



## Universidades Lusíada

Fernandes, Daniel Filipe Marques

### **Desenvolvimento de um sistema de ignição pirotécnica**

<http://hdl.handle.net/11067/5975>

#### **Metadata**

**Issue Date** 2021

**Abstract** Na indústria da pirotecnia, a composição e logística de um espetáculo pirotécnico é essencial de forma a proporcionar aos espetadores um espetáculo que demonstre uma boa coordenação dos artigos que são apresentados. Como tal, os sistemas de disparo sem fios têm um papel fundamental no que toca à coordenação e controlo do espetáculo, contribuindo ainda para aumentar a segurança dos operadores pirotécnicos. Este trabalho apresenta o desenvolvimento de um sistema de disparo de ignitores elétricos,...

In the pyrotechnics industry, the composition and logistics of a pyrotechnic show is essential to provide spectators with a show that demonstrates good coordination of the articles that are presented. As such, wireless firing systems play a key role in terms of coordination and control of the show, also contributing to increasing the safety of pyrotechnic operators. This work shows the development of a system for firing electric igniters, are responsible for the ignition of pyrotechnic articles...

**Keywords** Engenharia, Indústria pirotécnica

**Type** masterThesis

**Peer Reviewed** no

**Collections** [ULF-FET] Dissertações

This page was automatically generated in 2022-07-22T02:26:12Z with information provided by the Repository



**UNIVERSIDADE LUSÍADA – NORTE**  
***Campus de Vila Nova de Famalicão***

**Desenvolvimento de um Sistema de Ignição Pirotécnica**

**Daniel Filipe Marques Fernandes**

Dissertação para obtenção do Grau de Mestre  
em Engenharia Eletrónica e Informática

Janeiro de 2021



**UNIVERSIDADE LUSÍADA – NORTE**  
***Campus de Vila Nova de Famalicão***

## **Desenvolvimento de um Sistema de Ignição Pirotécnica**

**Daniel Filipe Marques Fernandes**

Dissertação para obtenção do Grau de Mestre  
em Engenharia Eletrónica e Informática

Orientadores: Prof. Doutor Vítor Pereira  
Prof. Doutor Pedro Reis

Janeiro de 2021

## **Agradecimentos**

Com o fim deste projeto tenho a agradecer a todos os colegas e professores que me acompanharam durante toda a minha vida acadêmica.

Quero agradecer aos meus pais, uma vez que sem o apoio deles não seria possível chegar onde cheguei.

Um especial agradecimento à minha namorada, que sempre me apoiou nos momentos difíceis que surgiram durante o desenvolvimento do projeto, pela paciência que teve para comigo e por acreditar em mim e nos meus sonhos. Agradeço também aos seus pais.

Agradecer ainda à minha irmã, afilhadas, cunhado, aos meus padrinhos e à restante família que sempre me apoiou.

À empresa Douro Pirotecnia, que se disponibilizou para ajudar e acompanhar o desenvolvimento deste projeto.

Por fim, quero deixar um agradecimento aos meus orientadores, Professor Doutor Vítor Pereira e Professor Doutor Pedro Reis, que sempre se mostraram disponíveis para ajudar no desenvolvimento deste projeto.

## Índice

Agradecimentos.....	i
Resumo .....	iv
Abstract .....	v
Índice de Figuras.....	vi
Índice de Tabelas .....	ix
Lista de Abreviaturas, Acrónimos e Siglas .....	x
1. Introdução .....	1
1.1. A empresa Douro Pirotecnia.....	1
1.2. Enquadramento e motivação .....	2
1.3. Objetivos.....	2
1.4. Estrutura da dissertação .....	3
2. Enquadramento Teórico.....	4
2.1. Indústria pirotécnica .....	4
2.2. Funcionamento de um artigo pirotécnico .....	6
2.3. Funcionamento de um ignitor elétrico.....	7
2.4. Organização de um espetáculo pirotécnico .....	12
2.5. Sistemas existentes no mercado .....	15
2.5.1. Sistema Cobra Firing Systems LLC. ....	16
2.5.2. Sistema Liuyang Happiness Firing Systems CO., LTD .....	17
2.5.3. Sistema Galaxis Showtechnik GmbH.....	19
2.5.4. Sistema StarFire.....	22
2.6. Comunicação sem fios .....	23
3. Componentes e Dispositivos .....	26
3.1. MOSFET IRF9530.....	26
3.2. Multiplexer/Demultiplexer .....	28
3.3. Sensor DHT11.....	30
3.4. HMI NEXTION.....	31
3.5. Acoplador ótico PC817 .....	32
3.6. Células de Iões de Lítio (18650).....	33
3.7. Reguladores e conversores de tensão.....	35
3.8. <i>Battery management system</i> .....	38
3.9. Arduino Mega 2560 .....	39
3.10. Módulo Rádio Ai-Tinker Ra-02.....	40

3.11.	Módulo SD Card Velleman .....	41
3.12.	Conversores de nível lógico Sparkfun.....	42
3.13.	Transístor 2N3904.....	43
3.14.	Fusíveis de proteção .....	44
4.	Desenvolvimento do Projeto.....	45
4.1.	Descrição do desenvolvimento do projeto.....	45
4.2.	Testes .....	69
5.	Descrição do Sistema de Ignição Pirotécnica .....	73
5.1.	Software.....	74
5.1.1.	Encriptação .....	74
5.1.2.	Descrição do funcionamento do Emissor .....	75
5.1.3.	Descrição do funcionamento do Recetor .....	89
5.1.4.	Configurações e instruções do HMI (Emissor e Recetor) .....	96
5.1.5.	Estrutura dos pacotes de dados .....	101
5.2.	Hardware .....	104
5.2.1.	Esquema elétrico do emissor .....	104
5.2.2.	Esquema elétrico do recetor .....	108
5.2.3.	Esquema elétrico da régua .....	111
5.2.4.	Descrição de cada pino do conector DB25.....	115
6.	Conclusões e Trabalho Futuro .....	117
	Referências .....	121
	Apêndice A - Pseudocódigo referente ao programa dos módulos Emissor e Recetor .....	124
	Apêndice B - Características do HMI Nextion 7" Enhanced Series.....	137
	Apêndice C - Características do HMI Nextion 2.8" Basic Series .....	138
	Apêndice D - Características do Arduino Mega 2560 Pro mini .....	139
	Apêndice E - Tabela de designação e descrição de cada componente dos esquemas elétricos do módulo emissor (Figura 99 e Figura 100) .....	140
	Apêndice F - Tabela de designação e descrição de cada componente dos esquemas elétricos do módulo recetor e régua (Figura 100, Figura 101 e Figura 103) .....	141

## Resumo

Na indústria da pirotecnia, a composição e logística de um espetáculo pirotécnico é essencial de forma a proporcionar aos espetadores um espetáculo que demonstre uma boa coordenação dos artigos que são apresentados. Como tal, os sistemas de disparo sem fios têm um papel fundamental no que toca à coordenação e controlo do espetáculo, contribuindo ainda para aumentar a segurança dos operadores pirotécnicos.

Este trabalho apresenta o desenvolvimento de um sistema de disparo de ignitores elétricos, responsáveis pela ignição dos artigos pirotécnicos. O sistema é constituído por emissor, recetores e réguas de terminais, nas quais são ligados os ignitores elétricos.

Este projeto foi desenvolvido na ótica do utilizador, com o intuito de ser fácil de operar, flexível e com fiabilidade. Foi elaborado inicialmente um planeamento de certas características que seriam fundamentais no projeto; contudo, ao longo do mesmo foram adicionadas outras características, de forma a tornar o sistema ainda mais completo. Foram desenvolvidos três modos de controlo do espetáculo, de forma a tornar o sistema mais flexível: modo automático, modo semiautomático e modo manual.

No que toca ao controlo, o sistema tem a possibilidade de controlar (disparar) até 960 números totalmente independentes no modo automático e até 16384 números independentes nos modos semiautomático e manual. O emissor possui a capacidade de analisar os dados de vários recetores remotamente.

Em vez dos terminais (onde se ligam os ignitores) estarem no próprio recetor, sendo esta a arquitetura usada na maior parte dos sistemas existentes no mercado, foi desenvolvido um sistema de réguas com terminais que são conectadas ao recetor através de um cabo. Esta característica possui diversas vantagens, uma vez que fará com que as réguas possam estar mais perto dos ignitores, fazendo com que haja economia no fio gasto para ligar os ignitores aos recetores. Outra vantagem é que permite que o recetor fique mais afastado dos artigos pirotécnicos, oferecendo assim mais segurança ao operador.

**Palavras-chave:** Indústria Pirotécnica, Ignitor Elétrico, Sistema de Ignição Pirotécnica, LoRa, Arduino

## **Abstract**

In the pyrotechnics industry, the composition and logistics of a pyrotechnic show is essential to provide spectators with a show that demonstrates good coordination of the articles that are presented. As such, wireless firing systems play a key role in terms of coordination and control of the show, also contributing to increasing the safety of pyrotechnic operators.

This work shows the development of a system for firing electric igniters, are responsible for the ignition of pyrotechnic articles. The firing system consists of emitter, receivers, and terminal strips, where the electrical igniters that ignite the pyrotechnic articles are connected.

This project was developed from the user's perspective, to be easy to operate, flexible and reliable. Initially, a planning of certain characteristics that would be fundamental in the project was prepared; however, with its development, other features were added to make the system even more complete.

Three modes of show control were developed to make the system more flexible: automatic mode, semi-automatic mode, and manual mode.

As far as control is concerned, the system has the possibility to control (fire) up to 960 totally independent numbers in automatic mode and up to 16384 independent numbers in semi-automatic and manual mode. The emitter the ability to analyse the data from multiple receivers remotely.

Instead of the terminals (where the electric igniters are connected) being on the receiver itself, which is the architecture used in most of the existing systems on the market, a system of strips with terminals was developed, which are connected to the receiver via a cable.

This feature has several advantages, since it will allow the strips to be closer to the igniters, saving the wire used to connect the igniters to the receivers. Another advantage is that it allows the receiver to be further away from the pyrotechnic articles, thus offering more security to the operator.

**Keywords:** Pyrotechnic Industry, Electric Ignitor, Pyrotechnic Ignition System, LoRa, Arduino



## Índice de Figuras

FIGURA 1. LOGOTIPO DA EMPRESA DOURO PIROTECNIA (FONTE: DOURO PIROTECNIA ).....	1
FIGURA 2. PREPARAÇÃO DE UMA COR (FONTE: YOUTUBE ) .....	5
FIGURA 3. PREPARAÇÃO DE TIROS (FONTE: YOUTUBE ) .....	5
FIGURA 4. ESTÁGIOS DE UM ARTIGO PIROTÉCNICO (BALONA) .....	6
FIGURA 5. LANÇAMENTO DE UM FOGUETE DE CANA (COMPOSTO POR UM CANUDO QUE FAZ IMPULSÃO) (FONTE: FLORÊNCIO PEREIRA, HELDER FERREIRA, RUI CAMACHO ).....	7
FIGURA 6. EXEMPLO DE UM IGNITOR ELÉTRICO (FONTE: META PYRO ) .....	8
FIGURA 7. IGNIÇÃO DE UM IGNITOR ELÉTRICO (FONTE: META PYRO ) .....	8
FIGURA 8. INSERÇÃO DE UM IGNITOR ELÉTRICO NUM ARTIGO PIROTÉCNICO.....	9
FIGURA 9. EXEMPLO DE ASSOCIAÇÃO EM SÉRIE DE IGNITORES ELÉTRICOS NO MESMO TERMINAL (ADAPTADO DA FONTE: META PYRO , P.5).....	10
FIGURA 10. EXEMPLO DE UMA ASSOCIAÇÃO EM PARALELO DE IGNITORES ELÉTRICOS NO MESMO TERMINAL (ADAPTADO DA FONTE: METAPYRO , P.5) .....	10
FIGURA 11. MONTAGEM DE UM ESPETÁCULO PIROTÉCNICO .....	14
FIGURA 12. RECETOR DB24R-X COM IGNITORES LIGADOS EM ALGUNS NÚMEROS .....	14
FIGURA 13. MÓDULO EMISSOR 18R2 DA EMPRESA COBRA FIRING SYSTEMS LLC. (FONTE: COBRA FIRING SYSTEMS LLC ) .....	17
FIGURA 14. EMISSOR DBR02-X DA EMPRESA LIUYANG HAPPINESS FIRING SYSTEMS CO.,LTD (FONTE: AMAZON ) .....	18
FIGURA 15. MÓDULO RECETOR DB24R-X DA EMPRESA LIUYANG HAPPINESS FIRING SYSTEMS CO.,LTD (FONTE: AMAZON ) ...	19
FIGURA 16. MÓDULO EMISSOR PFC ADVANCE BLACK EDITION DA EMPRESA GALAXIS SHOWTECHNIK GMBH (FONTE: GALAXIS SHOWTECHNIK GMBH).....	20
FIGURA 17. SOFTWARE GALAXIS PYROTEC COMPOSER DA EMPRESA GALAXIS SHOWTECHNIK GMBH (FONTE: GALAXIS SHOWTECHNIK GMBH ) .....	20
FIGURA 18. MÓDULO RECETOR PFE ADVANCED 10 OUTPUTS DA EMPRESA GALAXIS SHOWTECHNIK GMBH (FONTE: GALAXIS SHOWTECHNIK GMBH ) .....	21
FIGURA 19. RÉGUA DE TERMINAIS MATRIX MODULE 100 + MÓDULO RECETOR PFE ADVANCED 10 OUTPUTS DA EMPRESA GALAXIS SHOWTECHNIK GMBH (FONTE: GALAXIS SHOWTECHNIK GMBH ).....	21
FIGURA 20. MÓDULO DE CONTROLO STARFIRE DIGITAL FIRING SYSTEM (FONTE: ACE PYRO ) .....	22
FIGURA 21. MÓDULO RECETOR STARFIRE FIELD MODULE (FONTE: ACE PYRO ) .....	23
FIGURA 22. RÉGUAS DE TERMINAIS FIRING SYSTEM RAILS (FONTE: STARFIRESALES ) .....	23
FIGURA 23. ESTRUTURA DO PACOTE LORA (FONTE: SEMTECH , P.29).....	25
FIGURA 24. SIMBOLOGIA MOSFET TIPO N E TIPO P OU CANAL N E CANAL P (FONTE: PHYSICS ) .....	27
FIGURA 25. MOSFET IRF9530 (FONTE: AQUÁRIO ).....	28
FIGURA 26. MULTIPLEXER 16 CANAIS CD74HC4067 (FONTE: SPARKFUN ).....	29
FIGURA 27. SENSOR DE TEMPERATURA E HUMIDADE DHT11 ( <a href="https://ardushop.ro/681-LARGE_DEFAULT/DHT11-DIGITAL-TEMPERATURE-AND-HUMIDITY-SENSOR-MODULE.JPG">HTTPS://ARDUSHOP.RO/681-LARGE_DEFAULT/DHT11-DIGITAL-TEMPERATURE-AND-HUMIDITY-SENSOR-MODULE.JPG</a> ) .....	30
FIGURA 28. HMI NEXTION 7" ENHANCED SERIES (FONTE: NEXTION).....	32
FIGURA 29. HMI NEXTION 2,8" BASIC SERIES (FONTE: NEXTION).....	32
FIGURA 30. ESQUEMA ELÉTRICO INTERNO DO ACOPLADOR ÓTICO PC817 (FONTE: ALL TOP NOTCH ) .....	33
FIGURA 31. ACOPLADOR ÓTICO PC817 DA SHARP (FONTE: MAKERS ).....	33
FIGURA 32. CÉLULA 18650 SAMSUNG 18650-35E (FONTE: BALTRADE ) .....	34
FIGURA 33. CÉLULA 18650 PANASONIC NCR18650B (FONTE: ESTUDIOPT ) .....	34
FIGURA 34. TÍPICA APLICAÇÃO DO REGULADOR LINEAR LM117 PARA OBTENIR UMA TENSÃO DE SAÍDA REGULADA PELO POTENCIÓMETRO R2 (FONTE: TEXAS INSTRUMENTS ).....	35
FIGURA 35. MÓDULO STEP DOWN LM2596 (FONTE: BOTNROLL ) .....	36
FIGURA 36. CIRCUITO TÍPICO COM LM2576 COM SAÍDA AJUSTÁVEL (FONTE: MOUSER ) .....	36
FIGURA 37. MÓDULO CONVERSOR STEP DOWN MINI-360 (FONTE: MAUSER ) .....	37
FIGURA 38. MÓDULO STEP DOWN XHM401 (FONTE: AMAZON ) .....	37
FIGURA 39. BMS 6S (FONTE: BANGOOD ).....	38
FIGURA 40. BMS 3S (FONTE: BANGOOD ).....	39
FIGURA 41. ARDUINO MEGA 2560 PRO MINI (FONTE: DIGIPART ) .....	40
FIGURA 42. MÓDULO DE RÁDIO RA-02 (FONTE: ELECTRONICS-LAB ) .....	41
FIGURA 43. MÓDULO VMA304 DA VELLEMAN PARA CARTÕES SD (FONTE: VELLEMAN ) .....	42
FIGURA 44. MÓDULO CONVERSOR NÍVEL LÓGICO BIDIRECIONAL (FONTE: ELETROFUN ) .....	42
FIGURA 45. ESQUEMA COM SENTIDOS DAS CORRENTES NOS TRANSÍSTORES BIPOLARES NPN E PNP (FONTE: RAFFAELE ILARDO ).....	43
FIGURA 46. DESCRIÇÃO DOS PINOS E ESQUEMA DO TRANSÍSTOR BIPOLAR PNP 2N3904 (FONTE: COMPONENTS101 ) .....	44

FIGURA 47. BATERIA COMPOSTA POR SEIS CÉLULAS EM SÉRIE PANASONIC NCR18650B E UMA BMS 6S .....	46
FIGURA 48. CIRCUITO DE TESTE DESENVOLVIDO PARA DUAS SAÍDAS DA PRIMEIRA RÉGUA DE TERMINAIS .....	47
FIGURA 49. MONTAGEM DA PLACA DESENVOLVIDA BASEADA NO CIRCUITO DA FIGURA 48 .....	49
FIGURA 50. RÉGUA DE SEIS TERMINAIS, COM SHUNT EM CADA TERMINAL, DE FORMA A PODER OBSERVAR O TESTE DE CONTINUIDADE DE CADA TERMINAL .....	49
FIGURA 51. CIRCUITO DE TESTE RETIFICADO COM OS MOSFETS TIPO P.....	51
FIGURA 52. MÓDULO DE CATÕES MICROSD .....	51
FIGURA 53. MÓDULO PARA CARTÕES SD VELLEMAN VMA304.....	53
FIGURA 54. MAPEAMENTO DOS PONTOS DE TESTE DE COMUNICAÇÃO .....	54
FIGURA 55. PONTO 1 ONDE FOI COLOCADO O MÓDULO EMISSOR NO PRIMEIRO TESTE AOS MÓDULOS DE RÁDIO RA-02 .....	55
FIGURA 56. PONTO 2 – PRIMEIRO LOCAL DE COLOCAÇÃO DO RECETOR PARA TESTES DO MÓDULO DE RÁDIO RA-02 .....	55
FIGURA 57. PONTO 3 - SEGUNDO LOCAL DE COLOCAÇÃO DO RECETOR PARA TESTES DO MÓDULO DE RÁDIO RA-02.....	56
FIGURA 58. PONTO 4 - TERCEIRO LOCAL DE COLOCAÇÃO DO RECETOR PARA TESTES DO MÓDULO DE RÁDIO RA-02 .....	56
FIGURA 59. PONTO 5 - QUARTO LOCAL DE COLOCAÇÃO DO RECETOR PARA TESTES DO MÓDULO DE RÁDIO RA-02 .....	57
FIGURA 60. PONTO 6 - QUINTO LOCAL DE COLOCAÇÃO DO RECETOR PARA TESTES DO MÓDULO DE RÁDIO RA-02 .....	57
FIGURA 61. CONSUMO DE CORRENTE DO EMISSOR, COM TENSÃO DE 7 V DE ENTRADA, BRILHO MÁXIMO E QUANDO ESTE SE ENCONTRA NO MENU PRINCIPAL .....	58
FIGURA 62. CONSUMO DE CORRENTE DO RECETOR, COM TENSÃO DE 5 V DE ENTRADA, SEM AS RÉGUAS DE TERMINAIS APLICADAS, COM O BRILHO NO MÁXIMO E QUANDO ESTE SE ENCONTRA NA PÁGINA INICIAL.....	58
FIGURA 63. CIRCUITO COM O CONVERSOR BUCK LM2576, COM OUTPUT DE 7 V REGULADO PELO POTENCIÓMETRO (O CIRCUITO FOI ELABORADO DE ACORDO COM O DATASHEET DO FABRICANTE).....	60
FIGURA 64. TESTE COM CONVERSOR LINEAR LM117 REGULADO PARA UMA TENSÃO DE SAÍDA DE 7 V (CIRCUITO ELABORADO DE ACORDO COM DATASHEET DO FABRICANTE).....	61
FIGURA 65. COMPARAÇÃO ENTRE OS QUATRO CONVERSORES DE TENSÃO (MINI-360, LM2596, LM2576 E LM317), ONDE APRESENTA A EFICIÊNCIA DE CADA UM EM FUNÇÃO DA TENSÃO (9 V ATÉ 12,6 V). (TESTES REALIZADOS EM CADA UM COM UMA SAÍDA FIXA DE 7 V E 500 MA).....	61
FIGURA 66. COMPARAÇÃO ENTRE OS QUATRO CONVERSORES DE TENSÃO (MINI-360, LM2596, LM2576 E LM317), ONDE APRESENTA A EFICIÊNCIA DE CADA UM EM FUNÇÃO DA TENSÃO (18 V ATÉ 25.2 V). (TESTES REALIZADOS EM CADA UM COM UMA SAÍDA FIXA DE 5 V E 166 MA).....	63
FIGURA 67. ELABORAÇÃO DA PLACA ELETRÓNICA NO SOFTWARE EAGLE .....	64
FIGURA 68. IMPRESSÃO DO PAINEL DA RÉGUA DE TERMINAIS .....	64
FIGURA 69. MONTAGEM FINAL DA RÉGUA DE TERMINAIS.....	65
FIGURA 70. RÉGUA DE TERMINAIS CONCLUÍDA.....	65
FIGURA 71. MONTAGEM DA BATERIA COM A BMS NO CHASSI DO RECETOR E O CONVERSOR BUCK XHM401 .....	66
FIGURA 72. MONTAGEM DE TODOS OS COMPONENTES DO RECETOR .....	66
FIGURA 73. MÓDULO RECETOR FINALIZADO .....	66
FIGURA 74. ELABORAÇÃO E MONTAGEM DA PLACA ELETRÓNICA DO EMISSOR.....	67
FIGURA 75. MÓDULO EMISSOR FINALIZADO .....	67
FIGURA 76. ESQUEMA GERAL DO PROJETO DESENVOLVIDO .....	74
FIGURA 77. PÁGINA 0 - APRESENTAÇÃO DO EMISSOR COM O NOME DA APLICAÇÃO .....	75
FIGURA 78. PÁGINA1 - MENU PRINCIPAL .....	77
FIGURA 79. PÁGINA 2 - MODO AUTOMÁTICO .....	79
FIGURA 80. PÁGINA 3 - DISPARO AUTOMÁTICO.....	81
FIGURA 81. PÁGINA 4 - SEMIAUTOMÁTICO .....	82
FIGURA 82. PÁGINA 5 -CONFIGURAÇÕES.....	83
FIGURA 83. PÁGINA 8 - INFO RECETORES .....	84
FIGURA 84. PÁGINA 6 - DADOS DOS RECETORES .....	85
FIGURA 85 PÁGINA 7 - DADOS DO RECETOR.....	86
FIGURA 86. PÁGINA 9 - MODO MANUAL .....	87
FIGURA 87. PÁGINA 10 - AVISOS E ERROS.....	88
FIGURA 88. PÁGINA 11 - PROTEÇÃO PARA OS MODOS DE DISPARO .....	89
FIGURA 89. FLUXOGRAMA BÁSICO DO FUNCIONAMENTO DO RECETOR.....	92
FIGURA 90. PÁGINA 0 - APRESENTAÇÃO DO RECETOR COM O NOME DA APLICAÇÃO .....	93
FIGURA 91 PÁGINA 1 -PÁGINA PRINCIPAL DO RECETOR .....	94
FIGURA 92. PÁGINA 3 - "ALTERAR CANAL" .....	94
FIGURA 93. PÁGINA 4 - "ALTERAR ID EMISSOR" .....	95
FIGURA 94. PÁGINA 5 - "ALTERAR ID RECETOR" .....	95

FIGURA 95. PÁGINA 6 - CONFIGURAÇÕES .....	96
FIGURA 96. PAYLOAD DO PACOTE ENVIADO DO EMISSOR PARA O RECETOR .....	102
FIGURA 97. PAYLOAD DO PACOTE ENVIADO DO RECETOR PARA O EMISSOR QUANDO É FEITO UM PEDIDO DE INFORMAÇÕES BÁSICAS .....	103
FIGURA 98. PAYLOAD ENVIADO DO RECETOR PARA O EMISSOR QUANDO É FEITO UM PEDIDO DE INFORMAÇÕES DETALHADAS ..	104
FIGURA 99. ESQUEMA ELÉTRICO DO MÓDULO EMISSOR .....	107
FIGURA 100. ESQUEMA ELÉTRICO DO PAINEL DO MÓDULO EMISSOR .....	107
FIGURA 101. ESQUEMA ELÉTRICO DO MÓDULO RECETOR.....	110
FIGURA 102. ESQUEMA ELÉTRICO DO PAINEL DO MÓDULO RECETOR .....	110
FIGURA 103. RELAÇÃO DE TRANSFERÊNCIA DE CORRENTE (CTR) EM FUNÇÃO DA CORRENTE $IF$ , DO ACOPLADOR ÓTICO PC817 (ALTERADO DA FONTE: FARNELL) .....	114
FIGURA 104. ESQUEMA ELÉTRICO DA RÉGUA (DENTRO DO QUADRADO AZUL ESTÁ REPRESENTADO O ESQUEMA DE UM NÚMERO, OS RESTANTES 15 SÃO REPETIDOS, COM A DIFERENÇA DE LIGAÇÃO A OUTROS PINOS DOS MUX E DEMUX) .....	115
FIGURA 105. PINOUT DO CONECTOR DB25 FÊMEA E MACHO (FONTE DOITYOURSELFCHRISTMAS).....	116

## Índice de Tabelas

TABELA 1. CARACTERÍSTICAS TÉCNICAS DE ALGUNS DOS IGNITORES ELÉTRICOS PRESENTES NO MERCADO.....	12
TABELA 2. TABELA VERDADE PARA SELECIONAR CADA UMA DAS 16 SAÍDAS/ENTRADAS RELATIVA AO MULTIPLEXER CD74HC4067 (FONTE: TEXAS INSTRUMENTS , P.2).....	29
TABELA 3. ESPECIFICAÇÕES DAS CÉLULAS 18650 SAMSUNG INR 18650-35E E PANASONIC NCR18650B.....	34
TABELA 4. CARACTERÍSTICAS DO MÓDULO DE RÁDIO RA-02 (FONTE: LCSC ) .....	41
TABELA 5. EFICIÊNCIA OBTIDA COM CONVERSOR BUCK MINI-360, COM 5 INPUTS DE TENSÃO DE FORMA A SIMULAR A DESCARGA DA BATERIA DE 12,6 V, COM UM OUTPUT FIXO DE 7 V E 500 MA.....	60
TABELA 6. EFICIÊNCIA OBTIDA COM CONVERSOR BUCK LM2596, COM 5 INPUTS DE TENSÃO DE FORMA A SIMULAR A DESCARGA DA BATERIA DE 12,6 V, COM UM OUTPUT FIXO DE 7 V E 500 MA.....	60
TABELA 7. EFICIÊNCIA OBTIDA COM CONVERSOR BUCK LM2576, COM 5 INPUTS DE TENSÃO DE FORMA A SIMULAR A DESCARGA DA BATERIA DE 12,6 V, COM UM OUTPUT FIXO DE 7 V E 500 MA.....	60
TABELA 8. EFICIÊNCIA OBTIDA COM CONVERSOR LINEAR LM317, COM 5 INPUTS DE TENSÃO DE FORMA A SIMULAR A DESCARGA DA BATERIA DE 12,6 V, COM UM OUTPUT FIXO DE 7 V E 500 MA .....	61
TABELA 9. EFICIÊNCIA OBTIDA COM CONVERSOR BUCK MINI-360, COM 5 INPUTS DE TENSÃO DE FORMA A SIMULAR A DESCARGA DA BATERIA DE 25,2 V, COM UM OUTPUT FIXO DE 5 V E 166 MA .....	62
TABELA 10. EFICIÊNCIA OBTIDA COM CONVERSOR BUCK LM2596, COM 5 INPUTS DE TENSÃO DE FORMA A SIMULAR A DESCARGA DA BATERIA DE 25,2 V, COM UM OUTPUT FIXO DE 5 V E 166 MA .....	62
TABELA 11. EFICIÊNCIA OBTIDA COM CONVERSOR BUCK LM2576, COM 5 INPUTS DE TENSÃO DE FORMA A SIMULAR A DESCARGA DA BATERIA DE 25,2 V, COM UM OUTPUT FIXO DE 5 V E 166 MA .....	62
TABELA 12. EFICIÊNCIA OBTIDA COM CONVERSOR BUCK LM317, COM 5 INPUTS DE TENSÃO DE FORMA A SIMULAR A DESCARGA DA BATERIA DE 25,2 V, COM UM OUTPUT FIXO DE 5 V E 166 MA.....	62
TABELA 13. DADOS RELATIVOS À COMUNICAÇÃO ENTRE O RECEPTOR E O EMISSOR .....	70
TABELA 14. DADOS RELATIVOS À COMUNICAÇÃO ENTRE O EMISSOR E O RECEPTOR .....	71
TABELA 15. DESIGNAÇÃO DE CADA PINO DO CONECTOR DB25 .....	116

## Lista de Abreviaturas, Acrónimos e Siglas

<b>AC</b>	<i>Alternating Current</i>
<b>AES</b>	<i>Advanced Encryption Standard</i>
<b>ANEPE</b>	Associação Nacional de Empresas de Produtos Explosivos
<b>APIPE</b>	Associação Portuguesa dos Industriais de Pirotecnia e Explosivos
<b>BMS</b>	<i>Battery Management System</i>
<b>BW</b>	<i>Bandwidth</i>
<b>CCA</b>	<i>Copper-clad aluminium wire</i>
<b>CCS</b>	<i>Chirp Spread Spectrum</i>
<b>CEO</b>	<i>Chief Executive Officer</i>
<b>CR</b>	<i>Coding Rate</i>
<b>DAE</b>	Departamento de Armas e Explosivos
<b>DC</b>	<i>Direct Current</i>
<b>DMUX</b>	<i>Demultiplexer</i>
<b>EEPROM</b>	<i>Electrically Erasable Programmable Read-Only Memory</i>
<b>EUA</b>	Estados Unidos da América
<b>FDM</b>	<i>Fused Deposition Modeling</i>
<b>GND</b>	<i>Ground</i>
<b>GPIO</b>	<i>General-Purpose Input/Output</i>
<b>HMI</b>	<i>Human-Machine Interface</i>
<b>I/O</b>	<i>Input/Output</i>
<b>ICSP</b>	<i>In-Circuit Serial Programming</i>
<b>ID</b>	<i>Identifier</i>
<b>IGBT</b>	<i>Insulated Gate Bipolar Transistor</i>
<b>IoT</b>	<i>Internet of Things</i>
<b>LED</b>	<i>Light Emitting Diode</i>
<b>Li-ion</b>	<i>Lithium-ion</i>
<b>LoRa</b>	<i>Long Range</i>
<b>MOSFET</b>	<i>Metal Oxide Semiconductor Field Effect Transistor</i>
<b>MUX</b>	<i>Multiplexer</i>
<b>NTC</b>	<i>Negative Temperature Coefficient</i>

<b>PC</b>	<i>Personal Computer</i>
<b>PSP</b>	Polícia de Segurança Pública
<b>PWM</b>	<i>Pulse-Width Modulation</i>
<b>RSSI</b>	<i>Received Signal Strength Indicator</i>
<b>RX</b>	<i>Receiver</i>
<b>SD</b>	<i>Secure Digital Card</i>
<b>SF</b>	<i>Spreading Factor</i>
<b>SNR</b>	<i>Signal-to-noise ratio</i>
<b>SPI</b>	<i>Serial Peripheral Interface</i>
<b>SRAM</b>	<i>Static Random-Access Memory</i>
<b>TTL</b>	<i>Transistor–Transistor Logic</i>
<b>TX</b>	<i>Transmitter</i>
<b>UART</b>	<i>Universal Asynchronous Receiver-Transmitter</i>

## 1. Introdução

Este capítulo serve como introdução ao tema: apresenta uma contextualização e os objetivos principais do projeto. É ainda feita uma apresentação da empresa, a qual ajudou e orientou em alguns aspetos técnicos durante a elaboração do projeto. Por fim, é apresentada a estrutura da dissertação.

### 1.1. A empresa Douro Pirotecnia

A Douro Pirotecnia é uma empresa de pirotecnia que já segue na terceira geração. A sua história começou em 1929 com o nome “Oficina de José da Silva Macedo” que era localizada em Figueiró no concelho de Amarante. Nos anos 80, já na segunda geração, a empresa alterou a designação para “Pirotecnia Macedo”. Em 2008, na terceira geração, devido às normas impostas, foram criadas novas instalações na freguesia de Mancelos no concelho de Amarante, com novas tecnologias e com uma enorme segurança, com o objetivo de fazer a empresa crescer no mercado e enfrentar novos desafios nacionais e internacionais. Em 2009 o atual CEO registou a marca “Douro Pirotecnia” (Figura 1) e foi concedido o alvará de exploração. A empresa conta com mais de 80 empregados indiretos (credenciados para o lançamento de fogo de artifício) e 10 empregados diretos. Produz vários tipos de espetáculos; contudo, os mais comuns são os tradicionais multiposto. Este tipo de espetáculo é composto pelos mais variados artigos pirotécnicos em vários pontos, com o intuito de preencher o céu, oferecendo um espetáculo envolvente. Outro tipo de espetáculo que a empresa realiza com alguma frequência é o piromusical, no qual é conjugado o fogo de artifício com uma ou várias músicas. Nos últimos dez anos o número médio de espetáculos anuais produzidos pela empresa é cerca de 350. Com o intuito de alargar os seus produtos e serviços para além de Portugal continental, a empresa fundou na Ilha Terceira, nos Açores, um ponto de armazenamento e venda (Douro Pirotecnia, s.d.).



Figura 1. Logotipo da empresa Douro Pirotecnia (fonte: Douro Pirotecnia <sup>1</sup>)

---

<sup>1</sup> [www.douropirotecnia.pt](http://www.douropirotecnia.pt)

## **1.2. Enquadramento e motivação**

Com o aparecimento nos anos 90 das primeiras empresas com sistemas de disparo através de uma máquina, o número de acidentes no ramo da pirotecnia reduziu. Com elas puderam-se desenvolver espetáculos pirotécnicos mais compostos e atrativos para o público. Poucas empresas inicialmente tinham acesso, devido aos escassos sistemas existentes, e como a maior parte eram desenvolvidos no estrangeiro, muitos só chegavam cá passado alguns anos. Com o avanço da tecnologia e a popularidade da internet, várias empresas começaram a tornar os sistemas mais eficientes, mais rápidos e começaram a surgir alguns sistemas controlados sem fios, o que permitiu aumentar a segurança e a flexibilidade da montagem pirotécnica. Atualmente existem várias empresas de renome com vários anos de experiência na área, que desenvolvem sistemas completos, sendo por vezes um pouco complexos. Com o conhecimento no terreno, e com ideias de como seria um sistema ideal para pequenos e médios espetáculos pirotécnicos, surgiu a ideia de criar um sistema que fosse mais simples de operar, que tivesse boa fiabilidade e que fosse sobretudo flexível.

De referir que atualmente a maior parte dos operários pirotécnicos não têm um grande nível de escolaridade, sendo por isso essencial um sistema com uma curva de aprendizagem pequena.

## **1.3. Objetivos**

Com este projeto pretende-se desenvolver e apresentar um sistema com uma solução mais simples, de forma a reduzir a curva de aprendizagem do utilizador final e tornar o sistema mais flexível, comparado com a maior parte dos existentes no mercado, e a um preço acessível. Os objetivos estabelecidos são os seguintes:

- Pesquisa e estudo de alguns sistemas existentes no mercado;
- Criar um sistema composto por emissor, recetores sem fios e réguas de terminais, fácil de operar e compreender;
- Interface simples, apenas com o essencial para um utilizador não muito experiente;
- Emissor com três modos de operação: manual, semiautomático e totalmente automático;



- Modo automático com a possibilidade de inserir os tempos para cada número em cada canal;
- Modo *scan* que permita que o utilizador insira todos os ID's (*Identifier*) dos recetores presentes no espetáculo, de forma a poder verificar, à distância, todos os dados essenciais, como: tensão da bateria, humidade, temperatura e o nível do sinal e ruído da comunicação;
- Criar a possibilidade de que no modo *scan* o utilizador possa saber quais os ignitores que estão ligados às régua de cada recetor;
- Criar recetores com possibilidade de ligar régua de terminais através de um cabo;
- Recetores com capacidade de tensão e corrente suficientes para uma boa fiabilidade de disparo de vários ignitores em cada terminal;

#### 1.4. Estrutura da dissertação

O presente projeto consiste no estudo e desenvolvimento de um sistema de disparo de fogo de artifício sem fios. Para além deste capítulo, esta dissertação possui cinco outros:

**Capítulo 2:** Enquadramento Teórico – Neste capítulo são abordados conceitos da indústria pirotécnica, a explicação do funcionamento dos artigos pirotécnicos, o funcionamento dos ignitores elétricos e uma breve explicação da comunicação sem fios. É apresentada ainda uma breve descrição de alguns dos sistemas de disparo existentes no mercado.

**Capítulo 3:** Componentes e Dispositivos – Neste capítulo são descritos de forma geral cada um dos principais componentes utilizados no projeto.

**Capítulo 4:** Desenvolvimento do Projeto – Este capítulo descreve e justifica os vários passos dados durante elaboração do projeto.

**Capítulo 5:** Descrição do Sistema de Ignição Pirotécnica – É descrito o funcionamento dos módulos elaborados, o funcionamento do emissor e recetor, configurações e instruções usadas no próprio *Human-Machine Interface* (HMI), bem como a estrutura dos pacotes de dados utilizados. Apresentação e descrição dos esquemas elétricos do módulo emissor, recetor e régua.

**Capítulo 6:** Conclusões e Trabalho Futuro. São apresentadas as principais conclusões do trabalho bem assim como algumas propostas de trabalho futuro.

## **2. Enquadramento Teórico**

Neste capítulo é apresentado um estudo sobre alguns componentes teóricos relevantes no projeto. São citados conceitos sobre a pirotecnia, alguns dos sistemas de disparo presentes no mercado e uma parte da componente da comunicação sem fios.

