figura 29, apenas é necessário escolher o sistema operativo a instalar no Raspberry Pi, escolher o cartão SD, previamente inserido no computador e carregar em write. Para este projeto decidiu-se a instalação do Raspberry Pi lite, pois, para o problema em causa, não é necessário um ambiente de Desktop. Concluída a escrita no cartão, o sistema encontra-se pronto a utilizar.

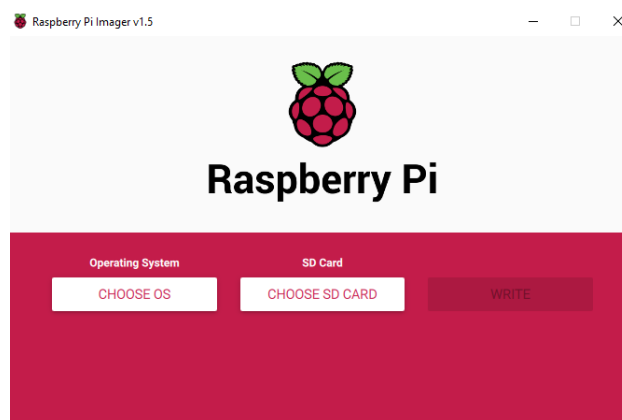


Figura 29- Raspberry Pi Imager.

No passo seguinte, é necessário ligar o Raspberry Pi à Rede a uma rede sem fios. Para isto, utilizou-se um monitor, um adaptador microHDMI, um teclado, e um adaptador tipo C e HDMI.

Quando é ligado Raspberry, após configuradas algumas definições, como a alteração da palavra-passe e nome de utilizador padrão.

Na linha de comandos é digitado o comando `>> sudo raspi-config`, abrindo um menu de configuração. Aqui seleciona-se a opção `System Options` e de seguida a opção `Wireless LAN`.

Configurada a rede sem fios do Raspberry, é necessário estabelecer uma ligação Secure Socket Shell (SSH), a fim de permitir a configuração do Raspberry remotamente, dispensando assim utilização dos periféricos nos testes seguintes. Para isto, basta voltar ao menu de configuração, seleccionar a opção `Interface Options` e ativar a opção `SSH`. Após este passo, já é possível a utilização do Raspberry a partir do computador. Para aceder ao Raspberry pelo computador, basta abrir a linha de comando do computador e digitar o comando `ssh Nome do dispositivo`. Após colocação das credenciais de acesso, têm-se acesso à linha de comandos do Raspberry Pi.

Ainda no Raspberry é necessário instalar a framework Django, que é um framework para desenvolvimento rápido para web, e o Apache, que é um servidor *Hypertext Transfer Protocol* (HTTP) [23]. A vantagem de utilizar a Framework Django é que, para além de fornecer uma base de dados integrada, esta é acompanhada por um layout pré-definido. O servidor Apache é necessário para disponibilizar a Framework Django no Web Browser sem a necessidade de ter um ambiente virtual constantemente ativo. Embora o Apache dispense a utilização de um ambiente virtual, para instalar a framework Django, é recomendado criar um ambiente virtual, a fim de evitar problemas com outros projetos ou sistema operativo.

Para instalar a framework Django é necessário, primeiramente, criar um diretório e criar um ambiente virtual, digitando o comando `>>virtualenv testenv`, onde `testenv` é o nome do ambiente virtual a criar. De seguida, ativa-se o ambiente virtual digitando o comando `source testenv/bin/activate`. Com o ambiente virtual ativo, procede-se à instalação da framework através do comando `pip3 install django`. A partir daqui, o procedimento para a Iluminação é o mesmo que o das horas, pelo que apenas tem que se repetir o processo referido abaixo outra vez para o controlo de Iluminação. De seguida digita-se o comando `django-admin.py startproject Horas.` para criar um projeto novo. O ponto final no fim do comando serve para dizer que o projeto `Horas` deve ser instalado no diretório atual. Com o projeto criado, é necessário proceder à alteração do ficheiro `settings.py`, criado pelo Django durante a criação do novo projeto, utilizando o comando `nano ~/teste/teste/settings.py`. Aberto o ficheiro, vai-se à secção `INSTALLED_APPS` e digita-se o ficheiro que se quer adicionar, neste caso o ficheiro `teste`, exemplificado na Figura 30.

```

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'Horas',
    'Iluminação',
]

```

Figura 30 – Introdução do nome ficheiro “Horas” e “Iluminação” na Secção “INSTALLED_APPS”.

No fim do mesmo ficheiro, digita-se também o comando “`STATIC_ROOT = os.path.join(BASE_DIR, "static/")`”, a fim de garantir que o `STATIC_ROOT` foi definido e guarda-se o ficheiro. De seguida, é necessário migrar as alterações para que a base de dados funcione. Para migrar as alterações, digita-se o comando “`./manage.py makemigrations`” e, de seguida o comando “`./manage.py migrate`”. O passo seguinte da instalação da framework, é criação de um utilizador. Para criar um utilizado basta digitar o comando “`./manage.py createsuperuser`”. Ao executar este comando, vai ser requerido ao utilizado para introduzir um nome de utilizador e uma palavra-passe. Para concluir a instalação, faz-se a configuração dos ficheiros estáticos utilizados pelo administrador, utilizando o comando “`./manage.py collectstatic`”. Agora é necessário instalar o servidor Apache. Para isso basta digitar o comando “`sudo apt-get install apache2 -y`”. Para conectar a framework ao servidor HTTP é necessário a instalar mais duas extensões do Apache2, obtidas através dos comandos “`sudo apt-get install apache2-dev -y`” e “`sudo apt-get install libapache2-mod-wsgi`”. Após instaladas as duas extensões é necessário editar o ficheiro “`000-default.conf`”, digitando o comando “`sudo nano /etc/apache2/sites-available/000-default.conf`”. Aberto o documento é colocada a entrada disposta na Figura 31 no separador “`<VirtualHost *:80>`”.

```

Alias /static /home/pi/teste/static
<Directory /home/pi/teste/static>
    Require all granted
</Directory>

<Directory /home/pi/teste/teste>
    <Files wsgi.py>
        Require all granted
    </Files>
</Directory>

WSGIDaemonProcess teste python-path=/home/pi/teste python-home=/home/pi/teste/testenv
WSGIProcessGroup teste
WSGIScriptAlias / /home/pi/teste/teste/wsgi.py

```

Figura 31- Entradas a adicionar no ficheiro “000-default.conf”.

De seguida, é necessário garantir que o Apache tem acesso à base de dados criada pela framework Django. Para isso é necessário executar os comandos “`chmod g+w ~/teste/db.sqlite3`”, “`chmod g+w ~/teste`”, que permitem que a base de dados e o conteúdo da pasta testes possa ser escrito, “`sudo chown :www-data db.sqlite3`” e “`sudo chown :www-data ~/teste`”, que permitem que o Apache tenha permissões para escrever na base de dados e na pasta teste. Para que a base de dados possa ser acedida por outro dispositivo, abre-se o ficheiro `settings.py`, utilizando o comando “`nano ~/teste/teste/settings.py`”, adiciona-se à entrada “`ALLOWED_HOSTS`” o IP do Raspberry. Por fim, basta reiniciar o serviço Apache através do comando “`sudo service apache2 restart`”. Bastando digitar o IP do dispositivo no browser obtém-se a página disposta na Figura 32.

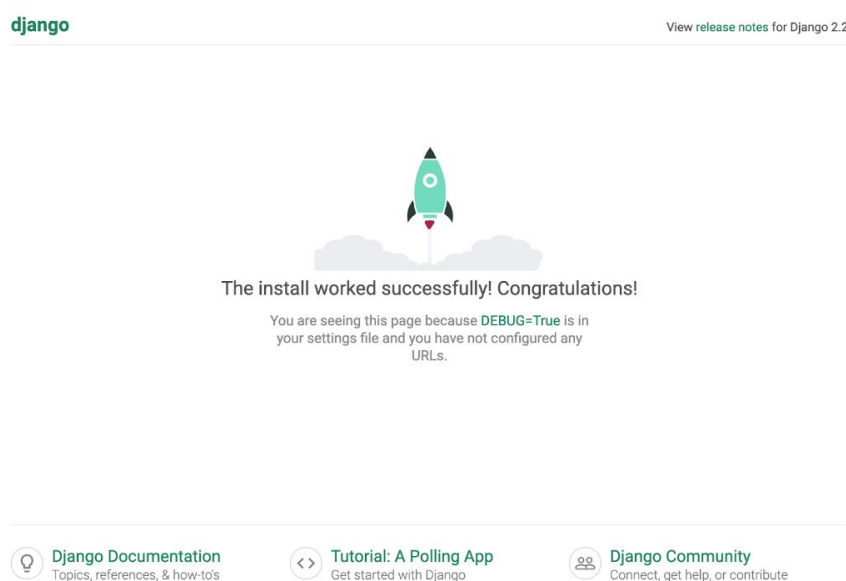


Figura 32- Página Inicial.

É importante referir também que já se encontra a área de administração criada, onde o utilizador poderá fazer as alterações ao sistema, bastando digitar o IP do dispositivo + “`/admin`” no browser.

A partir daqui o projeto poderá ser concebido diretamente num computador onde, ao contrário do sistema operativo instalado no Raspberry Pi, existe um ambiente de trabalho gráfico, permitindo que a interface seja desenhada mais rapidamente. Para a criação e edição de ficheiros

criados do sistema é utilizado o Sublime Text. Para controlo do Raspberry, é utilizado o prompt de comandos, fornecida pelo sistema operativo Windows.

3.2. Interface de Controlo do Dispositivo

A interface do controlo tem como objetivo controlar dois sistemas independentes: o relógio com mecanismo de toque e a iluminação da torre, permitindo que o utilizador possa alterar facilmente as definições do dispositivo, podendo assim ter total autonomia. A interface terá de permitir ao utilizador adicionar, modificar ou eliminar novas rotinas de toque, melodias a tocar nessas novas rotinas, adicionar, modificar ou remover toques de horas, execução de toques imediatos, no sistema do relógio, e adicionar, modificar ou eliminar rotinas de iluminação, no sistema de iluminação da torre. Anteriormente, instalou-se a framework Django, criando uma área de administração. É na área de administração que o utilizador vai fazer todas as alterações às rotinas de toque e de iluminação. Todos os dados introduzidos na área de administração serão guardados numa base de dados. Esta base de dados será lida pelo programa principal que, de acordo com as informações lidas pela base de dados, irá enviar as informações para as saídas do Raspberry. Outra informação a ter em conta é que as informações na base de dados estão guardadas em tabelas, duas para o controlo de horas, uma para os toques semanais e outra para os toques anuais, e duas para o controlo de iluminação, uma para rotinas semanais e outra para rotinas anuais.

3.2.1. Controlo de Horas

Conectado o Sublime Text ao Raspberry via conexão SSH, procede-se à configuração das tabelas para controlo de horas. Estas tabelas vão ser criadas no projeto criado anteriormente e que foi denominado de Horas e serão guardadas na base de dados. Estas tabelas devem estar disponíveis na área de Administração, onde será possível adicionar, editar e remover dados destas.

Antes de criar as tabelas, é necessário analisar que variáveis devem estar nestas e que devem ser guardadas. Para isso, é analisado o esquema de funcionamento do sistema, apresentado na Figura 33.