### **2.1. Indústria pirotécnica**

A pirotecnia propriamente dita consiste na arte do manuseamento de várias substâncias químicas, com a finalidade de obter produtos de entretenimento, com som e cor.

A indústria da pirotecnia tem uma grande importância na economia portuguesa devido ao enorme número de festas e romarias existentes durante todo o ano. Esta indústria evoluiu ao longo dos anos no que toca à segurança no fabrico e no próprio lançamento. Inicialmente a maior parte dos artigos pirotécnicos eram de cana, o que obrigava os operários a exporem-se a um elevado risco. Na última década os artigos pirotécnicos de cana tenderam a desaparecer devido à legislação imposta. As balonas de tubo vieram substituir praticamente todo o tipo de foguetes de cana, e têm a enorme vantagem de o operário poder lançar os artigos pirotécnicos com alguma distância, o que fez com que o risco para os operários baixasse bastante, diminuindo assim o número de acidentes.

Em Portugal existem cerca de 80 empresas e mais de 10.000 trabalhadores (Antena Minho, 2020). Estas empresas geram uma faturação anual a rondar os 40 milhões de euros (Luso, 2019).

As principais empresas em Portugal no ramo são: Douro Pirotecnia, Grupo Luso Pirotecnia, Pirotecnia Oleirense, GJR e Macedos. Cerca de 10% de todos os espetáculos são realizados no estrangeiro. Em Portugal existem duas associações de empresas de Pirotecnia: a Associação Nacional de Empresas de Produtos Explosivos (ANEPE) e a Associação Portuguesa dos Industriais de Pirotecnia e Explosivos (APIPE). Ambas têm os mesmos objetivos: defenderem os direitos das empresas, criar cooperação das empresas em espetáculos com maior dimensão e tentar resolver problemas que surjam com as empresas associadas. Seria extremamente complicado apenas uma empresa comunicar diretamente com órgãos superiores, como ministros e a própria polícia. Com as associações esta dificuldade torna-se menor e, através de um porta-voz, este terá a responsabilidade de comunicar e resolver problemas.

O Departamento de Armas e Explosivos (DAE) da Polícia de Segurança Pública (PSP) é responsável por garantir a segurança e o cumprimento das normas estabelecidas no fabrico, armazenamento e comercialização dos produtos pirotécnicos. É também a entidade responsável pela fiscalização dos eventos pirotécnicos, com o intuito de verificar se as normas impostas estão a ser cumpridas.

A indústria pirotécnica é ainda uma indústria conservadora, sendo que os conhecimentos técnicos foram passados de geração em geração e aprendidos e modificados com o experimento de novas técnicas, de modo a melhorar a técnica e ao mesmo tempo oferecer novos produtos para o mercado. De referir ainda, que a produção dos artigos pirotécnicos, atualmente, continua a ser bastante manual (Figura 2 e Figura 3).



Figura 2. Preparação de uma cor (fonte: Youtube <sup>2</sup>)



Figura 3. Preparação de tiros (fonte: Youtube <sup>2</sup>)

Desde março de 2020 que a pirotecnia tem sentido imenso o impacto causado pela pandemia, vendo reduzida a sua faturação em cerca de 95%, de acordo com o *Chief Executive Officer* (CEO) de uma empresa, estando este sector a passar por tempos difíceis.

---

<sup>2</sup> [https://www.youtube.com/watch?v=NKOLTqkqf3E&t=30s&ab\\_channel=DouroPirotecnia](https://www.youtube.com/watch?v=NKOLTqkqf3E&t=30s&ab_channel=DouroPirotecnia)

## 2.2. Funcionamento de um artigo pirotécnico

Existem diversos artigos pirotécnicos, sendo a maior parte artigos cuja impulsão é dada através de tubos, artigos estes denominados por balonas de tubo e candelas romanas (com vários disparos do mesmo tubo espaçados por um determinado intervalo de tempo, com apenas um acionamento). Existem ainda repuxos e várias peças de fogo preso. Estes fazem efeitos de cor, de barulho ou ambos. Em termos de efeitos existem milhares disponíveis no mercado. Em relação ao acionamento deles, são todos feitos através de uma chama num condutor pirotécnico. No caso das balonas (Figura 4), esta chama entra em contacto com o primeiro estágio, que se denomina condutor pirotécnico e este faz a condução do fogo até ao segundo estágio. No segundo estágio é a detonação da parte inferior do artigo, o que faz criar uma enorme pressão no fundo do tubo onde este está inserido, e faz com que ele suba devido à pressão sob ele. O terceiro estágio é ativado por meio de um rastilho, que é incendiado no momento da explosão do segundo estágio. Este estágio pode ser o último, no caso da balona de um só tiro (bomba). No caso de uma balona com mais estágios, temos normalmente uma abertura do artigo no ar e nos estágios seguintes procedem os vários efeitos de cor, com sons, tiros ou assobios, por exemplo.

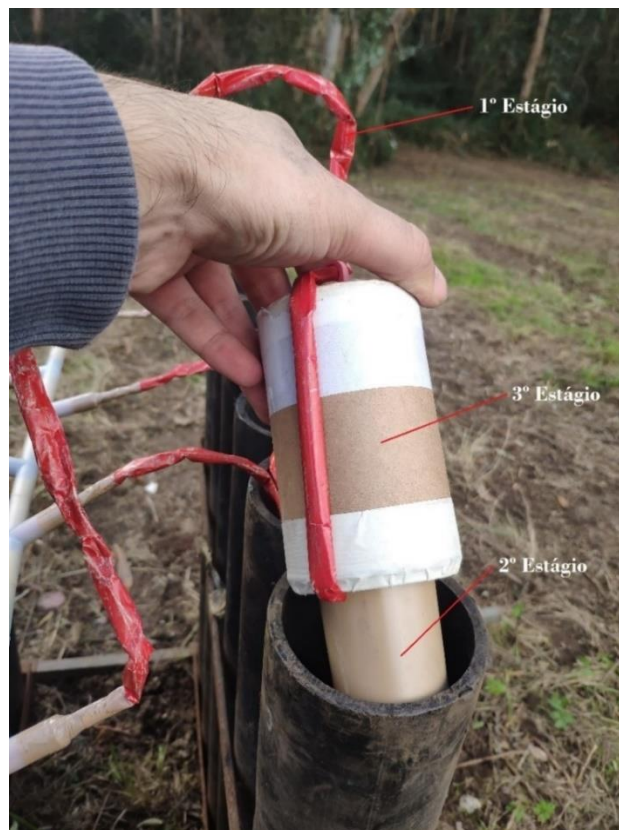


Figura 4. Estágios de um artigo pirotécnico (balona)

Antigamente em vez das balonas de tubo, as mais comuns atualmente, existiam os foguetes com canudo (foguetes de cana) (Figura 5). Devido à legislação atual, os foguetes com canudo estão proibidos em praticamente todo o território português. Para além de perigoso para o fogueteiro, uma vez que este tipo de artigo era acionado na própria mão, existia sempre um perigo associado à queda da cana do artigo pirotécnico nas redondezas ao lançamento.



*Figura 5. Lançamento de um foguete de cana (composto por um canudo que faz impulsão) (fonte: Florêncio Pereira, Helder Ferreira, Rui Camacho <sup>3</sup>)*

### **2.3. Funcionamento de um ignitor elétrico**

Os ignitores elétricos (Figura 6) são usados para colocar em combustão os condutores pirotécnicos compostos por uma ponte de arame revestida com uma composição pirotécnica. Os ignitores elétricos aumentam o nível de segurança dos operadores em espetáculos pirotécnicos, uma vez que estes já não necessitam de estar próximos do artigo pirotécnico para o “ativar” manualmente, através, por exemplo, de um isqueiro ou tocha (Pyrotechic Innovations, s.d.). Estes são compostos na tocha ou cabeça por um conjunto de misturas químicas que, quando expostas a uma corrente elétrica (a recomendada pelo fabricante), num certo período de tempo, entram em combustão (Figura 7) (S.C.

---

<sup>3</sup> <https://museus.madeira.gov.pt/Exposicao?exbID=18>

Mukhopadhyay, 2008). Existem vários tipos de ignitores, sendo que o que varia é o seu comprimento (fio até aos terminais da tocha), o seu formato, e a sua composição química, que fará com que tenha comportamentos diferentes com a passagem da corrente elétrica. Estes por sua vez vão incendiar os artigos pirotécnicos que nele estão ligados (Figura 6).



Figura 6. Exemplo de um ignitor elétrico (fonte: Meta Pyro <sup>4</sup>)

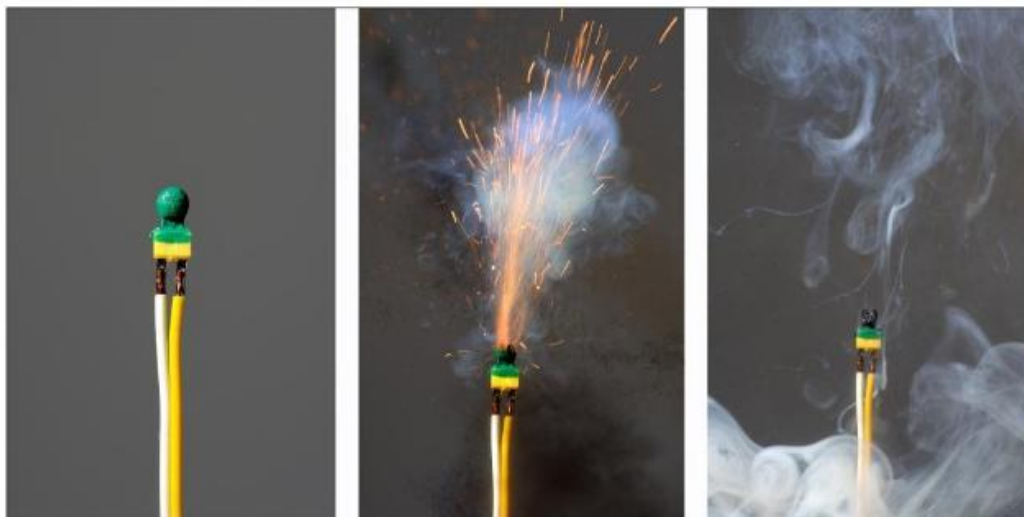


Figura 7. Ignição de um ignitor elétrico (fonte: Meta Pyro <sup>5</sup>)

<sup>4</sup> [http://www.meta-pyro.cz/file/3ff85d32f08fb23bb2a654dec22084c8/1012/20170810\\_111537.jpg](http://www.meta-pyro.cz/file/3ff85d32f08fb23bb2a654dec22084c8/1012/20170810_111537.jpg)

<sup>5</sup> <http://www.meta-pyro.cz/file/1a2f78289b8a1d86bfc0eb0bc1bdd60/992/jiskra%20%20EPX2.jpg>





*Figura 8. Inserção de um ignitor elétrico num artigo pirotécnico*

Num espetáculo pirotécnico, os ignitores podem ser ligados em série ou paralelo numa única saída (terminal). Para tal, será necessário a empresa pirotécnica pedir informações ao fabricante, para que possa, de acordo com a máquina de ignição, fazer o cálculo do número máximo de ignitores ligados a uma saída da máquina de ignição, com o objetivo de garantir a ignição dos mesmos. Para garantir a ignição, a corrente percorrida em cada ignitor tem de ser igual ou superior à corrente de ignição recomendada pelo fabricante. Abaixo estão representados dois exemplos com ligação de seis ignitores num terminal. A Figura 9 representa um exemplo típico de uma ligação em série de seis ignitores num único terminal, enquanto a Figura 10 representa uma ligação em paralelo com seis ignitores num único terminal.

A corrente percorrida por cada ignitor numa ligação em série num único terminal é igual à corrente total e é o quociente entre a tensão dos terminais e a resistência total do circuito. A resistência total é obtida através da soma de todas as resistências do circuito.

No caso da ligação em paralelo a corrente percorrida por cada ignitor é o quociente entre a queda de tensão nos ignitores e a resistência do ignitor. A queda de tensão nos ignitores é dada pelo produto entre a corrente total do circuito e a resistência equivalente dos ignitores em paralelo. A resistência equivalente dos ignitores em paralelo é o quociente entre um e o somatório dos quocientes entre um e a resistência de cada ignitor. A corrente total é o quociente entre a tensão nos terminais e a resistência total do circuito.

A resistência total do circuito é dada pela soma da resistência das linhas, da resistência interna da máquina e da resistência equivalente à associação dos ignitores em paralelo. De seguida são apresentadas algumas fórmulas que permitem determinar a corrente percorrida em cada ignitor ((1) e (7)).

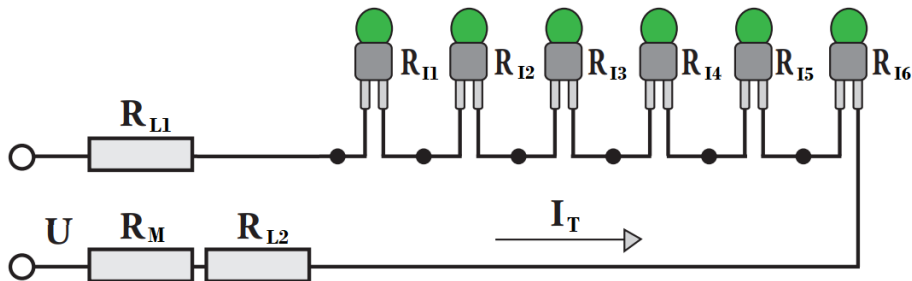


Figura 9. Exemplo de associação em série de ignitores elétricos no mesmo terminal (Adaptado da fonte: Meta Pyro <sup>6</sup>, p.5)

$$R_{\text{total}} = R_{L1} + R_{L2} + R_M + R_{I1} + R_{I2} + R_{I3} + R_{I4} + R_{I5} + R_{I6} \quad (1)$$

$$I_T = \frac{U}{R_{\text{total}}} \quad (2)$$

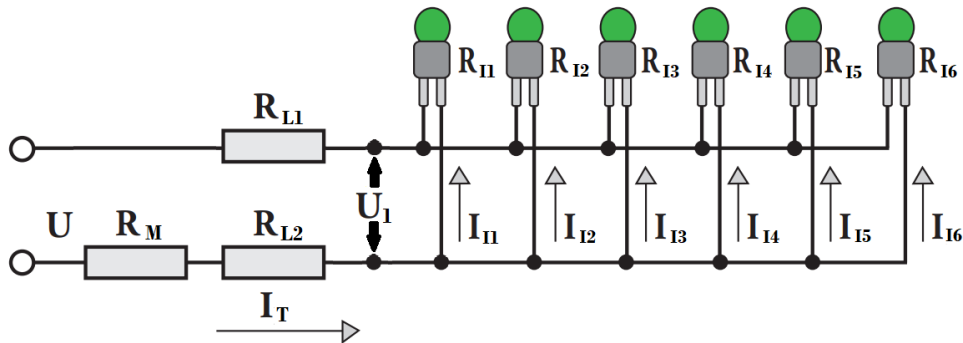


Figura 10. Exemplo de uma associação em paralelo de ignitores elétricos no mesmo terminal (Adaptado da fonte: MetaPyro <sup>6</sup>, p.5)

$$R_{\text{total}} = R_{L1} + R_{L2} + R_M + \frac{1}{\frac{1}{R_{I1}} + \frac{1}{R_{I1}} + \frac{1}{R_{I3}} + \frac{1}{R_{I4}} + \frac{1}{R_{I5}} + \frac{1}{R_{I6}}} \quad (3)$$

$$I_T = \frac{U}{R_{\text{total}}} \quad (4)$$

<sup>6</sup> [http://www.meta-pyro.cz/file/0af80cba71bdddac3cb841ee339b1edc/936/MetaPyro\\_EN\\_tisk.pdf](http://www.meta-pyro.cz/file/0af80cba71bdddac3cb841ee339b1edc/936/MetaPyro_EN_tisk.pdf)



$$R_{TI} = \frac{1}{\frac{1}{R_{I1}} + \frac{1}{R_{I1}} + \frac{1}{R_{I3}} + \frac{1}{R_{I4}} + \frac{1}{R_{I5}} + \frac{1}{R_{I6}}} \quad (5)$$

$$U_1 = R_{TI} \times I_T \quad (6)$$

$$I_{I1} = \frac{U_1}{R_{I1}}, I_{I2} = \frac{U_1}{R_{I2}} \dots I_{I6} = \frac{U_1}{R_{I6}} \quad (7)$$

Onde:

$R_{total}$  = Resistência total da associação

$R_{I1}$  até  $R_{I6}$  = Resistência de cada ignitor

$R_{L1}$  e  $R_{L2}$  = Resistência do condutor elétrico até aos terminais dos ignitores

$R_M$  = Resistência interna da máquina

$I_{I1}$ ...  $I_{I6}$  = Corrente que passa pelo ignitor

$I_T$  = Corrente total do circuito

$U_1$  = Tensão aos terminais da associação paralela

$R_{TI}$  = Resistência equivalente dos ignitores em paralelo

Para garantir a ignição em associações de ignitores em série, o valor de  $I_T$  tem que ser igual ou superior ao valor da corrente de ignição recomendada pelo fabricante, para poder existir a ignição de cada ignitor. No caso das associações em paralelo, o valor da corrente que atravessa cada ignitor tem que ser superior ao valor da corrente de ignição recomendada pelo fabricante para poder existir a ignição de cada ignitor. O valor da corrente de ignição recomendada é normalmente fornecido pelo fabricante ou pelas empresas que comercializam este tipo de artigos.

Na Tabela 1 são apresentadas as características técnicas de alguns ignitores presentes no mercado. A corrente máxima de teste é a corrente que cada fabricante recomenda para fazer um teste de continuidade a um ignitor. A corrente de não fogo é a corrente na qual foi testado o ignitor e não houve qualquer ignição num período de tempo (por vezes não é indicado pelo fabricante). A corrente de ignição é a corrente que o fabricante recomenda para que seja garantida a ignição do ignitor.

Para efeito de cálculo, a resistência do fio de cobre dos ignitores, com um núcleo de 0,5 mm de espessura num metro, tem aproximadamente 0,092  $\Omega$  (Meta Pyro, s.d.).

Tabela 1. Características técnicas de alguns dos ignitores elétricos presentes no mercado

Nome do Artigo	Resistência	Corrente máxima de teste	Corrente de não fogo	Corrente de ignição	Temperatura
EMP-U-EPX2-SO <sup>7</sup>	0,48 $\Omega$ - 0,6 $\Omega$	-	0,45 A (5 min)	2,12 A (4 ms)	-20 °C até +40 °C
EMP-A-EPX2-NO <sup>7</sup>	1,5 $\Omega$ - 1,9 $\Omega$	-	0,18 A (5 min)	1 A (4 ms)	-20 °C até +40 °C
J-TEK Igniter <sup>8</sup>	1 $\pm$ 0,2 $\Omega$	40 mA	0,3 A (30 s)	0,75 A (mínimo) 1 A (recomendado)	-
Wireless Fireworks E-Mach Igniters <sup>9</sup>	1,3 $\pm$ 0,15 $\Omega$ - 1,7 $\pm$ 0,15 $\Omega$	-	0,18 A	800 mA (5 ms)	-
MJG Firewire Initiator <sup>10</sup>	1 $\pm$ 0,2 $\Omega$	40 mA	0,3 A	0,6 A (mínimo) 0,75 A (recomendado)	-
N28B <sup>11</sup>	1,6 $\pm$ 0,3 $\Omega$	10 mA	0,2 A	0,37 A (mínimo) 1 A (2 ms)	-

## 2.4. Organização de um espetáculo pirotécnico

Um espetáculo pirotécnico normalmente carece antecipadamente de várias autorizações, como a autorização concedida pela autoridade policial da área, licença especial de ruído e uma declaração da corporação de bombeiros local. Para além disto é necessário existir um seguro de responsabilidade civil e um outro para os intervenientes na montagem do espetáculo. Nos casos de lançamento nas redondezas de um aeroporto ou aeródromo, é necessária uma informação positiva por parte do serviço de controlo de tráfego aéreo. Todos os operários pirotécnicos necessitam de ser credenciados pelo Departamento de Armas e Explosivos (DAE) da PSP para poderem operar com fogo de artifício<sup>12</sup>.

<sup>7</sup> <http://www.meta-pyro.cz/technical-specifications>

<sup>8</sup> <https://electricmatch.com/pyrotechnics/see/9/5/j-tek-igniter>

<sup>9</sup> Dan, Comunicação eletrónica pessoal, 24/10/2020

<sup>10</sup> [https://www.cobrafiringsystems.com/initiator\\_3ft](https://www.cobrafiringsystems.com/initiator_3ft)

<sup>11</sup>

[https://www.monetti.net/pdf/en/PRODUCTS\\_PYROTECHNIC\\_ACCESSORIES/2004%20A00%20N28%20EN.pdf](https://www.monetti.net/pdf/en/PRODUCTS_PYROTECHNIC_ACCESSORIES/2004%20A00%20N28%20EN.pdf)

<sup>12</sup> [http://www.aniet.pt/fotos/editor2/biblioteca/nt\\_n\\_1\\_2018.pdf](http://www.aniet.pt/fotos/editor2/biblioteca/nt_n_1_2018.pdf)

Antecipadamente, a empresa estuda e compõe espetáculo de acordo com o desejo do cliente e o orçamento. Depois são organizados conjuntos de vários artigos pirotécnicos, onde é composta uma escala, com números por ordem crescente. Em cada número é demonstrado o(s) artigo(s) pirotécnico(s) por cada posto de lançamento (local de colocação do artigo no terreno). Depois disto a produção rastilha os artigos (com esperas entre eles) e organiza-os por cada número e posto. A seguir, são etiquetados com o respetivo número e posto, de forma a que no local da montagem do espetáculo seja mais fácil a sua identificação e organização.

O local onde o espetáculo irá ocorrer, terá de ser escolhido de acordo com o tipo de espetáculo e com o calibre dos artigos. Por exemplo, um espetáculo com artigos com efeitos mais baixos, necessitam de um local com boa visibilidade para que o público possa usufruir do mesmo. Em relação aos calibres dos artigos, o departamento de armas e explosivos da PSP fornece uma tabela com as áreas de segurança ao público, por cada calibre do artigo pirotécnico. Quanto maior for o calibre maior terá de ser a distância ao público, de forma a criar e garantir a segurança dos espectadores.

Reunidas todas as condições, na hora da montagem, o local é vedado com fita de segurança, de forma a que dentro da área só permaneçam os operários. Depois disto um responsável começa a fazer a distribuição de todo o tipo de grades (compostas por vários tubos de lançamento com um determinado calibre), onde são inseridos os artigos pirotécnicos pelo terreno. Esta distribuição é seguida através da escala do espetáculo. Cada número poderá conter um ou mais artigos, iguais ou diferentes, saindo normalmente em sequência através de rastilhos, o que faz com que exista uma espera de tempo entre artigos do mesmo número. Para além disto, o espetáculo é dividido por postos, que normalmente são três, de forma a espalhar e obter uma frente de fogo de artifício maior, podendo assim fazer cruzamentos de fogo de artifício. Depois de todos os tubos espalhados pelo local de lançamento, procede-se à colocação dos artigos pirotécnicos pelos tubos (estes normalmente identificados com uma etiqueta). Após a colocação dos artigos (Figura 11), procede-se à colocação do ignitor elétrico na ponta do condutor pirotécnico de cada artigo ou conjunto de artigos. Por último serão feitas as ligações dos ignitores às respetivas máquinas de ignição recetoras (normalmente um recetor por posto) (Figura 12).



Figura 11. Montagem de um espetáculo Pirotécnico



Figura 12. Recetor DB24r-X com ignitores ligados em alguns números

Para o disparo dos artigos, dependendo das máquinas de disparo, podem existir três possibilidades de controlo, que são:

- **Manual**- Neste modo o operador tem de configurar o canal da máquina e qual será o número que quer disparar. O primeiro número é acionado no início e os restantes são acionados quando os artigos do último número terminarem o seu efeito.
- **Semiautomático**- Este modo é semelhante ao anterior, contudo o incremento do número a ser disparado é feito pela máquina. Assim o operador só necessita de pressionar apenas um botão para efetuar o próximo número a disparar.
- **Automático**- Este modo já requer uma logística superior aos outros, pois com antecedência tem de ser analisado o tempo de cada número para elaborar um script de tempos. Este script contém o tempo de espera entre

números de disparo. No momento da queima o operador só tem de pressionar o botão para começar o espetáculo, onde é disparado logo o primeiro número. A partir daí a máquina (emissor) vai decrementando os tempos definidos pelo operário no script, e estes quando acabam é disparado o próximo número.

A variável “canal” é preciso ter em conta para entender a maneira de gerar mais números independentes quando as máquinas recetoras só possuem um número limitado de saídas. Por exemplo, se o operador necessitar de disparar 30 números num espetáculo, e a máquina recetora só possuir 24 terminais, o operador terá que utilizar duas máquinas recetoras, e definir uma num canal e a outra noutra canal. Basicamente durante o espetáculo o operador dispara os 24 números de ignição de uma máquina e após isso muda o canal para o da outra máquina e controla os seis números restantes.

## **2.5. Sistemas existentes no mercado**

Antigamente todo o tipo de acionamento de artigos pirotécnicos em espetáculos era feito de forma manual, através de isqueiro, cigarro ou qualquer fonte de fogo. Era, portanto, um processo bastante perigoso, pois a ignição do artigo pirotécnico era feita próxima do explosivo (P. Vrignat, 2015). Como consequência, muitos operários sofreram perdas de membros e queimaduras devido a possíveis erros do operador ou até mesmo defeito do produto. Também não era possível fazer uma cronografia perfeita entre os artigos pirotécnicos e na sua “queima” eram necessários vários fogueteiros.

Com isto, nos anos 90 começaram a surgir algumas empresas de sistemas de disparo que desenvolveram os primeiros sistemas de ignição eletrónicos, sendo que estes foram melhorados ao longo do tempo. Inicialmente eram sistemas muito arcaicos, feitos através de baterias e botões que faziam o “curto-circuito” para cada ignitor. Os sistemas mais antigos também não tinham a possibilidade de verificar se existia algum problema na linha do ignitor (verificar continuidade no terminal) ou até mesmo defeito do próprio.

Atualmente já existem máquinas com disparo sem fios, com testes de continuidade em cada terminal, resoluções de disparo na ordem de 0.01 s bem como programas informáticos que ajudam na gestão e organização dos tempos de disparo de um espetáculo pirotécnico. Sem a alta precisão e capacidade de disparo destas máquinas de última

geração, não seria possível elaborar espetáculos piromusicais (fogo a acompanhar a música).

Nas secções seguintes são apresentados alguns sistemas de ignição pirotécnica existentes no mercado, dos quais alguns são utilizados na empresa Douro Pirotecnia.

### **2.5.1. Sistema Cobra Firing Systems LLC.**

O 18R2 (Figura 13) é um controlador (módulo emissor) de disparos. Apresenta o modo manual e totalmente automático, este último com a possibilidade de inserir os tempos para espetáculos completamente automatizados. Possui também memória de disparo (que informa o utilizador quais os números que foram disparados durante o espetáculo). Com este sistema é possível reproduzir áudio em perfeita sincronia com outro módulo, o Áudio Box, módulo este independente, que se pode ligar a um sistema de som, de modo a criar um espetáculo piromusical bem organizado. Para além disto, o módulo foi projetado de forma a ser o mais simples possível para o utilizador. Possui 100 canais distintos, uma precisão de disparo de 1/100 s e dispõe da capacidade de controlar até 1800 terminais independentes. Este sistema funciona na frequência de 2,4 GHz, e tem um alcance de mais de 450 m em campo aberto, com a deteção da qualidade do sinal. Os dados são criptografados na comunicação.

Tem a possibilidade de no decorrer do *script* (tempos entre disparos) no modo automático, acionar um número manualmente, sem interromper o espetáculo. Este sistema utiliza transceptores bidirecionais. Isto significa que através do módulo controlador (emissor) é possível saber informações dos recetores, como por exemplo: quais são os terminais que possuem ignitores conectados, qual é o estado do sinal, o estado da bateria e se o módulo recetor está armado (pronto para começar o espetáculo). Com a versão mais recente de *firmware*, foi adicionada a tecnologia MESH, que basicamente permite que todos os dispositivos no campo participem na comunicação de sinal, funcionando assim como repetidores. A autonomia é aproximadamente 7 h (COBRA Firing Systems, s.d. a).

A empresa possui vários módulos recetores disponíveis com 18, 36 e 72 terminais. A alimentação dos recetores pode ser feita de três maneiras. Pelas três baterias de 9 V, oferecendo uma corrente de pico de 3 A, por uma bateria LiPo recarregável de 14,8 V com uma corrente de pico de 6 A ou por uma alimentação externa de 12-24 V com 9 A. Tem também a possibilidade de mudar de canal (COBRA Firing Systems, s.d. b).



Figura 13. Módulo emissor 18R2 da empresa Cobra Firing Systems LLC. (Fonte: Cobra Firing Systems LLC<sup>13</sup>)

### 2.5.2. Sistema Liuyang Happiness Firing Systems CO., LTD

O emissor (Figura 14) possui dois modos de disparo, o manual (composto por 24 números de disparo) e o sequencial (que é dividido em partes). Neste último modo, as sequências dos disparos são feitas através de um tempo fixo de 0,1 s até 9,9 s entre os números de 1 a 12 e de 13 a 24.

No modo manual, ao pressionar o botão do número, acenderá um LED (*Light Emitting Diode*) a indicar que esse botão já foi pressionado. Pode operar a uma distância de até 500 m.

É alimentado por uma bateria de 12,6 V. Possui 5 h de autonomia em trabalho e 12 h em *standby*. Para além dos 24 botões de números de disparo, possui também um seletor de canais (0-99). Os canais servem basicamente para aumentar números independentes do espetáculo. Por exemplo, quando o botão “channel” é ajustado para “0” o emissor controla todos os 24 números dos recetores que estejam com o canal “0” definido, e ao ajustar o emissor para o canal “1” controla os 24 números dos recetores que estejam definidos com o canal “1”. Em relação à autonomia aguenta até 12 h em *standby* e em operação cerca de 5 h.

<sup>13</sup> <https://www.cobrafiringsystems.com/common/images/products/main/18R2.png>



Figura 14. Emissor DBR02-X da empresa Liuyang Happiness Firing Systems CO.,LTD (fonte: Amazon <sup>14</sup>)

Os módulos recetores (Figura 15) possuem 24 terminais, que operam no máximo com 12,6 V e 5 A. O recetor tem um ajuste de canal e uma chave de proteção com duas posições. A posição “test”, na qual o módulo mesmo que receba ordem de disparo de um emissor não executa o disparo, serve apenas para verificar quais ignitores estão ligados, pois o recetor possui um *Light Emitting Diode* (LED) em cada terminal a indicar se o ignitor está ligado. Quando a chave é rodada para a posição “fire”, o recetor fica armado e pronto para disparo. Possui ainda um botão para seleccionar o canal em que vai trabalhar, e um botão “study” para emparelhar o recetor com um emissor, de forma a que ele só execute os dados recebidos pelo emissor emparelhado. Em relação à autonomia, aguenta até 36 h em *standby* e em operação cerca de 5 h (Liuyang happiness firing system factory, s.d.) (Pyro Firing System, s.d.).

<sup>14</sup> [https://images-na.ssl-images-amazon.com/images/I/71g%2B75zZv%2BL.\\_AC\\_SL1500\\_.jpg](https://images-na.ssl-images-amazon.com/images/I/71g%2B75zZv%2BL._AC_SL1500_.jpg)





Figura 15. Módulo recetor DB24r-X da empresa Liuyang Happiness Firing Systems CO.,LTD (fonte: Amazon <sup>15</sup>)

### 2.5.3. Sistema Galaxis Showtechnik GmbH

O emissor PFC Advanced Black Edition (Figura 16) surgiu em 2015. É à prova de água, possui um ecrã tátil, legível em todas as condições de luz ambiente, apresenta uma interface multilinguagem, tem a possibilidade de analisar todos os dados dos recetores, tal como alcance de rádio e nível da bateria, e ainda quais os terminais que estão preenchidos com ignitores em cada recetor. Dispõe ainda da possibilidade de desligar os recetores remotamente, ativar o modo de suspensão e reativar os recetores.

Conta com uma capacidade máxima de 999 terminais de disparo independentes, com a capacidade de operar durante 6 h. Apresenta modo de disparo manual, semiautomático e automático. No modo automático de disparo existem as funções: parar, pausar e avançar. Tem um alcance de rádio até 800 m e até 2000 m com cabo de extensão de antena, colocando-a assim mais alta. Pesa cerca de 3,7 Kg e a bateria é de 12 V com 2 Ah. É possível adquirir um amplificador opcional de sinal de rádio, caso o cliente deseje, contudo só é vendido para clientes dos Estados Unidos da América (EUA) (proibida em vários países). Suporta até 999 recetores ligados, sendo que a comunicação é bidirecional.

---

<sup>15</sup> [https://images-na.ssl-images-amazon.com/images/I/7168ymVv4dL.\\_AC\\_SL1500\\_.jpg](https://images-na.ssl-images-amazon.com/images/I/7168ymVv4dL._AC_SL1500_.jpg)



o número três físico, por o número 80 virtual. A tensão de disparo é de 24 V e a corrente de queima de 50 A de pico de 17 A contínuos. A função de teste também está integrada, mostrando no módulo emissor a resistência de cada terminal do recetor (0 a 99  $\Omega$ ). A bateria presente é de chumbo-ácido 24 V, 2 Ah. Para mostrar algumas informações ele contém um pequeno ecrã 2x16 (Galaxis Showtechnik, s.d. b).

Para expansão dos dez terminais dos recetores, a empresa apresenta dois módulos de expansão, um com 50 terminais e outro com 100 terminais (Figura 19), sendo ambos de encaixe direto no recetor (Galaxis Showtechnik, s.d. c).



Figura 18. Módulo recetor PFE Advanced 10 Outputs da empresa Galaxis Showtechnik GmbH (fonte: Galaxis Showtechnik GmbH <sup>18</sup>)



Figura 19. Régua de terminais Matrix Module 100 + módulo recetor PFE Advanced 10 Outputs da empresa Galaxis Showtechnik GmbH (fonte: Galaxis Showtechnik GmbH <sup>19</sup>)

---

<sup>18</sup> <https://www.firing-system.com/project/pfe-advanced-10-outputs/>

<sup>19</sup> <https://www.firing-system.com/project/matrix-module/>

#### 2.5.4. Sistema StarFire

O módulo de controlo StarFire Digital Firing System Controller (Figura 20) possui três modos de disparo: manual, automático e semiautomático. Consegue controlar até 254 módulos e possui um ecrã de 10,4". Não carece do uso de um computador para o poder operar. Quanto à sua precisão, consegue realizar 100 disparos num segundo, possibilitando o disparo simultâneo de vários números. O fabricante apresenta ainda um *software* para *Personal Computer* (PC) incluído, com possibilidade de inserir e editar tempos de disparo automático para o emissor.

Dispõe de teste de continuidade em cada terminal dos recetores, onde indica para o utilizador quatro possibilidades: "Bom", "Marginal", "Mau" e "Sem conexão". Serve para o operador ter uma noção se o terminal está subcarregado, podendo o recetor não conseguir disparar. Possui ainda um interruptor de emergência o qual, quando pressionado, interrompe o disparo, desarma todos os módulos e bloqueia quase todas as funcionalidades. O sistema da StarFire utiliza comunicação por fio e tem a possibilidade de utilizar até oito canais diferentes através de um fio *duplex*. Cada canal pode controlar cerca de 30 a 40 módulos recetores, dependendo do comprimento do fio e da própria secção.



Figura 20. Módulo de controlo StarFire Digital Firing System (fonte: ACE Pyro<sup>20</sup>)

Os módulos recetores StarFire Modules (Figura 21) têm 32 terminais. Os disparos são feitos por descarga capacitiva, com uma saída de 24 V e 6 A. Os módulos possuem uma conexão às régua de terminais (Figura 22) que o fabricante disponibiliza através de um cabo de

---

<sup>20</sup> <http://www.acepyro.com/images/StarFire-controller.jpg>

diferentes tamanhos. Pode ser adquirido um módulo separadamente para realizar a comunicação entre o emissor e os vários recetores sem fios. Esse módulo serve como uma ponte de comunicação sem fios entre o emissor e os vários recetores (StarFire, s.d.).

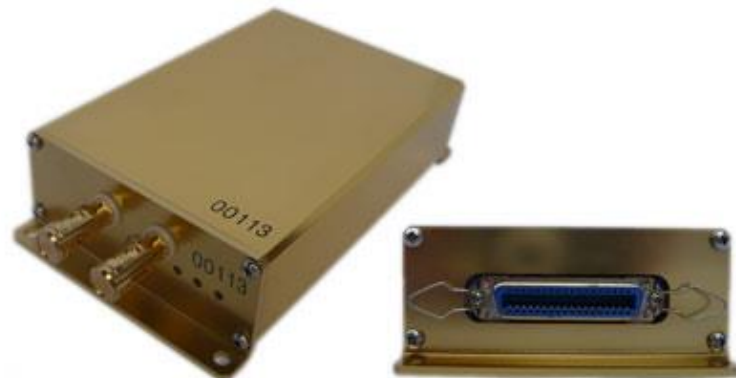


Figura 21. Módulo recetor StarFire Field Module (fonte: ACE Pyro <sup>21</sup>)

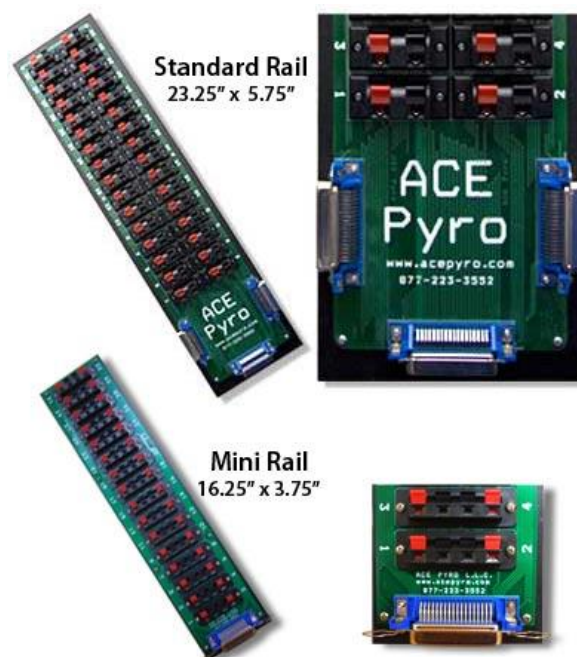


Figura 22. Réguas de terminais Firing System Rails (fonte: StarfireSales <sup>22</sup>)

## 2.6. Comunicação sem fios

Um dos objetivos principais deste projeto foi fazer com que a comunicação do sistema de disparo fosse realizada sem fios. Isto traz ao projeto uma enorme vantagem, uma vez que

<sup>21</sup> <http://www.acepyro.com/images/module.jpg>

<sup>22</sup> <http://starfiresales.com/images/rails-wh.jpg>

não se necessita de cabos para a transmissão de dados, sendo esta realizada através de ondas eletromagnéticas.

LoRa (*Long Range*) é uma tecnologia de modulação de espectro de propagação, desenvolvida pela empresa Semtech. A tecnologia especifica a camada física baseada em *chirp spread spectrum* (CSS) com correção integrada de erros. LoRa tem como principais propriedades a alta robustez, longo alcance, imunidade a ruídos e baixo consumo. Opera nas bandas ISM, 433 MHz e 868 MHz na união europeia e 433 MHz e 915 MHz nos EUA.

O rádio LoRa possui quatro parâmetros de configuração: frequência, *spreading factor* (factor de propagação), *bandwidth* (largura de banda) e *coding rate* (taxa de codificação). A frequência usada para a transmissão é configurável e, dependendo do módulo de rádio o ajuste poderá ser maior ou menor.

O *spreading factor* (SF) é a razão entre a taxa de símbolo e a taxa de *chip*. É configurável de sete a doze. Um SF mais alto aumenta a relação sinal ruído (SNR), a sensibilidade, o alcance e o tempo no ar durante a transmissão do pacote. O número de chips por símbolo é calculado através de  $2^{SF}$ . Por exemplo, para um SF de sete cada símbolo terá 128 *chips*. Cada aumento do SF reduz para metade a taxa de transmissão, dobra o tempo da transmissão no ar e aumenta o consumo energético.

*Bandwidth* (BW) é a faixa de frequências na banda de transmissão. Um BW mais alto fornece uma taxa de dados maior, ou seja, diminui o tempo da transmissão no ar, contudo, com o aumento do BW a sensibilidade baixa, devido a ruído adicional. Pode ser configurado em três escalas, 125 kHz, 250 kHz e 500 kHz. A taxa de *chips* é igual à largura de banda, ou seja, se BW foi 125 kHz, teremos uma taxa de 125 *kchips*.

*Coding rate* (CR) é a taxa usada pelo módulo LoRa, que oferece proteção contra surtos de interferência. Em suma, quanto mais alto o CR, maior é a proteção, diminuindo a taxa de erro. Contudo aumenta o tempo da transmissão no ar. Os rádios com diferentes valores de *coding rate* podem-se comunicar, porque o CR é armazenado no cabeçalho do pacote enviado (Bor, Vidler, & Utz Roeding, 2016).

Nas tecnologias de transmissão sem fios um dos aspetos a realçar é o *Link Budget*. O *link Budget* é uma forma de calcular a potência do sinal recebido que, para o cálculo, leva em consideração todos os ganhos e perdas do emissor até ao recetor. A equação (8) demonstra o cálculo geral para determinar a potência recebida em dBm.

$$\text{Potência recebida (dBm)} = \text{Potência Transmitida (dBm)} + \text{Ganhos(dB)} - \text{Perdas (dB)} \quad (8)$$

Quanto maior for o *link budget* maior será o alcance dos pacotes transmitidos. Como a tecnologia LoRa possui uma sensibilidade de recepção de sinais bastante alta, o *link budget* também será maior (Poole, s.d.).

Existem dois tipos de formatos de pacotes possíveis na transmissão: o explícito e o implícito. De uma forma geral, o que os diferencia é o cabeçalho. O formato explícito inclui, apesar de curto, algumas informações, como o número de bytes do *payload*, taxa de codificação (CR) e o CRC utilizado. O modo implícito é normalmente usado apenas em cenários em que o *payload*, a taxa de codificação e a presença de CRC são fixas, cenário este que pode ser vantajoso, pois reduz significativamente os tempos de transmissão. Neste modo o cabeçalho é removido do pacote e tanto o emissor como o receptor têm de possuir os mesmos valores de configuração.

A estrutura do pacote (Figura 23) é constituída por:

**Preâmbulo:** é usado para sincronizar o receptor com o fluxo de dados de entrada. Por padrão o pacote é configurado com uma sequência de doze símbolos. O comprimento do preâmbulo pode ser alterado para uma determinada aplicação mais específica. Para alterar este valor, basta alterar o registo *PreambleLength* de seis a 65535, resultando em comprimentos de 6 + 4 a 65535 + 4 símbolos. Os quatro símbolos dizem respeito ao *overhead* fixo do preâmbulo.

**Payload:** Contém a mensagem a transmitir.

**CRC:** Usado para verificar erros (desativada por defeito) (SEMTECH).

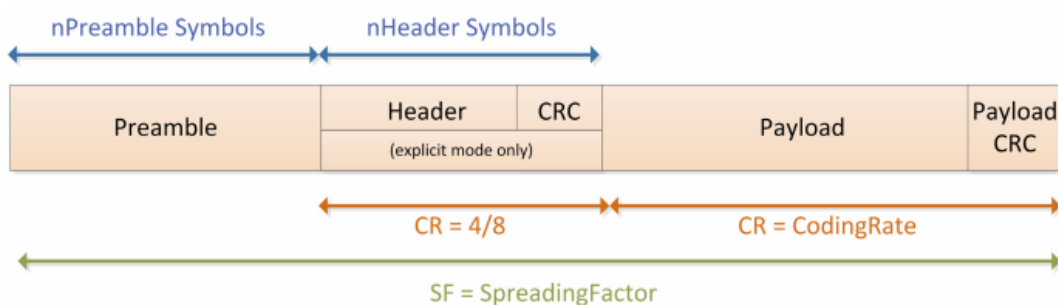


Figura 23. Estrutura do pacote LoRa (fonte: SEMTECH<sup>23</sup>, p.29)



### 3. Componentes e Dispositivos

Neste capítulo são abordados os principais componentes utilizados no projeto e é apresentado um resumo geral de cada um.

Inicialmente foi realizada uma lista de material necessária para o projeto de forma a prosseguir com o desenvolvimento do mesmo. Cada componente foi escolhido de acordo com os requisitos da aplicação final, bem como o preço e a disponibilidade do componente ou dispositivo. O feedback da comunidade de vários fóruns ou até mesmo de lojas, foi um dos fatores importantes na escolha, uma vez que dava maior confiança e grau de certeza na escolha de um determinado componente ou dispositivo. Alguns dos componentes testados não foram descritos, com o intuito de focar essencialmente os que foram utilizados no projeto final desenvolvido.

#### 3.1. MOSFET IRF9530

Um *Metal Oxide Semiconductor Field Effect Transistor* (MOSFET) é um dispositivo controlado por tensão, usado principalmente para comutações eletrônicas ou ampliações. Possui normalmente três terminais (Figura 24): a porta, fonte e dreno (ou *gate*, *source* e *drain*). Como é um dispositivo unipolar, a condução da corrente dá-se num sentido apenas. A tensão na gate controla a condutividade entre a fonte e o dreno. A condutividade elétrica varia de acordo com a tensão aplicada na gate. Comparando com o transistor bipolar (BJT) normal, tem uma grande vantagem porque não requer corrente de entrada para controlar uma grande quantidade de corrente. Comparando com o *Insulated Gate Bipolar Transistor* (IGBT), o MOSFET possui uma velocidade de comutação menor, o que faz ser mais rápido a comutar. Existem dois tipos de MOSFETs, os de tipo P e os de tipo N, que se distinguem pela polarização. No caso do tipo P, a *source* é conectada ao positivo e quanto mais negativo o  $V_{GS}$  (tensão entre a *gate* e a *source*), maior a corrente que flui entre a *source* e o *dreno*. No caso do tipo N, a *source* é conectada ao *ground* (GND) e quanto maior for a tensão  $V_{GS}$  maior a corrente que flui entre a *source* e o dreno. A tensão entre a *source* e a gate é a tensão da *source* a subtrair pela tensão da *gate*. Para o canal se formar é necessário um valor mínimo de  $V_{GS}$ . Esse valor é denominado por tensão de limiar (*threshold*) ou  $V_{GS(th)}$  e é fornecido no datasheet do fabricante.



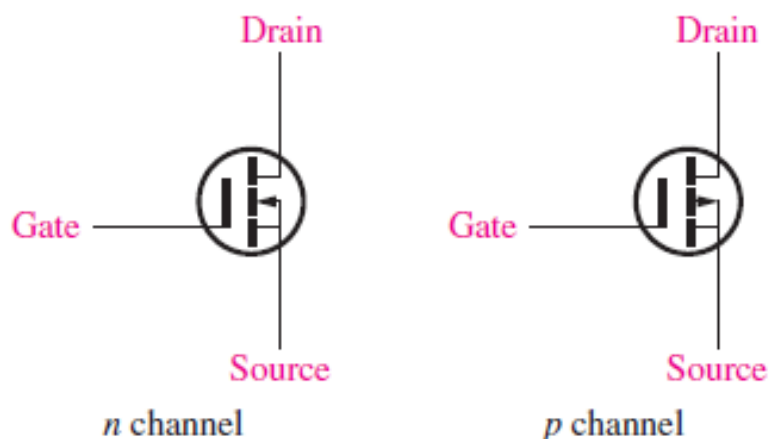


Figura 24. Simbologia MOSFET tipo N e tipo P ou canal N e canal P (fonte: Physics<sup>24</sup>)

Existem 3 modos de operação dos MOSFETs:

- Estado de corte: Acontece quando a tensão entre a *source* e a *gate* é inferior à tensão  $V_{GS(th)}$  no caso do tipo N ( $V_{GS} < V_{GS(th)}$ ). No caso do tipo P é quando a tensão  $V_{GS}$  é superior à tensão  $V_{GS(th)}$  ( $V_{GS} > V_{GS(th)}$ ). Este estado faz com que o MOSFET se comporte como isolante, não existindo corrente a fluir entre a *source* e o dreno, mesmo que exista uma diferença de potencial entre eles. Comporta-se como um interruptor aberto.
- Estado de condução linear: Nos MOSFETs tipo N, acontece quando  $V_{GS} > V_{GS(th)}$  e  $V_{DS} < V_{GS} - V_{GS(th)}$  ( $V_{DS}$  = Tensão entre o dreno e a *source*). Neste caso o MOSFET está em condução, funcionando como uma chave fechada. A corrente do dreno para a fonte é controlada através da tensão na *gate*. No caso dos MOSFETs tipo P,  $V_{GS} < V_{GS(th)}$  e  $V_{DS} \geq V_{GS} - V_{GS(th)}$ , ou seja, entra em funcionamento, contudo a corrente flui entre a *source* e o dreno. No caso dos MOSFETs tipo P, quanto menor a tensão  $V_{GS}$ , maior a corrente entre a *source* e o dreno, ou seja, o MOSFET comporta-se como uma resistência controlada pela tensão da *gate*.
- Saturação: Neste caso, o MOSFET permite o fluxo de corrente máximo entre o dreno e a *source* no caso do tipo N, e vice-versa, no caso do tipo P.

<sup>24</sup> <https://physicsabout.com/wp-content/uploads/2016/06/e-mosfet-schematic-symbol.png>

Comporta-se como um interruptor fechado. Para operar neste modo, no caso dos MOSFETs tipo N, o  $V_{GS} > V_{GS(th)}$  e  $V_{DS} > V_{GS} - V_{GS(th)}$ . No caso do MOSFET tipo P, o  $V_{GS} < V_{GS(th)}$  e  $V_{DS} \leq V_{GS} - V_{GS(th)}$  (Aqeel, 2018).

O MOSFET utilizado no projeto é o IRF9530 da International Rectifier (Figura 25), é do tipo P, e opera com correntes até 12 A. A tensão máxima entre a *source* e o dreno é de 100 V. Já a tensão máxima entre *gate* e *source* é de 20 V. A sua resistência quando trabalha como interruptor fechado é aproximadamente 0,3  $\Omega$  (Internation Rectifier).



Figura 25. MOSFET IRF9530 (fonte: Aquário <sup>25</sup>)

### 3.2. Multiplexer/Demultiplexer

O Multiplexer/Demultiplexer CD74HC4067 (Figura 26) é um módulo bastante prático e com uma boa eficiência. É bidirecional e normalmente é usado em projetos que exijam maior quantidade de pinos do microcontrolador. Com apenas seis pinos é possível fazer uma extensão para dezasseis entradas/saídas. Para controlo das mesmas, possui quatro pinos que controlam o fluxo de um pino de sinal que serve de entrada ou saída para os outros dezasseis, permitindo ser usado em qualquer direção, como interfaces digitais e comunicação série. Para além destes dois pinos, ele tem um de entrada/saída e outro para o colocar em funcionamento. As especificações adicionais CD74HC4067 incluem tensão de operação de 2 a 6 V, resistência interna de 70  $\Omega$  quando alimentado a 4,5 V e temperatura de funcionamento de -55 °C a 125 °C (SparkFun, s.d.).

Na Tabela 2 é apresentada a tabela verdade, mostrando as combinações para a seleção de cada porta entrada/saída.

---

<sup>25</sup> [https://www.aquario.pt/lmgs/produtos/product\\_21095/IRF9530.jpg](https://www.aquario.pt/lmgs/produtos/product_21095/IRF9530.jpg)

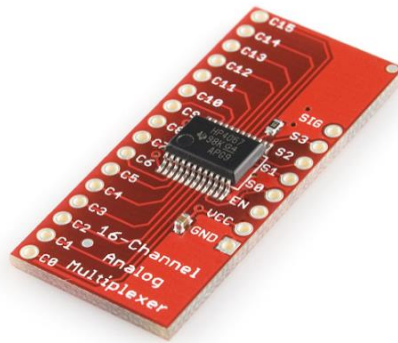


Figura 26. Multiplexer 16 canais CD74HC4067 (fonte: SparkFun <sup>26</sup>)

Tabela 2. Tabela verdade para selecionar cada uma das 16 saídas/entradas relativa ao Multiplexer CD74HC4067 (fonte: Texas Instruments <sup>27</sup>, p.2)

S0	S1	S2	S3	$\bar{E}$	SELECTED CHANNEL
X	X	X	X	1	None
0	0	0	0	0	0
1	0	0	0	0	1
0	1	0	0	0	2
1	1	0	0	0	3
0	0	1	0	0	4
1	0	1	0	0	5
0	1	1	0	0	6
1	1	1	0	0	7
0	0	0	1	0	8
1	0	0	1	0	9
0	1	0	1	0	10
1	1	0	1	0	11
0	0	1	1	0	12
1	0	1	1	0	13
0	1	1	1	0	14
1	1	1	1	0	15

<sup>26</sup> <https://cdn.sparkfun.com//assets/parts/2/4/2/0/09056-01.jpg>

<sup>27</sup>

[https://www.ti.com/lit/ds/symlink/cd74hc4067.pdf?ts=1604792245278&ref\\_url=https%253A%252F%252Fwww.google.com%252F](https://www.ti.com/lit/ds/symlink/cd74hc4067.pdf?ts=1604792245278&ref_url=https%253A%252F%252Fwww.google.com%252F)

### 3.3. Sensor DHT11

O DHT11 (Figura 27) é um sensor bastante utilizado na comunidade Arduino devido ao seu baixo custo e facilidade de implementação. Internamente possui dois transdutores resistivos, sendo que um mede a humidade e o outro a temperatura. Possui a capacidade de medir a humidade entre 20% até 80% com uma precisão de  $\pm 5\%$ . O sensor possui uma resolução de 16 bits. No que toca à temperatura, consegue medir entre 0 °C até 50 °C, com um erro de  $\pm 2$  °C a 25 °C. Possui quatro pinos: o primeiro é de VCC que suporta tensões entre 3,3 a 5,5 V DC (*Direct Current*); o segundo pino é usado para dados; um terceiro que não é usado e o outro e o quarto é o GND. Em relação ao consumo, em *standby* consome 60  $\mu$ A e quando está em funcionamento consome 0,3 mA. Para deteção das variáveis humidade e temperatura, o sensor possui essencialmente três componentes: o primeiro é um circuito integrado que é responsável por comunicar com o microcontrolador e processar a variação dos outros dois componentes. O segundo componente tem como função a medição de humidade. Internamente possui dois eléctrodos, existindo, entre eles, um substrato de retenção de humidade. Quando a humidade varia, a resistência entre os dois eléctrodos altera. Essa mudança é medida e processada pelo circuito integrado. O terceiro componente tem como função medir a temperatura. É simplesmente um sensor de temperatura NTC (*Negative Temperature Coefficient*), normalmente feito de cerâmica ou polímeros. Basicamente a resistência dele diminui com o aumento da temperatura. Este valor resistivo é lido e interpretado pelo circuito integrado (Dejan, s.d.). Existem várias bibliotecas dedicadas ao sensor DHT11, sendo possível, de forma simples e rápida, fazer a leitura dos dados de temperatura e humidade.

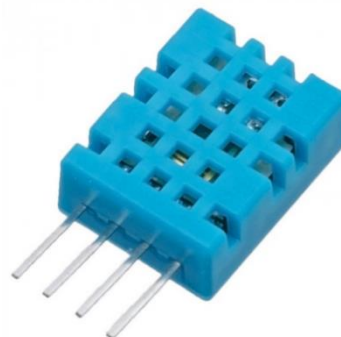


Figura 27. Sensor de temperatura e humidade DHT11 ([https://ardushop.ro/681-large\\_default/dht11-digital-temperature-and-humidity-sensor-module.jpg](https://ardushop.ro/681-large_default/dht11-digital-temperature-and-humidity-sensor-module.jpg))

### 3.4. HMI NEXTION

Nextion é uma solução de Interface Homem Máquina (*Human-Machine Interface*, HMI, na literatura anglo saxónica) que combina um processador *onboard*, um ecrã tátil e memória. É compatível com qualquer microcontrolador ou dispositivo que possua portas *Universal Asynchronous Receiver-Transmitter* (UART). É indicado para aplicações *Internet of things* (IoT) e, com a sua facilidade de programação, é das melhores soluções do mercado. Para a programação e desenvolvimento de ambiente gráfico, o fabricante disponibiliza um software gratuito chamado “*Nextion Editor*”, bastante fácil e intuitivo, onde se pode desenvolver de forma rápida toda a interface, arrastando e soltando componentes gráficos (gráficos, botões, texto, *slider* etc.). O objetivo principal deste fabricante foi reduzir a carga de trabalho no desenvolvimento da interface gráfica (Nextion, s.d.).

Este HMI tem uma grande vantagem quando comparado a outras interfaces existentes no mercado, pois efetua o armazenamento e processamento de toda a informação, reduzindo assim a utilização do microcontrolador utilizado. De uma forma geral, a troca de informação entre a interface e o microcontrolador é baseada em comandos e trocas de variáveis. A comunicação, como foi dito antes, é efetuada através da porta UART existente no próprio HMI, sendo que dispõe dos pinos de transmissão (TX) e receção (RX) como também dos pinos de alimentação (5 V e GND). Alguns modelos que a empresa Nextion comercializa oferecem suporte a *General Purpose Input/Output* (GPIO) (possuem entradas e saídas), de forma a poderem operar pequenos projetos sem a necessidade de um microcontrolador externo. De forma a gravar e armazenar dados, como também atualizar o *firmware*, possui uma entrada para cartões microSD. De realçar que a atualização do ambiente gráfico é feita a partir de um ficheiro criado no programa do fabricante e armazenado no cartão. O cartão depois de inserido no HMI e ligado à alimentação, automaticamente descarrega o ficheiro e instala-o no HMI. Após isto é só retirar o cartão do HMI, desligar e voltar a ligar a alimentação.

Neste projeto foram usados dois modelos, um para o emissor (Figura 28) e outro para o recetor (Figura 29). No Apêndice B e Apêndice C estão descritas as principais características de ambos os dispositivos.



Figura 28. HMI Nextion 7" Enhanced Series (fonte: Nextion<sup>28</sup>)



Figura 29. HMI Nextion 2,8" Basic Series (fonte: Nextion<sup>29</sup>)

### 3.5. Acoplador ótico PC817

O acoplador ótico é um dispositivo que faz o isolamento elétrico entre duas partes, normalmente com tensões ou potências diferentes. Para tal, o seu interior de um lado é composto por um fototransistor e do outro por um sinal luminoso, um LED. A grande vantagem deste dispositivo é proteger circuitos mais sensíveis que normalmente se ligam a entradas/saídas de um microcontrolador. Dentro dele, como foi dito anteriormente, existe um diodo emissor de luz e um fototransistor, devidamente posicionados, de forma que quando o LED é ligado, o fototransistor é acionado, fazendo assim a comutação totalmente isolada entre dois circuitos (Apogeweb, 2018).

O acoplador ótico utilizado no projeto é o PC817 da Sharp (Figura 31), que trabalha com uma tensão máxima entre o coletor e o emissor de 80 V, e possui uma tensão alta de

---

<sup>28</sup> <https://cdn.nextion.tech/wp-content/uploads/2017/07/Nextion-Enhanced-Generic-7.0-HMI-Touch-Display-1.jpg>

<sup>29</sup> <https://cdn.nextion.tech/wp-content/uploads/2017/07/Nextion-2.8-HMI-LCD-Touch-Display-2.jpg>

isolamento entre um circuito e outro (5 kV). O LED interno cria uma queda de 1,2 a 1,4 V quando submetido uma corrente de 20 mA. A corrente máxima que é suportada pelo LED é cerca de 50 mA e a corrente máxima do coletor é de 50 mA. Abaixo (Figura 30) é apresentado o esquema interno do acoplador ótico PC817 (SHARP, 2003).

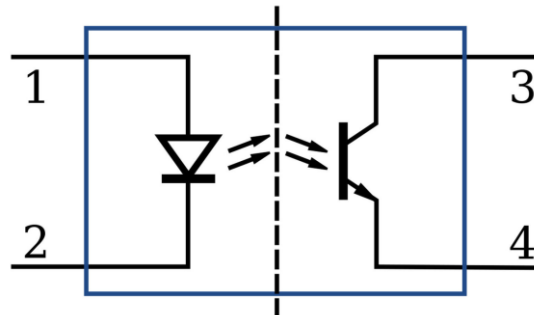


Figura 30. Esquema elétrico interno do Acoplador Ótico PC817 (fonte: All Top Notch <sup>30</sup>)

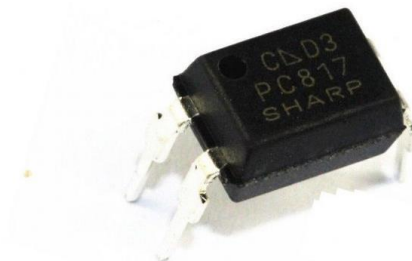


Figura 31. Acoplador Ótico PC817 da Sharp (fonte: Makers <sup>31</sup>)

### 3.6. Células de Iões de lítio (18650)

Neste projeto é necessária uma bateria que possua uma grande densidade de energia (de forma a ocupar o menor volume possível) e que conte com uma corrente de descarga alta. Para tal, das várias alternativas no mercado, as células *Lithium-ion* (Li-ion) demonstram ser uma boa opção, pois são capazes de fornecer grandes quantidades de energia (S.T.A., s.d.). No caso dos módulos recetores desenvolvidos, são necessárias células com uma alta capacidade de armazenamento de energia e que possuam uma corrente de descarga também alta. Assim, foram escolhidas as células da Panasonic NCR18650B (Figura 33), com a capacidade de 3350 mAh, 3,6 V de tensão nominal e com uma corrente de descarga de 6,7 A contínuos. No caso do emissor, são utilizadas células da Samsung INR18650-35E (Figura 32), com 3400 mAh e corrente de descarga contínua até 8 A. Como atualmente o

<sup>30</sup> <https://alltopnotch.co.uk/wp-content/uploads/imported/1/5-x-PC817-Transistor-Output-Optocoupler-Photocoupler-DIP4-EL817-LTV817-232255693541-3.JPG>

<sup>31</sup> <https://www.makers.pt/wp-content/uploads/2019/03/pc817sharp.jpg>

preço no mercado português das células da Samsung é menor em comparação com as da Panasonic, e as da Samsung oferecem maior corrente de descarga, são para ambos os módulos a melhor escolha.

Tabela 3. Especificações das células 18650 SAMSUNG INR 18650-35E e PANASONIC NCR18650B <sup>33</sup>

Especificações	SAMSUNG INR 18650-35E <sup>32</sup>	PANASONIC NCR18650B <sup>33</sup>
<b>Tensão de carga</b>	4,2 V	4,2 V
<b>Tensão nominal</b>	3,6 V	3,6 V
<b>Capacidade</b>	3400 mAh	3350 mAh
<b>Tensão de corte</b>	2,65 V	2,5 V
<b>Corrente de carga</b>	1700 mA	1625 mA
<b>Corrente descarga Max</b>	8 A	6,7 A
<b>Contínua</b>		
<b>Peso</b>	50 g	47,5g
<b>Tempo do carregamento</b>	+4 h	+4 h



Figura 32. Célula 18650 Samsung 18650-35E (fonte: Baltrade <sup>34</sup>)



Figura 33. Célula 18650 Panasonic NCR18650B (fonte: ESTUDIOPT <sup>35</sup>)

<sup>32</sup> <https://www.orbtronic.com/content/samsung-35e-datasheet-inr18650-35e.pdf>

<sup>33</sup> <https://www.batteryspace.com/prod-specs/NCR18650B.pdf>

<sup>34</sup> [https://shop.baltrade.eu/img/product/big/akumulator\\_18650\\_Li\\_ion\\_Samsung\\_INR18650\\_35E\\_3500mAh.png](https://shop.baltrade.eu/img/product/big/akumulator_18650_Li_ion_Samsung_INR18650_35E_3500mAh.png)

<sup>35</sup> <https://www.estudiopt.pt/images/thumbnails/734/600/detailed/43/estudiopt15672.jpg>



### 3.7. Reguladores e conversores de tensão

O LM117 é um regulador linear ajustável com a capacidade de operar até 1,5 A de corrente máxima e possui proteção contra curto-circuito. Oferece também proteção contra sobrecarga térmica. A tensão de entrada máxima é de 40 V e a tensão mínima é de 4,2 V. A tensão de saída pode ir de 37 V de tensão máxima e 1,25 V de tensão mínima. Trabalha num intervalo de temperatura entre -55 °C e 150 °C (Texas Instruments, 2020). É recomendado usar um dissipador de calor quando a potência dissipada pelo LM117 é grande. O regulador normalmente é usado em circuitos mais simples que exijam pouca corrente e quando a diferença de tensão entre a entrada e a saída é reduzida. Na Figura 34 é apresentado um circuito típico composto pelo LM117, dois condensadores e um potenciômetro que regula a saída  $V_{OUT}$ .

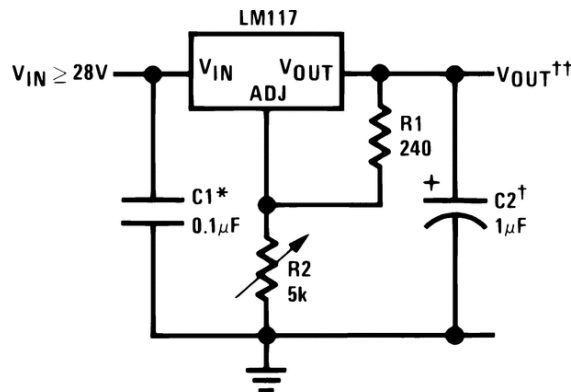


Figura 34. Típica aplicação do regulador linear LM117 para obter uma tensão de saída regulada pelo potenciômetro R2 (fonte: Texas Instruments <sup>36</sup>)

O módulo LM2596 (Figura 35) é um conversor *step down* (conversor *buck*), e é composto pelo conversor LM2596. Possui a capacidade de operar com tensões de entrada entre 3,2 V até 40 V e converter para uma saída regulada de 1,5 V até 35 V com 3 A de corrente máxima. Possui um oscilador de frequência fixa de 150 kHz. O seu consumo em *standby* é cerca de 80  $\mu$ A (Botnroll, s.d.).

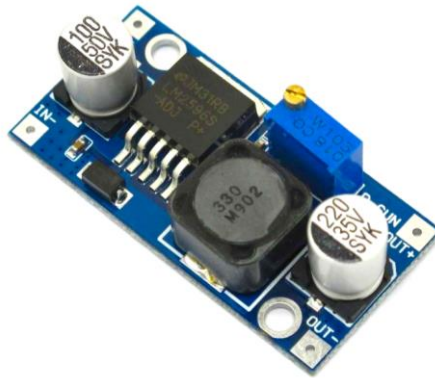


Figura 35. Módulo step down LM2596 (fonte: Botnroll<sup>37</sup>)

O LM2576 é um regulador de tensão *step down* (conversor *buck*), com uma elevada eficiência, capaz de suportar tensões de entrada entre 7 V e 40 V e regular para tensões de saída entre 1,23 V até 37 V. Suporta corrente de saída até 3 A. Possui um oscilador de frequência fixa de 52 kHz. O seu consumo em standby é cerca de 50  $\mu$ A. Trabalha num intervalo de temperaturas entre -40 °C e 125 °C (ON Semiconductor). Na Figura 36 é apresentado um circuito típico para obter uma tensão de saída regulada. A tensão é regulada, alterando os valores de R1 e R2.

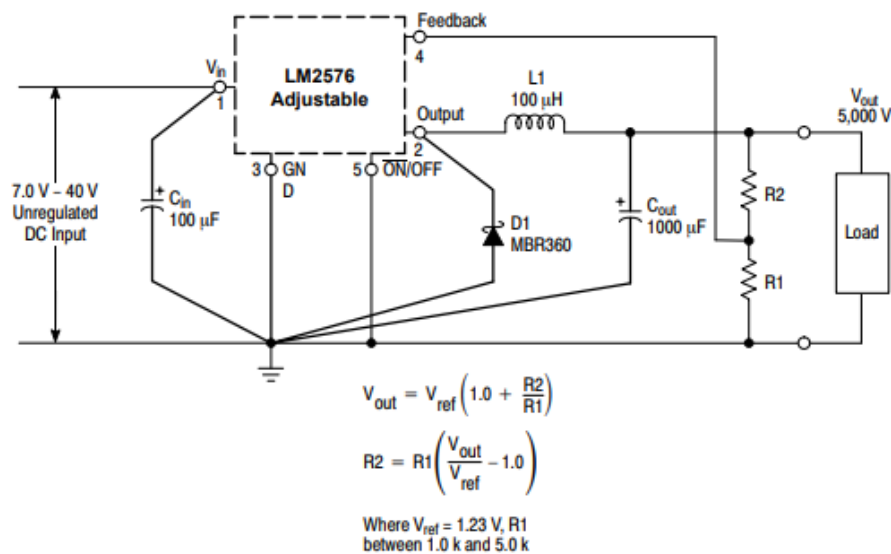


Figura 36. Circuito típico com LM2576 com saída ajustável (fonte: mouser<sup>38</sup>)

O Mini-360 (Figura 37) é um conversor *step down* (conversor *buck*). Ele é composto pelo conversor MP2307DN e suporta tensões de entrada recomendadas entre 4,75 V até 23 V (contudo o fabricante do MP2307DN indica que a tensão máxima pode chegar até 26 V). O módulo consegue regular a saída entre 1 V e 17 V com uma corrente máxima de 3 A.

<sup>37</sup> [https://www.botnroll.com/13542-large\\_default/lm2596-step-down.jpg](https://www.botnroll.com/13542-large_default/lm2596-step-down.jpg)

<sup>38</sup> [https://pt.mouser.com/datasheet/2/308/LM2576\\_D-1810688.pdf](https://pt.mouser.com/datasheet/2/308/LM2576_D-1810688.pdf)

Trabalha com uma frequência de comutação fixa de 340 kHz. Possui uma elevada eficiência, podendo esta chegar a mais de 95%. O intervalo de temperaturas recomendadas para o seu correto funcionamento situa-se entre -40 °C e 85 °C (CircuitRocks, s.d.).

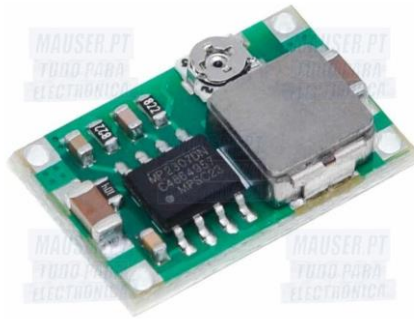


Figura 37. Módulo conversor step down Mini-360 (fonte: Mauser <sup>39</sup>)

O XHM401 (Figura 38) é um conversor *step down* (conversor *buck*). O módulo utilizado é composto pelo conversor XL4016, e segundo os dados da loja que comercializa o módulo com este conversor, suporta tensões de entrada entre 4 V e 40 V. A saída pode ser ajustada entre 1,25 V e 36 V. Possui uma elevada eficiência, mais de 94%. Apresenta a capacidade de trabalhar com uma corrente máxima de 8 A. A frequência de comutação é de 180 kHz. O módulo *step down* utilizado no projeto possui proteção contra sobrecarga de corrente, contra sobreaquecimento e contra curto-circuito e trabalha com um intervalo de temperatura entre -45 °C até 85 °C. O módulo já possui um retificador MBR10200 (Desertcart, s.d.).

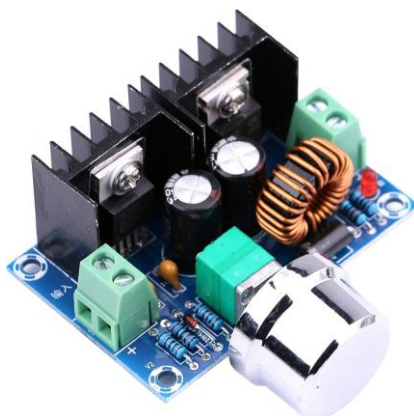


Figura 38. Módulo step down XHM401 (fonte: Amazon <sup>40</sup>)

<sup>39</sup> <https://mauser.pt/images/27519b620d1562fc97fa746a7e1091a7.jpg>

<sup>40</sup> [https://images-na.ssl-images-amazon.com/images/I/71D41an20-L.\\_AC\\_SL1500\\_.jpg](https://images-na.ssl-images-amazon.com/images/I/71D41an20-L._AC_SL1500_.jpg)

### 3.8. Battery management system

Uma *Battery management system* (BMS) é um dispositivo de segurança recomendado em projetos que utilizem baterias de íões de lítio. Este dispositivo é responsável por monitorizar a operação de carga e descarga, protegendo assim as células. Protege as células na carga, de forma a que carreguem todas à mesma tensão. Protege no carregamento, onde a tensão, se estiver acima de um determinado limite, a BMS abre o circuito. Protege também na descarga, no momento que existe um aumento excessivo de corrente, ou até mesmo um curto-circuito, abrindo o circuito. Em casos de tensões baixas, a BMS abre o circuito de forma a poder preservar as células (Engineering, s.d.). Para o projeto foram usadas duas BMS, uma para 25,2 V 6s1p (seis células em série) e outra para 12,6V 3s1p (três células em série). Ambas possuem balanceamento (impede os desequilíbrios de tensões nas células), proteção de carga e proteção de descarga.

A BMS 6S (Figura 39) aceita 25,2 V de tensão de carga, trabalha com uma corrente máxima de 8 A, possui balanceamento e proteção contra sobrecarga.

A BMS 3S (Figura 40) aceita tensões de carga de 12,6 V, trabalha com uma corrente máxima de 40 A, possui balanceamento e proteção contra sobrecarga.

Ambas são BMS genéricas, sendo que são usadas apenas por precaução de não danificar as células e por aumentar a longevidade das mesmas.

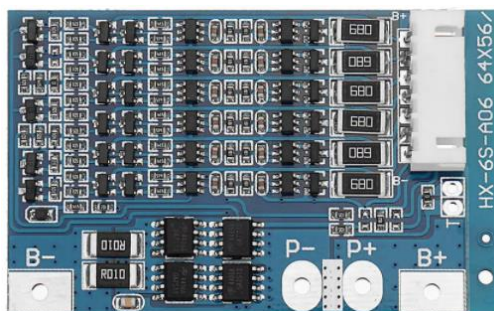


Figura 39. BMS 6S (fonte: Bangood <sup>41</sup>)

<sup>41</sup> [https://pt.banggood.com/6S-22\\_2V-Li-ion-18650-Lithium-Battery-BMS-Charger-Protection-Board-With-Balance-Integrated-Circuits-p-1321865.html?rmmds=search&cur\\_warehouse=CN](https://pt.banggood.com/6S-22_2V-Li-ion-18650-Lithium-Battery-BMS-Charger-Protection-Board-With-Balance-Integrated-Circuits-p-1321865.html?rmmds=search&cur_warehouse=CN)

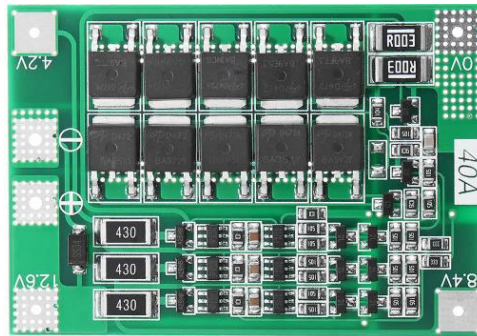


Figura 40. BMS 3S (fonte: Bangood <sup>42</sup>)

### 3.9. Arduino Mega 2560

O Arduino Mega com o microcontrolador ATmega2560 foi idealizado para projetos com um nível de complexidade maior, normalmente projetos com um número elevado de entradas e saídas. Conta com 54 pinos de entrada/saída digitais e 16 entradas analógicas.

Principais características do Arduino Mega 2560:

- Microcontrolador ATmega2560 com um clock de 16 MHz;
- Regulador de tensão 5 V;
- Regulador de tensão 3,3 V;
- Quatro portas série;
- Uma porta I2C;
- Uma porta *Serial Peripheral Interface* (SPI);
- Dezasseis portas analógicas;
- 54 portas digitais *Input/Output* (I/O);

Existem três formas de alimentar o Arduino: pela porta USB, pela entrada  $V_{IN}$  (relativamente à qual o fabricante recomenda tensões de entrada entre 7 V a 12 V) ou através da porta 5 V, com os respetivos 5 V.

A placa possui ainda um botão de RESET e um conector *In-Circuit Serial Programming* (ICSP) para gravar *firmware*.

As portas digitais do Arduino Mega podem suportar até 20 mA. Todos os pinos digitais e analógicos funcionam com tensões entre 0 a 5 V.

---

<sup>42</sup> [https://pt.banggood.com/3S-40A-Li-ion-Lithium-Battery-Charger-Protection-Board-PCB-BMS-For-Drill-Motor-11\\_1V-12\\_6V-Lipo-Cell-Module-With-Balance-p-1394450.html?cur\\_warehouse=CN](https://pt.banggood.com/3S-40A-Li-ion-Lithium-Battery-Charger-Protection-Board-PCB-BMS-For-Drill-Motor-11_1V-12_6V-Lipo-Cell-Module-With-Balance-p-1394450.html?cur_warehouse=CN)

A placa possui um outro microcontrolador, o ATmega16U2, responsável pela comunicação entre o microcontrolador ATmega2560 e o PC (Murta, 2018).

No projeto é utilizada uma *board* mais compacta (Arduino Mega 2560 Pro Mini) (Figura 41) comparativamente à original do Arduino Mega, contendo na mesma o microcontrolador ATmega2560; no entanto, em vez do microcontrolador ATmega18U2 possui o CH340G. O Apêndice D descreve as principais características do Arduino Mega 2560 Pro Mini.