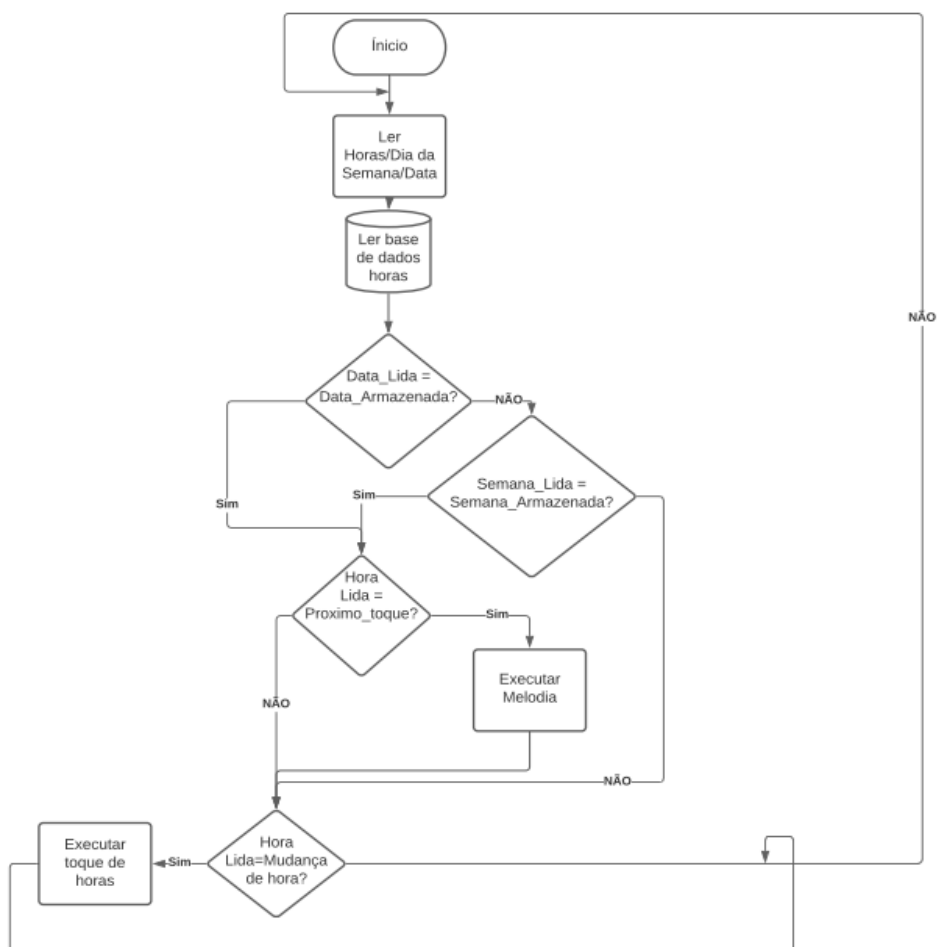


Figura 33 – Esquema de funcionamento do programa principal.

De acordo com a Figura 33, o programa principal irá fazer leitura da data, dia da semana e horas do Raspberry. Depois de lidas as informações do Raspberry, são comparadas com as informações existentes na base de dados, previamente introduzidas pelo utilizador. Se a leitura realizada pelo sistema coincidir com os dados existentes na base de dados, é executada a melodia escolhida. Para tal, são criadas duas tabelas, uma para o funcionamento de toques Anual e outra para toque Semanal/Diário.

Para introduzir as duas tabelas na área de Administração é necessário o ficheiro “models.py”, existente no projeto Horas. Neste ficheiro introduz-se uma classe e, dentro desta, introduzem-se as variáveis que a tabela deverá ter. Na Figura 34 é possível verificar que, para a tabela de Toques Semanal/Diário, as variáveis que devem ser colocadas na base dados são os dias da semana, a hora de toque e a melodia escolhida. Analisando em detalhe a classe, é possível verificar que os dias de toque é um campo de seleção múltipla, podendo assim escolher vários dias da semana. Para um toque diário devem ser escolhidos todos os dias da semana. Já o campo “horas” é um campo de hora e o campo “melodia” é um campo de escolha simples, escolhendo uma das opções

existentes na lista. É importante salientar que ao contrário do campo “dias”, o campo “melodia” não tem uma lista pré-definida. Isto porque este campo vai buscar as variáveis a uma pasta criada previamente. Nesta pasta encontram-se várias melodias. Assim, quando houver a necessidade de adicionar novas melodias ou eliminar existentes, não é necessário editar o ficheiro “models.py”, bastando apenas adicionar os novos ficheiros na pasta e reiniciando o servido Apache.

```
class Semana(models.Model):

    introduzido = models.ForeignKey(User, on_delete=models.CASCADE)

    semana = (('Segunda', 'Segunda-Feira'),
              ('Terca', 'Terça-Feira'),
              ('Quarta', 'Quarta-Feira'),
              ('Quinta', 'Quinta-Feira'),
              ('Sexta', 'Sexta-Feira'),
              ('Sabado', 'Sábado'),
              ('Domingo', 'Domingo'),)

    dias = MultiSelectField(choices = semana, default = '')

    hora = models.TimeField(null=True, blank=True)

    melodia = models.CharField(
        max_length = 20,
        choices = testes,
    )

    def __str__(self):
        return str(self.dias) + ' | ' + str(self.hora) + ' | ' + str(self.melodia)
```

Figura 34- Classe definida para Semana/Diário.

Os campos para a tabela Anual apresentada na Figura 35, segue procedimento análogo com os campos criado para a tabela Semanal/Diário, mudando o campo “dias”, que é um campo de seleção múltipla, para o campo “data”, que é um campo de inserção de data. Existe também a possibilidade de colocação de um pequeno comentário, facilitando assim a organização dos toques anuais.

```

class Ano(models.Model):
    introduzido = models.ForeignKey(User, on_delete=models.CASCADE)

    data = models.DateField(null=True, blank=True)

    hora = models.TimeField(null=True, blank=True)

    melodia = models.CharField(
        max_length = 20, |
        default = '', |
        choices = testes,
    )

    cometário = models.CharField(max_length=50, default = '',)

    def __str__(self):
        return str(self.data) + ' | ' + str(self.hora) + ' | ' + str(self.melodia)

```

Figura 35 - Classe definida para Anual.

Para adicionar as duas tabelas à área de Administração, basta adicionar ao ficheiro “admin.py” o comando “admin.site.register(Semana)” e “admin.site.register(Ano)”. A partir daqui, como demonstra a Figura 36, as tabelas encontram-se já disponíveis na área de Administração, sendo já possível a sua edição.

Figura 36 - Área de Administração com as tabelas inseridas.

De seguida, com as tabelas criadas, é necessário que o programa principal seja capaz de ler estes dados. Para tal, é criado na raiz do sistema um programa em Python, denominado “lerdados.py”, que seja capaz de ler a base de dados e enviá-los para uma matriz. É esta matriz que o programa principal deve ler e interpretar os dados e executar com base nestes. Na Figura 37 encontra-se disposta a rotina de leitura da tabela “Horas_ano” da base de dados, gerada anteriormente e exportação para uma matriz. Como é possível verificar na Figura 37, é criada uma função, denominada “lerhorasanos”. Esta função será chamada mais tarde na rotina principal, quando necessária. Dentro desta função, é importado o módulo “sqlite3”, a fim de ser possível o uso das funções de leitura das tabelas da base de dados. É importada a função “np” do módulo “numpy”, a fim de ser possível a conversão do conteúdo da leitura das tabelas, que vêm em forma de tuples¹. De seguida, é estabelecida a conexão à base de dados, utilizando o comando “sqliteConnection = sqlite3.connect('db.sqlite3)”. Estabelecida a conexão, é selecionada a tabela “Horas_ano”, através da utilização do comando “sqlite_select_Query = """SELECT * FROM Horas_Semana"""”, e é enviada a informação para uma variável, em forma de tuple, através do comando “record = cursor.fetchall()”. Para converter esta variável para uma variável matricial, é utilizado o comando “leranos=np.asmatrix(record)”. Por fim, é encerrada a sessão e é enviada a variável para o programa. O mesmo procedimento é utilizado para a tabelas das semanas.

```
def lerhorasanos():
    import sqlite3
    import numpy as np

    try:
        #conversão dos ficheiros da base dados para uma matriz; atualizar imediatamente
        #colocar isto a fazer de minuto a minuto mais tarde
        sqliteConnection = sqlite3.connect('db.sqlite3')
        cursor = sqliteConnection.cursor()
        # Selecionar tabela
        sqlite_select_Query = """SELECT * FROM Horas_Ano """
        cursor.execute(sqlite_select_Query)
        # lê dados da tabela
        record = cursor.fetchall()

        cursor.close()
        #converte dados para array
        leranos=np.asmatrix(record)
        #print(leranos)

    except sqlite3.Error as error:
        # print("Error while connecting to sqlite", error)

    finally:
        if (sqliteConnection):
            sqliteConnection.close()

    return(leranos)
```

Figura 37 – Rotina para leitura de uma tabela proveniente da base de dados.

¹ -Coleção de objetos, imutável, separados por vírgulas. [28]

Tendo a leitura da base de dados operacional, procede-se à construção do programa principal, que vai ser responsável por receber os dados das tabelas e irá enviar para outras rotinas, que por sua vez executam os dados fornecidos ao tempo pretendido. Para tal, é criado um ficheiro no diretório principal do Raspberry, onde se encontra também o ficheiro “lerdados.py”. Inicialmente, são importados os módulos “datetime”, “time”, “lerdados”, “diretorios”, a função “np” do módulo “numpy”, e a função “GPIO” do módulo “RPi.GPIO” como é exposto na Figura 38. É importante relembrar que o módulo “lerdados” e “diretorios” não são nativos do Python, ou seja, são criados pelo utilizador. Neste caso, o módulo “lerdados” foi criado anteriormente e exposto na Figura 37 e o módulo “diretorios” será criado após processamento dos dados da tabela, sendo a ponte entre o programa principal e as rotinas de toque.

```
from datetime import datetime
import time
import lerdados
import diretorios
import numpy as np
import RPi.GPIO as GPIO
from time import sleep
```

Figura 38 - Módulos a importar.

De seguida é insere-se o comando “while True:”. Este comando permite que o código Python que se encontra dentro deste comando seja executado continuamente até que a condição seja falsa. Neste caso, como foi inserido “True”, esta irá ser executada indefinidamente. Nas Figuras 39, 40 e 41 é possível verificar como será executado o programa. Começa-se por executar a função “now” do módulo “datetime” e os dados são armazenados na variável “now”. De seguida, separa-se as informações existentes na variável “now” por data, hora e dia da semana. De seguida, são chamadas as funções de leitura das tabelas da base dados através dos comandos “anos=lerdados.lerhorasanos()” e “semana=lerdados.lerhorassemana()”. Estes comandos chamam a função correspondente, existente no ficheiro “lerdados.py”, devolvendo a informação e guardando na variável correspondente do programa principal, sob a forma de matriz. Guardadas as informações na variável, é necessário ler as informações existentes nesta, começando pelos toques anuais. Para isso é utilizado um ciclo “while”. Neste ciclo, é comparada a data lida pelo sistema e a data existente na base de dados. Se as datas coincidirem, é feita uma comparação de horas atuais com o valor na base de dados. Se coincidirem, é gravada na variável “melodia” o ficheiro Python, correspondente a uma melodia selecionada pelo utilizador e gravada na base dados, e é chamada a função “diretorios” do módulo “diretorios”, exposta na Figura 42. O ciclo “while” irá funcionar até que a variável “check” seja 1 ou que seja percorrida toda a matriz sem dados correspondentes. Caso não tenha sido encontrada correspondência na tabela de toques anuais, continua-se para a leitura da base de dados de toques semanais. O procedimento é similar ao de leitura de toques anuais, requerendo apenas mais um

passo. Para a leitura dos dias da semana da tabela, é necessário separar a informação que vem num campo da matriz: os dias da semana a tocar. Para tal, utiliza-se o comando “separado=dias.split(',')”, separando a informação sempre que encontrando uma virgula. De seguida, esta informação é convertida para um array e é adicionado ao ciclo while já existente de leitura da tabela de dias da semana, outro ciclo while para percorrer cada dia da semana gravado na variável “separado”. Por fim, é verificado se os minutos da hora recolhida pelo sistema. Caso seja o minutos, é executado o toque de horas, correspondendo o número de badaladas às horas correspondentes. Caso seja 30 minutos é executada um toque único. Apenas não será executado o toque de horas se a hora atual seja maior que as dez da noite (22h) ou menor que as oito da manhã (8h). Esta interrupção tem apenas como finalidade o cumprimento do dever de boa vizinhança, procurando evitar fazer barulho durante o descanso da população da zona.

```
while True:

    now = datetime.now()
    current_time = now.strftime("%H:%M:%S")
    current_date = now.strftime("%Y/%m/%d")
    diasemana = now.strftime("%A")
    splithoras=current_time.split(':')
    #print(splithoras)
    splithoras=np.asarray(splithoras)
    minuto=splithoras[1]

    if minuto != minuto1:
        minuto1=minuto
        anos=lerdados.lerhorasanos()
        semana=lerdados.lerhorassemana()
        medida=len(anos)
        a=0
        check=0

        while a<len(anos) or check==1:
            if current_date==anos[a,2]:
                if current_time==anos[a,3]:
                    melodia=anos[a,5]
                    diretorios.diretorios()
                    check=1

            a=a+1
```

Figura 39 – Programa Principal – Leitura de dados.

```

if check != 1:
    a=0
    while a<len(semana) or check==1:
        dias=semana[a,3]
        separado=dias.split(',')
        separado=np.asarray(separado)
        i=0
        while i < len(separado) or check==1:
            diaseparado=separado[i]
            if diasemana==diaseparado:
                if current_time==semana[a, 4]:
                    melodia=semana[a,1]
                    diretorios.diretorios()
                    check=1
            i=i+1
        a=a+1

```

Figura 40 - Programa Principal – Atribuição da melodia.

```

if minuto=='00':
    hatual=now.strftime("%H")
    hatual=int(hatual)
    if hatual<22 and hatual > 8:
        horas = now.strftime("%I")
        toquehoras= int(horas)
        toque=0
        while toque < toquehoras:
            GPIO.output(26, GPIO.HIGH)
            sleep(1)
            GPIO.output(26, GPIO.LOW)
            sleep (1)
            toque=toque+1

if minuto == '30':
    if hatual<22 and hatual > 8:
        GPIO.output(26, GPIO.HIGH)
        sleep(1)
        GPIO.output(26, GPIO.LOW)
        sleep (1)

```

Figura 41 - Programa Principal – Atribuição do número de toques.

Como referido anteriormente, o módulo “diretorios” é a rotina que faz a ligação entre o programa principal, e as rotinas de toque. Este módulo tem apenas uma função, apresentada na Figura 42. Nesta função, é importada a variável melodia do programa principal para a variável “thoras”. De seguida, importa-se o módulo “sys”, que vai direcionar o sistema para a pasta onde se encontram as rotinas Python, utilizando logo de seguida o comando “sys.path.append('/home/pi/teste/toque_codigo')”. De seguida são importados todos os módulos lá existentes. Para efeitos de testes foram criadas rotinas simples de toque, exemplificado na Figura 43. É importante referir que os nomes destes ficheiros são lidos

automaticamente pela framework django, pelo que os nomes dos ficheiros aqui serão iguais aos apresentados pela base de dados.

```
def diretorios():
    import maincode
    thoras=maincode.melodia

    import sys
    sys.path.append('/home/pi/teste/toque_codigo')
    #print(thoras)
    import led_blink
    import led_blink1
    import toque_horas

    if thoras=="led_blink.py":
        led_blink.melodia()
    if thoras=="led_blink1.py":
        led_blink1.melodia()
    if thoras=="toque_horas.py":
        toque_horas.melodia()
```

Figura 42 – Rotina “diretorios”, que direciona o programa principal às rotinas de toque.

```
def melodia():
    import RPi.GPIO as GPIO
    from time import sleep

    GPIO.setwarnings(False)
    GPIO.setmode(GPIO.BOARD)
    GPIO.setup(37, GPIO.OUT, initial=GPIO.LOW)

    GPIO.output(37, GPIO.HIGH)
    sleep(1)
    GPIO.output(37, GPIO.LOW)
    sleep (1)
    GPIO.output(37, GPIO.HIGH)
    sleep(1)
    GPIO.output(37, GPIO.LOW)
    sleep (1) |
```

Figura 43 – Exemplo de melodia de toque.

3.2.2. Toque Imediatos de Melodias

A execução de melodias de forma imediata é, para além da possibilidade de editar os toques anuais e semanais, um dos elementos-chave deste sistema. Existem várias possibilidades de executar esta tarefa. Uma delas é introdução de botões no *HyperText Markup Language* (HTML) na área de administração. No entanto, são apresentados dois problemas. Um destes problemas advém de como a área foi gerada automaticamente pela framework django, os HTML desta página encontram-se fragmentados, sendo difícil a sua implementação. A outra desvantagem é que a sua implementação poderia provocar conflitos com o programa principal ao acederem em simultâneo às saídas do Raspberry. Outra solução para o problema considerado é a criação de outra tabela de base de dados. O programa principal lê constantemente essa tabela, e executa caso haja algum dado nessa tabela. Depois de executada a melodia, é eliminada a entrada, ficando essa tabela vazia. Como é de esperar, caso o programa principal esteja a executar outra melodia, essa entrada fica em espera até que seja terminada a melodia anterior. Na Figura 44 é exposto o esquema de funcionamento do sistema atualizado, com a implementação de toques imediatos.

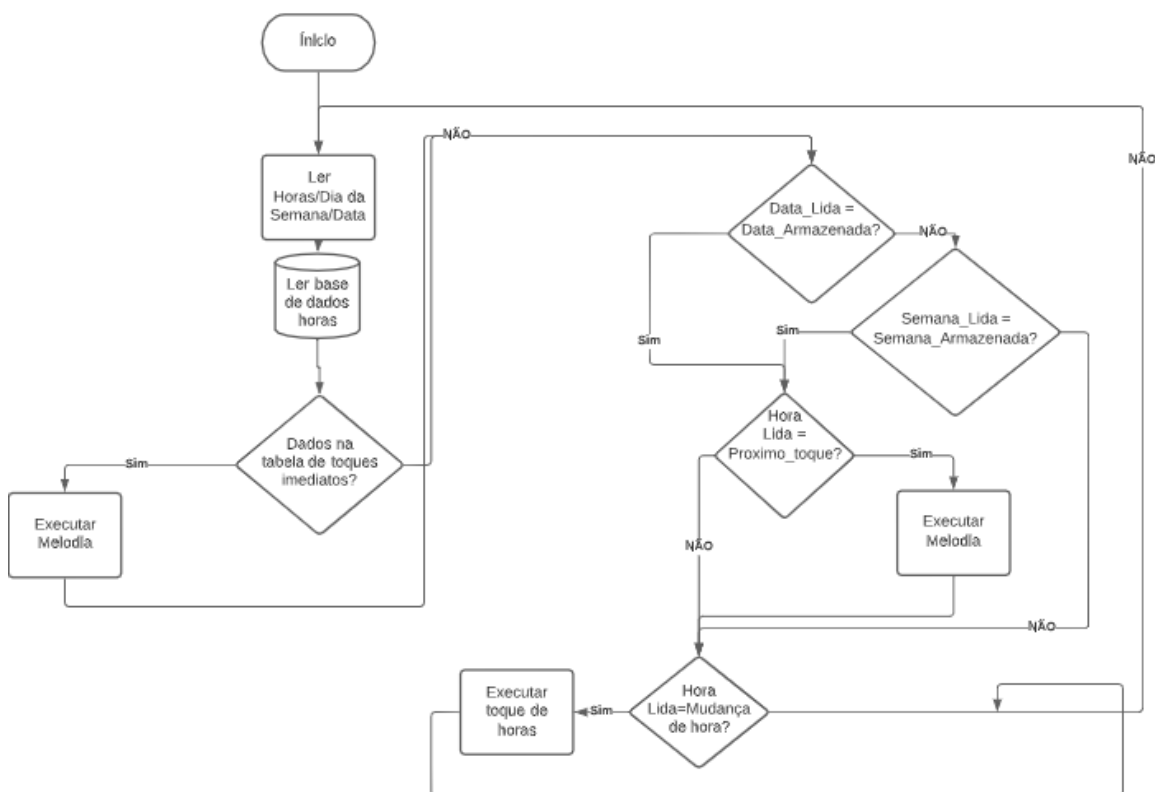


Figura 44 - Esquema de funcionamento do sistema com toques imediatos e horários.

Para que o utilizador tenha possibilidade de escolher a melodia a executar, é necessário criar a tabela de toques imediatos e torná-la disponível na área de administração. Para criar uma nova tabela na base de dados, basta ir ao ficheiro “models.py” e criar uma nova classe, como é

demonstrado na Figura 45. Nesta tabela apenas está disponível para edição o campo de melodia, onde é escolhida uma melodia com base nas melodias já existentes.

```
class Imediato(models.Model):
    melodia = models.CharField(
        max_length = 27,
        choices = testes,
    )
    def __str__(self):
        return str(self.melodia)
```

Figura 45 – Rotina a adicionar para a criação da tabela de toque imediatos.

De seguida, é necessário adicionar esta tabela à área de administração. Para tal, é necessário importar o modelo e registar a tabela na área no ficheiro “admin.py”, como é demonstrado na Figura 46. A partir daqui a tabela passa a estar disponível na área de administração do sistema.

```
from django.contrib import admin

# Register your models here.
from .models import Semana
from .models import Ano
from .models import Imediato

admin.site.site_header = 'Administration'
admin.site.register(Semana)
admin.site.register(Ano)
admin.site.register(Imediato)
```

Figura 46 - Registo da tabela na área de administração.

De seguida, é necessário que o programa principal leia esta tabela. Para a leitura é adicionada uma nova função no ficheiro “lerdados.py”, exposta na Figura 47, sendo este bastante similar a outras já criadas.

```

def lerimediatos():
    import sqlite3
    import numpy as np

    try:
        sqliteConnection = sqlite3.connect('db.sqlite3')
        cursor = sqliteConnection.cursor()
        # Selecionar tabela
        sqlite_select_Query = """SELECT * FROM Horas_Imediato"""
        cursor.execute(sqlite_select_Query)
        # lê dados da tabela
        record = cursor.fetchall()
        cursor.close()
        lerimediatos=np.asmatrix(record)

    finally:
        if (sqliteConnection):
            sqliteConnection.close()

    return(lerimediatos)

```

Figura 47 – Leitura da tabela existente na base de dados.

Já na rotina principal, exposta na Figura 48, é introduzida no ciclo “while” principal, o conteúdo da leitura feita da tabela da base dados numa variável. Se a variável tiver um tamanho superior a 0, esta é convertida para um formato de array e é retirada a melodia deste array, executando logo de seguida a função “diretorios”, que por sua vez reencaminha para a rotina da melodia a executar. O sistema irá constantemente fazer a procura de dados nessa tabela, pelo que os dados, após serem processados, devem ser removidos. Caso contrário, o sistema irá executar o toque imediato constantemente até que este seja removido. Para a remoção da entrada, é introduzida outra função no ficheiro “lerdados.py”. Como é possível verificar na Figura 49, na função remove, é feita a conexão com a base de dados do sistema. De seguida é dado o comando “sqlite_Delete_Query = """DELETE FROM Horas_Imediato WHERE ID"""”, fazendo que a linha ID seja removida da tabela. É utilizada uma variável em vez de uma constante, pois sempre que é introduzido um novo toque imediato, é introduzido um novo ID, independentemente do número de linhas existentes nesta tabela. Removida a entrada, é terminada a conexão com a base dados.

```

imediatos=lerdados.lerimediatos()
imediatos=imediatos[0]

if imediatos.size > 0 :
    imediatos=np.asarray(imediatos)
    imediatos=imediatos[0]
    ID=imediatos[0]
    imediatos=imediatos[1]
    melodia=imediatos
    diretorios.diretorios()
    lerdados.remove()

```

Figura 48 - Rotina de toques imediatos na rotina principal.

```
def remover():
    import sqlite3
    import numpy as np

    try:
        sqliteConnection = sqlite3.connect('db.sqlite3')
        cursor = sqliteConnection.cursor()
        sqlite_Delete_Query = """DELETE FROM Horas_Imediato WHERE ID"""
        cursor.execute(sqlite_Delete_Query)
        sqliteConnection.commit()
        cursor.close()

    finally:
        if (sqliteConnection):
            sqliteConnection.close()
```

Figura 49 - Rotina de remoção da entrada existente na tabela de toques imediatos.