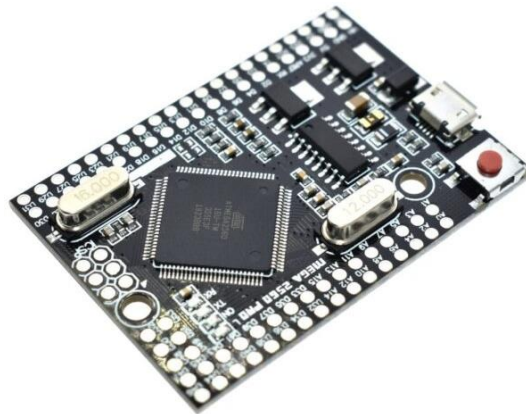


Figura 41. Arduino Mega 2560 Pro Mini (fonte: Digipart<sup>43</sup>)

### 3.10. Módulo Rádio Ai-Tinker Ra-02

O módulo de rádio Ra-02 (Figura 42) da empresa Ai-Tinker, permite a comunicação através da modulação LoRa, o que faz com que tenha um longo alcance aliado a um baixo consumo de energia. Comparativamente a outras propostas semelhantes no mercado, este dispositivo destaca-se pelo seu baixo custo e por já possuir um adaptador para a antena. O módulo é baseado no chip SX1278 da SEMTECH e, na configuração atual, permite uma potência máxima de transmissão até  $18 \pm 1$  dBm e trabalha na frequência de 433 MHz. Permite vários tipos de modulações, como FSK, GFSK, MSK, GMSK, OOK e LoRa. A alimentação típica é de 3,3 V, contudo pode ser ligada a tensões entre 2,5 V e 3,7 V. No que toca ao seu consumo de energia, consome apenas 93 mA (a transmitir a 433 MHz) quando ligado a 3,3 V. Possui ainda o modo de *standby* com consumo de apenas 1,6 mA. Além disto apresenta um conjunto de pinos I/O que podem ser configurados para trabalhar como interrupções (Ai-Thinker, 2017). As antenas utilizadas possuem um ganho de 5 dBi. De seguida, na Tabela 4, estão as principais características do módulo.

---

<sup>43</sup> <https://digipart.pk/wp-content/uploads/2020/05/Mega-2560-PRO-MINI-5V-Embed-CH340G-ATmega2560-16AU-with-male-pinheaders-Compatible-for-arduino-Mega-4-600x600.jpg>

Tabela 4. Características do módulo de rádio Ra-02 (fonte: LCSC<sup>44</sup>)

<b>Chip Transmissor</b>	SX1278
<b>Modulações de comunicação por rádio suportadas</b>	FSK, GFSK, MSK, GMSK, LoRa e OOK
<b>Frequência de comunicação</b>	De 410 MHz até 525 MHz
<b>Interface de comunicação</b>	SPI
<b>Potência máxima de transmissão</b>	18 +-1 dBm
<b>Tensão de alimentação</b>	2,5 V até 3,7 V



Figura 42. Módulo de rádio RA-02 (fonte: ELECTRONICS-LAB<sup>45</sup>)

### 3.11. Módulo SD Card Velleman

Para o armazenamento de informação, principalmente para guardar os tempos de disparo do modo automático, utilizou-se um cartão SD. Este dispositivo de armazenamento de dados, com memória *flash*, é utilizado por todo o tipo de tecnologia em que seja necessário guardar dados, como telemóveis, câmaras fotográficas, etc.

Este módulo possui integrado um leitor de cartões SD que comunica com o microcontrolador através do barramento SPI (pinos MOSI, SCK, MISO e CS). A memória responsável pelo armazenamento dos dados do cartão é a memória flash (um *chip* do tipo *Electrically Erasable Programmable Read-Only Memory* (EEPROM)), o que faz com que a gravação e a eliminação dos dados seja feita eletricamente. Ou seja, é uma tecnologia não volátil sendo que as informações ficam armazenadas sem necessidade de uma alimentação elétrica permanente. O módulo utilizado é o modelo VMA304 da Velleman (Figura 43). Pode ser alimentado com 3,3 V ou 5 V, suporta formatos de arquivo FAT16 e FAT32 e ainda cartões de armazenamento até 2 GB. Comparado a outros existentes no mercado, este

<sup>44</sup> [https://datasheet.lcsc.com/szlcsc/1811151537\\_Ai-Thinker-Ra-02\\_C90039.pdf](https://datasheet.lcsc.com/szlcsc/1811151537_Ai-Thinker-Ra-02_C90039.pdf)

<sup>45</sup> <https://www.electronics-lab.com/wp-content/uploads/2020/01/Ra-02.jpg>



apresenta documentação e um preço acessível. Foi necessária uma expansão de memória externa devido à baixa memória interna do microcontrolador ATmega2560 e, também, do número limitado de escritas em memória que ele suporta (Velleman, 2017).



Figura 43. Módulo VMA304 da Velleman para cartões SD (fonte: Velleman <sup>46</sup>)

### 3.12. Conversores de nível lógico Sparkfun

O conversor de nível lógico bidirecional da Sparkfun (Figura 44) é um dispositivo pequeno, que é capaz de reduzir os sinais de 5 V para 3,3 V e aumentar de 3,3 V para 5 V ao mesmo tempo.

Cada conversor de nível lógico possui a capacidade de converter quatro pinos com nível lógico alto em quatro pinos com nível lógico baixo e vice-versa.

No caso do projeto ele é necessário para estabelecer a comunicação entre o módulo de rádio, que trabalha com nível lógico de 3,3 V, e o microcontrolador, que trabalha com nível lógico de 5V. O conversor, possui quatro MOSFETs de canal N e algumas resistências *pull-up* para realizar a mudança de nível lógico bidirecional (SparkFun, s.d.).

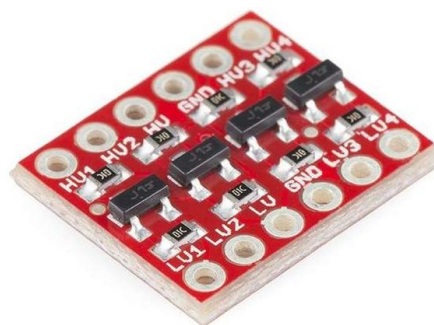


Figura 44. Módulo conversor nível logico bidirecional (fonte: eletrofun <sup>47</sup>)

<sup>46</sup> <https://www.velleman.eu/images/products/29/vma304.jpg>

<sup>47</sup> [https://www.electrofun.pt/1449-large\\_default/conversor-nivel-logico-bidirecional.jpg](https://www.electrofun.pt/1449-large_default/conversor-nivel-logico-bidirecional.jpg)



### 3.13. Transistor 2N3904

Um transistor é um semicondutor linear que tem como função controlar a corrente com a aplicação de um sinal elétrico de potência reduzida. O transistor possui três terminais: base, emissor e coletor. Existem dois tipos: o bipolar e o efeito de campo. O transistor bipolar usa uma pequena corrente para controlar uma grande corrente. Existem dois tipos de transistores bipolares: o NPN e PNP. A diferença entre eles consiste na polarização das junções durante a operação. De uma forma geral, os transistores bipolares, através de uma pequena corrente, controlam a corrente que vai do coletor para o emissor (PNP) ou do emissor para o coletor (NPN). A corrente que controla este processo vai da base para o emissor (PNP), ou do emissor para a base (NPN) (Figura 45). De acordo com a simbologia, a seta aponta contra a direção do fluxo de elétrons. O transistor pode ser usado como um interruptor, ou seja, consegue controlar uma grande carga, a partir de uma corrente reduzida (fornecida por exemplo por uma saída de um microcontrolador).

O transistor pode trabalhar na zona de corte, sendo que a corrente que circula entre o emissor e o coletor, ou vice-versa, é igual a zero.

Quando trabalha na zona ativa, basicamente controla a corrente que circula entre o emissor e o coletor ou vice-versa.

Se tiver em saturação a corrente entre emissor e coletor ou vice-versa é máxima (corrente da carga).

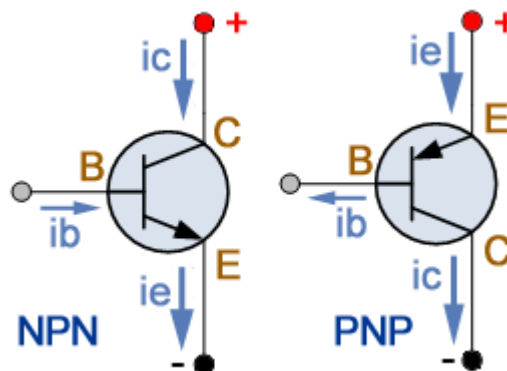


Figura 45. Esquema com sentidos das correntes nos transistores bipolares NPN e PNP (fonte: Raffaele Ilardo <sup>48</sup>)

No projeto é utilizado o transistor 2N3904, do tipo NPN, para fazer o controle da base do MOSFET tipo P IR9530. Este transistor suporta uma corrente de coletor máxima ( $i_c$ ) de 200

<sup>48</sup> <http://www.raffaeleilardo.it/pnonp1.png>

mA. Suporta uma tensão máxima do emissor para a base de 6 V, do coletor para o emissor de 40 V, e da base para o coletor de 60 V. O ganho de corrente no máximo é de 300. O seu encapsulamento é o TO-92 (Components101, 2017). Na Figura 46 é apresentada a descrição de cada um dos pinos.

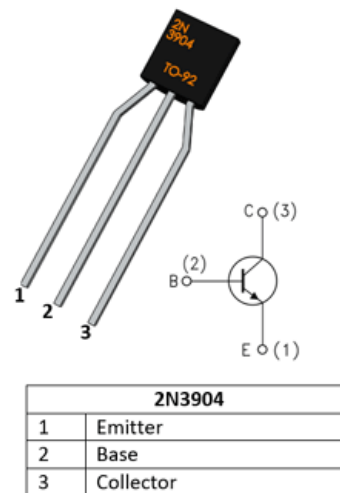


Figura 46. Descrição dos pinos e esquema do transistor bipolar PNP 2N3904 (fonte: Components101 <sup>49</sup>)

### 3.14. Fusíveis de proteção

Os fusíveis são dispositivos usados para proteção de componentes elétricos contra curtos-circuitos e sobrecargas. Consistem num metal ou fio de baixa resistência que tem a finalidade de fechar um determinado circuito. Quando a corrente que flui sobre ele ultrapassa o valor de corrente máxima suportada pelo fusível, o fio derrete e interrompe o circuito, protegendo assim diversos componentes sensíveis. A escolha de um fusível deve ser de acordo com a corrente máxima do circuito, com uma margem, normalmente de 35% a mais. Outros fatores a ter em conta incluem a sua tensão máxima, se o circuito é *Alternating Current* (AC) ou DC, o intervalo de temperatura suportada e se o efeito é rápido ou lento. Os lentos são usados em casos onde possam existir picos de corrente em determinados momentos (por exemplo: arranque de motores). Os de ação rápida normalmente são usados onde os picos de corrente sejam críticos para determinados componentes (Grote, s.d.).

<sup>49</sup> [https://components101.com/sites/default/files/component\\_pin/pinout-of-2N3904-npn-transistor\\_3.png](https://components101.com/sites/default/files/component_pin/pinout-of-2N3904-npn-transistor_3.png)

## 4. Desenvolvimento do Projeto

Neste capítulo são apresentadas, por ordem cronológica, todas as etapas do desenvolvimento do projeto. São explicados ainda os testes finais realizados com os módulos desenvolvidos.

### 4.1. Descrição do desenvolvimento do projeto

Inicialmente foi realizado um estudo sobre as várias opções de HMI existentes no mercado e sobre vários microcontroladores. O HMI escolhido foi da empresa Nextion devido à presença de um microcontrolador interno, ser intuitivo de programar e o fabricante possuir uma IDE intuitiva para se poder desenvolver a componente gráfica.

Em relação ao microcontrolador para controlo do projeto (módulo emissor e módulo recetor), o ATmega2560 mostrou ser um dos mais competentes, visto oferecer uma quantidade significativa de entradas/saídas. O Arduino Mega 2560 Pro Mini foi a placa de desenvolvimento escolhida uma vez que o seu microcontrolador é o ATmega2560.

Após isto, foi analisado o funcionamento da IDE do HMI Nextion, analisando as principais características da mesma. Foram ainda realizados os primeiros testes de comunicação do ATmega2560 e foi configurado no ficheiro de configurações da biblioteca Nextion.h a porta série 2 como predominante para a comunicação.

Após conclusão dos primeiros testes de comunicação entre o ATmega2560 e o Nextion, foi pensado um plano de como seriam guardados os tempos por cada conjunto canal + número. A solução final passou por criar duas matrizes, uma de inteiros para armazenar o tempo em segundos e outra do tipo byte, apenas para armazenar as décimas de segundo. Optou-se por esta solução com o intuito de economizar espaço, pois caso fosse utilizada uma matriz do tipo *double*, para armazenar o tempo de cada conjunto, seriam ocupados quatro bytes e, com a solução final, são apenas ocupados três bytes (dois bytes do número inteiro para armazenar os segundos e um byte para armazenar as décimas de segundo). Devido à limitação da memória *Static Random-Access Memory* (SRAM) do ATmega2560 foi estabelecido trabalhar com matrizes [15][64], ou seja, possibilitam a utilização de 64 números diferentes por cada quinze canais. Para aumentar o número de canais a solução passaria por trocar a matriz (que armazena os segundos) do tipo inteiro para o tipo byte, contudo, esta alternativa só daria para armazenar 255 segundos por cada conjunto, o que

seria limitativo. Com a configuração [15][64] o sistema tem a capacidade de disparar em modo automático cerca de 960 números diferentes (15 canais  $\times$  64 números).

Depois foi elaborada a componente gráfica de uma página no HMI e a própria programação para a inserção de tempos por cada conjunto canal + número no modo automático.

Em seguida foi realizado um estudo do comportamento e funcionamento de alguns ignitores elétricos presentes no mercado, onde foi analisado, em cada ignitor, a sua corrente de disparo recomendada e a corrente máxima de teste (serve para verificar continuidade em cada terminal com garantia de não entrarem em ignição) recomendada pelos fabricantes.

Para alimentar o circuito de disparo a bateria ideal é uma que contenha uma corrente de descarga de pelo menos 5 A, o que permite disparar num terminal cerca de três a quatro ignitores em paralelo (dependendo do modelo do ignitor). No que toca à tensão, na maioria das montagens de espetáculos pirotécnicos, a utilização de ligações em série em cada terminal do recetor é o mais habitual. Como tal, a utilização de uma bateria com pelo menos 15 V é essencial, o que permite disparar cerca de dez ignitores ligados em cada terminal (dependendo do modelo do ignitor).

Foram analisadas algumas baterias presentes no mercado com o intuito de escolher a solução de alimentação que encaixasse melhor no projeto. As células 18650 da Panasonic NCR18650B, demonstravam ter características excelentes, uma vez que a loja que as comercializa, indicava que as mesmas suportavam uma corrente de descarga contínua até 8 A. Como cada célula carregada possui 4,2 V, optou-se por elaborar um conjunto de seis células em série (Figura 47) fazendo subir a tensão para 25,2 V nos terminais do conjunto, quando completamente carregadas.

Para proteção e balanceamento das baterias foi adicionada uma BMS 6s ao conjunto.

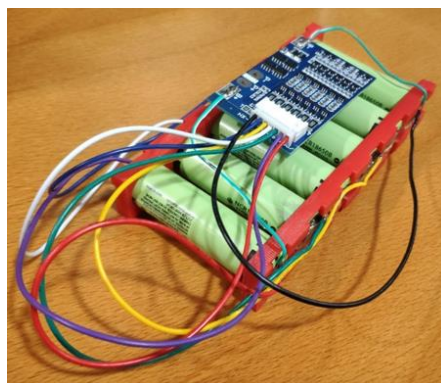


Figura 47. Bateria composta por seis células em série Panasonic NCR18650B e uma BMS 6s

Após tudo isto foi estudado um circuito que fizesse a deteção e o disparo de um ignitor num terminal. Depois da pesquisa dos componentes necessários, foi testado o circuito no *software* Proteus, de forma a analisar o seu comportamento, sendo, de seguida, testado numa *protoboard*.

Com o intuito de aumentar as saídas/entradas do Arduino, de forma a suportar vários números, a solução passou por usar um *multiplexer/demultiplexer*. Assim, em cada régua de terminais optou-se por usar dois dispositivos *multiplexer/demultiplexer* de dezasseis canais.

O circuito da Figura 48 serve para controlo e monitorização de dois terminais (saídas para conectar ignitores). No circuito, os pinos de controlo de seleção de porta do MUX (*multiplexer*) e DEMUX (*demultiplexer*) são partilhados de forma a poupar pinos I/O do Arduino. O circuito possui pinos de 5 V, 15 V, GND, quatro pinos de controlo do MUX e DEMUX, um pino de entrada de sinal do DEMUX e outro para a saída do MUX. Os pinos *enable* do MUX e DEMUX estão conectados ao GND de forma a estarem sempre ativos.

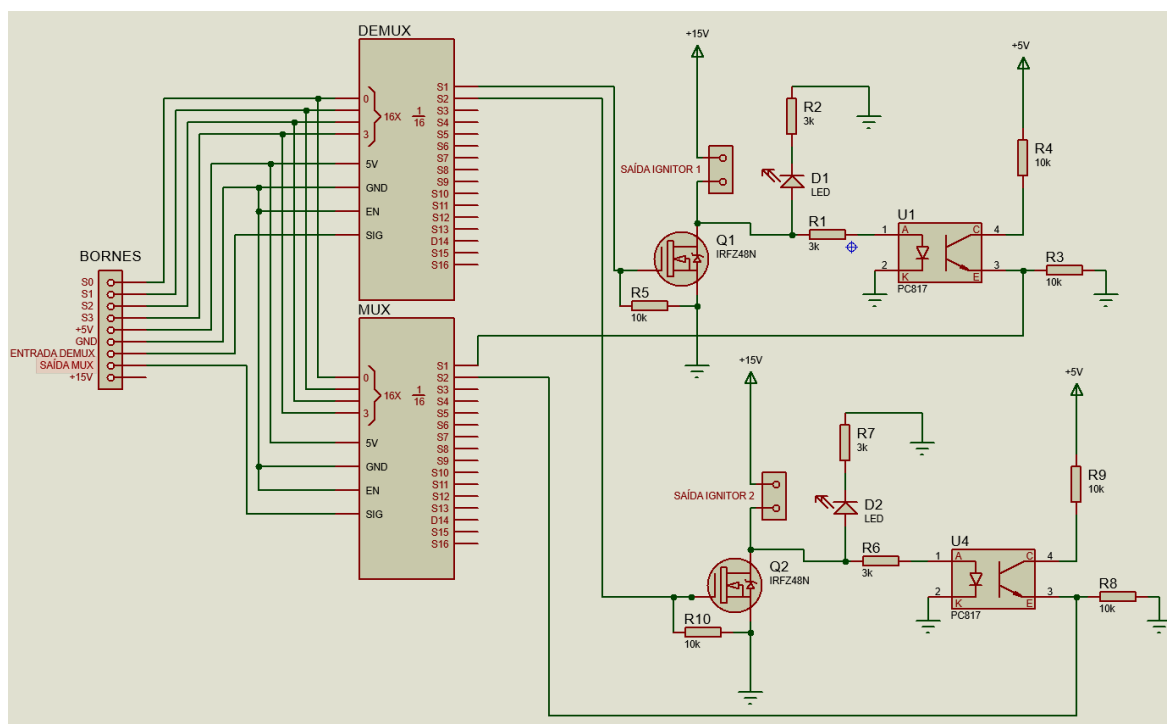


Figura 48. Circuito de teste desenvolvido para duas saídas da primeira régua de terminais

Para ligar a régua ao Arduino foi usado um cabo VGA (quinze pinos), opção que pareceu adequada uma vez que possui doze condutores independentes e três condutores comuns. Seis condutores foram usados para os MUX/DEMUX, um para os 5 V e os restantes para a alimentação com 15 V e GND. Basicamente o cabo VGA faz a ponte de dados para o Arduino e também o transporte de alimentação para o circuito da régua.

Para primeiro protótipo de testes foi desenvolvida uma placa com um circuito para uma régua que contém apenas seis terminais.

O circuito consiste no disparo dos MOSFETs tipo N (IRFZ48N)(Q1) colocando-os em saturação (o que faz disparar o ignitor, ligado entre o dreno do MOSFET e o pino dos 15 V) através de uma saída do *demultiplexer*. Para monitorizar a continuidade de cada terminal, um LED vermelho (D1) acende sempre que um ignitor é colocado (o ignitor fecha o circuito). Para monitorizar através do Arduino a presença de um ignitor num determinado terminal, foi usado um acoplador ótico (U1), no qual o seu LED interno acende sempre que um ignitor é ligado (fecha o circuito). Assim, o fototransístor do acoplador ótico entra em condução e coloca 5 V numa das entradas do *multiplexer* (depois, a partir do Arduino, é possível saber se contém um ignitor conectado nesse terminal). No circuito não está representada, contudo foi adicionada uma resistência de potência de 3  $\Omega$  25 W conectada ao pino dos 15 V de forma a limitar a corrente do circuito (inicialmente foram realizados testes com saída até 15 V 5 A devido à limitação de corrente máxima da fonte de alimentação de bancada). Foi modelado o chassi para a régua de seis terminais, no *software* Autodesk Inventor, de forma a acomodar a placa eletrónica desenvolvida (Figura 49). Esta caixa foi criada com recurso à impressão 3D do tipo *Fused Deposition Modeling* (FDM).

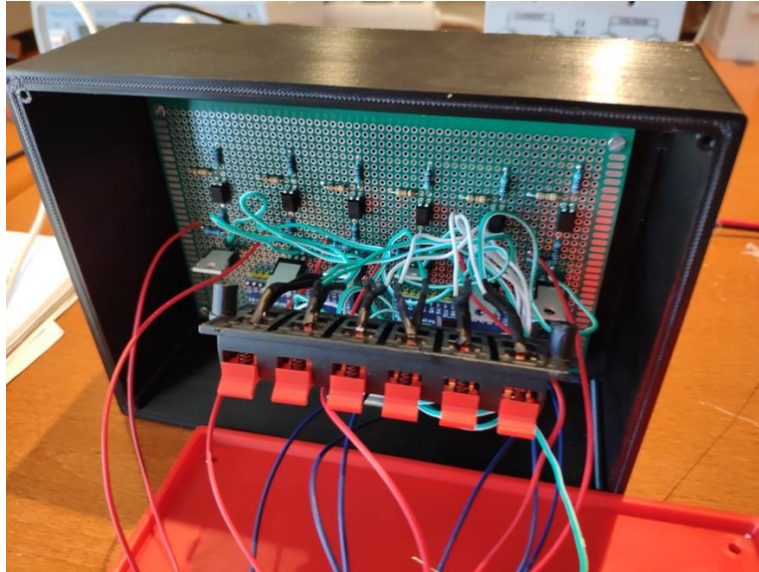


Figura 49. Montagem da placa desenvolvida baseada no circuito da Figura 48

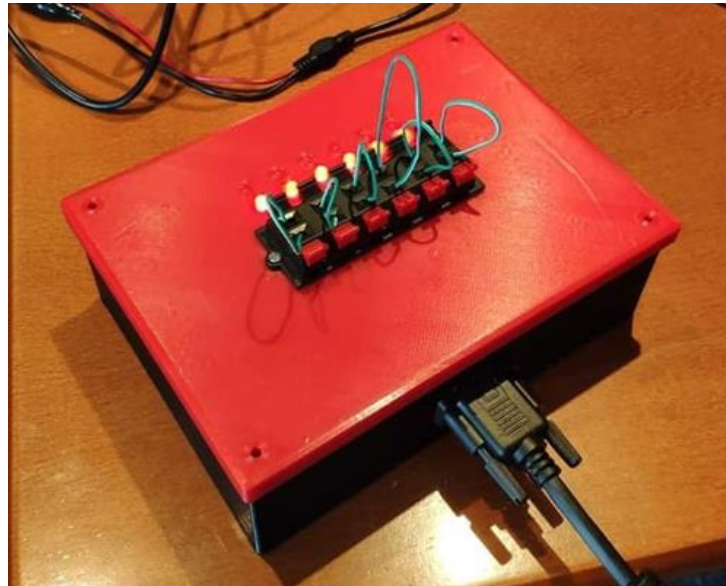


Figura 50. Régua de seis terminais, com shunt em cada terminal, de forma a poder observar o teste de continuidade de cada terminal

Após alguns testes da régua desenvolvida (Figura 50) realizados com um Arduino Uno, através de um pequeno programa, este demonstrou conseguir controlar cada MOSFET através de um *demultiplexer* e também detetar através do *multiplexer* se o ignitor estava presente no terminal. Contudo, foi detetado um erro nesta primeira régua desenvolvida, pois o ignitor ao ser colocado num dos terminais da régua, ficava logo com um dos seus terminais ligado ao VCC (+15 V), pois o MOSFET presente era do tipo N, ou seja, o ignitor estava a servir como “ponte” entre o dreno do MOSFET e o VCC.

Este erro era grave, pois caso o ignitor estivesse com um dos terminais no VCC e tocasse com o outro terminal num ponto que tivesse ligação à massa (GND), este entrava logo em ignição, pois ficava com uma diferença de potencial nos seus terminais. Para além deste problema, foi também detetado um problema com o cabo VGA, pois este não possuía condutores com uma secção suficiente para suportar correntes altas (cerca de 4 a 5 A) quando o ignitor é disparado. Este último problema foi detetado pois durante um teste verificou-se um ligeiro aquecimento (2 a 3 °C ao fim de 1 min com 4,5 A) dos condutores que transportavam essa mesma corrente.

Devido a estes problemas o circuito foi analisado novamente.

Depois de realizada uma pesquisa, tornou-se evidente a troca dos MOSFETs tipo N por MOSFETs tipo P e conseqüentemente o circuito original teve de ser alterado, no qual se adicionou um transistor bipolar NPN para realizar o controlo do MOSFET. O MOSFET foi escolhido de forma a suportar pelo menos 10 A (devido às baterias trabalharem com um máximo de 8 A) e de acordo com o stock disponível nas lojas de eletrónica presentes em Portugal. O MOSFET que correspondia aos requisitos impostos foi o IRF9530. Este consegue trabalhar com uma corrente máxima de 12 A o que é excelente pois o sistema no máximo trabalhará com 7,5 A a 8 A em cada terminal.

Em relação ao cabo a ser utilizado, o DB25 mostrou ser uma excelente alternativa, uma vez que possui muitos mais condutores, possibilitando a sua junção, de forma a dividir a corrente fornecida pelos vários condutores ao(s) ignitor(es) no momento do disparo.

Depois do circuito inicial ser modificado, este foi testado novamente no *software* Proteus, onde se realizaram vários testes na *proto-board*, com apenas dois terminais, de forma a poder comprovar se os resultados obtidos no *software* eram semelhantes aos obtidos na realidade. Para controlo do circuito foi utilizado um Arduino Uno e quatro botões (Figura 51). Dois dos botões foram configurados para aceder ao estado dos terminais (indicar se o ignitor estava colocado) e os outros dois para conceber o disparo dos ignitores nos terminais.



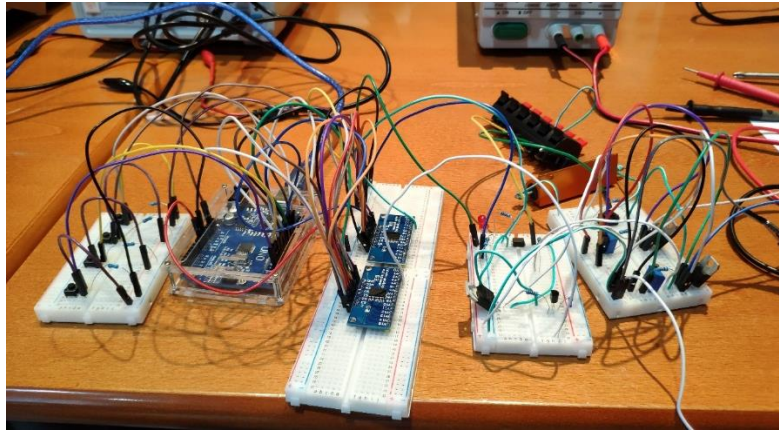


Figura 51. Circuito de teste retificado com os MOSFETs tipo P

Os resultados obtidos com o circuito foram positivos pois este funciona de acordo com o esperado.

Como o microcontrolador ATmega2560 tem limitação de 4 kB de memória EEPROM seria bastante agressivo efetuar imensas escritas sempre que fosse necessário gravar as matrizes de tempo, pois a memória EEPROM tem um número limitado de escritas. Para contornar este problema a solução passou por gravar essas matrizes num cartão microSD através de um módulo de cartões microSD (Figura 52).

Inicialmente o processo passou por perceber, através de exemplos fornecidos pela biblioteca SD, o funcionamento do processo da escrita e leitura de dados.

O processo de escrita e leitura de tempos (relativos ao modo automático) num cartão microSD, regra geral, traz vantagens ao sistema porque permite, por exemplo, no futuro, criar um pequeno programa em Windows Forms de forma a permitir ao utilizador inserir os tempos de um espetáculo pirotécnico num programa de computador e gravar esses dados no cartão microSD.

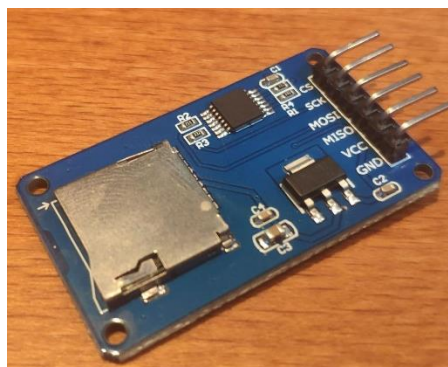


Figura 52. Módulo de cartões microSD

Para a comunicação sem fios do módulo emissor para os recetores, foi analisado o módulo de rádio EBYTE E32 433 MHz 1 W, onde se pesquisaram todas as características e capacidades do mesmo. Realizou-se um teste de distância, inicialmente através de exemplos da biblioteca EBYTE32.h modificados, onde através de antenas de ganho +5 dBi se conseguiu uma transmissão até 5 km numa zona com topografia irregular. O fabricante refere que em testes com condições de campo aberto e antena de ganho 5 dBi a comunicação pode ser conseguida até 7 a 8 km.

Em termos de distâncias é um módulo ideal, contudo, passado algum tempo foi verificado que o mesmo não detém capacidade de análise da intensidade do sinal recebido (RSSI), nem é possível determinar a relação sinal-ruído (SNR) da própria comunicação. Por este motivo, com estes módulos não era possível perceber se eles estariam na zona limite da comunicação e com isso comprometeria a qualidade e a fiabilidade da comunicação.

Assim, depois de descartar a possibilidade de usar o módulo EBYTE E32, foi realizada uma nova pesquisa onde o módulo Ai-Thinker RA-02 chamou à atenção pelo feedback de quem o testou. Para além disso este tem disponíveis as funções de *Received Signal Strength Indicator* (RSSI) e *signal-to-noise ratio* (SNR). Para comunicar com o microcontrolador este módulo utiliza o barramento SPI.

Inicialmente foram realizados alguns testes, através de exemplos da biblioteca LoRa.h, onde se testou se a comunicação entre dois módulos RA-02 era concebida com sucesso. Aqui ocorreu um problema, pois verificou-se que o módulo microSD e o RA-02 juntos originavam conflitos na comunicação com o microcontrolador. Com alguma pesquisa chegou-se à conclusão de que o módulo microSD apresentava um problema de arquitetura, o que faz com que o módulo ignore o pino *select chip*, e com isto, quando o módulo RA-02 utilizava o barramento SPI, originava conflitos. A solução passou por testar um novo módulo SD de outra marca e arquitetura. Com a aquisição de um novo módulo para cartões SD (Figura 53), o problema de conflitos entre os dois módulos ficou resolvido.

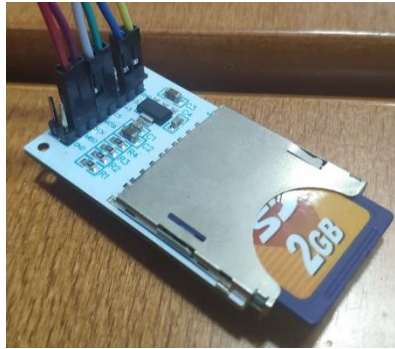


Figura 53. Módulo para cartões SD Velleman VMA304

Após isto, os módulos de rádio RA-02 foram submetidos a testes, onde se analisou a comunicação unidirecional entre dois módulos, em campo aberto, com os parâmetros padrão (Frequência = 433 MHz, Potência = 17 dBm, BW = 125 kHz, SP = 7 e CR = 4/5).

Do lado do emissor, o programa enviava uma mensagem genérica por segundo, composta pelo texto *"Hello World"* e o ID da mensagem (a cada envio este era incrementado).

Do lado do recetor o pacote era recebido e apresentava através do monitor série presente na IDE do Arduino, num computador, a mensagem, ID da mensagem e o valor da intensidade do sinal recebido.

A comunicação foi testada em cinco pontos (Figura 54) e manteve-se estável até uma distância de 4,2 km. No último ponto (6,3 km de distância entre o emissor e o recetor) já existiram algumas perdas de pacotes. O teste foi efetuado numa zona rural e de campo aberto para evitar interferências.

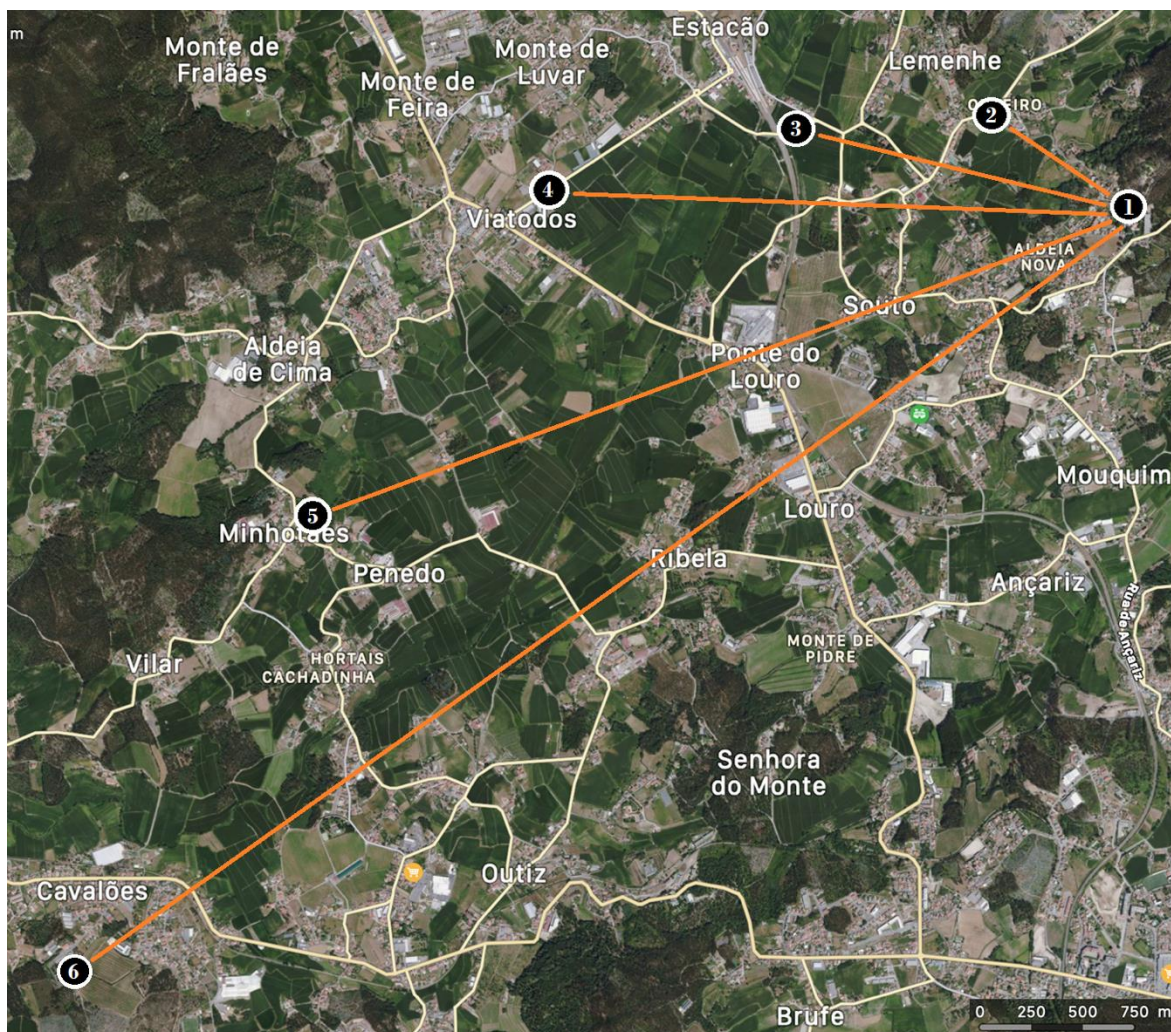


Figura 54. Mapeamento dos pontos de teste de comunicação

A Figura 54 diz respeito ao mapeamento dos pontos de testes de comunicação e abaixo estão descritas as respetivas coordenadas referentes à mesma figura:

Ponto 1: 41°26'54.4"N 8°31'20.0"W;

Ponto 2 (860 m do Ponto 1): 41°27'11.2"N 8°31'48.7"W;

Ponto 3 (1,7 km do Ponto 1): 41°27'08.9"N 8°32'29.6"W;

Ponto 4 (2,82 km do Ponto 1): 41°26'59.4"N 8°33'21.3"W;

Ponto 5 (4,2 km do Ponto 1): 41°26'08.2"N 8°34'10.8"W;

Ponto 6 (6,27 km do Ponto 1): 41°24'56.6"N 8°35'00.7"W;



A Figura 55 diz respeito ao ponto 1 da Figura 54, onde o recetor foi colocado, num ponto alto, de forma a melhorar a comunicação.



*Figura 55. Ponto 1 onde foi colocado o módulo emissor no primeiro teste aos módulos de rádio RA-02*

A Figura 56 foi captada durante a realização do teste onde o recetor se situava a uma distância de 860 m do emissor, ou seja, o recetor encontrava-se no ponto 2 da Figura 54. Aqui o recetor recebeu os pacotes enviados pelo emissor de forma correta com um RSSI médio de -108 dBm.



*Figura 56. Ponto 2 – Primeiro local de colocação do recetor para testes do módulo de rádio RA-02*

A Figura 57 foi captada durante a realização do teste onde o recetor se situava a uma distância de 1,7 km do emissor, ou seja, o recetor encontrava-se no ponto 3 da Figura 54. Aqui o recetor recebeu os pacotes enviados pelo emissor de forma correta com um RSSI médio de -111 dBm.



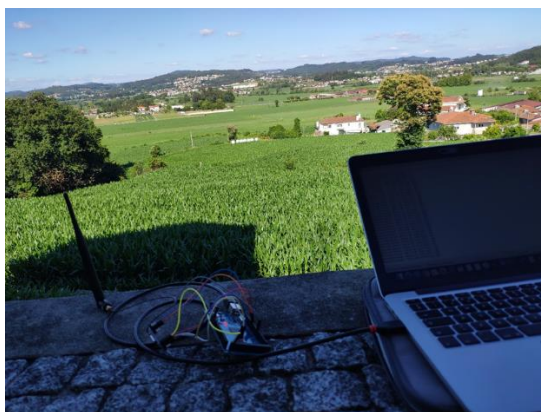
*Figura 57. Ponto 3 - Segundo local de colocação do recetor para testes do módulo de rádio RA-02*

Já a Figura 58 foi captada durante a realização do teste onde o recetor se situava a uma distância do emissor de 2,82 km, ou seja, o recetor encontrava-se no ponto 4 na Figura 54. Aqui o recetor recebeu os pacotes enviados pelo emissor de forma correta com um RSSI médio de -108 dBm.