Por fim, é dada a ordem para girar, sendo acionada a porto de saída do dispositivo durante um determinado período de tempo.

```
D-B-I
#mudança de minuto
GPIO.output(27, GPIO.HIGH)
sleep(5)
GPIO.output(27, GPIO.LOW)
```

Figura 50 – Ativação do porto de saída do dispositivo.

3.2.3. Controlo de Iluminação

O controlo de iluminação é um fator importantíssimo na gestão de património e de controlo de custos. Para o caso de estudo em questão, é pretendido a utilização da iluminação exterior apenas em ocasiões especiais, pelo que é necessário a utilização de um modelo de estrutura similar ao de toque de horas, isto é, a utilização da base de dados para gestão de iluminação. É criada uma tabela de gestão semanal. A integração deste controlo ao programa principal é feita de acordo com fluxograma apresentado na Figura 51.

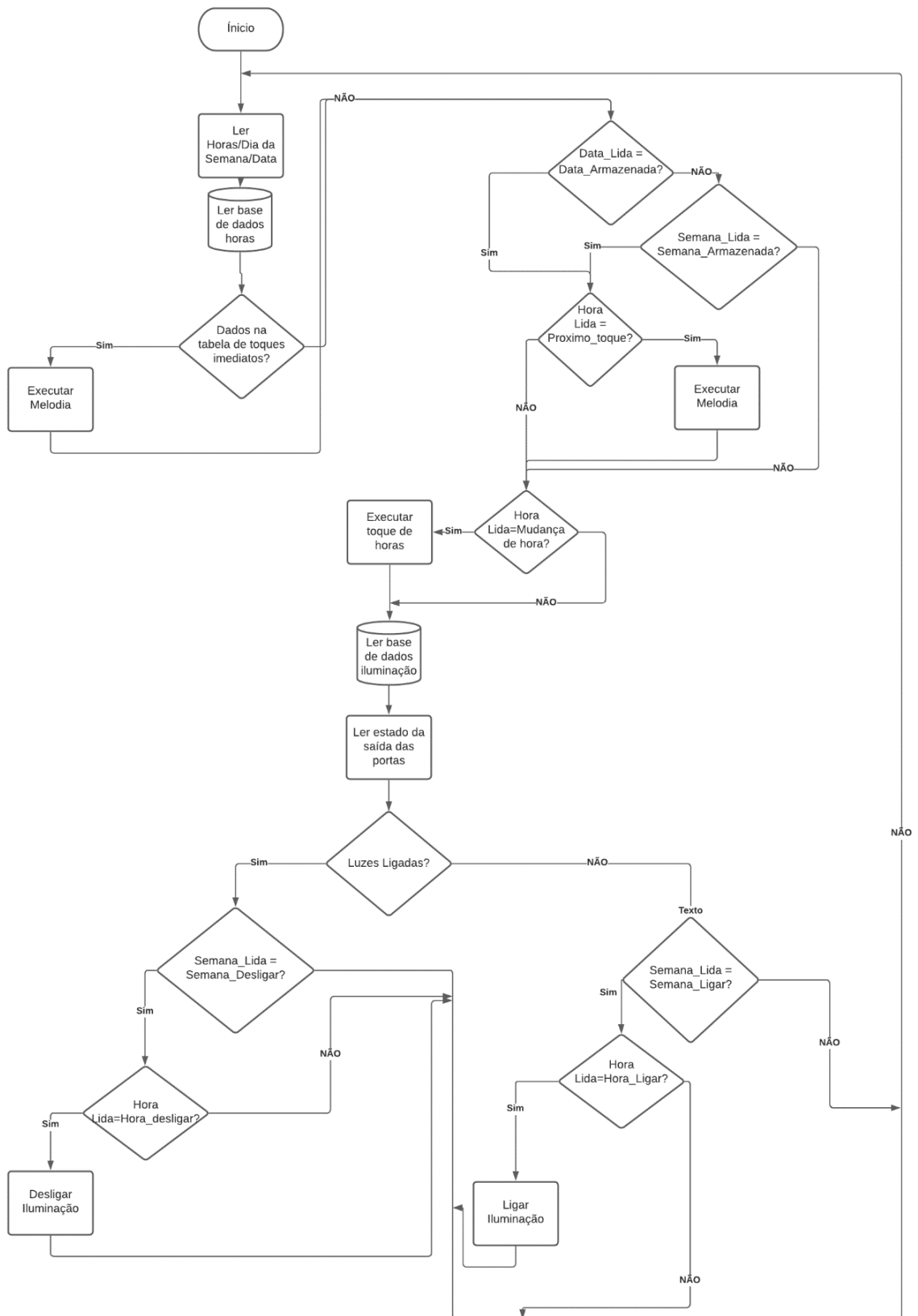


Figura 51 – Programa principal com a integração de controlo de iluminação.

Para adicionar a tabela de gestão semanal é necessário recorrer ao ficheiro “models.py”, embora desta vez seja no diretório criado para a iluminação e adicionar os parâmetros pretendidos, como explicitado na Figura 51. Como é possível verificar na Figura 52, os parâmetros a utilizar são os dias em que a iluminação deve ser ligada e em que intervalos de horários deve permanecer ligada.

```
class Semanais(models.Model):
    Introduzido = models.ForeignKey(User, on_delete=models.CASCADE)

    semana = (('Monday', 'Segunda-Feira'),
              ('Tuesday', 'Terça-Feira'),
              ('Wednesday', 'Quarta-Feira'),
              ('Thursday', 'Quinta-Feira'),
              ('Friday', 'Sexta-Feira'),
              ('Saturday', 'Sábado'),
              ('Sunday', 'Domingo'),)

    Dias = MultiSelectField(choices = semana, default='')

    Ligar= models.TimeField(null=True, blank=True)
    Desligar= models.TimeField(null=True, blank=True)

    def __str__(self):
        return str(self.Dias) + ' | ' + str(self.Ligar) + ' | ' + str(self.Desligar)
```

Figura 52 – Criação da tabela no ficheiro “models.py”.

De seguida é necessário registar a base de dados na área de administração, como exposto na Figura 53, ficando assim disponível para o utilizador registar dados.

```
admin.site.site_header = 'Administration'
admin.site.register(Semanais)
```

Figura 53 – Registo da tabela na área de administração.

Estando disponível na área de administração, é necessário que o programa principal seja capaz de ler esta tabela, pelo que é criada outra função no ficheiro “lerdados.py”, como apresentado na Figura 54.

```
def ilusemanais():  
    import sqlite3  
    import numpy as np  
  
    try:  
        sqliteConnection = sqlite3.connect('db.sqlite3')  
        cursor = sqliteConnection.cursor()  
        # Selecionar tabela  
        sqlite_select_Query = """SELECT * FROM Iluminação_semanais"""  
        cursor.execute(sqlite_select_Query)  
        # lê dados da tabela  
        record = cursor.fetchall()  
        cursor.close()  
        luzsemana=np.asmatrix(record)  
  
    finally:  
        if (sqliteConnection):  
            sqliteConnection.close()  
  
    return(luzsemana)
```

Figura 54 - Leitura da tabela “Iluminação_Semanais”.

De seguida, na rotina principal, como exemplificado na Figura 55, é introduzido no ciclo “while” principal, o conteúdo da leitura feita da tabela da base dados numa variável. De seguida, é utilizado outro ciclo “while” para efetuar a leitura da variável, linha a linha. É utilizada o comando “separado=dias.split(',')” para separa os dias da semana, que ocupam a mesma posição na tabela e são guardados num array. De seguida, é percorrido o array para verificar se algum dos dias guardados corresponde ao dia lido. Caso corresponda, é verificado se a iluminação se encontra ativa ou não, através do uso de uma variável. Caso se encontre desligada, será verificado se a hora lida é a idêntica que a hora armazenada, procedendo à sua ativação em caso de igualdade. Se se encontrar ligada, é feita a mesma verificação, desligando se a hora for a mesma.

```
#iluminação
b=0
check=0
lusemanas=lerdados.ilusemanais()
print(lusemanas)
while b<len(lusemanas) and check!=1:
    dias=lusemanas[b,4]
    print(dias)
    separado=dias.split(',')
    separado=np.asarray(separado)
    print(separado)
    i=0
    while i < len(separado) and check!=1:
        diaseparado=separado[i]
        print(diaseparado)
        if diasemana==diaseparado:
            print(current_time)
            print(lusemanas[b,3])
            print(hluz)
            if hluz==0:
                if current_time==lusemanas[b, 3]:
                    print('oi')
                    GPIO.output(21, True)
                    hluz=1
                    #ativar saída
                    check=1
            if hluz==1:
                if current_time==lusemanas[b, 2]:
                    #desativar saída
                    GPIO.output(21, False)
                    hluz=0
                    check=1
        i=i+1
    b=b+1
```

Figura 55 – Rotina de Iluminação.

3.3. Desenho do Hardware do Dispositivo

Para a construção da placa de circuito impresso (PCB – *Printed Circuit Board*) é utilizada a ferramenta de desenho assistido por computador (CAD – *Computer Aided Design*), Proteus, sendo que todos os esquemas de circuito e o desenho da PCB foram desenvolvidos neste software.

Antes de se desenhar o sistema, tendo em vista o aproveitamento máximo de componentes do sistema atual, é necessário compreender como é feito o acionamento dos martelos de picar. O sinal é emitido do dispositivo principal para um comutador, sendo que o comutador difere consoante o martelo a tocar. O sinal é emitido através de um relé existente no dispositivo OPUS para o comutador, sendo o sinal de 230 V alternado. Os comutadores encontram-se num quadro elétrico existente no cimo da torre, como apresentado na Figura 56.

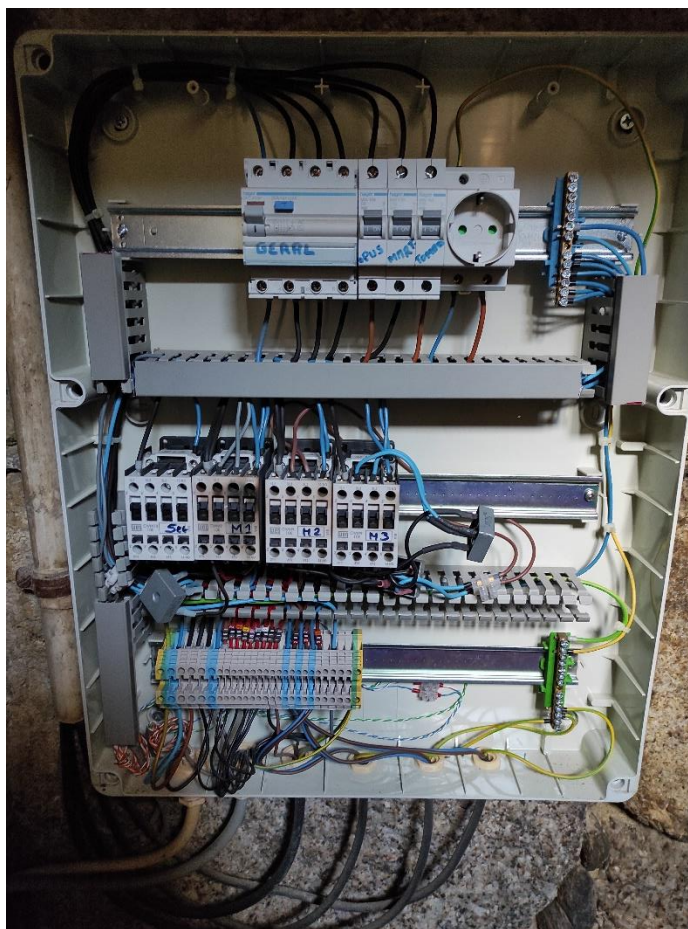


Figura 56 – Quadro elétrico, aberto, existente no cimo da torre, junto ao conjunto sineiro.