*Figura 58. Ponto 4 - Terceiro local de colocação do recetor para testes do módulo de rádio RA-02*

A Figura 59 foi captada durante a realização do teste onde o recetor se situava a uma distância do emissor de 4,20 km, ou seja, o recetor encontrava-se no Ponto 5 da Figura 54. Aqui o recetor recebeu os pacotes enviados pelo emissor de forma correta com um RSSI médio de -110 dBm.



*Figura 59. Ponto 5 - Quarto local de colocação do recetor para testes do módulo de rádio RA-02*

A Figura 60 foi captada durante a realização do teste onde o recetor se situava a uma distância de 6,27 km do emissor, ou seja, no ponto 6 da Figura 54. A esta distância, cerca de 40% dos pacotes foram perdidos ou continham erros, o RSSI médio dos pacotes recebidos corretamente foi de -114 dBm.



*Figura 60. Ponto 6 - Quinto local de colocação do recetor para testes do módulo de rádio RA-02*

Após concluídos os testes, o módulo de rádio RA-02 demonstrou ser bastante capaz para o propósito.

No decorrer do projeto a interface gráfica, em ambos os HMI (módulos emissor e recetor), foi-se desenvolvendo, à medida que nos microcontroladores ATmega2560 também se desenvolveram os programas para os módulos emissor e recetor, referentes aos vários modos de disparo; bem como o modo de obter informações dos módulos recetores a partir do módulo emissor.

Como os módulos emissor e recetor possuem baterias para alimentar diversos componentes e dispositivos, foi necessário realizar um estudo de vários dispositivos reguladores e conversores de tensão presentes no mercado, para assim encontrar o mais

eficiente. Quanto mais eficiente for, menos energia será desperdiçada com perdas na regulação de tensão, aumentando assim a autonomia tanto do emissor como do recetor.

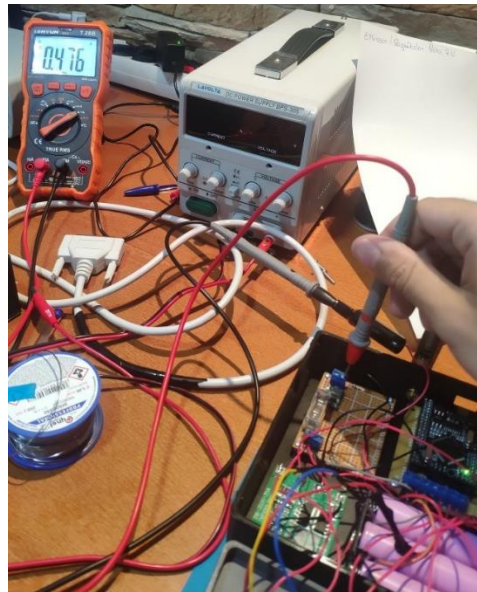


Figura 61. Consumo de corrente do emissor, com tensão de 7 V de entrada, brilho máximo e quando este se encontra no Menu Principal

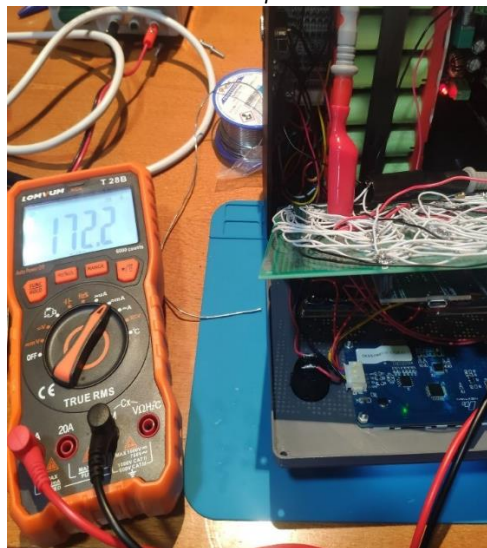


Figura 62. Consumo de corrente do recetor, com tensão de 5 V de entrada, sem as réguas de terminais aplicadas, com o brilho no máximo e quando este se encontra na página inicial

Para uma escolha mais acertada do conversor ou regulador de tensão para o emissor e recetor, foram tidos em conta, a tensão de entrada, a tensão de saída, o consumo de corrente de entrada e o consumo de corrente de saída. Com isto, os testes foram concebidos com resistências, nas saídas do regulador e conversores, de maneira a obter correntes semelhantes às que foram medidas na Figura 61 e Figura 62, ao longo da descida de tensão na entrada dos conversores ou regulador (de forma a simular a descarga das baterias). A corrente medida na Figura 61 e Figura 62, é a corrente que ambos os módulos têm na maior parte do tempo, quando estão em funcionamento. O módulo recetor possui



dois conversores: um tem a função de regular a tensão de entrada para 5 V de modo a alimentar o Arduino e outros componentes, e outro regula a tensão de entrada para 18 V para essencialmente fazer o disparo dos ignitores.

No regulador e conversores existem, durante a regulação de tensão, perdas de potência, que podem ser calculadas a partir da equação (9).

$$P_{Erda} = P_{Entrada} - P_{Saída} \quad (9)$$

$$P_{Saída} = I_{Saída} \times V_{Saída} \quad (10)$$

$$P_{Entrada} = I_{Entrada} \times V_{Entrada} \quad (11)$$

$$Eficiência\ do\ conversor/regulador = \frac{P_{Saída} \times 100}{P_{Entrada}} \quad (12)$$

Onde:

$P_{Erda}$  = Potência perdida

$P_{Entrada}$  = Potência na entrada

$P_{Saída}$  = Potência na saída

$I_{Saída}$  = Corrente na saída

$I_{Entrada}$  = Corrente na entrada

$V_{Saída}$  = Tensão na saída

$V_{Entrada}$  = Tensão na entrada

As tabelas abaixo apresentam os valores de eficiência obtidos entre os vários conversores e regulador, onde contém a potência de entrada e de saída, e como conclusão a sua eficiência. A eficiência foi calculada a partir da equação (12).

Os testes foram feitos em quatro componentes:

- LM317 (regulador linear)
- LM2576 (conversor *buck*)
- LM2596 (conversor *buck*)
- Mini-360 (conversor *buck*)

Testes com entrada entre 9 V até 12,6 V (de forma a simular um ciclo de descarga da bateria de 12,6 V do módulo emissor) e com uma saída com tensão regulada para 7 V com uma corrente de 500 mA (auxílio de resistências na saída com total de 14  $\Omega$ ).

Tabela 5. Eficiência obtida com conversor buck Mini-360, com 5 inputs de tensão de forma a simular a descarga da bateria de 12,6 V, com um output fixo de 7V e 500 mA

Entrada			Saída			Perdas	Eficiência
Tensão	Corrente	Potência	Tensão	Corrente	Potência	Potência	0-100%
12,6 V	0,29 A	3,65 W	7 V	0,5 A	3,5 W	0,15 W	95,89%
11,7 V	0,31 A	3,63 W	7 V	0,5 A	3,5 W	0,13 W	96,42%
10,8 V	0,34 A	3,67 W	7 V	0,5 A	3,5 W	0,17 W	95,36%
9,9 V	0,37 A	3,66 W	7 V	0,5 A	3,5 W	0,16 W	95,62%
9 V	0,40 A	3,6 W	7 V	0,5 A	3,5 W	0,1 W	97,22%

Tabela 6. Eficiência obtida com conversor buck LM2596, com 5 inputs de tensão de forma a simular a descarga da bateria de 12,6 V, com um output fixo de 7 V e 500 mA

Entrada			Saída			Perdas	Eficiência
Tensão	Corrente	Potência	Tensão	Corrente	Potência	Potência	0-100%
12,6 V	0,32 A	4,03 W	7 V	0,5 A	3,5 W	0,53 W	86,85%
11,7 V	0,35 A	4,10 W	7 V	0,5 A	3,5 W	0,60 W	85,36%
10,8 V	0,37 A	4 W	7 V	0,5 A	3,5 W	0,5 W	87,5%
9,9 V	0,40 A	3,96 W	7 V	0,5 A	3,5 W	0,46 W	88,38%
9 V	0,44 A	3,96 W	7 V	0,5 A	3,5 W	0,46 W	88,38%

Tabela 7. Eficiência obtida com conversor buck LM2576, com 5 inputs de tensão de forma a simular a descarga da bateria de 12,6 V, com um output fixo de 7 V e 500 mA

Entrada			Saída			Perdas	Eficiência
Tensão	Corrente	Potência	Tensão	Corrente	Potência	Potência	0-100%
12,6 V	0,34 A	4,28 W	7 V	0,5 A	3,5 W	0,53 W	81,78%
11,7 V	0,36 A	4,21 W	7 V	0,5 A	3,5 W	0,6 W	83,13%
10,8 V	0,4 A	4,32 W	7 V	0,5 A	3,5 W	0,5 W	81,02%
9,9 V	0,43 A	4,25 W	7 V	0,5 A	3,5 W	0,46 W	82,35%
9 V	0,48 A	4,32 W	7 V	0,5 A	3,5 W	0,46 W	81,02%

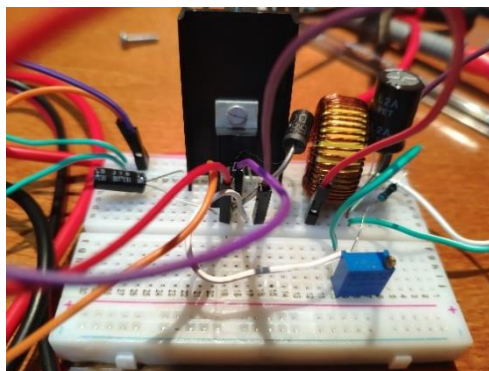


Figura 63. Circuito com o conversor buck LM2576, com output de 7 V regulado pelo potenciômetro (o circuito foi elaborado de acordo com o datasheet do fabricante)

Tabela 8. Eficiência obtida com conversor linear LM317, com 5 inputs de tensão de forma a simular a descarga da bateria de 12,6 V, com um output fixo de 7 V e 500 mA

Entrada			Saída			Perdas	Eficiência
Tensão	Corrente	Potência	Tensão	Corrente	Potência	Potência	0-100%
12,6 V	0,5 A	6,3 W	7 V	0,5 A	3,5 W	2,8 W	55,56%
11,7 V	0,5 A	5,85 W	7 V	0,5 A	3,5 W	2,35 W	59,83%
10,8 V	0,5 A	5,4 W	7 V	0,5 A	3,5 W	1,9 W	64,81%
9,9 V	0,5 A	4,95 W	7 V	0,5 A	3,5 W	1,45 W	70,71%
9 V	0,5 A	4,5 W	7 V	0,5 A	3,5 W	1 W	77,78%

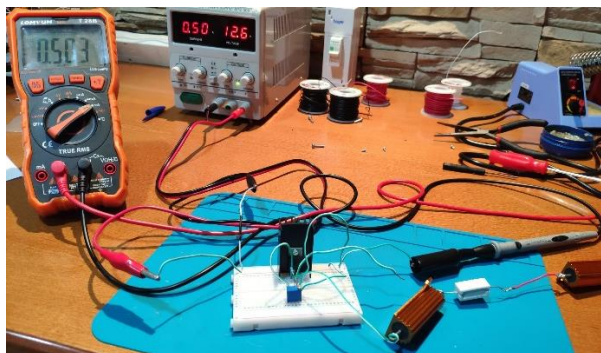


Figura 64. Teste com conversor linear LM117 regulado para uma tensão de saída de 7 V (circuito elaborado de acordo com datasheet do fabricante)

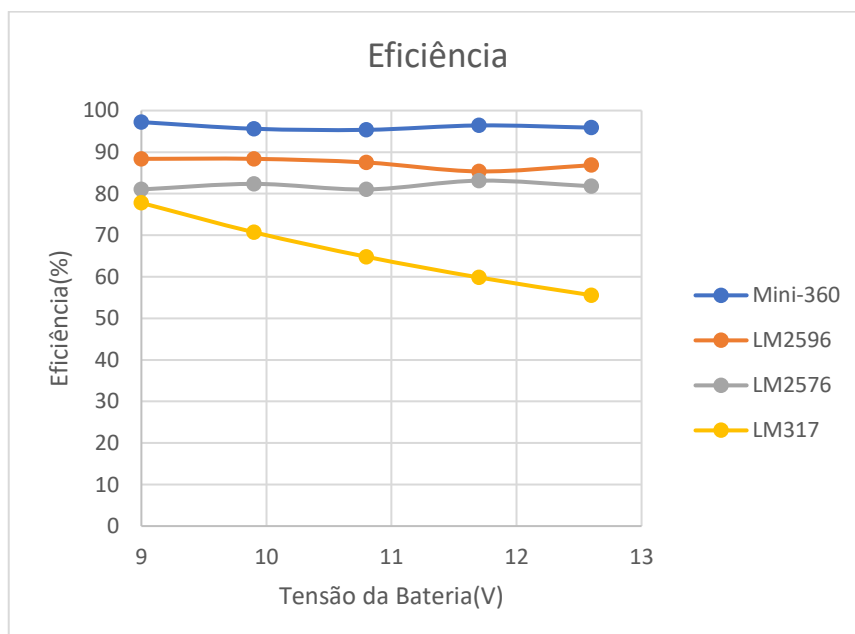


Figura 65. Comparação entre os quatro conversores de tensão (Mini-360, LM2596, LM2576 E LM317), onde apresenta a eficiência de cada um em função da tensão (9 V até 12,6 V). (Testes realizados em cada um com uma saída fixa de 7 V e 500 mA)

Testes com entrada entre 18 V até 25,2 V (de forma a simular a descarga da bateria de 25,2 V do módulo recetor) e uma saída com tensão regulada para 5 V e 166 mA (auxílio de resistências com um total de 30,1  $\Omega$ )

Tabela 9. Eficiência obtida com conversor buck Mini-360, com 5 inputs de tensão de forma a simular a descarga da bateria de 25,2 V, com um output fixo de 5 V e 166 mA

Entrada			Saída			Perdas	Eficiência
Tensão	Corrente	Potência	Tensão	Corrente	Potência	Potência	0-100%
25,2 V	52 mA	1,31 W	5 V	166 mA	0,83 W	0,48 W	63,35%
23,4 V	54 mA	1,26 W	5 V	166 mA	0,83 W	0,43 W	65,87%
21,6 V	57 mA	1,23 W	5 V	166 mA	0,83 W	0,4 W	67,47%
19,8 V	61 mA	1,20 W	5 V	166 mA	0,83 W	0,37 W	69,1%
18 V	70 mA	1,26 W	5 V	166 mA	0,83 W	0,43 W	65,87%

Tabela 10. Eficiência obtida com conversor buck LM2596, com 5 inputs de tensão de forma a simular a descarga da bateria de 25,2 V, com um output fixo de 5 V e 166 mA

Entrada			Saída			Perdas	Eficiência
Tensão	Corrente	Potência	Tensão	Corrente	Potência	Potência	0-100%
25,2 V	46 mA	1,16 W	5 V	166 mA	0,83 W	0,33 W	71,56%
23,4 V	50 mA	1,17 W	5 V	166 mA	0,83 W	0,34 W	70,94%
21,6 V	53 mA	1,14 W	5 V	166 mA	0,83 W	0,31 W	72,80%
19,8 V	58 mA	1,15 W	5 V	166 mA	0,83 W	0,32 W	77,40%
18 V	62 mA	1,12 W	5 V	166 mA	0,83 W	0,29 W	74,44%

Tabela 11. Eficiência obtida com conversor buck LM2576, com 5 inputs de tensão de forma a simular a descarga da bateria de 25,2 V, com um output fixo de 5 V e 166 mA

Entrada			Saída			Perdas	Eficiência
Tensão	Corrente	Potência	Tensão	Corrente	Potência	Potência	0-100%
25,2 V	53 mA	1,34 W	5 V	166 mA	0,83 W	0,51 W	61,94%
23,4 V	56 mA	1,31 W	5 V	166 mA	0,83 W	0,48 W	63,36%
21,6 V	60 mA	1,30 W	5 V	166 mA	0,83 W	0,47 W	63,85%
19,8 V	65 mA	1,29 W	5 V	166 mA	0,83 W	0,46 W	64,34%
18 V	71 mA	1,29 W	5 V	166 mA	0,83 W	0,46 W	64,34%

Tabela 12. Eficiência obtida com conversor buck LM317, com 5 inputs de tensão de forma a simular a descarga da bateria de 25,2 V, com um output fixo de 5 V e 166 mA

Entrada			Saída			Perdas	Eficiência
Tensão	Corrente	Potência	Tensão	Corrente	Potência	Potência	0-100%
25,2 V	166 mA	4,18 W	5 V	166 mA	0,83 W	3,35 W	19,86%
23,4 V	166 mA	3,88 W	5 V	166 mA	0,83 W	3,05 W	21,39%
21,6 V	166 mA	3,58 W	5 V	166 mA	0,83 W	2,75 W	23,18%
19,8 V	166 mA	3,29 W	5 V	166 mA	0,83 W	2,46 W	25,23%
18 V	166 mA	2,99 W	5 V	166 mA	0,83 W	2,16 W	27,76%

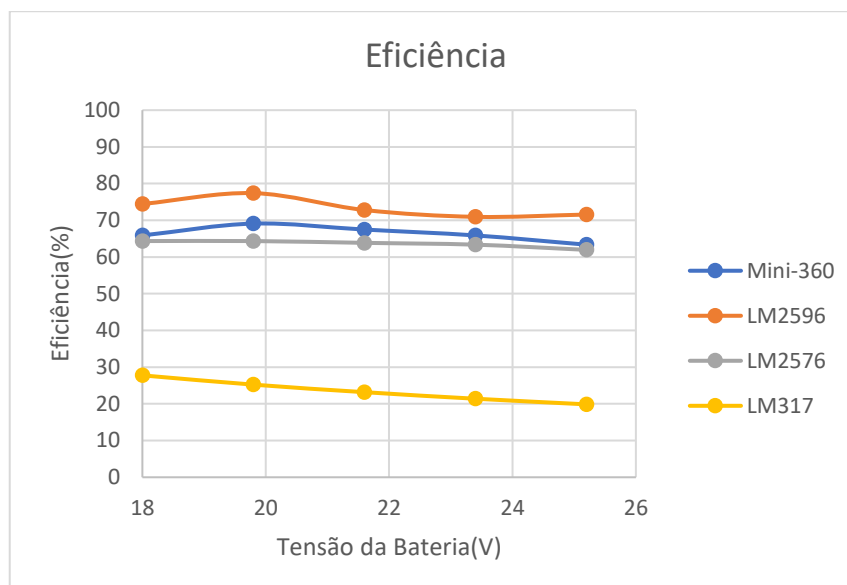


Figura 66. Comparação entre os quatro conversores de tensão (Mini-360, LM2596, LM2576 E LM317), onde apresenta a eficiência de cada um em função da tensão (18 V até 25.2 V). (Testes realizados em cada um com uma saída fixa de 5 V e 166 mA)

Como se pode ver pelos resultados obtidos, no caso do emissor, o conversor *buck* Mini-360 obteve a melhor eficiência, com resultados superiores a 95%. No caso do recetor, o conversor *buck* LM2596 obteve a melhor eficiência, com valores superiores a 70%.

Em seguida, foi desenhado no *software* Eagle (Figura 67), a nova placa da régua de terminais. Como o circuito elétrico projetado possuía mais de 150 componentes, foi decidido desenhar e realizar a placa em circuito impresso, de forma a reduzir o tamanho da régua, tornando-a mais leve e compacta. Depois do desenho concluído verificou-se que a placa teria de possuir pelo menos duas camadas, devido ao elevado número de ligações entre componentes, e isso dificultaria bastante o modo de produção e impediria uma futura modificação, caso fosse necessária. Assim, optou-se por se desenvolverem placas eletrónicas em placas circuito perfuradas.

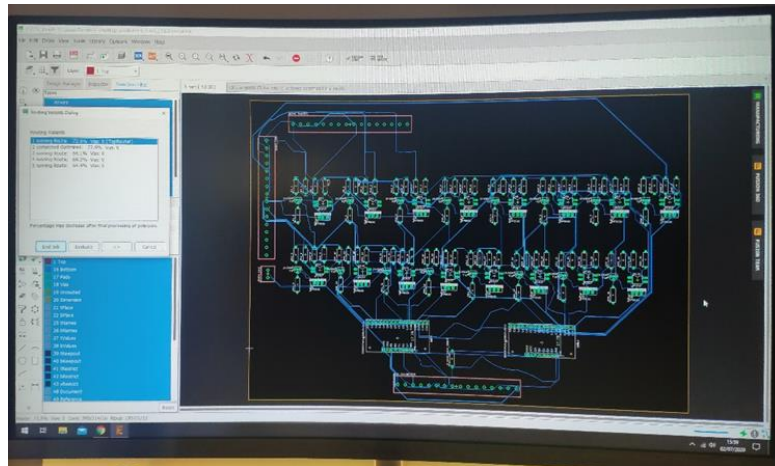


Figura 67. Elaboração da placa eletrônica no software Eagle

Durante a elaboração da placa testou-se a melhor posição de cada componente, de maneira a aproveitar melhor o espaço. Depois foi suficiente seguir o projeto anteriormente realizado até finalizar a placa. Após isto foram realizados vários testes, de forma a verificar a existência de algum erro ou curto-circuito.

Em seguida foi modelado e elaborado, com recurso à impressão 3D, o chassi da régua de terminais (Figura 68).

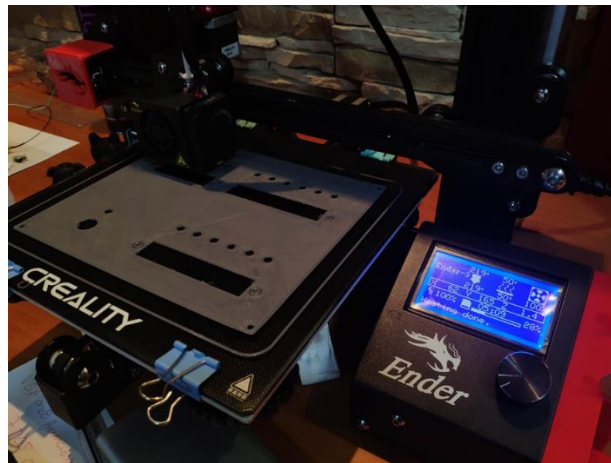
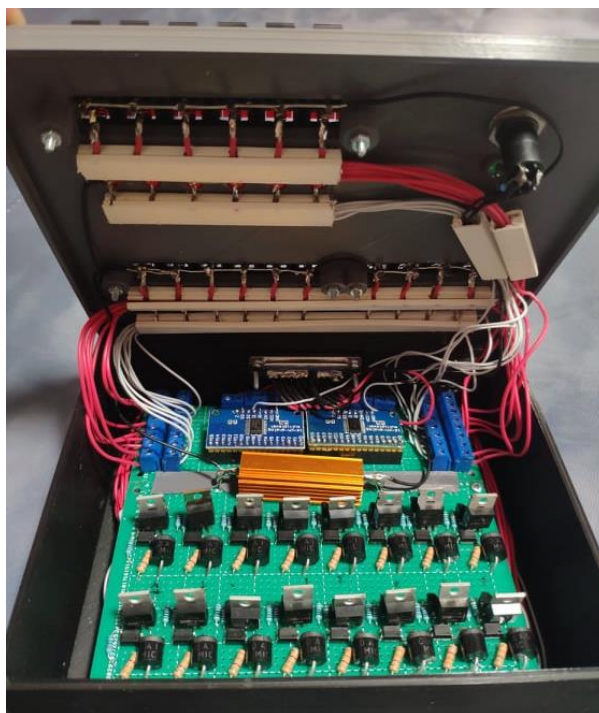


Figura 68. Impressão do painel da régua de terminais

Após a impressão, foi montado o circuito com todas as ligações no *chassi* da régua (Figura 69 e Figura 70).



*Figura 69. Montagem final da régua de terminais*



*Figura 70. Régua de terminais concluída*

Depois foi elaborada a placa eletrónica para o recetor. Neste caso é mais simples, e como tal não se elaborou a placa em circuito impresso. De forma a tornar a placa o mais compacta possível foi estudada a melhor posição de todos os componentes. Após o recetor estar pronto (Figura 71, Figura 72 e Figura 73), este foi testado e verificaram-se todos os componentes de forma a verificar se o seu comportamento estava normal.





Figura 71. Montagem da bateria com a BMS no chassi do recetor e o conversor buck XHM401

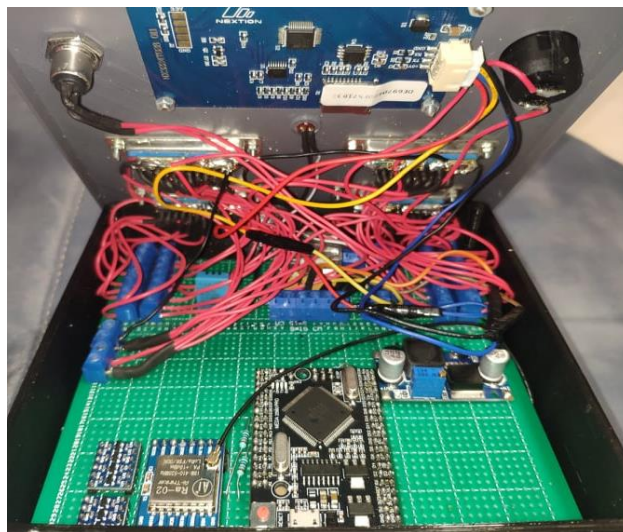


Figura 72. Montagem de todos os componentes do recetor



Figura 73. Módulo recetor finalizado



Isto, isto foi desenvolvida a placa eletrônica para o emissor (Figura 74). Esta é, em parte, semelhante à placa dos recetores, com a principal diferença de em vez de conter os terminais de saída DB25, possuir a ligação para o módulo de cartões SD. Além disto foi elaborada uma bateria com três células em série Samsung IRN18650-35E. De forma a proteger as células e aumentar a sua vida útil, foi adicionada uma BMS 3s à bateria.

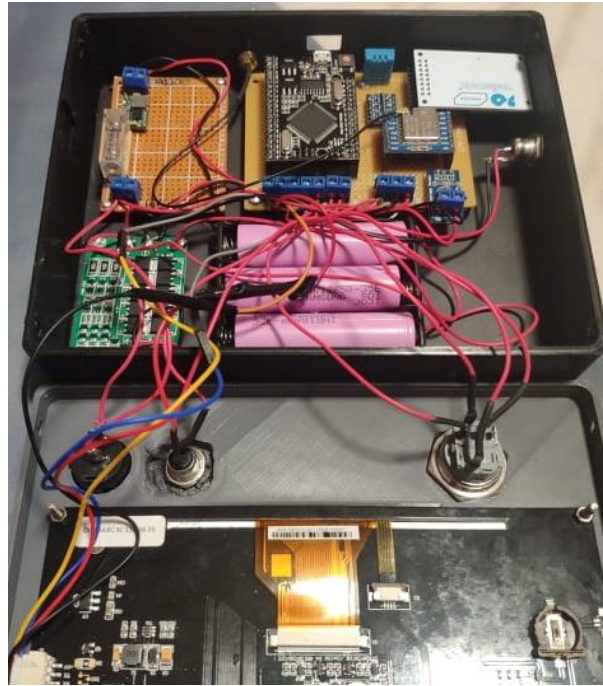


Figura 74. Elaboração e montagem da placa eletrônica do emissor

Após a placa e todos os componentes estarem inseridos no chassi do emissor (Figura 75), este foi testado, de forma a verificar alguma possível falha.



Figura 75. Módulo emissor finalizado

Após a elaboração do conjunto emissor, recetor e régua foi possível testar o sistema completo de forma a verificar possíveis falhas.

Na primeira régua de terminais elaborada, entre os 18 V C/P (18 V controlados por uma chave no módulo recetor) e a *source* de cada MOSFET existe um díodo de 10 A (10A10) de forma resolver um problema que surgiu inicialmente. Este problema surgia caso não existisse um ignitor ligado em alguns terminais da régua, caso a chave do recetor estivesse na posição “1” (não fornecendo 18 V à *source* de cada MOSFET da régua) e caso fosse dada ordem de disparo de um terminal que contivesse um ignitor. Nestas condições, nos acopladores óticos (dos terminais da régua) que não continham ignitor, a corrente que passava por eles, fluía para o terminal (que continha ignitor) onde era dada a ordem de disparo, podendo assim trazer problemas de fiabilidade e segurança. A solução passou pela colocação de um díodo entre os 18 V C/P e a *source* de cada MOSFET, o que impede a circulação de uma pequena corrente entre o dreno e a *source* dos MOSFETs, nos terminais que não contêm ignitor, nas situações em que é dada a ordem de disparo num terminal que possua ignitor.

No desenvolvimento do *software* do recetor, verificou-se que o caso descrito anteriormente, poderia ser solucionado através de *software*, não sendo então necessária a utilização dos díodos em todas as *sources* dos MOSFET's.

A solução de *software* passou por monitorizar a posição da chave do recetor através de um transístor bipolar NPN, e caso a chave esteja na posição zero (posição segura), não fornecendo os 18 V C/P à *source* de cada MOSFET, não executa a ordem de disparo.

Como tal, na seguinte régua desenvolvida foi aplicado apenas um díodo na entrada dos 18 V C/P da régua, pois assim é evitado o retorno da corrente para o transístor bipolar NPN do recetor, quando a chave física está na posição zero (aberta).

Em relação às células utilizadas no módulo recetor, foi descoberto um erro cometido pela loja que as comercializa, pois a loja indicava que as mesmas suportavam correntes de descarga continuas até 8 A. Este erro foi detetado quando se realizou uma pesquisa com o objetivo de compreender o ciclo da descarga. Nesta pesquisa descobriu-se que as mesmas suportavam no máximo uma corrente de descarga de 6,7 A. Com isto, seria arriscado estar a utilizar estas células, uma vez que podiam surgir problemas no futuro, como a redução da vida útil das mesmas. A loja foi alertada do erro e as células Panasonic NCR18650B

utilizadas inicialmente no recetor desenvolvido foram substituídas pelas Samsung 18650-35E. Estas últimas, segundo o *datasheet* do fabricante, suportam até 8 A de corrente contínua de descarga e 13 A de corrente de pico, o que é suficiente.

#### 4.2. Testes

Durante a elaboração desta dissertação foram desenvolvidos um emissor, um recetor e duas réguas, de modo a ser possível testar e comprovar o funcionamento do sistema projetado.

Após os módulos estarem concluídos, foi realizado um conjunto de testes relativos à fiabilidade da comunicação. Foi analisada a taxa de “disparos bem-sucedidos” e os valores de RSSI e SNR para várias distâncias entre o emissor e o recetor. Para tal foi simulado um espetáculo de 64 números no modo automático, com tempos entre disparos de 0,2 s.

Pretendeu-se com este teste verificar e analisar a relação dos dados RSSI e SNR com a taxa de sucesso de disparos bem-sucedidos. O teste foi efetuado três vezes em cada ponto, para uma melhor fidedignidade dos dados recolhidos.

Foi efetuado ainda um teste em cada ponto, sobre o estado da comunicação do recetor para o emissor. Ou seja, foi executado no emissor o pedido de dados básicos do recetor e de dados mais detalhados. Foram analisados os valores de intensidade de sinal recebido (RSSI) e relação sinal-ruído (SNR) três vezes em cada ponto, e apresentada a média. Este teste foi feito pois os tamanhos dos pacotes do recetor para o emissor, nos dois modos de pedido de informações, possuem tamanhos diferentes.

Os testes de zero a 500 m foram realizados na estrada com as coordenadas 41°16'50.6"N 8°34'12.0"W. As localizações referentes aos pontos entre 860 m a 6270 m são exatamente as mesmas da Figura 54 do subcapítulo anterior, sendo que o recetor ficou em ambos os testes no ponto 1. Os restantes pontos 2,3,4,5 e 6 foram os pontos percorridos pelo emissor durante os testes. Ambos os módulos de rádio (LoRa) continham a seguinte configuração: Frequência = 433 MHz, Potência = 19 dBm, BW = 125 kHz, SP = 7 e CR = 4/5.

Tabela 13. Dados relativos à comunicação entre o recetor e o emissor

<b>Distância entre o recetor e o emissor</b>	<b>RSSI (dBm médio) Dados básicos</b>	<b>SNR (médio) Dados básicos</b>	<b>Recebeu os 3 Pacotes?</b>	<b>RSSI (dBm médio) Dados Detalhados</b>	<b>SNR (médio) Dados Detalhados</b>	<b>Recebeu os 3 Pacotes?</b>
<b>0 m</b>	-13	9	SIM	-22	9,75	SIM
<b>10 m</b>	-69	9,5	SIM	-59	9,5	SIM
<b>30 m</b>	-72	9,5	SIM	-64	9,5	SIM
<b>50 m</b>	-75	9	SIM	-66	10,5	SIM
<b>100 m</b>	-87	7,5	SIM	-81	8	SIM
<b>200 m</b>	-89	6,5	SIM	-82	7,25	SIM
<b>500 m</b>	-89	7	SIM	-82	7,5	SIM
<b>860 m</b>	-91	4	SIM	-84	4	SIM
<b>1700 m</b>	-97	-8,5	SIM	-90	-9,5	SIM
<b>2820 m</b>	-97	-7	SIM	-89	-7	SIM
<b>4200 m</b>	-98	-9,25	APENAS 1	-90	-9,75	APENAS 2
<b>6270 m</b>	-	-	NÃO	-	-	NÃO

Os testes realizados de aquisição dos dados do recetor (do recetor para o emissor) em ambos os testes (pedidos de dados básicos e pedidos de dados detalhados) demonstraram ser competentes até uma distância de 2820 m. No ponto seguinte (4200 m) só um dos pedidos enviados pelo emissor, chegou de volta. Os valores de intensidade de sinal (RSSI) e relação sinal-ruído (SNR) foram diminuindo, de forma pouco relevante, ao longo da distância. Estes resultados podem estar relacionados com a própria topografia do terreno, interferências ou problemas na antena do recetor (sendo que este recebia o pedido do emissor e enviava um pacote com a resposta, contudo o emissor não o conseguia receber). O ideal seria testar os módulos nas posições inversas e verificar se os problemas de comunicação se mantinham.

Outro dado analisado no teste, foi a verificar se o tamanho do pacote interfere na intensidade de sinal recebido. Pelos dados adquiridos o pacote com os dados detalhados,

que possui 70 bytes, apresenta sempre um RSSI mais elevado comparado ao pacote de dados básicos que possui apenas 4 bytes.

Tabela 14. Dados relativos à comunicação entre o emissor e o recetor

Distância entre o emissor e o recetor	RSSI (dBm médio)	SNR (médio)	Disparos bem-sucedidos (%)
0 m	-14	10	100%
10 m	-54	10	100%
30 m	-58	10	100%
50 m	-63	9,75	100%
100 m	-76	10	100%
200 m	-78	9,75	100%
500 m	-78	9,5	100%
860 m	-83	5,5	100%
1700 m	-99	-4	100%
2820 m	-95	-5,25	100%
4200 m	-99	-6	100%
6270 m	-103	-7	100%

Os testes de disparo realizados (emissor para recetor) obtiveram resultados bastante positivos, uma vez que todas as simulações de disparo foram bem-sucedidas a qualquer distância. Como era de esperar, o valor da intensidade de sinal recebido (RSSI) bem como a relação sinal-ruído (SNR) foi diminuindo com a distância. Infelizmente não foi possível detetar qual a distância máxima na qual o sistema no modo de disparo começa a falhar, de modo a perceber qual será o valor limite de RSSI e SNR. Contudo, a distância obtida é excelente, uma vez que normalmente este tipo de sistemas de disparo não são usados a tão grandes distâncias. Isto demonstra a boa fiabilidade da comunicação do sistema a distâncias mais curtas.

Além do teste de comunicação sem fios, realizou-se um teste de continuidade com alguns ignitores ligados nos terminais das réguas, onde o sistema demonstrou ser bastante preciso e prático. No que toca aos disparos de ignitores, vários foram testados, nos diferentes modos de disparo, e demonstrou ser bastante consistente e com “*delay*” impercetível (desde a ordem de disparo do emissor até à ignição do mesmo). A corrente máxima foi

analisada com um amperímetro num terminal da régua durante o disparo. O amperímetro apresentou valores de aproximadamente 7,3 A. Neste teste o amperímetro fechava a saída (curto-circuito) para simular uma resistência de zero na saída, uma vez que se forem colocados vários ignitores em paralelo numa única saída a resistência (dos condutores em paralelo) diminui, podendo ir para valores próximos de zero. O teste foi realizado com um cabo de 2 m DB25 (Nedis CCGP52100IV20).

Mais tarde foi testado com um cabo de 5m (Nedis CCGP52100IV50), contudo a corrente máxima baixou, uma vez que o material condutor é *Copper-clad aluminium wire* (CCA) em vez de cobre e com um maior comprimento. A resistência de potência ( $1 \Omega$ ) utilizada na régua, foi inicialmente escolhida tendo em conta a resistência no cabo de 2 m utilizado, uma vez que a resistência dos condutores em paralelo que transportavam a corrente era de aproximadamente  $0,75 \Omega$ . No cabo de 5 m a resistência já sobe para  $3,4 \Omega$  aproximadamente. A resistência de cada condutor do cabo de 2 m é muito alta, no entanto, tal como o fabricante indica no *datasheet* este é de cobre.

O ideal seria adquirir no futuro diferentes bobines de cabo de 25 condutores e, de acordo com o comprimento do cabo, aumentar ou diminuir a secção de cada condutor. Com esta alteração já é possível utilizar vários comprimentos de cabo (tornando o sistema mais flexível) de forma a manter a corrente de disparo constante, independentemente do comprimento. Esta solução foi pensada inicialmente, contudo iria aumentar muito o custo do projeto em questão, uma vez que o custo por metro é bastante alto. Por isso optou-se por utilizar cabos genéricos DB25 encontrados atualmente no mercado.

## 5. Descrição do Sistema de Ignição Pirotécnica

Neste capítulo são abordados os aspetos informáticos e elétricos detalhados do projeto final desenvolvido.

O sistema de ignição pirotécnica (Figura 76) é composto por um emissor e até 256 recetores. Cada recetor suporta até quatro régua de terminais (onde se ligam os ignitores). Cada régua possui dezasseis terminais (dezasseis números independentes), sendo que a primeira régua diz respeito aos primeiros dezasseis números, e a última régua aos últimos dezasseis números. A ligação entre a régua de terminais e recetor é feita através de um cabo DB25 (25 condutores).

A comunicação entre o emissor e o(s) recetor(es) é feita sem fios através dos módulos RA-02 433MHz LoRa. A comunicação é bidirecional, sendo que para além do disparo dos ignitores (comunicação feita no sentido do emissor para o(s) recetor(es)) é possível saber as informações básicas ou detalhadas de cada recetor através do emissor. Cada recetor possui um ID próprio (ID do recetor) configurável através do menu do HMI do recetor; contudo, é necessário que o ID não seja repetido no espetáculo. Cada recetor só responde a pacotes do emissor caso o ID do emissor configurado no recetor seja igual ao ID local do emissor que enviou o pacote. O sistema trabalha com até 256 canais no modo manual e no semiautomático e até catorze no modo automático. Os canais servem para aumentar os números independentes por espetáculo. Por exemplo, caso um espetáculo necessite de 80 números independentes de disparo, basta configurar dois recetores com canais diferentes, sendo que o primeiro recetor dispara os primeiros 64 números e o segundo os restantes. Para além disto, o recetor possui dois modos de trabalho distintos: o modo de teste e o modo de fogo. No modo de teste (chave física na posição 0) o recetor só responde e executa pedidos de informações vindos por parte do emissor, sendo que os pedidos de disparo são ignorados. No modo de fogo (chave física na posição 1) o recetor apenas executa pedidos de disparo.

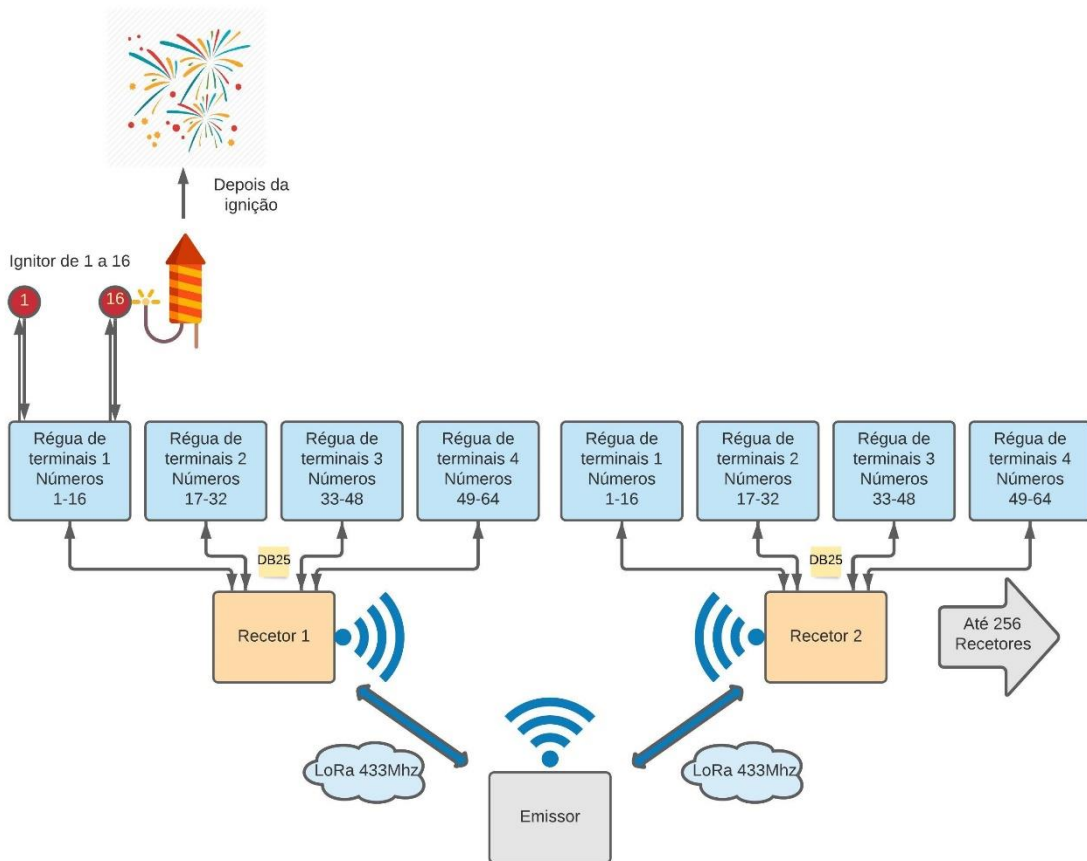


Figura 76. Esquema geral do projeto desenvolvido

## 5.1. Software

Neste subcapítulo são abordados os aspetos relativos ao funcionamento do sistema de ignição pirotécnica desenvolvido, onde são descritos o funcionamento e a descrição de cada página do HMI do módulo emissor, bem como do recetor.

### 5.1.1. Encriptação

Foi introduzida a encriptação *Advanced Encryption Standard* (AES) na comunicação do emissor com recetores, de modo a tornar mais segura a comunicação do emissor-recetor. Com isto é mais difícil que “desconhecidos tentem estragar” o espetáculo pirotécnico. A mensagem é encriptada por uma chave de 128 bits. No projeto foi utilizada uma biblioteca AESLib.h.



### 5.1.2. Descrição do funcionamento do Emissor

O HMI do emissor é composto por onze páginas. As páginas são compostas por vários componentes (botões, imagens, texto, etc.). Estes componentes são criados como objetos no programa do microcontrolador ATmega 2560, sendo que depois têm ao seu dispor um conjunto de funções que a biblioteca Nextion.h oferece de forma intuitiva. No decorrer do programa no ATmega2560, são feitos *gets* e *sets* de vários objetos e também rotinas de monitorização. Depois de declarados todos os componentes do HMI, são adicionados a uma lista. Depois, a cada objeto são associados um evento e uma função (por exemplo, evento no clique de um botão ou na libertação de um botão). Caso o evento associado ao botão ocorra, a função associada é chamada. Por exemplo, caso haja um clique no botão “Y” será chamada a função “X” associada a esse botão, sendo que a função é responsável pela troca de página e por atualizar os valores dos componentes numéricos “X1” e “X2”. Abaixo é apresentada uma descrição resumida de cada página, a sua função e algumas características, de forma a mostrar um panorama geral do funcionamento do emissor:

**Página 0-** Página de apresentação com o nome da aplicação (Figura 77). Esta é a primeira página apresentada, quando o emissor é ligado. Depois de pressionar o ecrã esta passa para a página 1 Menu Principal onde é apresentada a tensão da bateria num componente numérico (no futuro será atualizado para percentagem de bateria).



Figura 77. Página 0 - Apresentação do emissor com o nome da aplicação

**Página 1-** Menu Principal (Figura 78), possui 5 botões.

O primeiro botão “*Modo Automático*” dá acesso à página 2 onde é feita a inserção e modificação de tempos por cada canal + número do modo automático. Esta página só é acessada quando o cartão SD está inserido. Caso contrário, encaminha para a página 10 (página de apresentação de avisos e erros) onde apresenta o erro “Cartão SD não está inserido. Insira para poder aceder às várias funções”. No momento antes de apresentar a página 2, o ATmega2560 faz uma leitura de dados, de dois ficheiros de extensão .txt, presentes no cartão SD. Estes ficheiros contêm os tempos relativos ao conjunto canal + número. Um dos ficheiros armazena os segundos e o outro as décimas de segundo. Durante a leitura dos dados dos ficheiros do cartão SD é feita uma cópia desses valores para dois vetores bidimensionais (matrizes). Depois é apresentado, na página 2, por predefinição, o canal + número (0+1) e o tempo relativo desse conjunto (esse tempo provém das matrizes que armazenam o tempo).

O segundo botão “*Modo Manual*” dá acesso à página 9 “*Modo Manual*”. Antes do acesso ser feito é iniciado o módulo de rádio e é verificado se a chave física do painel do emissor (ligada a uma entrada digital do ATmega2560) está na posição 1 (nível *high*). Caso contrário, passa para a página 10 onde é apresentada uma mensagem de erro “Para poder aceder ao disparo vai ter de rodar a chave para o estado 1” ou “Falha no módulo de rádio”, caso exista um problema ao iniciar o módulo. Depois de verificadas estas condições, é apresentada a página 11 onde existe um *slider* de proteção. Quando o quadrado do componente *slider* (*h0*) estiver a 100% (totalmente no lado direito), a página 9 (Modo Manual) é apresentada.

O terceiro botão “*Semiautomático*” dá acesso à página 4 “*Semiautomático*”. Todas as verificações são iguais às do botão anterior, contudo, depois do *slider* da página 11 estar a 100%, é apresentada a página 4 “*Semiautomático*”.

O quarto botão “*Info Recetores*” dá acesso à página 8 “*Info Recetores*”.

O quinto botão “*Configurações*” dá acesso à página 5 “*Configurações*”. Quando a página 5 é apresentada são atualizados os valores do componente ID do emissor (*n0*) e da luminosidade (*h0*), com os dados armazenados no próprio ATmega2560. Estes dados estão armazenados na EEPROM e são acessados sempre que o emissor é ligado (programa arranca), armazenando esses valores em duas variáveis, de forma a serem utilizados durante o programa.

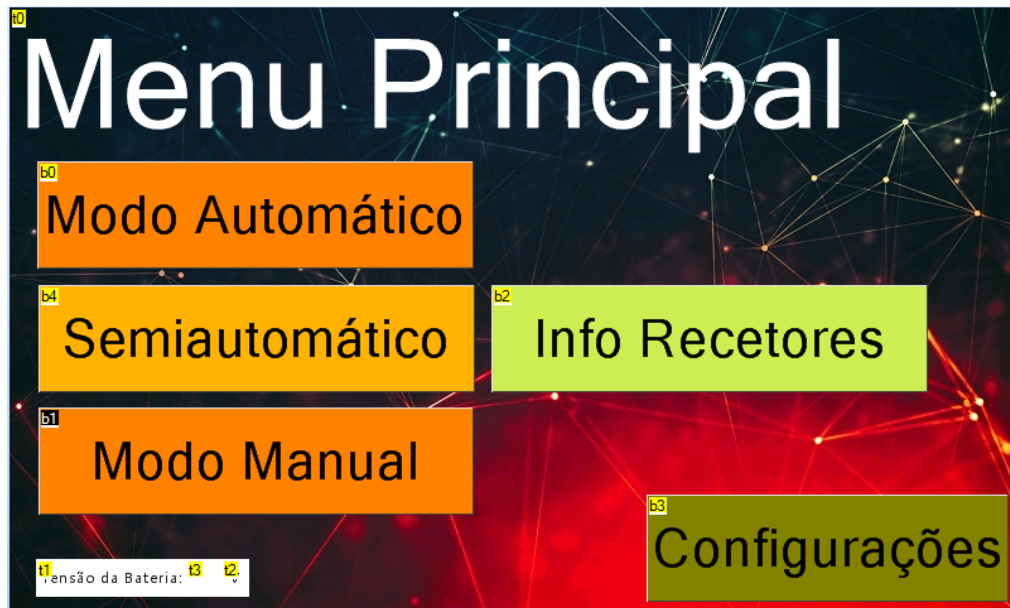


Figura 78. Página1 - Menu Principal

**Página 2-** Esta página (Figura 79) é usada para inserção de tempos por cada conjunto de canal + número. É constituída por dois componentes,  $n0$  e  $n1$ , onde se apresenta o valor do canal ( $n0$ ), que varia entre zero e catorze, e o número ( $n1$ ) que varia entre 1 e 64. O tempo é apresentado em dois componentes numéricos. O  $n2$  apresenta os segundos (zero a 9999) e o componente  $n3$  os décimos de segundo (zero, dois, quatro, seis e oito). Os dados de tempo apresentados provêm de dois vetores bidimensionais (matrizes) que armazenam os segundos e as décimas de segundo, por canal + número. Os botões  $b15$  “+100”,  $b10$  “+10”,  $b5$  “+1” e  $b8$  “+0.2” incrementam o tempo dos componentes numéricos correspondentes ao tempo  $n2$  e  $n3$  com uma resolução mínima de 0,2 s e máxima de 100 s. Estas modificações dos valores numéricos foram programadas na IDE do HMI, ou seja, funcionam de forma independente do microcontrolador ATmega2560.

O mesmo acontece com os botões  $b14$  “-100”,  $b11$  “-10”,  $b6$  “-1” e  $b9$  “-0.2”, sendo que estes decrementam o tempo. Os botões  $b1$  “+”,  $b3$  “+”,  $b2$  “-” e  $b4$  “-” servem para percorrer os vários canais e números. Estes estão sempre a ser monitorizados pelo ATmega2560 e, caso um seja pressionado, é verificado se o tempo presente nos componentes ( $n2$  e  $n3$ ) da página é igual ao das matrizes de tempo. Caso não seja, o tempo é substituído pelo presente nos campos  $n2$  e  $n3$ .

Para além disto, a seguir às verificações anteriormente relatadas, o canal ou o número é incrementado ou decrementado, consoante o botão pressionado. Depois os componentes numéricos da página ( $n2$  e  $n3$ ) são atualizados de acordo com o tempo presente nos índices

das matrizes de tempo. Exemplo: Caso  $n0=2$  e  $n1=1$ ,  $n2=MatrizSegundos[0][1]$  e  $n3=MatrizDecimos[0][1]$ .

Por predefinição, o canal + número visualizado quando esta página é apresentada é o canal zero e o número um (conjunto canal + número mais baixos). Os valores de tempo correspondentes também são apresentados. Como já foi referido, estas matrizes são preenchidas com valores provenientes de dois ficheiros txt presentes num cartão SD. Sempre que esta página é acedida a partir de outras páginas, a matriz atualiza com o valor dos ficheiros txt de tempos do cartão SD. Caso o cartão SD não esteja inserido é apresentada a página 10, indicando ao utilizador uma mensagem de erro.

O botão "*Menu Principal*" serve para voltar à página 1 "*Menu Principal*", onde o valor do componente numérico que apresenta a tensão da bateria é atualizado.

O botão "*Gravar SD*" está presente para gravar, nos ficheiros txt do cartão SD, os tempos presentes nas matrizes bidimensionais do ATmega2560. A gravação só é possível caso a chave física do emissor esteja na posição 1. Caso contrário, é apresentada a página 10 com um aviso de restrição de acesso.

O botão "*Reset Tempos*", tem como função eliminar os tempos das matrizes bidimensionais de tempo do ATmega2560. Esta função serve para eliminar os dados de tempo por cada conjunto (canal + número) relativos a um espetáculo anterior.

O botão  $b12$ , "*Disparo*", ao ser pressionado é verificado se a chave do painel está na posição 1, caso não esteja apresenta uma mensagem de erro na página 10. Caso a chave esteja na posição 1, é iniciado o cartão SD. Se existir uma falha ou o cartão SD não esteja inserido é apresentada uma mensagem de erro na página 10. Caso o cartão SD esteja inserido, os valores dos ficheiros txt de tempo serão armazenados nas matrizes de tempo. O módulo de rádio LoRa é assim iniciado de forma a verificar se está operacional e, se existir uma falha, é apresentada uma mensagem de erro na página 10. Caso as condições anteriores sejam concluídas com sucesso, é apresentada a página 11, que contém um slider, servindo como proteção de um clique accidental. Quando o slider ( $h1$ ) atingir o valor 100% é verificado nas matrizes de tempo o primeiro e o segundo canal+ número (caso existam) em que o tempo seja diferente de zero. Estes valores são guardados numa matriz de forma a serem utilizados a seguir.

Caso existam pelo menos dois conjuntos, canal + número com o tempo diferente de zero, a página 3 é apresentada. Caso contrário, é apresentada uma mensagem de erro na página 10.

Quando a página 3 é acedida é apresentado o primeiro conjunto canal + número (com índices mais baixos e tempo diferente de zero) e o respetivo tempo associado. É também apresentado, o segundo canal + número (tempos diferentes de zero). Depois disto o LED do botão físico “FIRE” de disparo começa a piscar. Os botões de pausa e avançar encontram-se invisíveis por predefinição.

Os tempos inseridos servem para “esperas” entre um ou vários artigos pirotécnicos e começam sempre no canal + número mais próximo de zero. Só serão disparados os conjuntos canal + número onde o tempo seja diferente de zero.

Assim, se o número um do canal zero tiver doze segundos de efeitos até acabar, o tempo a introduzir nesse número será de doze segundos.



Figura 79. Página 2 - Modo Automático

**Página 3-** Página (Figura 80) dedicada ao modo de disparo automático. Como referido anteriormente, para aceder a esta página o cartão SD tem de estar inserido para passar todos os tempos armazenados nos ficheiros txt de tempo do cartão, para as matrizes de tempo. Para além do cartão SD inserido, a chave física tem de estar na posição 1 e o módulo de rádio LoRa tem de ser iniciado com sucesso. Caso alguma destas condições não seja cumprida, será apresentada uma mensagem de erro na página 10. Se não existir erro nenhum, é apresentada a página 11 que possui um *slider* que, quando está a 100%,

apresenta a página 3. Esta página apresenta o número de disparo ( $n2$ ) e canal ( $n4$ ) que dizem respeito ao primeiro disparo. O componente numérico  $n3$  representa o número seguinte e o  $n5$  o canal seguinte. O tempo associado ao canal + número ( $n4+n2$ ) é apresentado em baixo,  $n0$  para os segundos e  $n1$  para as décimas de segundo. Este tempo é o tempo relativo entre o disparo atual e o disparo seguinte. O LED do botão físico “FIRE” de início de disparo piscará até o mesmo botão ser pressionado (botão está ligado a uma entrada digital do ATmega2560). Antes de este ser pressionado, os botões  $b1$  e  $bt0$  estarão invisíveis. Quando o primeiro disparo é efetuado (clique do botão físico), o primeiro número é encriptado através de uma função da biblioteca AESLib.h com a chave de encriptação (chave definida no programa do ATmega2560). Depois disso são enviados dois pacotes iguais para os recetores, contendo o ID do pacote, ID do recetor, ID do emissor (ID local), canal, modo (indica se é disparo ou pedido de informações ao recetor) e o número (para disparo) encriptado com 16 bytes de tamanho. Depois disto o tempo começa a ser decrementado de 0,2 s em 0,2 s, atualiza e apresenta essas mudanças nos componentes  $n0$  e  $n1$  do HMI. Quando este tempo chegar a zero, dá-se o disparo do segundo número, repetindo o processo do primeiro conjunto canal + número (onde desta vez contém um canal ou número diferente), com o envio dos pacotes para os recetores. Em seguida os componentes  $n2$  e  $n4$  da página do HMI são atualizados com o número e canal. São também atualizados os componentes  $n0$  e  $n1$  e respetivo tempo associado a esse conjunto, começando logo a decrementar o tempo. Para além disto o próximo conjunto canal + número será também atualizado nos componentes  $n5$  e  $n3$ . Durante o decremento do tempo é possível colocá-lo em pausa através do botão  $bt0$  ou até mesmo avançar para o próximo conjunto canal + número através do botão  $b1$  “>>/”, fazendo logo o próximo disparo, sem esperar que o tempo do conjunto atual chegue a zero.

O botão  $b4$  “Menu Anterior” não estará visível enquanto o tempo está a ser decrementado, sendo que só fica visível antes do primeiro disparo ou caso o tempo seja colocado em pausa. O botão de passar à frente  $b1$  “>>/” só ficará visível durante o decremento do tempo.

O tempo do último conjunto a ser disparado não será decrementado pois não existe um conjunto canal + número seguinte. Ou seja, no último conjunto canal + número é feito o seu disparo, os componentes  $n5$  e  $n3$  tornam-se invisíveis e a página 2 é apresentada. Na



página 2 é apresentado por predefinição o canal 0 e número 1 e o respetivo tempo do conjunto.

O clique no botão *b4* “Menu Anterior” apresenta a página 2 com o canal 0, número 1 e o respetivo tempo do conjunto.

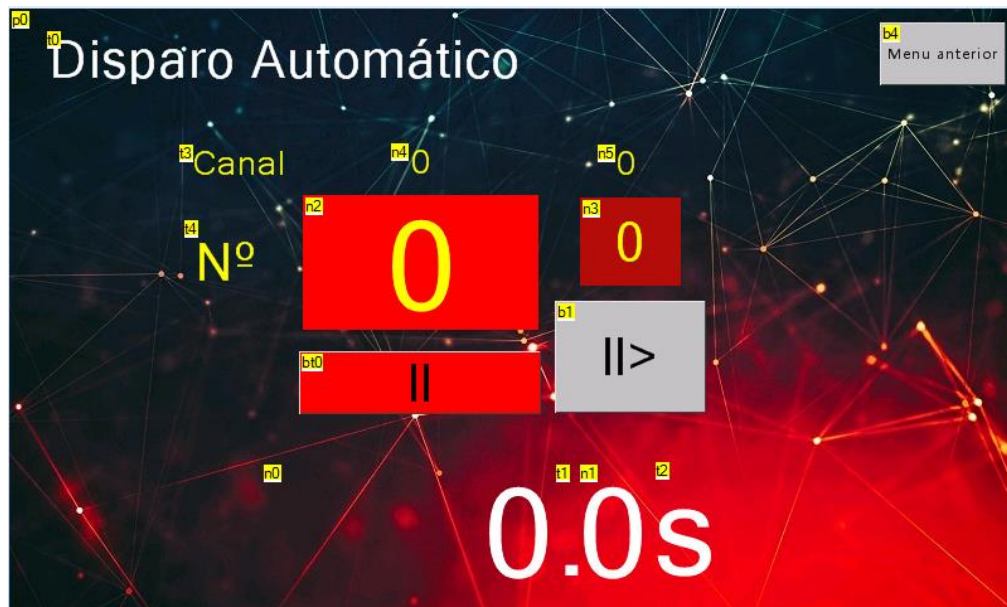


Figura 80. Página 3 - Disparo Automático

**Página 4-** Modo Semiautomático (Figura 81). Esta página é acedida a partir da página 1 “Menu Principal”. Como foi dito anteriormente, antes de ser acedida é necessário ter a chave na posição 1, o módulo de rádio iniciado com sucesso e o *slider* da página 11 a 100%. Para além disto, o LED do botão de disparo do painel do emissor acenderá (LED ligado a uma saída digital do ATmega2560).

A página é composta pelos botões *b0* e *b1* que, respetivamente, incrementam e decrementam o canal (*n0*), até ao valor máximo 255. Os botões *b4* e *b3* são semelhantes aos anteriores só que em vez de alterarem o canal alteram o número (*n1*), até ao valor máximo 64. Neste modo, o botão físico de disparo estará sempre a ser monitorizado e sempre que existir um clique (o estado passa de *LOW* para *HIGH*), é feito um get aos componentes que apresentam no HMI o canal e o número (*n0* e *n1*), armazenando os seus valores em variáveis. De seguida, o número de disparo é encriptado, e, à semelhança do modo automático, é enviado pelo módulo de rádio dois pacotes de dados com ordem de disparo. Enquanto o botão é pressionado, o emissor está a enviar constantemente o mesmo pacote a cada 100 ms, piscando também o LED do botão sempre que um pacote é enviado. Quando o botão é largado o número (*n1*) é incrementado e o botão volta a ser

monitorizado para o próximo disparo. Se o número ultrapassar o valor 64 (número máximo) e o canal ( $n0$ ) for inferior a 255, este é incrementado e o número ( $n1$ ) passará para um. O botão  $b2$  “Menu Principal” volta à página 1 “Menu Principal”, onde o valor do componente numérico que apresenta a tensão da bateria é atualizado.



Figura 81. Página 4 - Semiautomático

**Página 5-** Esta página (Figura 82) apresenta o valor do ID do emissor ( $n0$ ), luminosidade ( $n1$ ), temperatura ( $n3$ ) e humidade ( $n2$ ). Os valores da temperatura e humidade são atualizados de acordo com a temperatura interna do dispositivo quando a página é acedida. Os valores das variáveis ID do emissor e luminosidade são acedidos a partir das variáveis armazenadas na EEPROM do próprio ATmega2560. A variável ID do emissor pode ser modificada, através dos botões  $b0$ ,  $b1$ ,  $b3$  e  $b4$ , os quais incrementam e decrementam o valor ( $n0$ ) de 0 até 255.

A luminosidade pode ser alterada, em tempo real, através do componente *slider*  $h0$  (o que é conseguido através de uma instrução do HMI) sendo o valor associado a  $n1$ , de 1 a 100%, que indica o valor, em percentagem, de luminosidade selecionado pelo *slider*  $h0$ .

O botão  $b2$  (*Menu Principal*) quando pressionado, o ATmega2560 compara os valores anteriores das variáveis ID do emissor (ID local) e luminosidade e, caso sejam diferentes dos atuais presentes na página do HMI ( $h0$  e  $n0$ ), escreve no(s) endereço(s) de memória EEPROM o(s) novo(s) valor(es). Após isto apresenta a página 1 (Menu Principal) com o valor atualizado do componente numérico que apresenta a tensão da bateria.



De referir que quando o emissor é ligado, os dois endereços de memória EEPROM do ATmega2560, onde estão armazenados os valores (1 a 100) do nível de luminosidade e ID do Emissor (0 até 255), são acedidos e é realizada a atualização do nível de luminosidade do LCD do HMI através de um comando enviado do ATmega2560, através da porta série 2 para o HMI.

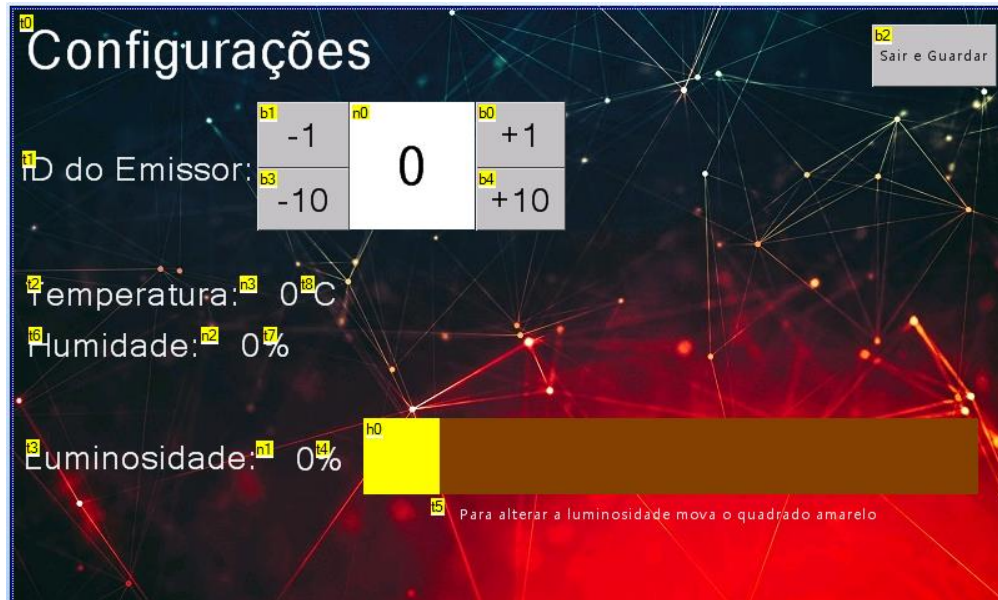


Figura 82. Página 5 -Configurações

**Página 8-** Esta página (Figura 83) tem como função inserir um intervalo de ID's dos recetores que se pretendem conhecer os seus dados genéricos (ID do recetor, canal, bateria, RSSI e SRN). O componente "ID min" ( $n0$ ) é alterável através dos botões  $b0$  e  $b1$ . O componente "ID max" ( $n1$ ) é alterável através dos botões  $b2$  e  $b3$ . O "ID max" ( $n1$ ) nunca pode ser inferior ao "ID min" ou maior do que 255. Já o "ID min" nunca é maior do que o "ID max" ou menor do que zero. O botão  $b5$  "Menu Principal" apresenta a página 1 "Menu Principal", com o valor atualizado do componente que apresenta a tensão da bateria. O clique sobre o botão  $b4$  "SCAN" faz com que o texto  $t5$  "A procurar... Por favor aguarde" se torne visível e após este clique os botões  $b4$  e  $b5$  desaparecem. Além disto, o ATmega2560 entra numa função onde faz um get aos objetos  $n0$  e  $n1$  (ID min e ID max), e envia um pacote de dados para cada um dos ID's relativos ao intervalo de ID's inseridos. Cada pacote contém o ID do pacote, ID do recetor, ID do emissor (ID local), canal, modo (indica se é disparo ou pedido de informações ao recetor) e um dado (0-255 para confirmação) encriptado com 16 bytes de tamanho. Entre cada envio, o emissor fica à escuta (durante 200 ms) de um pacote como resposta vindo do recetor com o ID local (ID

do recetor) igual ao do ID destino do pacote. Caso o receba, armazena os dados recebidos, separados por vírgulas, num ficheiro txt no cartão SD. Após percorrer todos os ID's, são analisados todos os dados do ficheiro txt. Os dados dos primeiros seis ID's desse ficheiro são armazenados, separados por um espaço em branco, em cada posição de um vetor de *strings*. Depois disto é apresentada a página 6 onde os dados de cada posição do vetor são passados aos componentes *t6* a *t11*. Ou seja, *t6*=vetor[0] até *t11*=vetor [5]. Para além disto é apresentado o número de ID's encontrados (*n0*). Caso esse número seja superior a seis, o botão *b2* "Próximos" fica visível.



Figura 83. Página 8 - Info Recetores

**Página 6-** Esta página (Figura 84) apresenta os dados dos ID's encontrados de acordo com o intervalo de valores inserido na página 8. A página é composta por seis componentes (caixas) de texto, *t6* a *t11*, nos quais são apresentados os dados de cada recetor encontrado. Esses dados, como referido anteriormente, provêm de um vetor de *strings*. Cada posição desse vetor representa os dados de um recetor. O botão *b0* "Menu Anterior" encaminha para a página 8. Os botões *b2* "Próximos" e *b3* "Anteriores" estão invisíveis por predefinição, contudo, se o número de ID's encontrados for superior a seis, o botão *b2* fica visível. Ao clicar sobre ele o botão *b3* "Anteriores" aparece, e são apresentados os próximos seis ID's, caso existam, ou os que existirem. Se forem menos do que seis, ficam as restantes caixas de texto em branco. Caso existam mais ID's, o botão *b2* "Próximos" permanecerá visível. Para isto funcionar, foi criada uma variável que armazena o número de páginas disponíveis de acordo com os ID's encontrados. Por exemplo, caso se encontrem catorze,

será verificado se o resto da divisão de ID's encontrados por seis é zero (número de páginas = ID's encontrados % 6). Neste caso, como o resto da divisão é diferente de zero, o número de páginas é igual ao número de ID's encontrados a dividir por seis e ao resultado acrescenta-se um (3 páginas). Caso o número de ID's encontrados seja igual a doze, o resto da divisão na primeira verificação será zero, ou seja, o número de páginas será igual ao número de ID's encontrados a dividir por seis (duas páginas). Com isto é possível navegar por várias páginas de acordo com os ID's encontrados.



Figura 84. Página 6 - Dados dos Recetores

Sempre que se navegar para uma página diferente (para percorrer os ID's encontrados) serão passados novos valores para o vetor de *strings* a partir do ficheiro txt do cartão SD. Para além disto, o clique sobre as caixas de texto (t6 a t11) está a ser monitorizado e, caso haja um clique, são enviados dois pacotes de dados (para uma maior fiabilidade da comunicação) compostos pelo: ID do pacote, ID do recetor (em questão), ID do emissor (ID local), canal, modo (indica se é disparo ou pedido de informações ao recetor) e um dado (0-255) encriptado com 16 bytes de tamanho (dado serve para o recetor verificar se o dado depois de descriptar, contém um valor entre 0 e 255). O emissor fica um tempo à escuta da resposta do recetor. Se não receber, mantém-se na mesma página. Caso receba, as variáveis são analisadas, é verificado se o ID de destino do pacote enviado pelo recetor é igual ao ID local (emissor) e encaminha para a página 7, caso as condições se verifiquem positivamente. Logo aparecerão na página 7 os dados mais específicos recebidos daquele ID (do recetor), com informações como ID do recetor (n4), humidade (t10), temperatura

(*t11*), tensão da bateria (*t12*), SNR (*t14*), RSSI (*t13*) e quais os terminais do recetor que contêm ignitor (*t4*), e os que não contêm(*t4*).

**Página 7-** Nesta página (Figura 85) são apresentados os dados mais específicos relativos a um ID do recetor. É constituída por vários componentes de texto e um numérico. Estes dados provêm de dados recebidos pelo emissor, num pedido (ao recetor) mais detalhado de informações. A informação é tratada no emissor quando o pacote de dados é recebido e é feito um conjunto de *sets* nos vários componentes da página. O botão *b0* “*Verificar Novamente*”, serve para enviar um novo pedido para o mesmo recetor, através de um pacote com a mesma estrutura, com um ID de pacote diferente do enviado na página 6. Espera um tempo (900 ms, tempo que leva a receber uma resposta do recetor) para que o recetor, receba o pacote, processe e envie a resposta para o emissor. Se o emissor receber resposta no tempo previsto, trata os dados, verificando se o ID de destino que o recetor enviou é igual ao ID local (ID do emissor). Se as condições anteriores se verificarem, atualiza todos os componentes numéricos e de texto de acordo com os novos dados recebidos. O botão *b1* “*Voltar*” faz o encaminhamento para a página 6, onde se apresentam os mesmos ID’s que estavam no momento antes de um deles ser pressionado (evento clique sobre um deles). Os dados manter-se-ão na página 6, porque os componentes que apresentam os ID’s presentes nessa página são globais, e só apagam o seu conteúdo quando este é substituído ou quando o emissor é desligado.



Figura 85 Página 7 - Dados do Recetor

**Página 9-** Modo Manual. As condições de acesso à página “*Modo Manual*” (Figura 86) são semelhantes às condições para aceder à página 4 “*Modo Semiautomático*”, só que através de outro botão (*b1*) na página 1 “*Menu Principal*”. A página é composta por 36 botões. O *b0* e *b1* que incrementam e decrementam o componente Canal (*n0*). Cada botão entre *b2* a *b33* representa um número de disparo. Eles estão constantemente a ser monitorizados pelo ATmega2560 quando a página 9 é acedida. Quando um dos botões de disparo é pressionado, é chamada a função associada ao botão. A função realiza um *get* e armazena em duas variáveis os valores dos componentes Canal (*n0*) e número do disparo associado ao botão. De seguida, o número de disparo é encriptado, e, à semelhança do modo automático, é enviado pelo módulo de rádio dois pacotes de dados com ordem de disparo. Em seguida, todos os botões continuarão a ser monitorizados.

Por predefinição são apresentados os números de um a 32 pelos botões *b2* a *b33*, contudo o botão *b35* serve para mudar o valor de todos os botões (*b2* a *b33*), incrementando o valor 32 a cada um e caso haja um clique, por exemplo no botão *b2*, a função de disparo vai interpretá-lo como se fosse o número 33. Para além disto o texto do botão *b35* “*Mudar 33-64*” irá alterar para “*Mudar 1-32*”, e ao ser pressionado novamente volta ao seu estado original. O botão *b34* serve para voltar à página 1 “*Menu Principal*”, onde o componente numérico que apresenta a tensão da bateria é atualizado.

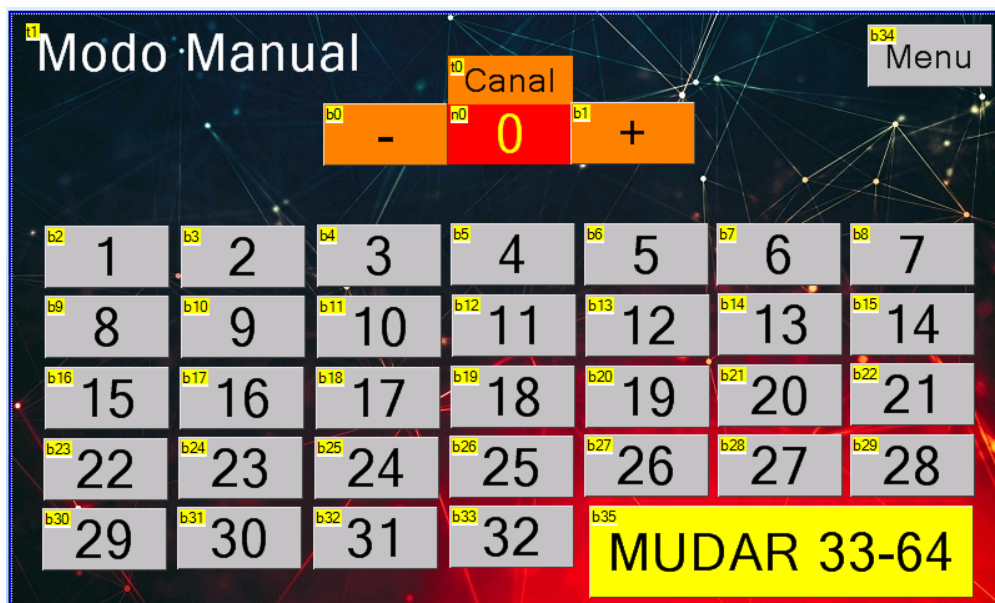


Figura 86. Página 9 - Modo Manual



**Página 10-** Página de avisos e erros. Esta página (Figura 87) é composta por um componente de texto (*t1*), onde é apresentada uma mensagem de aviso ou erro fornecida pelo ATmega2560, caso exista algum erro. Para sair desta página basta clicar na grande área do botão *b0* “Para sair carregue aqui”. O clique faz com que a página anterior seja apresentada. Caso a página anterior seja o “Menu Principal”, o valor do componente numérico que apresenta a tensão da bateria é atualizado. Caso seja a página “Modo Automático” as variáveis que apresentam o tempo são atualizadas de acordo com o canal 0 e número 1.



Figura 87. Página 10 - Avisos e erros

**Página 11-** Página de proteção para os modos de disparo. Esta página (Figura 88) é empregada antes de entrar nas páginas dedicadas ao disparo (*Modo Manual, Semiautomático e Disparo Automático*). O valor do *slider* (*h0*) está sempre a ser monitorizado pelo ATmega2560 e quando o seu valor for igual a 100 (quadrado do *slider* na extremidade direita), as páginas dos modos de disparo são abertas. Esta página tem como objetivo servir de proteção para toques acidentais nos botões dos diversos modos de disparo.