Como é possível verificar, existem 4 comutadores, sendo que um deles não se encontra ligado a nenhum dispositivo e os outros 3 encontram-se utilizados para o acionamento dos martelos eletromecânicos. É nestes comutadores que entra o sinal proveniente do dispositivo de controlo. Já o acionamento dos ponteiros do relógio é acionado diretamente do dispositivo de controlo. O controlo de iluminação da torre, como referido anteriormente, é controlado pela câmara, pelo que não existe qualquer tipo de controlo.

3.3.1. Placa de circuito impresso (PCB)

Analisando o quadro elétrico e o dispositivo de controlo existentes, é possível verificar que o dispositivo de controlo novo deve fornecer um sinal de tensão elétrica alternada de 230 V. No entanto, a tensão do funcionamento do Raspberry é de 5 V, pelo que este não pode acionar diretamente os comutadores, que têm uma tensão de 110-230 V. Para tal são utilizados relés. Neste caso, são usados relés com a referência SDR-05VDC-SL-C, permitindo uma tensão de

comutação máxima de 250 V e corrente máxima de 10 A [24]. Aos terminais da bobina do relé é colocado uma tensão contínua de 5 V DC. Ao comutador é ligado a 230V AC e a carga é colocada no terminal normalmente aberto, pois é pretendido que o relé conduza corrente quando seja dada ordem pelo controlador. Aos terminais da bobina é colocado um transistor a fim de ser providenciada corrente suficiente para ativar a bobina quando requerido. De acordo com o Datasheet do fabricante [24], a corrente necessária para ativar a bobina do relé é de 60 mA, pelo que é necessário um transistor que consiga fornecer pelo menos essa corrente. Neste caso, foi utilizado o transistor 2N2222, tendo este uma corrente máxima de 600 mA, de acordo com o Datasheet [25]. Na base do transistor é colocada uma resistência de 1 kΩ e é ligado ao terminal do Raspberry Pi. É colocado também um diodo aos terminais da bobina do relé a fim de impedir um pico de tensão quando a bobina é desligada, evitando assim que os transistores ou outros dispositivos queimem devido a tensões elevadas. O esquema de circuito para cada saída do dispositivo encontra-se exposto na Figura 57.

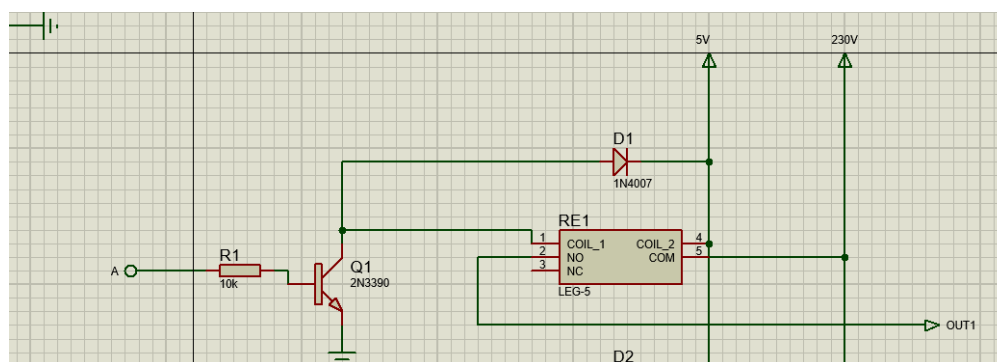


Figura 57 – Esquema de circuito para uma saída do dispositivo de controlo.

Como qualquer dispositivo eletrónico, este dispositivo precisa de uma fonte de alimentação. Neste caso, o dispositivo é alimentado pela rede elétrica. No entanto, embora as saídas do relé se encontrem ao mesmo valor de tensão que a da rede elétrica, o Raspberry, a bobina do relé e outros componentes eletrónicos não são compatíveis com a tensão da rede elétrica, operando a uma tensão muito mais baixa. Para alimentação dos componentes eletrónicos foi utilizada uma fonte de alimentação externa de 12 V com potência de 60 W, com a referência LDP-279-265-9. A vantagem de utilizar uma fonte de alimentação externa consiste na redução de temperatura de funcionamento da placa principal, e a facilidade de fazer despistes do dispositivo em caso de avaria, coisa que pode acontecer visto que a torre, durante as trovoadas, encontra-se a exposta a fenómenos eletromagnéticos de grande magnitude. No entanto, os componentes eletrónicos operam a uma tensão de 5 V, sendo necessário fazer a regulação para este valor de tensão. Para tal é utilizado um regulador LM338T, seguindo o esquema da Figura 58.

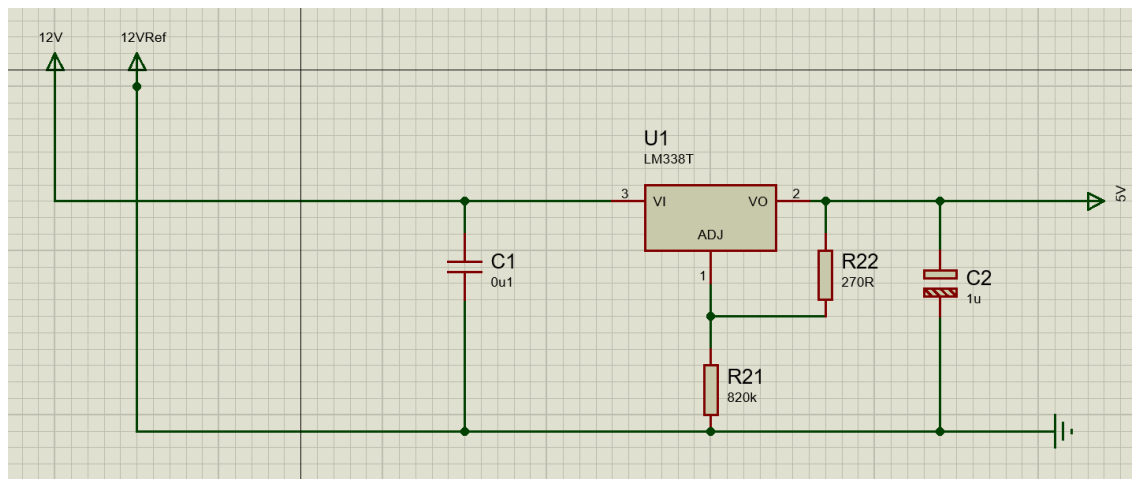


Figura 58 – Regulador de Tensão

Como é possível verificar na Figura 58, é colocado um condensador de cerâmica ao terminal de alimentação e o ADJ do regulador, tendo como finalidade a reduzir os ruídos indesejáveis vindos da fonte de alimentação, utilizando duas resistências, R21 e R22. A Resistência R22 é utilizada pois, de acordo com o fabricante [26], o regulador tem uma diferença potencial de cerca de 1,24 V entre o pin de saída e ADJ, sendo necessário a utilização de uma resistência de 270 Ω a fim de ser feita passagem de corrente elétrica de 4,6 mA pelas resistências R21 e R22. A resistência R21 é variável, isto é, dependendo da tensão a utilizar, a resistência terá um determinado valor, onde a soma das duas quedas de tensão das duas resistências terão de ser a tensão desejada. Como se pretende que a tensão de saída seja de 5 V, utilizando a Equação 1, fornecida pelo fabricante [26], conclui-se que a resistência a utilizar é de 820 Ω .

$$V_{out} = 1,25 * \left(1 + \frac{R21}{R22} \right) + V_{R21} \quad (1)$$

Tendo o esquema de circuito sido elaborado na ferramenta Proteus, procede-se ao desenho da PCB. Para a construção da PCB, basta carregar no botão “PCB Layout” da ferramenta e arrastar os componentes de acordo com a disposição desejada. De seguida, após colocados os componentes na placa, seleciona-se a opção “auto-routing”, permitindo que as ligações sejam feitas automaticamente e com os seus percursos otimizados, obtendo o resultado disposto na Figura 59.

É feita uma verificação no modelo 3D da placa, a fim de verificar se a placa e os componentes se encontram nas características pretendidas. Neste caso, não são utilizados componentes SMD, pelo que o modelo 3D não deverá apresentar componentes 3D, como apresenta a Figura 60.

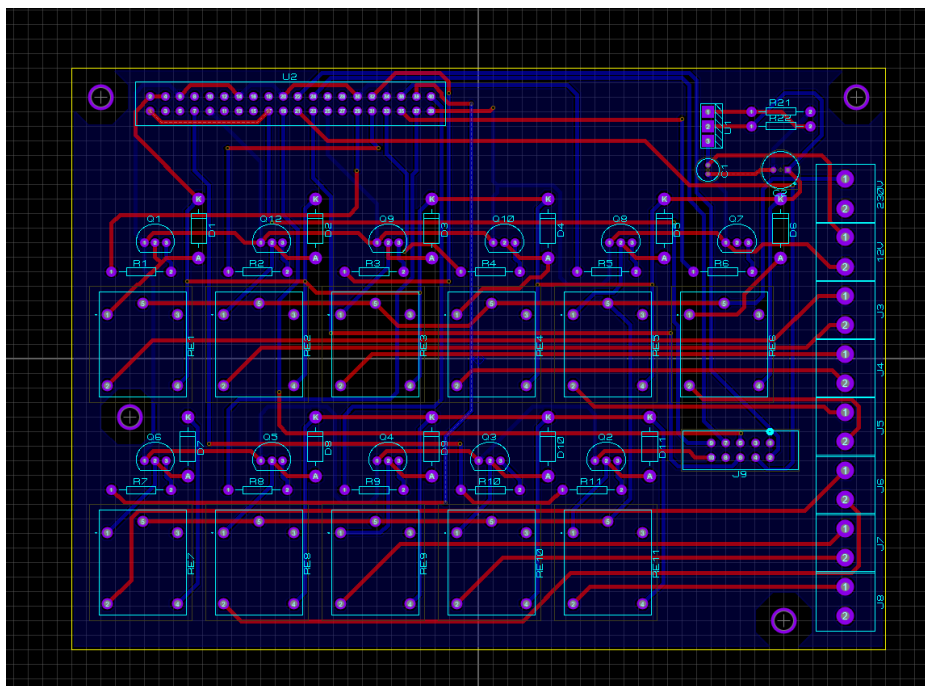


Figura 59 – Layout da PCB da placa principal.

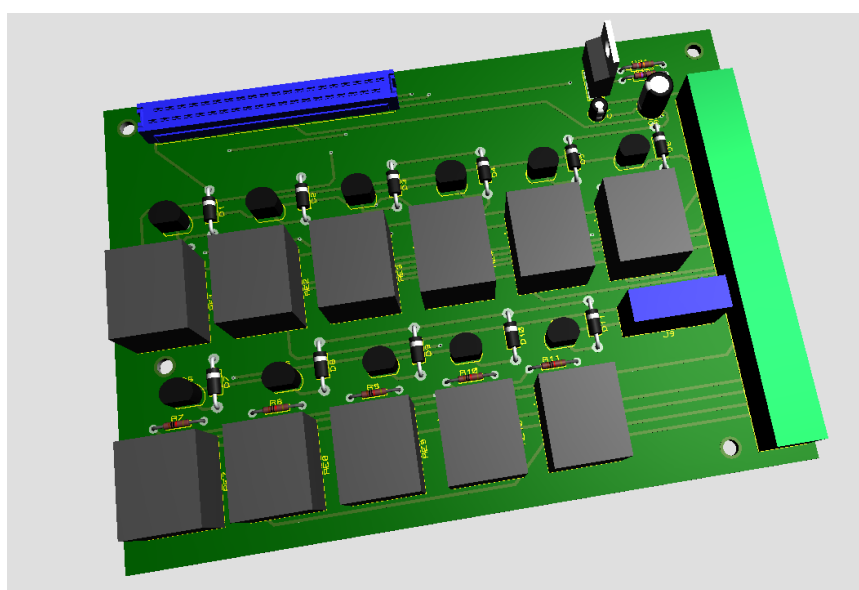


Figura 60 – Modelo 3D da placa de controlo.

É importante referir que as saídas deste dispositivo irão ligar aos comutadores já existentes na torre da igreja, mantendo assim grande parte da instalação já existente.

3.3.2. Caixa

A caixa é também um elemento bastante importante neste dispositivo pois previne a eletrónica de impactos e de poeiras, permitindo assim um aumento de tempo de vida útil.

Como o dispositivo não tem qualquer tipo de interface física com o utilizador, como por exemplo ecrãs ou botões, a caixa acaba por ser bastante simples de conceber, sendo que esta terá de conter a placa de controlo do sistema sineiro e uma fonte de alimentação.

Para o desenho da caixa é utilizada a Ferramenta CAD SOLIDWORKS, permitindo assim uma visualização realista da caixa e a sua construção, ao utilizar uma impressora 3D por exemplo.

Como referido anteriormente, a placa de controlo e a fonte de alimentação deverão encontrar-se dentro da caixa, pelo que é preciso verificar as medidas de ambos equipamentos. Para verificar as dimensões da placa de controlo basta abrir a ferramenta Proteus, utilizada para desenhar a PCB, e obter as medidas da PCB, como demonstra a Figura 61. É importante referir que é necessário obter também algumas medidas da furação da PCB pois a caixa terá apoios de fixação da placa, sendo que os apoios de fixação a furação da placa sejam coincidentes.

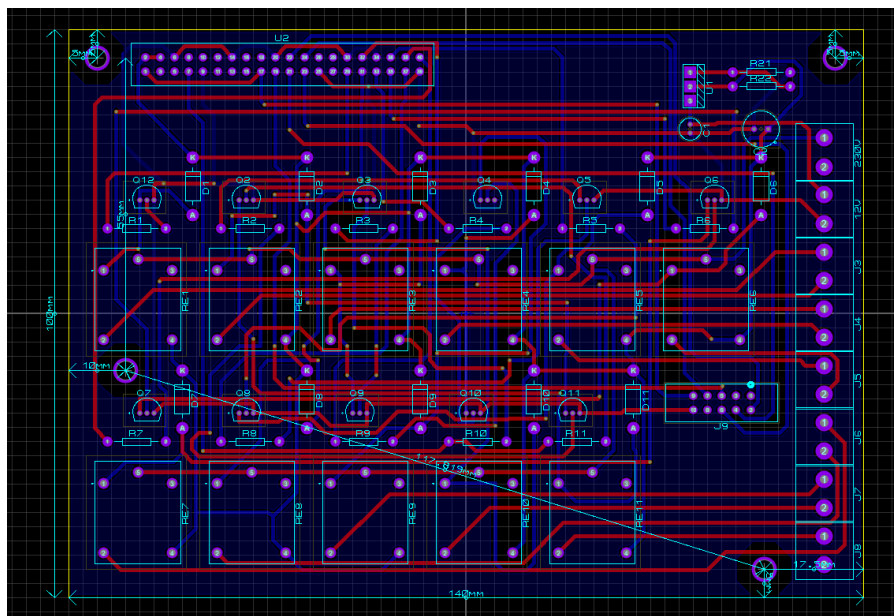


Figura 61 - Obtenção das dimensões da PCB e da respetiva furação.

É selecionada uma fonte de alimentação de 12V com potência máxima de 60W, suficiente para os requisitos energéticos do Raspberry Zero e a demais eletrónica, tendo como dimensões, de acordo com o fabricante [27], 41mm de largura, 162mm de comprimento e 33 mm de altura.

Tendo as medidas dos dispositivos, é necessário definir as dimensões da caixa. Para além das dimensões conjuntas da placa e da fonte de alimentação, é adicionado também uma folga de cerca de 5 mm para cada lado da caixa e 4 mm entre dispositivos, obtendo assim a caixa com comprimento 180 mm e largura de 160 mm. Já a altura ficou definida em 50 mm a fim de haver altura suficiente para albergar todos os componentes necessários para a execução deste projeto e descontar a espessura do fundo e do topo da caixa. São colocados também os apoios de fixação para a PCB, de acordo com as medidas obtidas e da fonte de alimentação, ficando o fundo da caixa de acordo com a Figura 62. É adicionado um furo para passagem de cabos elétricos.

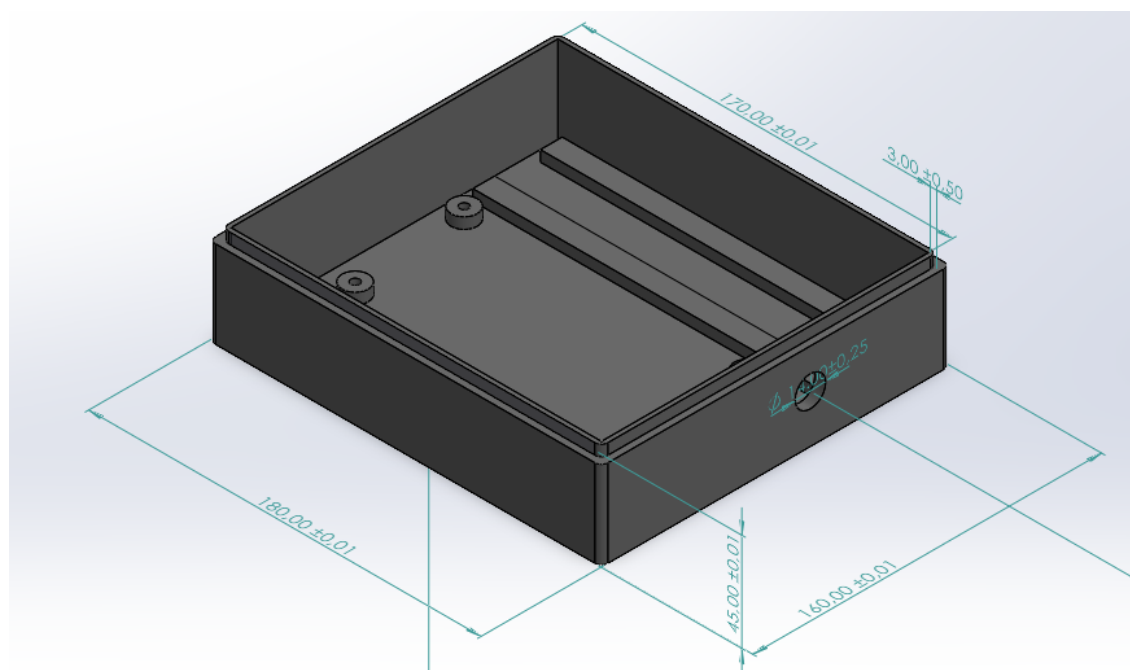


Figura 62 - Parte de baixo da caixa.

Para obter uma melhor ideia de como será a caixa com os componentes integrados é importado do Proteus o desenho 3D da PCB para o SOLIDWORKS e integrado na caixa, na qual é desenhada e integrada uma representação da fonte de alimentação com as mesmas dimensões, obtendo o resultado apresentado na Figura 63. Este procedimento final também importante a fim de verificar se as dimensões da caixa correspondem às necessidades exigidas pelos dispositivos.

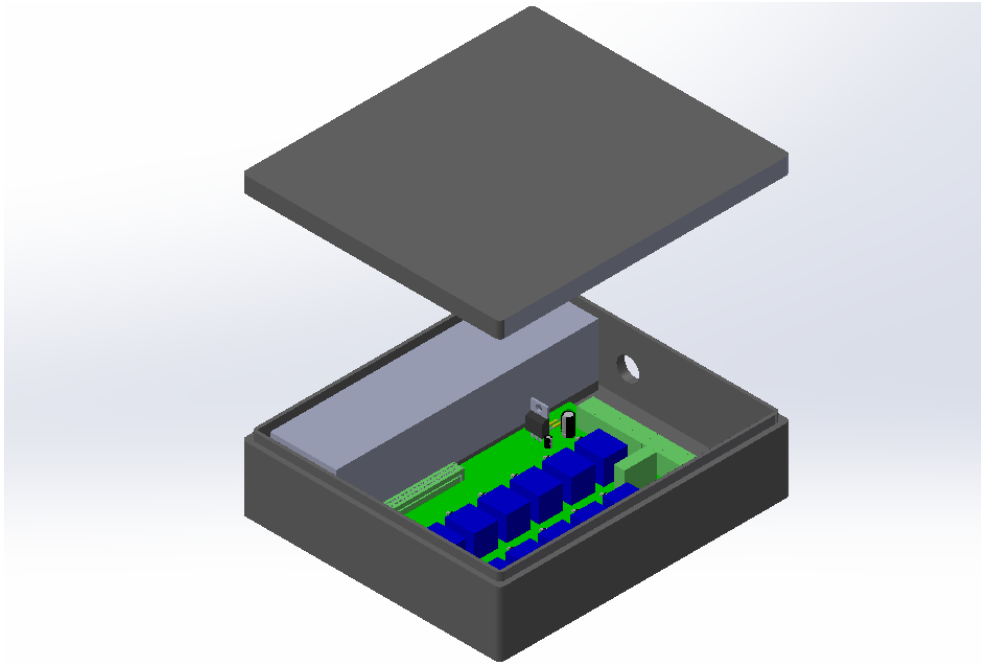


Figura 63 - Dispositivo final.

4. Testes de funcionamento

Neste capítulo, é apresentada montagem e teste na torre sineira do dispositivo eletrônico proposto para o problema em causa.

4.1. Montagem do dispositivo

A PCB desenvolvida para o dispositivo encontra-se exposta na Figura 64.

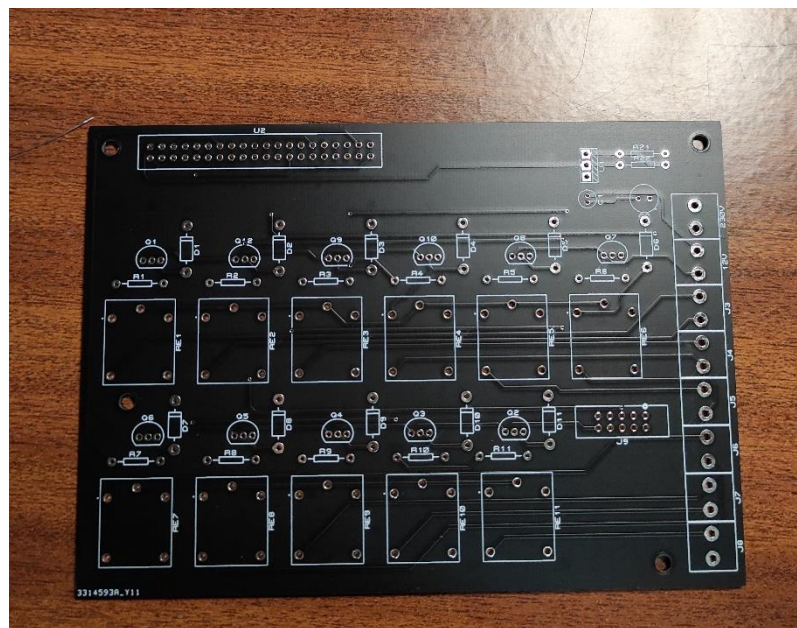


Figura 64 - PCB desenvolvida.