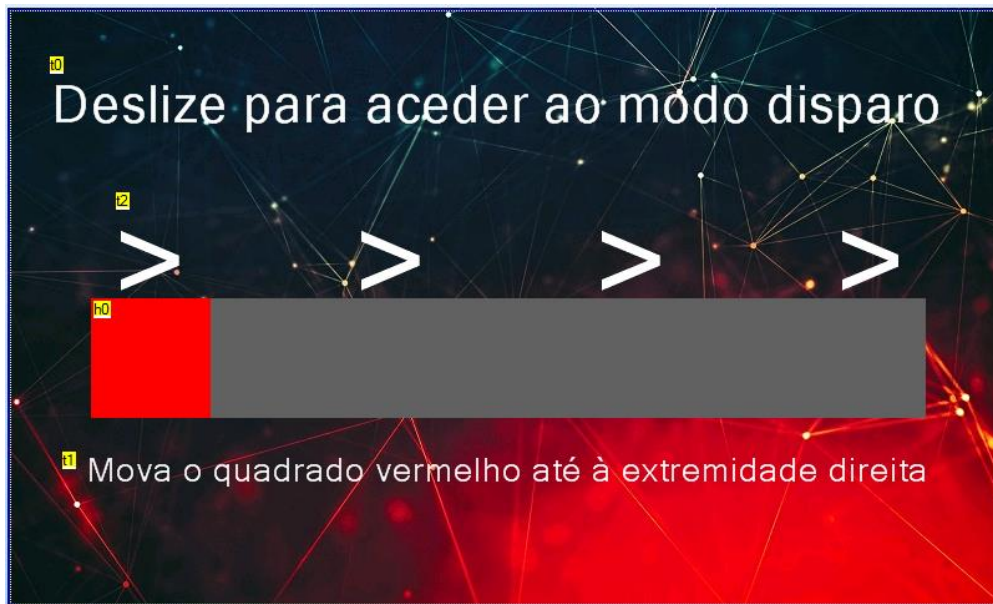


Figura 88. Página 11 - Proteção para os modos de disparo

### 5.1.3. Descrição do funcionamento do Recetor

O programa do recetor é composto essencialmente por um ciclo *loop* principal e várias funções. Dentro do ciclo *loop* é monitorizada a chave física de dois estados (zero ou um) do painel do recetor. O recetor possui dois modos de funcionamento, acedidos através da chave. Quando a chave está na posição zero (modo seguro), o ecrã do HMI está operacional; uma função verifica se foi recebido um evento clique de algum botão do HMI (associado a uma lista de objetos Nextion) e caso surja, executa a função associada a esse botão; também é possível alterar o valor dos componentes “Canal”, “ID Emissor” e “ID Recetor” presentes na página 1. Para além disto, o recetor apenas responde a pedidos provenientes do emissor que apresente o objetivo de obter informações acerca de algumas variáveis do recetor. Neste modo são ignoradas ordens de disparo por parte do emissor.

O programa em ambos os modos de funcionamento está sempre à escuta de pacotes que cheguem através de uma função da biblioteca LoRa. Caso receba um pacote, o ID local de origem do pacote recebido tem de ser igual ao valor presente na variável ID do emissor no recetor. Caso não seja ignora o pacote. Caso seja, verifica o ID do pacote e compara-o com o ID do pacote anterior. Caso o valor do ID do pacote seja igual ao ID do pacote anterior, rejeita o pacote. Este processo é realizado porque o emissor envia sempre duas vezes o mesmo pacote (para aumentar a fiabilidade), sendo uma forma de melhorar a fiabilidade da comunicação. Depois verifica o valor da variável “modo” do pacote recebido, que define qual é o objetivo do pacote. Caso contenha o valor cinco, significa, disparo, o que é

descartado neste modo de funcionamento do recetor. Caso seja dez refere-se ao pedido de dados básicos do recetor (canal e tensão da bateria). Se for quinze refere-se a um pedido mais específico de dados do recetor (canal, temperatura, humidade, tensão da bateria e quais os terminais que contêm ignitores).

A mensagem do pacote que contém um número é descriptada através de uma chave de 16 bytes armazenada no recetor (igual à chave que encriptou o número no emissor) e depois de descriptada é verificado se o valor está entre 0 e 255. Caso a condição se verifique continua para a próxima condição, se não, descarta e volta à escuta.

Se a variável “modo” do pacote recebido for dez e se o ID de destino do pacote for igual ao ID local (ID do recetor), o recetor envia um pacote com o ID local, ID de destino (ID do emissor), canal e tensão da bateria.

Se a variável “modo” for quinze e o ID de destino do pacote for igual ao ID do recetor, entra dentro de uma estrutura *for()* de 1 a 64, com incrementos unitários. Na estrutura, são percorridos todos os sinais das entradas dos *multiplexers* de cada uma das quatro régua. Ao percorrer cada uma das 64 entradas dos *multiplexers*, caso haja um sinal “HIGH”, é adicionado a uma *string* o valor um, caso contrário é adicionado o valor zero. No final temos uma *string* com 64 caracteres de zeros e uns, sendo que o caracter mais à esquerda da *string* diz respeito ao terminal da régua número 1 e o mais à direita ao terminal número 64. Por fim esta *string* é vinculada ao pacote de dados, e é adicionado também ao pacote, o ID local (ID do recetor), ID destino (ID do emissor), canal, e é enviada para o emissor.

Quando a chave está na posição 1 (modo disparo), o HMI entra em modo *sleep* (deixa de estar operacional), o ATmega2560 para de monitorizar os eventos provenientes do HMI e um LED vermelho do painel do recetor é ligado, para indicar que o modo disparo está ativo. Além disto, o recetor passa a responder apenas a ordens de disparo por parte do emissor. Todos os outros pedidos são ignorados.

A partir deste momento o recetor fica só a escutar os pacotes recebidos, e a verificar o estado da chave física (caso seja rodada para a posição 0 sai deste modo).

Quando recebe um pacote, este é analisado, e caso o ID de destino seja igual ao ID do emissor (armazenado no recetor), o ID do pacote seja diferente do ID do pacote anterior armazenado, o canal do pacote seja igual ao do canal definido no recetor; o modo seja o cinco (disparo), e a mensagem quando descriptada contenha um valor entre 1 e 64, a função de disparo é chamada e o número é lido e passado como parâmetro de entrada. Esta



função possui uma estrutura de decisão *switch/case*, possuindo para cada alternativa uma sequência de controlo da saída de cada *demultiplexer*. Depois disto, é colocado em funcionamento o respetivo *demultiplexer* da régua, que possui o número de disparo, e é colocado nível alto (5 V) na saída selecionada anteriormente, durante 50 ms (tempo suficiente para o ignitor entrar em ignição), de forma a colocar o transístor associado a esse número em saturação. Após isto, a saída passa para 0 V, o demultiplexer é desabilitado e os sinais de controlo do *demultiplexer* são colocados ao nível baixo (0 V). Em seguida, o programa volta para o *loop* principal.

A descriptação da mensagem é realizada a partir de uma função da biblioteca AESLib.h, usando uma chave de 16 bytes armazenada no programa do ATmega2560 do recetor.

Na Figura 89 é apresentado um fluxograma que demonstra o conceito básico de tratamento de pacotes recebidos pelo recetor.

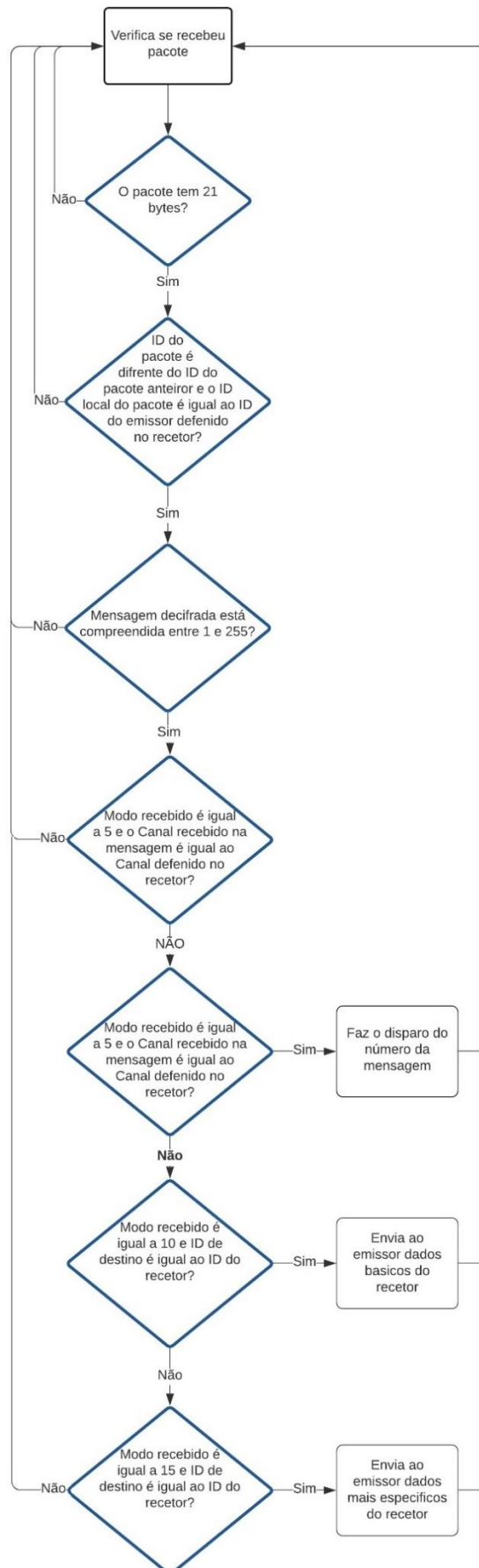


Figura 89. Fluxograma básico do funcionamento do recetor

O sistema é composto por seis páginas, sendo de seguida apresentadas as descrições de cada uma:

**Página 0-** Página de apresentação da aplicação do recetor (Figura 90). O evento clique sobre esta página é monitorizado pelo microcontrolador ATmega2560 e, quando existir um clique, apresenta a Página 1 “Página principal do recetor”, com os componentes ID emissor, ID recetor e canal, atualizados de acordo com os dados presentes em vários endereços de memória EEPROM do ATmega2560. É apresentado também o componente numérico que indica a tensão da bateria atualizada.



Figura 90. Página 0 - Apresentação do recetor com o nome da aplicação

**Página 1-** Esta página (Figura 91) é responsável pela apresentação de variáveis básicas essenciais para a comunicação e funcionamento do recetor, tais como canal ( $n2$ ), ID do emissor ( $n0$ ), ID do recetor ( $n1$ ) e tensão da bateria ( $t4$ ). As variáveis estão armazenadas na EEPROM do ATmega2560 e são acedidas no momento do arranque do programa. Estas são transferidas para variáveis globais de forma a serem usadas em várias funções no decorrer do programa e apresentadas no HMI. Podem ser alteradas através dos botões  $b1$ ,  $b2$  e  $b3$ . O botão  $b1$  "Alterar" apresenta a página 2 onde é apresentado, no componente numérico  $n0$  o valor do canal. O botão  $b3$  "Alterar" encaminha para a página 4 onde é apresentado, no componente numérico  $n0$ , o valor do ID do emissor.

O botão *b2* "Alterar" encaminha para a página 3 onde é apresentado, no componente numérico *n0* o valor do ID do recetor.



Figura 91 Página 1 -Página principal do Recetor

**Página 2-** Esta página (Figura 92) serve para alterar o valor do canal do recetor. Possui o campo numérico (*n0*) onde está apresentado o canal. O valor do canal (*n0*) pode ser alterado de 0 a 255 através dos botões *b1*, *b2*, *b3* e *b4*. Estes botões servem para incrementar e decrementar o valor do canal (*n0*) através de incrementos e decrementos de valor um ou dez. O botão *b0* serve para voltar à página 1 e para guardar as alterações. Neste caso, o valor do componente *n0* é comparado com o valor anterior do canal e, caso seja diferente, escreve o novo valor num endereço de memória da EEPROM. Caso contrário, nada faz, apenas volta à página 1. A escrita na EEPROM serve para a variável não ser perdida depois de desligar o recetor. Quando a página 1 é apresentada, o componente numérico que apresenta a tensão da bateria é atualizado.



Figura 92. Página 3 - "Alterar Canal"

**Página 3-** Esta página (Figura 93) tem como função, alterar o ID do emissor. A lógica do seu funcionamento é semelhante à página 2.

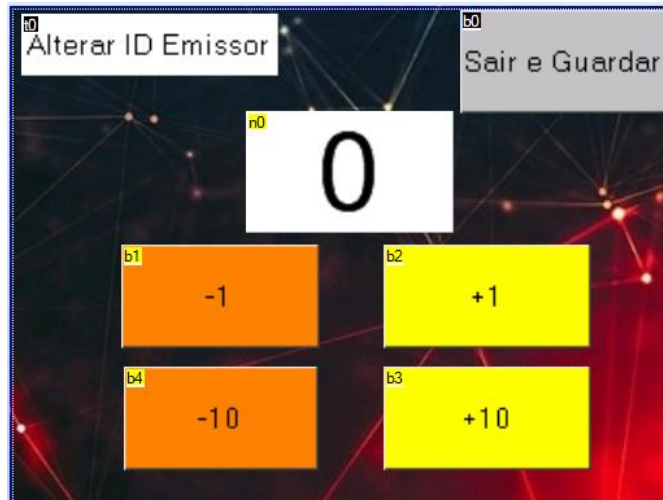


Figura 93. Página 4 - "Alterar ID Emissor"

**Página 4-** Esta página (Figura 94) tem como função alterar o ID do recetor (ID local). A lógica do seu funcionamento é semelhante à página 2.

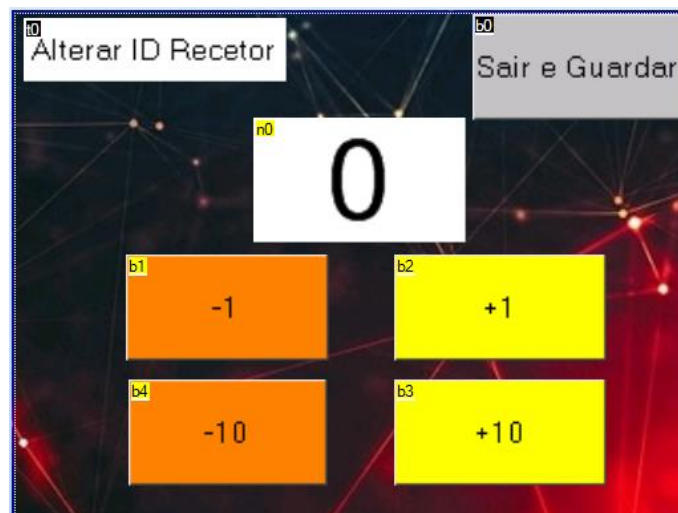


Figura 94. Página 5 - "Alterar ID Recetor"

**Página 5-** Esta página (Figura 95) serve para configurar o brilho do ecrã do HMI. O valor apresentado é guardado numa variável. Quando o ATmega2560 entra em funcionamento, o valor da luminosidade do ecrã armazenado num endereço de memória da EEPROM é acedido e é guardado numa variável global. O *slider* (*h0*) altera o brilho em tempo real, através de uma instrução implementada no próprio HMI. O valor do *slider* está limitado a percorrer de 1 a 100. O valor *n0* fornece a percentagem de luminosidade do LCD e atualiza em tempo real quando o *slider* se move, sendo apresentado o valor de um a 100%. Por fim o botão *b0* serve para voltar à página 1 e guardar o valor do *slider* num endereço de

memória da EEPROM do ATmega2560, caso o valor presente seja diferente do valor de luminosidade anterior. Caso contrário apenas vai para a página 1. Quando a página 1 é apresentada, o componente numérico que apresenta a tensão da bateria é atualizado.

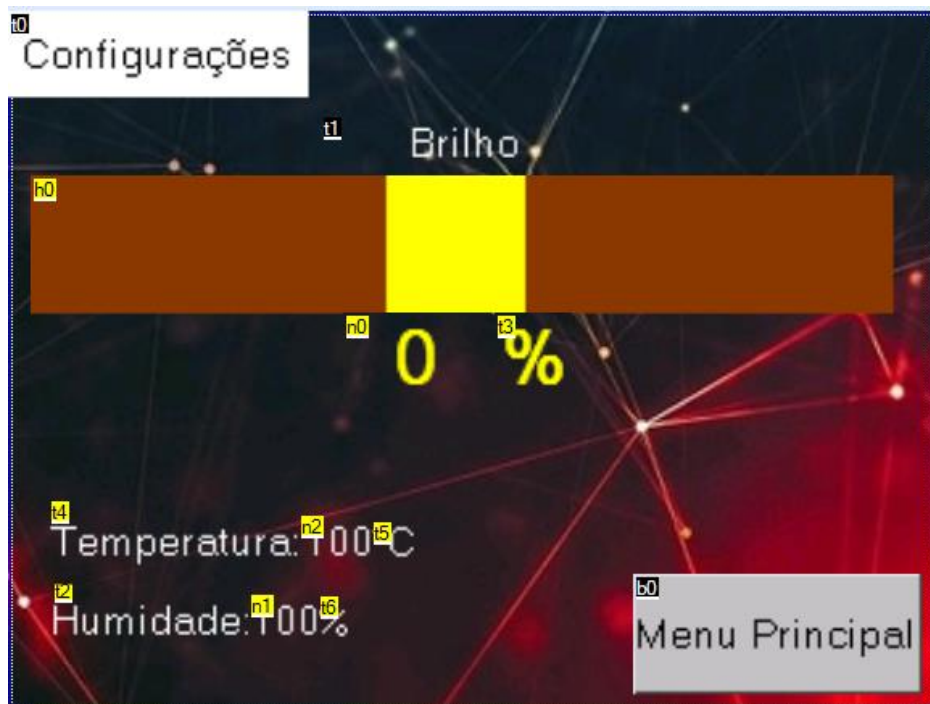


Figura 95. Página 6 - Configurações

#### 5.1.4. Configurações e instruções do HMI (Emissor e Recetor)

Cada componente do HMI tem associado dois ou mais eventos. Dentro desses eventos pode-se colocar um conjunto de instruções de forma a executar uma ou várias instruções quando esses eventos são gerados. O exemplo mais comum é o componente botão, que geralmente tem dois eventos associados:

O *“Touch Press Event”* que é gerado quando o botão é pressionado (passa do estado zero para o estado um). O *“Touch Release Event”* é gerado no momento em que o botão é largado (passa do estado 1 para o estado 0).

Nesses eventos existe a possibilidade de habilitar a função *“Send Component ID”* a qual envia, através da comunicação série para o microcontrolador (ATmega2560 por exemplo), o número da página, o ID do componente e o evento ocorrido (0- *Release* e 1-*Press*). Caso ocorra um evento *“Touch Press Event”* no componente com ID 1 na página 0 é enviado através da porta série para o microcontrolador a seguinte estrutura:

0x65 0x00 0x01 0x01 0xFF 0xFF 0xFF, onde 0x00 é o número da página, 0x01 é o ID do componente, 0x01 é o evento (0x01 *Press* e 0x00 *Release*).

Estes dados são recebidos no microcontrolador, através da porta série, e interpretados por funções e métodos associados à biblioteca Nextion, onde é chamada no microcontrolador, a função associada ao evento do componente.

De referir que o *baud rate* (em bits por segundo) usado para a comunicação entre o microcontrolador e o HMI é de 115200 bps sendo este o valor máximo que o fabricante do HMI recomenda. O *baud rate* padrão era de 9600 bps, contudo estava a trazer problemas na página “Disparo Automático”, pois atrasava o decréscimo de tempo porque são muitas as variáveis a serem atualizadas no HMI.

A seguir estão apresentadas todas as instruções, por página, definidas no próprio HMI do emissor:

Na página 0 “Página de apresentação”, a página tem habilitada no evento “*Touch Press Event*” a função “*Send Component ID*”.

Na página 1 “Menu Principal”, os botões “Modo Automático”, “Modo Manual”, “Semiautomático”, “Info Recetores” e “Configurações” têm todos habilitada a função “*Send Component ID*” no evento “*Touch Release Event*”.

Na página 2, os botões *b1* “+”, *b2* “-”, *b3* “+”, *b4* “-”, *b7* “Reset tempos”, *b13* “Gravar SD”, *b0* “Menu Principal” e *b12* “Disparo” têm habilitada a função “*Send Component ID*” no evento “*Touch Release Event*”.

Os botões *b15* “+100”, *b10* “+10”, *b5* “+1”, *b8* “+0.2”, *b14* “-100”, *b11* “-10”, *b6* “-1” e *b9* “-0.2” têm, associado ao evento “*Touch Press Event*”, um conjunto de instruções que incrementam e decrementam o valor dos componentes numéricos *n2* e *n3* de acordo com o valor indicado no botão. As instruções têm em conta os limites do componente *n2* com o valor 9999 como máximo e 0 como mínimo. Enquanto no componente *n3*, os valores que pode apresentar, vão de 0 a 8 com incrementos e decrementos de valor dois.

A seguir está apresentado um conjunto de instruções associadas ao evento de pressão do botão *b8* (*Touch Press Event*):



```

if(n3.val==8)
{
  if(n2.val<9999)
  {
    n2.val+=1
    n3.val=0
  }
}
else
{
  n3.val+=2
}

```

Na página 3 os botões *bt0* "||" (pausa) e *b1* ">>|" (seguinte) inicialmente estão invisíveis através do comando "vis" definido na abertura da página. O evento "Touch Release Event" associado ao botão *bt0* de dois estados, possui um conjunto de instruções que fazem mudar o seu texto para "|>" quando o botão é libertado, e quando esta ação volta a ocorrer, altera para o texto inicial "||". Como o botão *bt0* é de dois estados fixos, quando é pressionado e libertado fica no estado um com o texto apresentado "|>", e quando este é novamente pressionado e libertado o seu valor passa para zero e o seu texto para "||".

Na página 4 "SemiAutomatico" os botões *b0* "+" e "*b1*" possuem associado ao evento "Touch Press Event" um conjunto de instruções que fazem incrementar e decrementar o valor do componente numérico *n0* (responsável por apresentar o canal), sendo que inicialmente o valor padrão do componente *n0* é zero (que apresenta o número de disparo) e o componente *n1* é um (porque não existe o número zero no disparo). Estas instruções estão desenvolvidas de forma a que o valor de *n0* esteja compreendido entre 0 e 255.

Os botões *b4* "-" e *b3* "+" têm associado ao evento "Touch Press Event" um conjunto de instruções que incrementam e decrementam o valor do componente numérico *n1* (que apresenta o número de disparo). Estas instruções estão desenvolvidas de forma a que o componente *n1* possua valores entre um e 64 (número mais baixo e mais alto de disparo). Caso o botão *b3* "+" seja pressionado e o valor presente em *n1* seja 64 (número de disparo máximo), o *n0* será incrementado se for inferior a 255 (canal máximo) e o valor de *n1* passará para o valor um.



Caso o botão *b4* “-” seja pressionado e o valor presente em *n1* seja um (número de disparo mínimo), o *n0* será decrementado se for superior a zero (canal mínimo) e o valor de *n1* passará para o valor 64 (número máximo).

Estas instruções foram implementadas de forma a que o utilizador não tenha de trocar o canal e o número manualmente quando estiver no valor máximo do número de disparo apresentado em *n1*.

Na página 5 “Configurações” os botões *b1* “-1”, *b0* “+1”, *b3* “-10” e *b4* “+10” têm associado ao evento “*Touch Release Event*” um conjunto de instruções que fazem o incremento e decremento do valor do componente numérico *n0* (ID do emissor). As instruções estão estruturadas de forma a que o valor de *n0* esteja compreendido entre 0 e 255.

O componente *slider h0* tem associado aos eventos “*Touch Press Event*” e “*Touch Move Event*” duas instruções. A primeira altera o valor da luminosidade do LCD, e a segunda altera o valor do componente numérico *n0* (que apresenta o valor de 1 a 100 do *slider*). De referir que o *slider* foi configurado com valor mínimo 1 (brilho mínimo) e 100 (brilho máximo) como máximo.

No evento “*Touch Release Event*”, associado ao botão *b2* “Menu Principal”, foi habilitada a opção “*Send Component ID*”. A página 6 “Dados dos recetores” ao ser apresentada, os botões *b2* “Próximos” e *b3* “Anteriores” estão configurados como invisíveis. Os componentes de texto de *t6* a *t11* foram configurados como variáveis globais, de forma a guardar os seus dados, mesmo saindo da página em questão. Cada componente de texto (*t6* a *t11*) tem a função “*Send Component ID*” habilitada quando é gerado o evento “*Touch Release Event*”. O botão *b0* “Menu Anterior” possui uma instrução para apresentar a página 8 quando o evento “*Touch Release Event*” é gerado.

Na página 7 “Dados do recetor” o botão *b0* “Verificar Novamente” possui habilitada a função “*Send Component ID*” quando o evento “*Touch Realise Event*” é gerado. O botão *b1* “Voltar” possui a instrução de apresentar a página 6, quando o evento “*Touch Realise Event*” é gerado.

Na página 8 “Info Recetores” o componente texto *t5* “A procurar... Por favor aguarde...” por predefinição está definido como invisível.

Os botões *b0* “-“ e *b1* “+” incrementam e decrementam uma unidade, o componente numérico *n0* (ID MIN) quando o evento “*Touch Press Event*” é gerado.

De salientar que as instruções que fazem esse processo têm em conta que o valor de *n0* não pode ser abaixo de zero (ID mínimo possível) nem acima do valor do componente numérico *n1* (ID MAX).

Os botões *b2* “-“ e *b3* “+” incrementam e decrementam uma unidade o componente numérico *n1* (ID MAX) quando o evento “*Touch Press Event*” é gerado.

De salientar que as instruções que fazem esse processo têm em conta que o valor de *n1* não pode ser abaixo de *n0* nem acima de 255 (ID máximo possível).

O evento “*Touch Release Event*” do botão *b5* “Menu Principal” tem a função “*Send Component ID*” habilitada.

A função “*Send component ID*” está habilitada dentro do evento “*Touch Release Event*” do botão *b4* “SCAN”.

Na página 9 os botões *b0* “-“ e *b1* “+” incrementam e decrementam o valor do componente numérico *n0* (Canal) quando é gerado o evento “*Touch Press Event*” num deles. As instruções que incrementam e decrementam o valor de *n0* têm em conta que o valor apresentado tem de estar compreendido entre 0 e 255.

Os eventos “*Touch Press Event*” associados aos botões de *b2* a *b35* têm habilitada a função “*Send Component ID*”.

Na página 10, o evento “*Touch Release Event*” associado ao botão *b0* tem habilitada a função “*Send Component ID*”.

Na página 11 o valor do componente *h0* (*slider*) foi limitado a um intervalo de zero a 100. O evento “*Touch Release Event*” tem associado um conjunto de instruções onde coloca o valor do *slider* a 0, caso o valor do *slider* seja menor do que 100 (ou seja se o utilizador mover o quadrado do *slider* e largar, e este não estiver na extremidade esquerda, a 100%, volta para 0). Esta operação faz com que o utilizador tenha de manter o quadrado do *slider* sem largar até à extremidade esquerda (100%).

De seguida são apresentadas todas as instruções por página definidas no próprio HMI do recetor:

Na página 0, a função “*Send Component ID*” está habilitada no evento “*Touch Press Event*”. Na página 1, os botões *b0*, *b1*, *b2*, *b3* possuem a função “*Send Component ID*” habilitada no evento “*Touch Release Event*” e os componentes numéricos *n0*, *n1* e *n2* estão definidos como globais (de forma a manter o seu valor, mesmo quando é acedida uma outra página, e apresentada novamente esta página de seguida).

As páginas 2, 3 e 4 possuem a estrutura e lógica de funcionamento igual. Os botões *b1* “-1”, *b2* “+1”, *b4* “-10” e *b3* “+10” incrementam o valor do componente numérico *n0* com o valor apresentado nos botões. Estas instruções de incremento e decréscimo são realizadas sempre que o evento num dos botões “*Touch Release Event*” é gerado. As instruções têm em conta que o valor de *n0* tem de estar compreendido entre 0 e 255.

O botão *b0* “Sair e Guardar” possui a função “*Send Component ID*” habilitada no evento “*Touch Release Event*”.

Na página 5 o componente *slider h0* tem associado aos eventos “*Touch press Event*” e “*Touch Move Event*” duas instruções. A primeira altera o valor da luminosidade do LCD, e a segunda altera o valor do componente numérico *n0* (que apresenta o valor de 1 a 100 do *slider*). De referir que o *slider* foi configurado com valor mínimo 1 (brilho mínimo) e 100 (brilho máximo) como máximo. O botão *b0* “Menu Principal” tem habilitada a função “*Send Component ID*” no evento “*Touch Release Event*”.

#### **5.1.5. Estrutura dos pacotes de dados**

No projeto são utilizados três tipos de pacotes com diferentes dados. Um é usado na troca de mensagens entre emissor e recetor, e os outros dois no sentido do recetor para o emissor. Todos os pacotes utilizam o formato explícito.

*Payload* do pacote enviado do emissor para o recetor é composto por 21 bytes (Figura 96). O primeiro byte diz respeito ao ID da mensagem onde é incrementado a cada dois pacotes enviados. São enviados dois pacotes iguais para os recetores, de forma a garantir que pelo menos um dos pacotes chegue ao destino (daí a variável só ser incrementada a cada dois pacotes enviados). No recetor esse byte é armazenado numa variável, de forma a ser comparado com a variável recebida no pacote seguinte. No recetor, caso o ID do pacote recebido seja igual ao ID do pacote recebido anterior, o pacote é descartado.

O segundo byte é usado para o emissor comunicar apenas com um recetor específico, por exemplo, para pedir informações de estado de bateria. Este byte só é necessário nos modos de operação 10 e 15.

O terceiro byte corresponde ao ID do emissor. Este ID tem de estar inserido em todos os recetores, para que estes aceitem os pacotes recebidos.

O quarto byte é referente ao canal. O canal não representa frequência, mas sim um byte que faz com que o recetor “dispare” só em alguns recetores que estejam configurados com o mesmo canal.

O quinto byte diz respeito ao modo de operação do recetor. Caso seja cinco, o recetor entrará numa rotina de disparo, e dispara a saída que estará descrita na mensagem do pacote. Caso seja dez, o recetor responderá com algumas informações à cerca dele, como a tensão da bateria e o canal configurado. Por último, caso seja quinze, o recetor para além das variáveis enviadas anteriormente, enviará o estado dos terminais das réguas (indica por cada número de terminal se possui ignitor ou não).

Os últimos dezasseis bytes dizem respeito à mensagem criptografada que, quando descriptografada, contém o valor do terminal a disparar ou, no caso dos modos 10 e 15, um valor entre zero e 255 de modo a aumentar a segurança da comunicação.

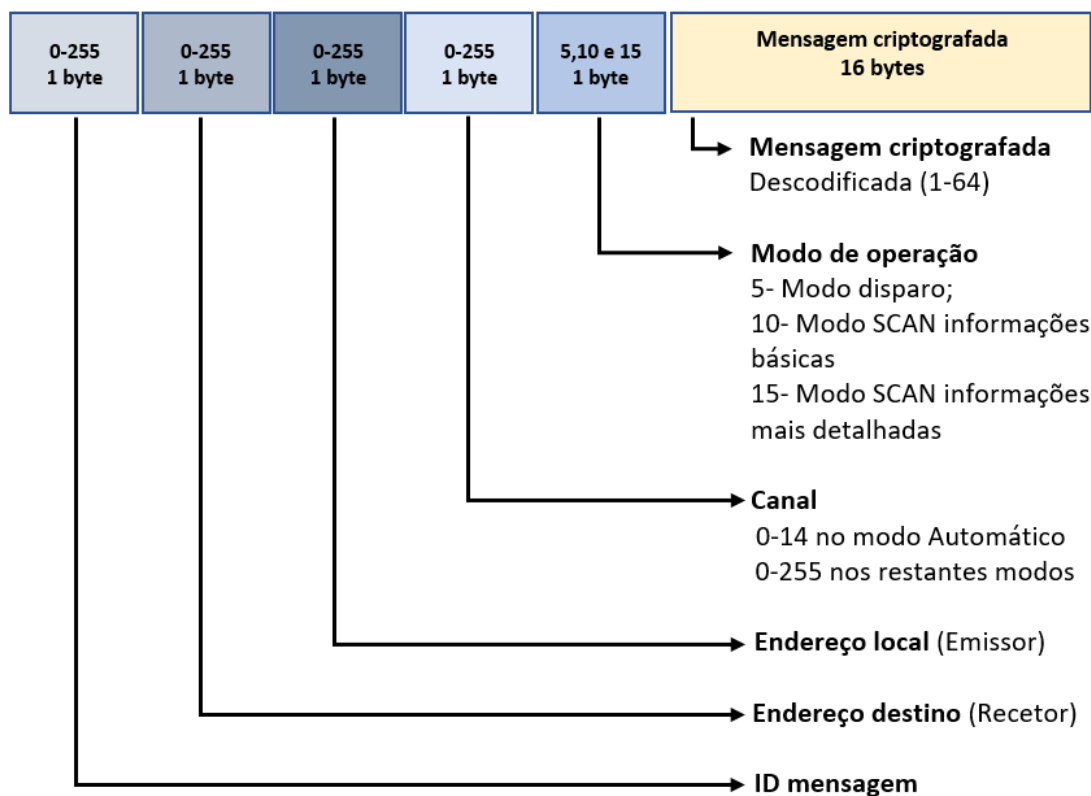


Figura 96. Payload do pacote enviado do emissor para o recetor

*Payload* do pacote enviado do recetor para o emissor quando o recetor recebe um pedido de informações gerais é composto por quatro bytes (Figura 97).

O primeiro byte é o endereço de destino, que servirá para o emissor responder, sendo que esse byte tem de ser igual ao byte do endereço local do emissor.

O endereço local serve para o emissor, quando receber este pacote, comparar se o ID local é igual ao ID de destino (ID do recetor) que o emissor fez o pedido de informações, caso não sejam iguais, ele descarta o pacote.

O byte canal e o byte bateria servem para apresentar no modo do teste as informações à cerca do recetor, onde aparecerá o canal no qual ele está configurado e a tensão da bateria.

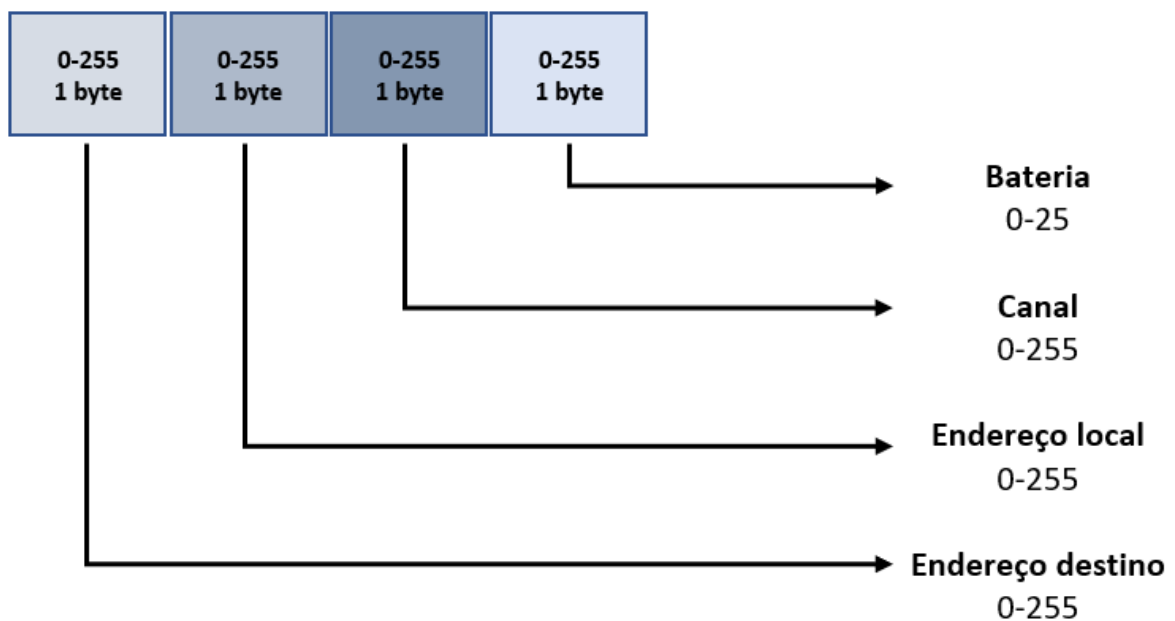


Figura 97. *Payload* do pacote enviado do recetor para o emissor quando é feito um pedido de informações básicas

*Payload* do pacote enviado do recetor para o emissor quando o recetor recebe um pedido de informações mais detalhadas é composto por 70 bytes (Figura 98).

O byte endereço destino, endereço local, canal e bateria têm exatamente as mesmas funções dos bytes com o mesmo nome do pacote anterior.

O byte temperatura e o byte humidade fornecem informações ao emissor à cerca da temperatura e humidade do recetor.

Os 64 bytes a seguir dizem respeito ao teste de continuidade nos terminais das réguas, ou seja, vão informar o emissor quais os terminais com ignitor ligado ou não. Para isso envia 64 bytes, onde cada um representa um número de um terminal, ou seja, o primeiro byte a

ser lido corresponde ao primeiro terminal, e pode ter o valor zero caso não tenha nada ligado, ou o valor um caso tenha algo ligado.

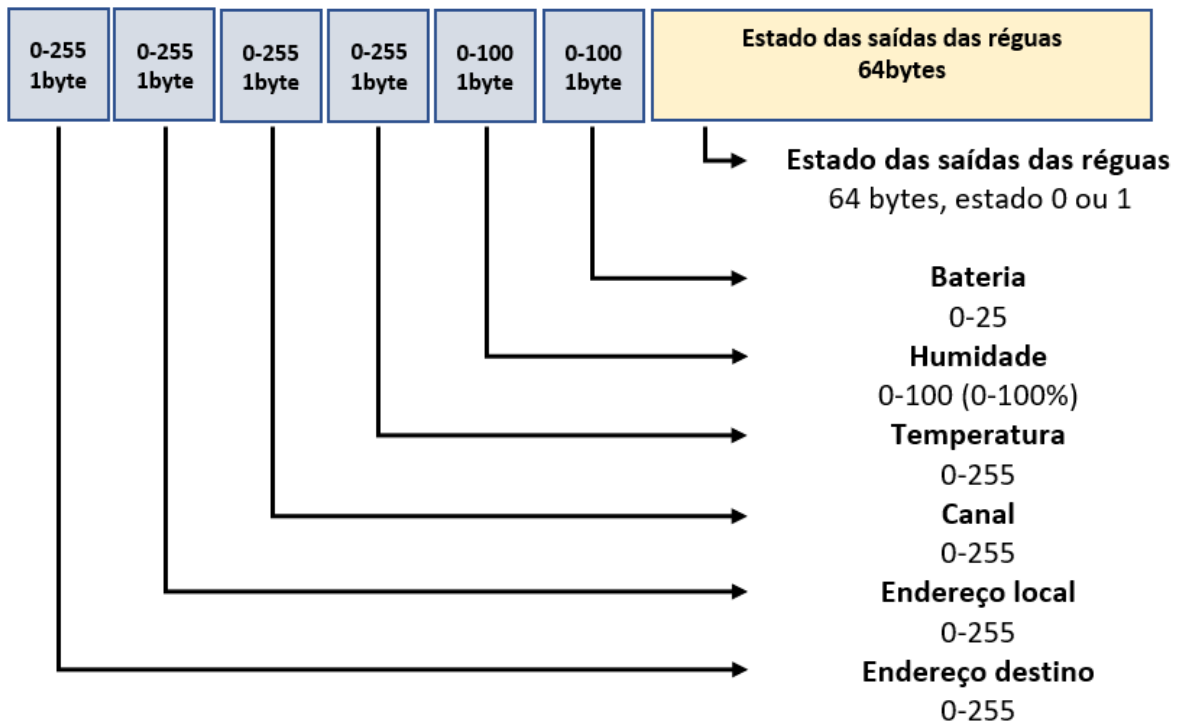


Figura 98. Payload enviado do recetor para o emissor quando é feito um pedido de informações detalhadas

## 5.2. Hardware

Neste subcapítulo são apresentados os esquemas elétricos dos módulos emissor e recetor (incluindo as réguas). É apresentado e comentado o esquema elétrico de uma maneira geral, onde cada nome de um componente dos esquemas tem a sua descrição associada em tabelas.