Após soldados os componentes necessários nos seus lugares predefinidos na PCB, obtém-se o resultado representado na Figura 65. É importante referir que, no momento da escrita desta dissertação de mestrado, ainda não foram implementados os martelos de picar os sinos, pelo que a utilização deste sistema fica restrita apenas aos sinos que têm martelo de picar operacional.

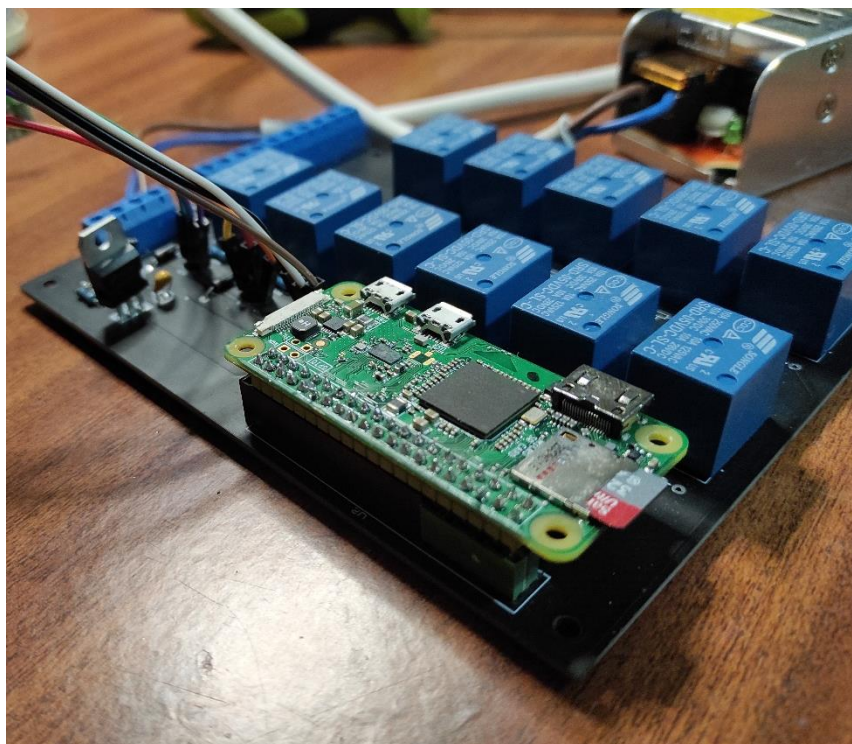


Figura 65 - Placa final.

Já para a construção da caixa, os ficheiros do SOLIDWORKS foram exportados em formato STL a fim de ser lidos por uma impressora 3D. Neste caso, o material da caixa a utilizar será PLA (*Polylactic acid*), utilizando uma impressora 3D

4.2. Testes

Antes de ser realizado qualquer tipo de testes, é necessário verificar o motivo pelo qual um martelo de picar e o mecanismo de horas não se encontram em funcionamento.

Como referido anteriormente, existia um martelo de picar que se encontrava com defeito. Ao analisar o martelo de picar e o quadro elétrico da Figura 56, constatou-se que o martelo tem como alimentação 230V em DC, daí a necessidade de utilização de uma ponte retificadora em cada comutador. Analisando mais detalhadamente o martelo, é possível afirmar o martelo não tem nenhum tipo de caixa ou motor que atue diretamente sobre o motor e tem uma mola de recolha. Após análise ao dispositivo, verificou-se que a mola de recolha não se encontrava a fazer a sua função: recolher o martelo à posição de origem. Após aperto da porca de afinação o martelo voltou a funcionar normalmente. A Figura 66 mostra o martelo na sua posição de origem após a sua afinação.



Figura 66 – Martelo na posição original.

Outro dispositivo que apresentava anomalias era o mecanismo dos mostradores. Este mecanismo transmite a hora para os 4 mostradores existentes no topo da torre. Após remoção do mecanismo para reparação, foi possível observar que o único motor existente no dispositivo está conectado a um sem-fim que por sua vez se encontra ligado a um conjunto de engrenagens e veios e, por sua vez, aos veios que transmitem as horas aos mostradores.

Após primeira análise do sistema, foi possível verificar que o sistema apresentava bastante oxidação, procedendo então à sua desmontagem e limpeza de todos os componentes e pintura da caixa, como mostra a Figura 67. Após limpeza dos vários componentes, verificou-se que o sistema voltou a funcionar, aplicando massa lubrificante a fim de aumentar a sua durabilidade com um funcionamento linear. A Figura 67 mostra o resultado final após manutenção.



Figura 67 - Mecanismo dos mostradores após manutenção.

De seguida, colocou-se o mecanismo no local inicial, faltando acertar cada mostrador do relógio. Esta tarefa não foi concluída pois as escadas de madeira existentes no topo da torre, à data de escrita desta dissertação, não permitiam as devidas condições de segurança.

Tendo todos os componentes que apresentavam defeito serem retificados, procedeu-se à montagem do dispositivo no local do dispositivo de controlo atual, apresentado na Figura 4.a. Os fios de ligação ao quadro da Figura 56 encontram-se identificados, pelo que a ligação ao sistema atual acaba por ser bastante simples. A Figura 67 mostra o dispositivo em testes. É importante referir que nesta primeira fase, o dispositivo não será conectado aos martelos de toque, mas sim a um ponto de luz. Este procedimento tem como finalidade verificar o correto funcionamento do dispositivo e evitar toques desnecessário dos sinos.

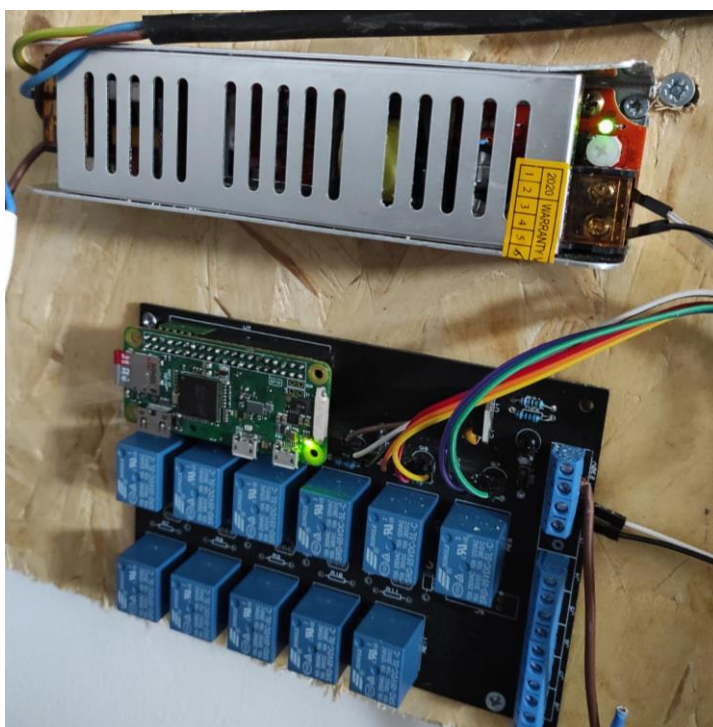


Figura 68 – Montagem de teste do dispositivo de controlo.

Um aspeto a ter em conta é que a rede WiFi do Raspberry Pi Zero tem de ser mudada para a rede WiFi existente na paróquia, sendo necessário enviar para o cartão SD do Raspberry Pi Zero um ficheiro com o nome “wpa_supplicant.conf” e com o código apresentado na Figura 69 no seu interior. À frente de “ssid = ” coloca-se o nome da rede e à frente de “psk = ” é colocada a palavra-passe e o Raspberry liga-se imediatamente à rede WiFi selecionada, podendo remotamente executar a melodia.

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
country=<Insert 2 letter ISO 3166-1 country code here>

network={
    ssid="<NetBoa>"
    psk="<1296027165>"
}
```

Figura 69 - Código de alteração de rede WiFi a conectar.

Para verificação do funcionamento do sistema, procedeu-se à configuração horária de alguns toques. O exemplo de um horário de teste do sistema encontra-se na Figura 70. Como é possível observar na Figura 70, para o exemplo de toques semanais, definiram-se os dias da semana para tocar, a hora pretendida e a melodia pretendida, neste caso, a melodia “testes”. Para testar que o sistema está em funcionamento, procedeu-se à ligação de umas luminárias LED às saídas da placa.

Change semana HISTORY

Introduzido: admin

Dias:

- Segunda-Feira
- Terça-Feira
- Quarta-Feira
- Quinta-Feira
- Sexta-Feira
- Sábado
- Domingo

Hora: 11:00 Now |
Note: You are 1 hour ahead of server time.

Melodia: led_blink1

Delete Save and add another Save and continue editing SAVE

Figura 70 – Configuração de um toque anual

Para os testes de toques anuais, procedeu-se da mesma forma que o teste anterior, como exposto na Figura 71. Como é possível verificar na Figura 71, os parâmetros definidos foram a data, hora e melodia.

WELCOME, ADMIN VIEW SITE / CHANGE PASSWORD / LOG OUT

led_blink.py | Dia de Corpo de Deus

Change ano HISTORY

Introduzido: admin

Data: 2021-06-04 Today | Note: You are 1 hour ahead of server time.

Hora: 16:00 Now | Note: You are 1 hour ahead of server time.

Melodia: led_blink

Comentário: Dia de Corpo de Deus

Delete Save and add another Save and continue editing SAVE

Figura 71 - Definição de toque semanal

Para os toques imediatos, bastou definir a melodia que desejada, conforme exposto na Figura 72.

Add imediato

Melodia: led_blink1

Save and add another Save and continue editing SAVE



Figura 72 - Definição de toque imediato.

A luminária LED acendeu e apagou 15 vezes, conforme o desejado, visto ser o que está definido na melodia de teste.

Para testar se o mostrador das horas está em funcionamento, foi ligado a luminária LED à saída definida para o mostrador das horas e verificou-se se em cada minuto decorrido o projetor acendia e apagava. Para o caso dos mostradores das horas, os testes e a integração ficaram por aqui, visto não se reunirem as condições de segurança necessárias no topo da torre, onde o dispositivo ficará localizado. A eventual integração do dispositivo será realizada quando se fizerem obras estruturais ao interior da torre.


Para o caso da iluminação, embora sejam feitos testes, a sua integração não será feita no momento da escrita desta dissertação, visto que os pontos de luz não foram trocados e o circuito elétrico revisto. Para testar a iluminação, basta recorrer ao separador da iluminação e definir os parâmetros de funcionamento. Para a iluminação semanal, como exposto na Figura 73, definiram-se os dias da semana em que se pretende que a iluminação funcione e o intervalo horário da iluminação.


Change semanais

Introduzido: admin  

Dias:

- Segunda-Feira
- Terça-Feira
- Quarta-Feira
- Quinta-Feira
- Sexta-Feira
- Sábado
- Domingo


Ligar: 02:17 Now | 
Note: You are 1 hour ahead of server time.

Desligar: 02:25 Now | 
Note: You are 1 hour ahead of server time.

Delete Save and add another Save

Figura 73 - Definição de tempo de funcionamento de Iluminação

Todos os parâmetros, tanto das melodias como da iluminação conseguem ser alterados na base de dados, bastando selecionar o horário de toque pretendido. Na Figura 74 é possível verificar alguns horários de toque definido para o toque semanal.

Select ano to change ADD ANO 

Action: 0 of 1 selected

<input type="checkbox"/>	ANO
<input type="checkbox"/>	2021-06-04 16:00:00 led_blink.py Dia de Corpo de Deus

1 ano

Figura 74 - Exemplo de lista de toques anuais.

4.3. Nota conclusiva

A recuperação dos dispositivos que apresentavam falhas de funcionamento, como por exemplo o martelo de picar ou o mecanismo que transmite as horas para o mostrador, permitiu analisar e verificar o princípio de funcionamento do mecanismo atual e analisar futuras alterações, a fim de melhorar o seu funcionamento e interface com o utilizador.

A aplicação do dispositivo em contexto real permite também que o utilizador final tenha uma primeira experiência com a sugestão de novo sistema e permite que sejam identificadas algumas anomalias, no código do sistema ou na base de dados, e permite que o utilizador final possa utilizar o sistema e dar algumas sugestões para melhorar o dispositivo.

5. Conclusões

5.1. Conclusões gerais

A ideia de desenhar e construir este novo sistema permitiu que a torre ganhasse, de novo, um lugar de destaque, relembrando que a necessidade de manutenção preventiva permite aumentar a durabilidade do dispositivo.

Embora não relacionado diretamente com a dissertação, a conceção deste projeto permitiu que fossem criadas novas ideias para utilização da torre, como por exemplo abrir ao público ou a criação de um museu da torre, visto ser um marco de destaque da cidade da Covilhã. Permitiu também o arranque de outras obras, como por exemplo a reconstrução das escadas em madeira existentes no topo da torre, fazendo com que quem sobe ao topo o possa fazer em segurança. Outra obra consequência direta deste projeto, foi a colocação de martelos de picar em todos os 9 sinos já colocados na torre mais os 3 sinos que não se encontram colocados, perfazendo assim um conjunto de 12 sinos com martelos de picar, e manutenção preventiva dos mesmos. Todas estas obras encontram-se em fase de orçamentação e análise, pelo que nenhuma destas arrancou no momento de escrita desta dissertação, embora esteja previsto a sua execução. A análise de um novo sistema de iluminação também se encontra em avaliação, à data da escrita desta dissertação.

Quanto ao dispositivo, pode-se afirmar que, para primeira versão, é perfeitamente capaz de atingir os objetivos propostos. É uma solução que consegue perfeitamente substituir o dispositivo atual, dando toda a liberdade ao utilizado para programar todos os horários diretamente do seu telemóvel ou computador ou executar melodias predefinidas imediatamente.

No entanto, embora seja um dispositivo bastante completo, cada melodia que é adicionada ao sistema tem de ser programada manualmente, fazendo com que o utilizador perca alguma liberdade de escolha de melodias ou dificultando a colocação das mesmas. Outro problema identificado tem a ver com a utilização dos comutadores já existentes. Embora a funcionar, os comutadores apresentam já alguns problemas de funcionamento, tornando o repique do sino irregular. Este problema, no entanto, pode ser resolvido com um novo comutador.

Apesar de ainda haver um longo caminho a percorrer, o dispositivo apresentado irá permitir que a torre sineira ganhe uma nova posição de destaque, onde a inovação é trabalhada em conjunto com a preservação de história.

5.2. Sugestões de trabalhos futuros

Tal como tudo, este dispositivo tem alguns problemas a resolver e podem ser adicionadas algumas funcionalidades a fim de aumentar a sua fidelidade e competitividade com outros dispositivos existentes ou desenvolvimento neste mercado tão pequeno.

Um dos aspetos a melhorar é o layout utilizado para apresentar as listagens de toques e de iluminação. Embora seja um layout que cumpre a função, é padronizado, pelo que a alteração do layout da página apresentada seria uma mais-valia para o dispositivo. Outra consideração a adicionar para facilitar a interação entre o dispositivo e o utilizador, seria a utilização de uma APP para gerir as listagens de toque do dispositivo, dispensando a necessidade de utilizar diretamente um navegador, digitar o IP do dispositivo e colocar o nome de utilizador e palavra-passe. Nas melodias de toque, um processo que convertesse automaticamente uma pauta para ficheiro Python também seria uma mais-valia para o sistema, evitando assim a necessidade de um programar cada melodia individualmente.

Como é possível verificar na Figura 56, o quadro do dispositivo atual é de grandes dimensões, podendo ficar ainda maior, caso sejam adicionados mais martelos para todos os sinos, recuperando a totalidade das notas reproduzidas. Uma possibilidade para um desenvolvimento futuro é o desenvolvimento de uma unidade única, com utilização de eletrónica de potência, com recurso à utilização de, por exemplo, transístores IGBT ou MOSFET, tornando assim o espaço requerido pelo funcionamento do dispositivo menor.

Outra sugestão é relativa à inclusão de uma unidade de energia de emergência, a fim de evitar que o dispositivo se desligue com o corte de energia e tenha de iniciar quando volta a energia. Por norma, a WiFi também demora mais tempo a iniciar, tendo o dispositivo de esperar para voltar a obter as horas pela internet para fique certo. Outro benefício de ter uma unidade de energia de emergência seria que os mostradores se manteriam em funcionamento, evitando assim os possíveis desacertos dos mostradores e em funcionamento.

6. Referências Bibliográficas

- [1] A. G. Borges, História da Freguesia de S. Pedro da Covilhã, Covilhã: Junta de Freguesia de São Pedro, 2009.
- [2] D. F. Pinto, “De Campanis Fundentis - A fábrica de Fundição de Sinos de Rio,” Faculdade de Letras da Universidade do Porto, Porto, 2019.
- [3] A. Chaves, A. D. Cranmer, I. Yglesias de Oliveira, J. Soerio de Oliveira, L. Rombouts, L. Marreiros, M. G. Cordeiro e V. Debut, Carrilhões de MAfra, 1^o ed., Mafra: Câmara Municipal de Mafra / Direção-Geral do Património Cultural, 2020, p. 98.
- [4] J. Wagner, C. Huey, K. Knaub, E. Volk e A. Jagarwal, “Modeling and analysis of a weight driven mechanical tower clock,” *Proceedings of the 2010 American Control Conference, ACC 2010*, pp. 634-639, 2010.
- [5] A Fundição de Sinos de Braga - Serafim da Silva jerónimos & Filhos, LDA , “Exactus,” A Fundição de Sinos de Braga - Serafim da Silva jerónimos & Filhos, LDA , [Online]. Available: <https://www.jeronimobraga.com/catalogo/relogio-de-torre-exactus/>. [Acedido em 30 dezembro 2020].
- [6] Bodet, “Sigma Master Clocks,” [Online]. Available: https://static.bodet-time.com/images/stories/Pdfs/EN/Brochures/653071_Master_clock_SIGMA.pdf. [Acedido em 31 dezembro 2020].
- [7] Bordet, “Características Técnicas Sigma H Master Clocks,” [Online]. Available: https://static.bodet-time.com/images/stories/Pdfs/EN/Technical-features/Sigma/643311_Master_clock_Sigma_H.pdf. [Acedido em 03 janeiro 2021].
- [8] Bodet, “Características Técnicas Sigma P Master Clock,” [Online]. Available: [https://static.bodet-time.com/images/stories/Pdfs/EN/Technical-](https://static.bodet-time.com/images/stories/Pdfs/EN/Technical-features/Sigma/643311_Master_clock_Sigma_P.pdf)

- features/Sigma/643321_Master_clock_programmable_Sigma_P.pdf. [Acedido em 03 janeiro 2021].
- [9] Bodet, “Características Técnicas Sigma C Master Clocks,” [Online]. Available: https://static.bodet-time.com/images/stories/Pdfs/EN/Technical-features/Sigma/643L71_Sigma_C_Master_clock.pdf. [Acedido em 3 janeiro 2021].
- [10] Bodet, “Características Técnicas - Sigma Mod Master Clock,” [Online]. Available: https://static.bodet-time.com/images/stories/Pdfs/EN/Technical-features/Sigma/643331_Modular_master_clock_Sigma_Mod.pdf. [Acedido em 03 janeiro 2021].
- [11] Bodet, “Sigma Software,” [Online]. Available: <https://static.bodet-time.com/images/stories/Pdfs/EN/Manuals/Sigma/607752-User-manual-Sigma-software.pdf>. [Acedido em 04 janeiro 2021].
- [12] Bodet, “Microquartz Manual de Operação,” [Online]. Available: https://static.bodet-time.com/images/stories/Pdfs/EN/Manuals/Sigma/Old/604505_Microquartz_S_Manual.pdf. [Acedido em 04 janeiro 2021].
- [13] Y. B. ., Z. M. Z. Zhiyong Zou, “Design of Wireless Bell Control System Based on Microcontroller,” em *2017 IEEE 2nd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, Chengdu, China, 2017.
- [14] R. Kannan Megalingam, V. K. B. . M. Muralidharan Nair, V. S. Venugopala Sarma e R. Sri Kumar , “Power Aware Automatic Microcontroller Based Smart, College Electric Bell System with Time Display,” em *Fifth International Conference on MEMS NANO, and Smart Systems*, 2009.
- [15] W. Mouakkassa, “Trasnportable Bell Arrangement”. Australia Patente AU 2002100205 A4, 14 03 2002.
- [16] F. K. [US] e S. G. L. [US], “Electronic carillon system and sequencer module therefor”. Estados Unidos Patente US5471006A, 1995.
- [17] EFAPEL, “Catálogo Geral,” [Online]. Available: <https://www.efapel.pt/media/Geral-5efc9164164ae.pdf>. [Acedido em 05 janeiro 2021].

- [18] Hager, “Ficha Técnica EE200,” [Online]. Available: https://www.hager.pt/files/download/o/30308_1/o/Ficha%20t%C3%A9cnica%20EE200.pdf. [Acedido em 04 janeiro 2021].
- [19] Philips, “DDRC820FR-CS-BT,” [Online]. Available: http://images.philips.com/is/content/PhilipsConsumer/PDFDownloads/Colombia/technical-sheets/ODLI20180227_001-UPD-es_CO-DDRC820FR-CS-BT_Data_Sheet.pdf. [Acedido em 07 janeiro 2021].
- [20] Texas Instruments, *Tiva™ TM4C123GH6PM Microcontroller DATASHEET*, Texas Instruments, 2014.
- [21] Espressif Systems, *ESP8266EX Datasheet*, Espressif Systems, 2020.
- [22] Raspberry, “Raspberry Pi Zero W,” [Online]. Available: <https://www.raspberrypi.org/products/raspberry-pi-zero-w/>. [Acedido em 29 janeiro 2021].
- [23] Mike, “The Anti-Kyte,” [Online]. Available: <https://mikesmithers.wordpress.com/2017/02/21/configuring-django-with-apache-on-a-raspberry-pi/>. [Acedido em 20 01 2021].
- [24] Songle Relay, “SRD-05VDC-SL-C-Datasheet”.
- [25] ON Semiconductor, “P2N2222A”.
- [26] Texas Instruments, “LM138 and LM338 5-Amp Adjustable Regulators”.
- [27] LEDUP, “Fonte Metal SLIM 5A 60W,” [Online]. Available: http://ledup.pt/index.php/ficha_tecnica/643. [Acedido em 14 05 2021].
- [28] GeeksforGeeks, “GeeksforGeeks,” [Online]. Available: <https://www.geeksforgeeks.org/tuples-in-python/>. [Acedido em 06 junho 2021].