### 5.2.1. Esquema elétrico do emissor

O emissor é composto por: HMI Nextion de 7"; um botão de pressão de dois estados; uma chave seletora de dois estados (aberto ou fechada); um botão ON/OFF; uma BMS 3s, três células 18650 SAMSUNG IRN18650-35E; um módulo conversor *buck* MP2307DN; um módulo rádio RA-02; um módulo cartão SD; uma resistência de 25 k $\Omega$ ; uma resistência de 38 k $\Omega$ ; duas resistências de 10 k $\Omega$ ; um sensor de temperatura e humidade DHT11; um fusível rápido 600 mA; terminais de ligação para o HMI e botões; dois conversores de nível

lógico; e ainda, para controlo, o Arduino Mega 2560 Pro mini, composto pelo microcontrolador ATmega2560.

Os parágrafos seguintes apresentam uma descrição geral dos componentes utilizados no esquema elétrico do módulo emissor. Nos esquemas elétricos (Figura 99 e Figura 100), cada componente possui uma referência. No Apêndice E, é apresentada a descrição de cada referência.

A alimentação do circuito é feita por três células 18650 (BAT1). Estas estão protegidas através de uma BMS 3s (BMS3S) que auxilia durante a descarga (deixa de alimentar o circuito quando a tensão das baterias é baixa) e também auxilia no carregamento e protege contra sobrecargas e curtos-circuitos. As células são carregadas por um carregador 12,6 V conectado à entrada DC (DC1)).

O botão ON/OFF (BT1) liga e desliga a alimentação para o resto do circuito.

Um fusível rápido (FU1) é responsável por proteger o circuito caso a corrente que passa por ele, seja superior à corrente que ele suporta.

O conversor *buck* (SD3A1) tem como principal finalidade converter a tensão de entrada (até 12,6 V), numa tensão mais baixa (7 V). Os 7 V alimentam o VCC do Arduino (ARD1) e o HMI (NX7). O divisor de tensão (R4 e R3) fornece a tensão das baterias (2.83 vezes menos devido ao divisor de tensão ser 2,83 para 1) a uma entrada analógica do ATmega2560. Quando a bateria (BAT1) está com uma tensão de 12,6 V (totalmente carregada) fornece à entrada analógica 4,45 V e 3,17 V quando está totalmente descarregada (8 V).

O módulo DHT11 (STH1) é um sensor de temperatura e humidade que comunica com o ATmega2560 através da entrada digital 20.

O módulo de cartões SD (SD1) tem como função, armazenar diversos dados, como tempos, por exemplo, num cartão SD. Comunica com o ATmega2560 através do protocolo SPI (SCK-pino 52, MISO-pino 50, MOSI-pino 51, CS-pino 2, pinos estes do ATmega2560).

O módulo RA-02 (RAD1) comunica também através do protocolo SPI (SCK-pino 52, MISO-pino 50, MOSI-pino 51, CS-pino 5, pinos estes do ATmega2560). Para além destes pinos, possui um pino interrupção e um pino de RESET conectados às portas digitais 3 e 4 do ATmega2560. Como o RA-02 (RAD1) trabalha com tensões de 3,3 V teve de se adicionar dois conversores de nível lógico (CNV1 e CNV2) em todos os seus pinos.

Nos terminais (TP1) estão ligados os diversos componentes e dispositivos do painel frontal do emissor. Os terminais de 7 V e GND alimentam o HMI (NX7). Os pinos 2TX e 2RX ligados

aos pinos 16 e 17 servem para o HMI (NX7) comunicar através da porta série com o ATmega2560. A chave (CH1) serve para restringir o acesso aos modos de disparo e gravação de tempos de disparo no cartão SD. Caso a chave (CH1) esteja na posição 1 (chave fechada), são fornecidos 5 V à entrada digital 23 do ATmega2560, caso contrário (chave aberta), através de uma resistência *pull-down* (R3), garante 0 V na mesma entrada.

O botão de pressão (BT2) tem como função dar o comando “disparo” no “Modo Semiautomático” ou começar o espetáculo no “Modo Automático”. Quando se encontra pressionado, fornece à entrada digital 22 do ATmega2560 o nível lógico 5 V (*HIGH*). Caso esteja em repouso, fornece 0 V (*LOW*) devido à resistência *pull-down* (R2).

O botão de pressão (BT2) possui um LED já com resistência interna. O LED é ligado e desligado a partir da saída digital 21 do ATmega2560, e tem como função indicar ao utilizador que o botão está disponível.



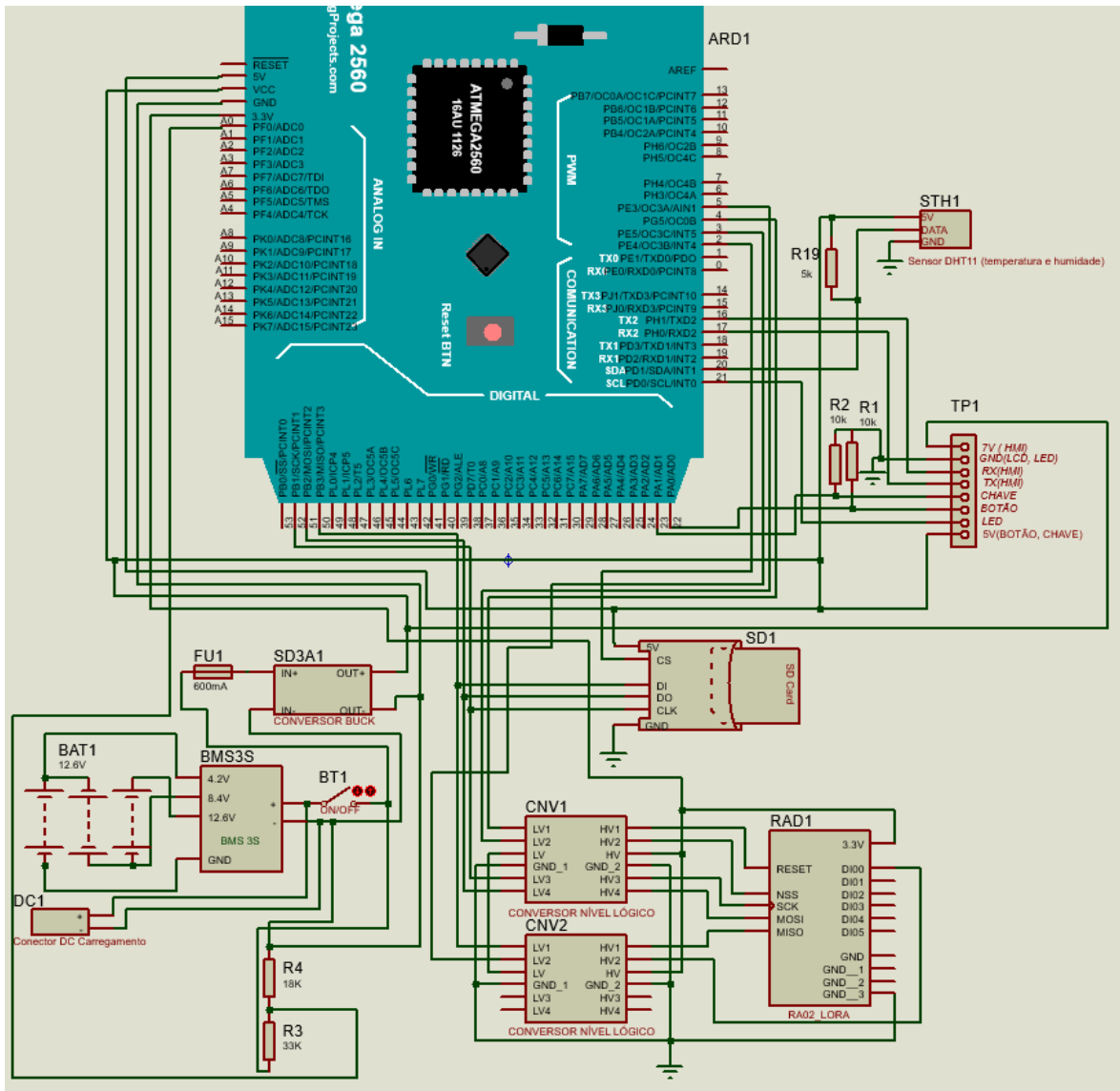


Figura 99. Esquema elétrico do módulo emissor

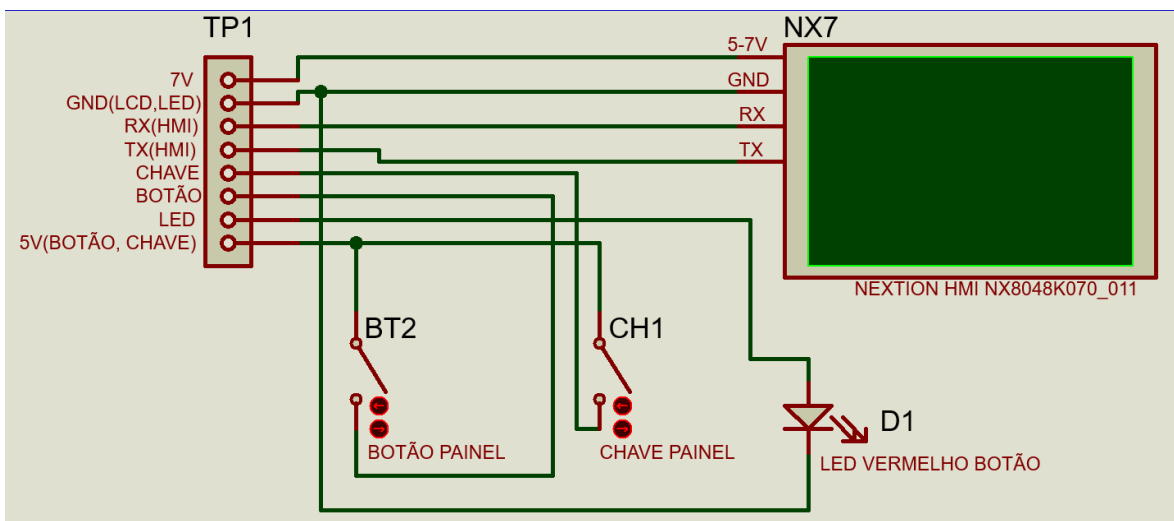


Figura 100. Esquema elétrico do painel do módulo emissor

### 5.2.2. Esquema elétrico do recetor

O recetor é composto por: seis células 18650; uma BMS 6s; um botão ON/OFF; dois conversores de nível lógico; seis resistências de 10 k $\Omega$ ; uma resistência de 270  $\Omega$ ; uma resistência de 11 k $\Omega$ ; um LED vermelho; uma resistência de 47 k $\Omega$ ; um sensor DHT11; HMI Nextion com 2,8"; dois fusíveis; um módulo de rádio RA-02; um módulo conversor *buck* LM2596 3 A; um módulo conversor *buck* XHM401 8 A; um transístor 2N3904; terminais; quatro conectores fêmea DB25 e um Arduino 2560 Pro Mini composto pelo microcontrolador ATmega2560 responsável pelo controlo.

De seguida é apresentada uma descrição geral dos componentes utilizados no esquema elétrico do módulo recetor. Cada componente possui uma referência nos esquemas (Figura 101 e Figura 102). No Apêndice F é apresentada a descrição de cada componente de acordo com a referência dos esquemas elétricos do módulo recetor e régua.

O recetor é alimentado por seis células 18650 (BAT2) ligadas em série, que fornecem um total de 25,2 V quando completamente carregadas, e cerca de 18 V quando descarregadas. O conjunto de células consegue trabalhar até uma tensão de 15,9 V, contudo para sua proteção e para fornecer uma tensão mais alta aos ignitores, considerou-se os 18 V sendo a tensão mínima.

A BMS 6S é responsável pela proteção das células contra curto-circuito e sobrecarga, e o seu equilíbrio durante as cargas e descargas.

O conversor buck (SD3A2) é responsável por baixar a tensão da bateria para 5 V. Estes 5 V alimentam o Arduino diretamente na porta de 5 V, LED's das régua, e outros dispositivos como o HMI Nextion.

O conversor *buck* (SD8A) é responsável por baixar a tensão das baterias para uma tensão de 18 V. Esta tensão é usada essencialmente nos terminais das régua (onde se ligam os ignitores).

Para proteção adicional, foram utilizados fusíveis rápidos na entrada de cada conversor.

O divisor de tensão (R13 e R12) possui uma relação de 5,27 para 1, serve para informar ao ATmega2560 através de uma entrada analógica (A0) com escala de 3,42 V a 4,78 V, o valor da tensão da bateria sendo que 25.2 V (bateria carregada) são convertidos para 4,78 V, e 3,42 V quando a bateria está com 18 V (bateria descarregada).

O Módulo de rádio RA-02 (RAD2) é o responsável pelas comunicações sem fios. À semelhança do que foi explicado anteriormente, na parte do emissor, este comunica com o ATmega2560 através do protocolo SPI (SCK-2, MISO-50, CS-3). Os pinos DIO, RESET e NSS estão ligados às portas digitais 3, 4 e 5 do ATmega2560. Como este módulo funciona com 3,3 V e o ATmega2560 com 5 V, foram adicionados dois conversores de nível lógico (CN3 e CN4).

O módulo de temperatura e humidade DHT11 (STH2) está ligado à entrada digital 2 do ATmega2560.

Os terminais 0DB25, 1DB25, 2DB25 e 3DB25 são usados como ponte para cada um dos quatro terminais fêmea DB25 (CDB25), juntamente com os terminais TP5 (estes fazem a ponte para todos os quatro terminais DB25). No esquema só foi representado um terminal DB25 (CDB25), de forma a se tornar mais fácil de interpretar. Contudo, os outros três terminais DB25 que não estão representados, são ligados aos terminais TP3, TP6 e TP7 e aos terminais comuns aos quatro (TP5). Cada um dos terminais DB25 (CDB25) é responsável por transferir alimentação, sinais e grandes correntes para as réguas, através de um cabo DB25.

O recetor apresenta quatro terminais DB25, sendo possível ligar quatro réguas de terminais (onde se ligam os ignitores).

O conjunto de terminais (TP4) está ligado ao HMI (NX2.8) e necessita dos 5 V, GND para alimentação, e dos pinos 2TX, 2RX para a comunicação.

O LED vermelho (D2) está conectado à saída digital 49 do ATmega2560. O LED será responsável por indicar ao operador se o modo de fogo está ativo.

Os 18V S/P indicam que são 18 V diretos da saída do conversor de tensão. Os 18V C/P indicam que possui uma chave que interrompe a tensão fornecida (esta chave quando fechada fornece 18 V).

O pino terminal (TP4) 18V S/P está ligado a um terminal da chave seletora (CH2) que se encontra no painel frontal do recetor, e o pino 18V C/P está conectado a outro pino dessa mesma chave. Desta forma pode fornecer os 18V C/P para as réguas, impedindo um disparo acidental por defeito do sistema quando este se encontra ligado. Basicamente quando é colocada a chave no modo de fogo (chave na posição 1), a chave fecha o circuito e o pino de 18V C/P fica com 18 V. Além disto, alimenta a base de um transístor (TR1) através de uma resistência de 10 k $\Omega$  (R5), fazendo com que este entre em saturação. O pino

emissor do transistor fica com aproximadamente 5 V, sendo que este está ligado na entrada analógica A14 do ATmega2560, com o objetivo de iniciar o modo de fogo. Ao ser iniciado, o HMI Nextion (NX2.8) é colocado em modo *sleep*, e o LED (D2) que indica ao utilizador se o modo fogo se encontra ativo é ligado.

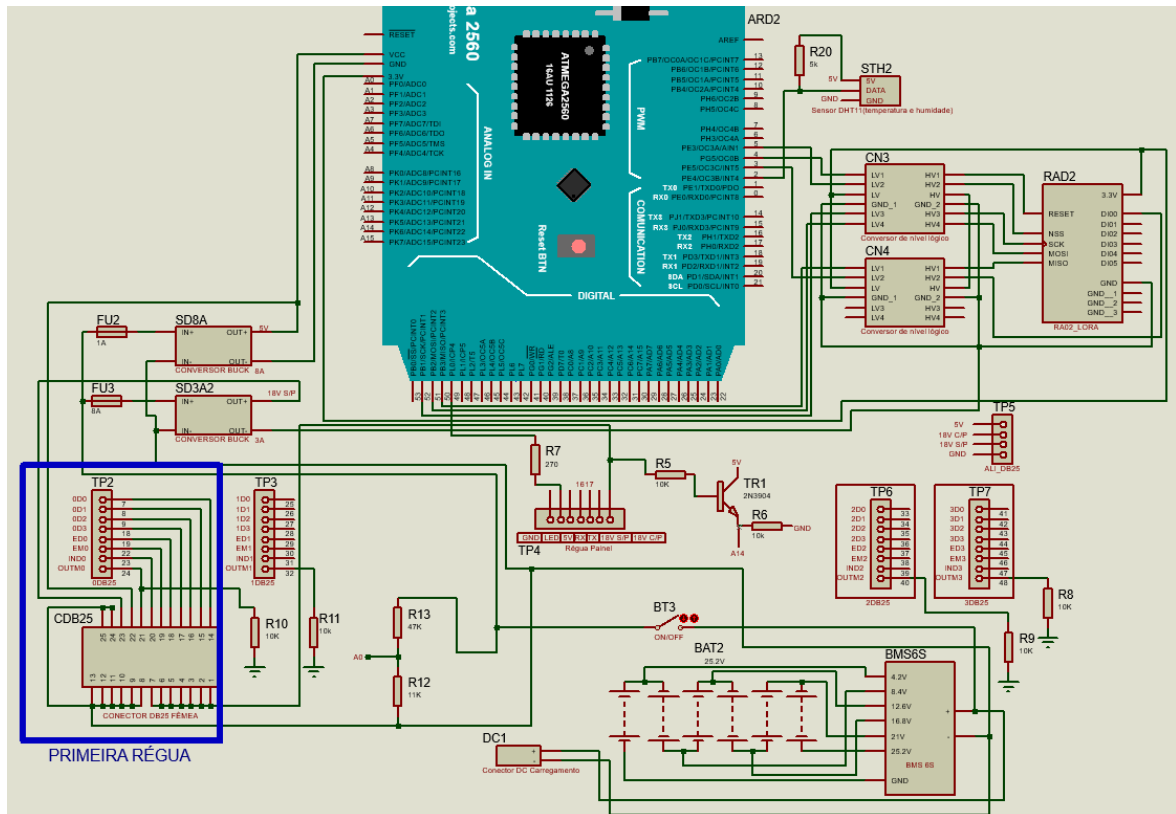


Figura 101. Esquema elétrico do módulo recetor.

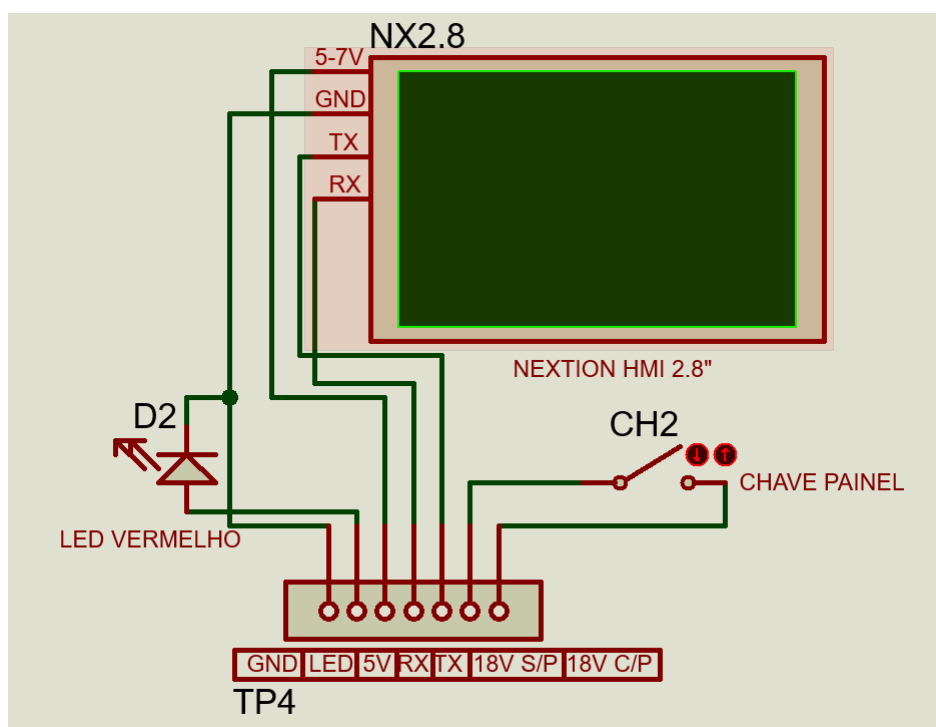


Figura 102. Esquema elétrico do painel do módulo recetor

### 5.2.3. Esquema elétrico da régua

Como referido anteriormente, em cada recetor é possível ligar até quatro régua de terminais (onde se ligam os ignitores).

Cada régua é composta por um conector DB25 fêmea; um *multiplexer* (MUX); um *demultiplexer* (DEMUX); dezasseis MOSFETs tipo P IRF9530; dezasseis transístors 2N3904; dezasseis LED's vermelhos; um LED verde; um botão; 48 resistências de 10 k $\Omega$ ; dezasseis resistências de 3,3 k $\Omega$ ; dezassete resistências de 270  $\Omega$ ; dezasseis resistências de 3,3 k $\Omega$  e blocos de terminais.

No esquema da Figura 104 só está representada um terminal de saída (para ligar um ignitor), que se encontra identificada dentro do retângulo azul, com a descrição "Primeiro Número", pois essa parte do esquema repete-se dezasseis vezes por régua, com a diferença da ligação ao MUX e DEMUX serem feitas por ordem nos pinos seguintes: E2, E3... E16 e S2, S3... S16.

Os pinos do conector DB25 (CDB25) estão ligados a cada pino do terminal (TP8). Os pinos D0 a D3 fazem a seleção de pinos no MUX (MX) e no DEMUX (DMX) através de uma combinação. O pino ED (*enable* DEMUX) e o borne EM (*enable* MUX) são responsáveis por colocar o DEMUX e o MUX em operação. O borne IND (*input* DEMUX), ligado ao pino sinal do DEMUX, serve para colocar o sinal de 5 V numa saída do DEMUX selecionada (5 V fornecidos por uma saída digital do Arduino do recetor). O borne OUTM (output MUX) ligado ao pino sinal do MUX, fornece o nível lógico alto ou baixo, consoante o estado da entrada selecionada. Os 5 V são usados para alimentação do MUX e DEMUX, como também para alimentar todos os LEDs presentes na régua.

O pino 18V S/P, que contém 18 V, é ligado através de uma resistência (R17) de 3,3 k $\Omega$  ao ânodo do LED interno do acoplador ótico (OPT), e o cátodo do LED é ligado a um dos terminais da régua (onde liga o ignitor). Caso exista um ignitor ligado, o LED acenderá, pois o outro terminal do ignitor está ligado a uma resistência de potência de 1  $\Omega$  (R19), conectada ao pino GND. O LED do acoplador ótico ao ligar dispara o fototransístor do acoplador ótico e este coloca os 5 V ligados no coletor, no emissor. Do lado do emissor, existe uma ligação a uma entrada do MUX (S1), colocando assim os 5 V nessa entrada. Este processo serve para indicar se existe um ignitor ligado quando o recetor fizer procura em todas as entradas do MUX. A resistência (R15) garante o nível baixo quando não existe

nenhum ignitor ligado naquela saída (*pull-down*). Os LEDs vermelhos (exemplo D4) permitem verificar se os ignitores estão bem ligados ou não, pois eles acendem com o disparo do transístor do acoplador ótico, e também ao pressionar o botão teste (BT5).

O botão teste (BT5) basicamente interrompe o GND(1) dos LEDs vermelhos de teste. Para além disto, existe um LED verde (D3) alimentado com 5 V, que indica se a régua está com a alimentação 5 V bem ligada. Este LED (D3) só acende quando se pressiona o botão de teste (BT5), pois o GND(1) também é controlado pelo botão.

O pino 18C/P que contém 18 V controlados pela chave do recetor (CH2), é usado para fazer o disparo do(s) ignitor(es) num dos terminais da régua (exemplo T1). O díodo (D1) é responsável por garantir que não há circulação de corrente para o transístor (TR1) quando a chave (CH2) não está na posição 1 (fechada). Cada saída do DEMUX está conectada, através de uma resistência de 10 k $\Omega$  (R14), à base de cada transístor (exemplo TR2). Quando na saída do DEMUX (exemplo E1) estiver com 5 V, o coletor do MOSFET (M1) fica com 0 V, o que fará com que a resistência interna do MOSFET entre o dreno e a *source* se torne praticamente nula, fornecendo assim aproximadamente 7,3 A, no máximo, ao(s) ignitor(es). De referir que existe uma resistência (R16) de 10 k $\Omega$  para garantir que a tensão  $V_{GS}$  (tensão entre a *gate* e a *source* do MOSFET) seja aproximadamente zero, isto faz com que o MOSFET fique estável.

Na régua o GND(1) é separado do GND, porque no momento do disparo, caso a resistência do terminal (onde se ligam um ou vários ignitores) seja muito baixa, aparecerá uma tensão ao terminal do GND da régua, e se o GND da parte do comando fosse compartilhado, iria fazer com que diminuísse a diferença de potencial, pondo em risco o correto funcionamento do circuito.

De referir que a corrente total de cada saída é limitada a aproximadamente 7,3 A de forma a que as células da bateria do recetor não sejam danificadas, uma vez que podem fornecer no máximo 8 A de forma contínua, segundo o fabricante. A limitação da corrente é feita através de uma resistência de potência em cada régua de 1  $\Omega$ , através da resistência interna do MOSFET ( $R_{DS(ON)} = 0,2 \Omega$ ) e da resistência do cabo DB25 2 m ( $\pm 0,8 \Omega$ ). A resistência restante das próprias ligações e soldas representa cerca de 0,3  $\Omega$  a 0,4  $\Omega$ . O conversor *step-down* (conversor *buck*) possui uma resistência interna de aproximadamente 0,05  $\Omega$ . Ou seja, pode-se contabilizar cerca de 2,4  $\Omega$  como resistência interna do dispositivo.

Na régua desenvolvida foi aplicado um díodo 10A10 nos 18V C/P, o que faz com que nos terminais da régua haja uma tensão de 17,5 V e uma corrente máxima de aproximadamente 7,3 A, devido à existência de uma queda de cerca de meio volt no díodo. Com esta tensão e esta corrente, teoricamente é possível num só terminal fazer a ignição de sete ignitores em paralelo e quinze ignitores em série (utilizando os ignitores J-TEK).

O MOSFET (M1) funciona nas zonas de saturação e corte, ou seja, quando é efetuado um disparo de um ignitor, o transístor (TR2) coloca o GND na gate do MOSFET, o que faz com que entre em saturação.

Para estar em corte,  $V_{GS} > V_{GS(th)}$ . Como  $V_{GS}$  é zero quando o transístor (TR2) não está em funcionamento e  $V_{GS(th)}$ , segundo o *datasheet* é cerca de -4 V, pode-se comprovar que nesta situação o mesmo se encontra no modo de corte.

Para estar em saturação,  $V_{GS} < V_{GS(th)}$  e  $V_{DS} \leq V_{GS} - V_{GS(th)}$ . Neste caso  $V_{GS}$  é aproximadamente no máximo -13,8 V,  $V_{DS}$  é aproximadamente -17,5 V (devido à queda no díodo (D1)) e  $V_{GS(th)}$  é de -4 V. Por isso pode-se provar que nestas condições o MOSFET se encontra na região de saturação (comporta-se como interruptor fechado). Foi considerado o valor máximo  $V_{GS}$  aproximado -13,8 V (valor verificado na simulação no software Proteus e comprovado com voltímetro no circuito testado numa *proto-board*) pois, como a resistência do ignitor é variável, e esta encontra-se no dreno do MOSFET, quando é praticamente zero (vários ignitores em paralelo na mesma saída), a queda de tensão no cabo que transporta os 18 V até à *source* dos MOSFETs é superior. Ou seja, quanto maior a resistência no terminal de saída (onde se ligam os ignitores), menor será a queda de tensão no cabo, sendo que  $V_{GS}$  se irá aproximar dos -17,5 V.

Como foi dito anteriormente, a resistência usada entre a *source* e a *gate* do MOSFET, de forma a mantê-lo estável entre os modos de funcionamento, é de 10 k $\Omega$ . Esta poderia até ter um valor mais elevado, uma vez que o controlo do MOSFET é feito por tensão. Contudo, sempre que fosse possível, foram utilizadas várias resistências de valores iguais, de forma a tornar o projeto mais económico.

Como o MOSFET é controlado através de tensão e a tensão de trabalho dele é superior à que o microcontrolador e o DEMUX suportam, foi utilizado um transístor bipolar para fazer esse controlo.

Quando é aplicado um sinal de 5 V na base do transistor (TR2), proveniente do DEMUX, o mesmo entra em saturação. Como a resistência na sua base é de 10 kΩ faz com que a corrente  $I_B$  (corrente da base) seja de 0,43 mA,  $I_B = \frac{U - U_{BE}}{R_B}$ , onde U é a tensão fornecida (5 V),  $U_{BE}$  é a queda de tensão entre a base e o emissor (cerca de 0,7 V) e  $R_B$  é a resistência da base (10 kΩ).

A corrente  $I_C$  (corrente no coletor) é de aproximadamente 1,8 mA,  $I_C = \frac{18 V}{10000 \Omega}$ .

Para o transistor trabalhar como chave foi necessária uma resistência de base menor do que 95 kΩ uma vez que o ganho do transistor é de 40. A resistência usada na base é de 10 kΩ, contudo podia ter-se usado uma resistência com um valor maior, no entanto, como foi dito acima, para tornar o projeto mais económico optou-se por utilizar resistências de valor igual.

Quando existe um ignitor num terminal da régua, passa uma corrente ( $I_F$ ) de cerca de 5 mA pelo LED do acoplador ótico (OPT), fazendo com que o ganho de corrente, segundo o datasheet seja perto de 480%. No fototransistor do acoplador ótico, a corrente do coletor é cerca de 9 mA.  $I_C = \frac{U_{Fonte} - U_{LED} - U_{CE}}{R_{LED}}$  onde  $U_{Fonte}$  é a tensão fornecida no coletor do fototransistor,  $U_{LED}$  é a queda de tensão do LED (cerca de 2,2 V),  $U_{CE}$  é a queda de tensão entre o coletor e o emissor do fototransistor (cerca de 0,25 V) e  $R_{LED}$  é a resistência que limita a corrente fornecida ao LED (270 Ω). Existe uma folga considerável, uma vez que o fabricante indica que seja ponderada uma perda de intensidade do LED ao longo dos anos.

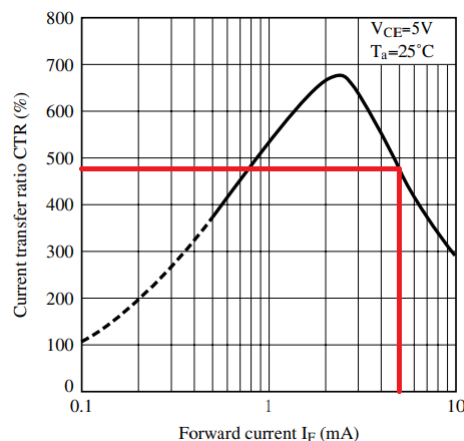


Figura 103. Relação de transferência de corrente (CTR) em função da corrente  $I_F$ , do acoplador ótico pc817 (Alterado da fonte: Farnell<sup>50</sup>)

<sup>50</sup> <https://www.farnell.com/datasheets/73758.pdf>



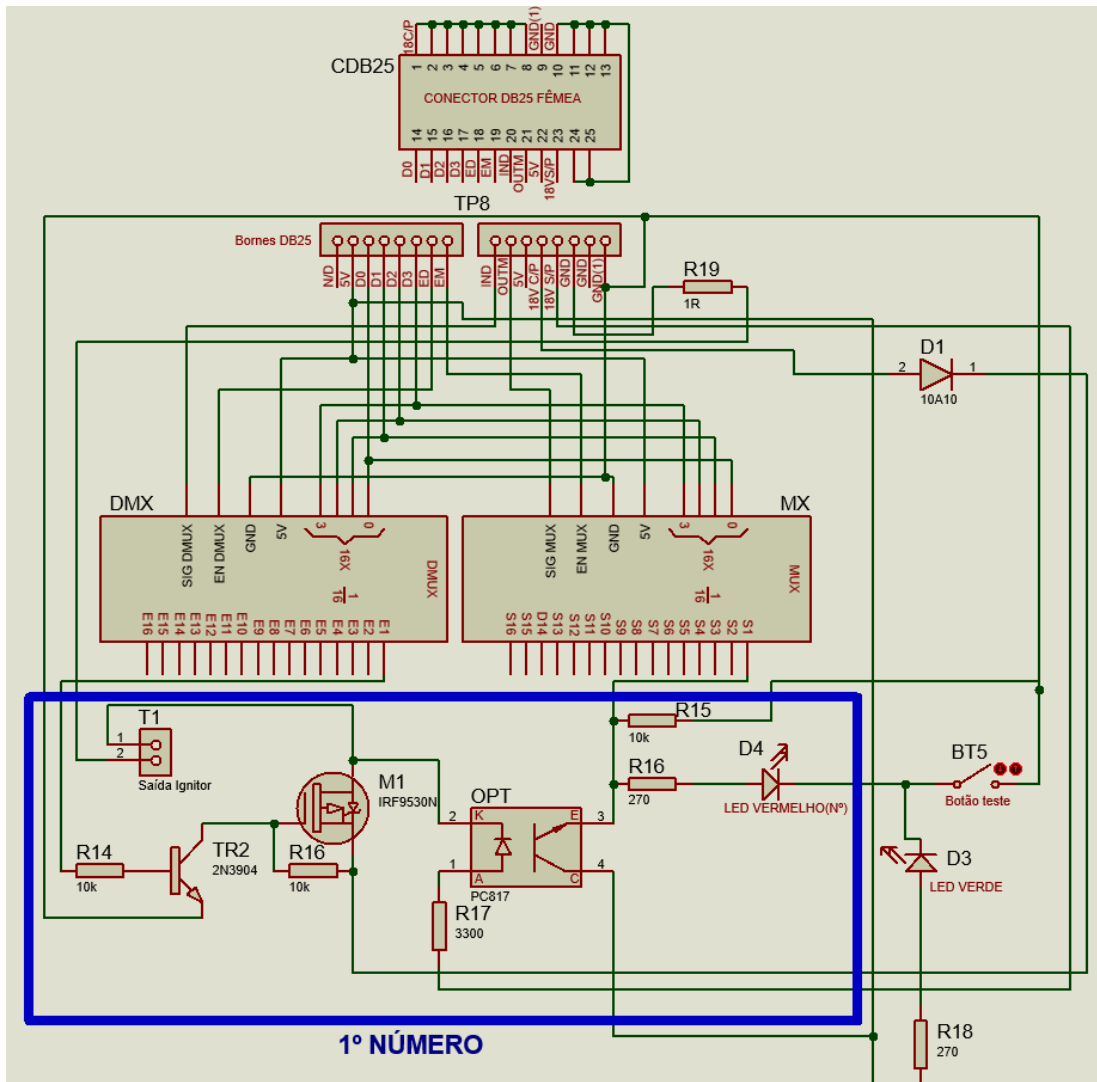


Figura 104. Esquema elétrico da régua (dentro do quadrado azul está representado o esquema de um número, os restantes 15 são repetidos, com a diferença de ligação a outros pinos dos MUX e DEMUX)

#### 5.2.4. Descrição de cada pino do conector DB25

O cabo usado para ligar as régua ao emissor é composto por 25 condutores. A Tabela 15 apresenta a função de cada condutor. Como é necessário transportar uma corrente alta pelo cabo, vários condutores transportam essa energia, de forma a dividir a corrente total por cada um deles. Ambas as pontas do cabo possuem conector fêmea. O recetor possui quatro conectores DB25 macho, de forma a poderem ser ligadas quatro régua. Cada régua possui um conector DB25 macho. Na Figura 105 é demonstrado o pinout do conector DB25 macho e fêmea.

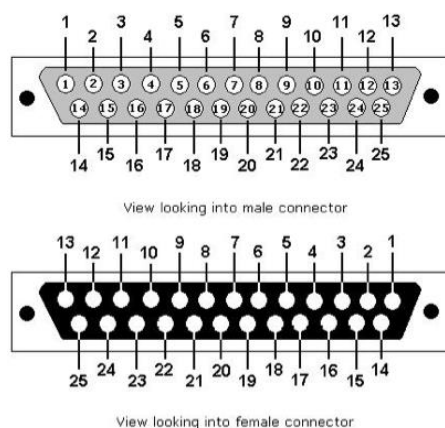


Figura 105. Pinout do conector DB25 fêmea e macho (fonte doityourselfchristmas<sup>51</sup>)

Tabela 15. Designação de cada pino do conector DB25

Pino	Função	Pino	Função
<b>1</b>	18V C/P	<b>14</b>	D0
<b>2</b>	18V C/P	<b>15</b>	D1
<b>3</b>	18V C/P	<b>16</b>	D2
<b>4</b>	18V C/P	<b>17</b>	D3
<b>5</b>	18V C/P	<b>18</b>	ED
<b>6</b>	18V C/P	<b>19</b>	EM
<b>7</b>	18V C/P	<b>20</b>	IND
<b>8</b>	GND	<b>21</b>	OUTM
<b>9</b>	GND	<b>22</b>	5 V
<b>10</b>	GND	<b>23</b>	18 V
<b>11</b>	GND	<b>24</b>	GND
<b>12</b>	GND	<b>25</b>	GND
<b>13</b>	GND		

<sup>51</sup> [https://www.doityourselfchristmas.com/wiki/images/5/5f/Db25\\_pinout.jpg](https://www.doityourselfchristmas.com/wiki/images/5/5f/Db25_pinout.jpg)

## 6. Conclusões e Trabalho Futuro

O sistema de ignição pirotécnica desenvolvido é composto por um emissor, recetor e régua e possui uma interface simples e fácil de operar.

A comunicação do emissor para o recetor contém um dos bytes do pacote encriptados por uma chave de 16 bytes, o que torna o sistema mais seguro.

O emissor possui três modos de disparo: automático, semiautomático e manual.

O modo semiautomático tem a possibilidade de disparar números apenas pressionando o botão "*Fire*" do painel. Os números de disparo são incrementados sempre que o botão é pressionado até ao valor máximo (64) sendo que depois, o canal é incrementado (de forma a disparar os próximos 64 números do recetor que estiverem com o mesmo canal definido). De referir que enquanto o botão não é libertado, é enviado o pacote de disparo a cada 100 ms. Neste modo é possível ainda mudar de canal (0-255) e o número de disparo através do HMI.

No modo manual são apresentados no HMI 32 botões com números de disparo (1-32), os quais o utilizador pode mudar os seus valores para 33-64. O canal (0-255) é configurável no HMI. Neste modo, ao ser pressionado um botão de disparo, é feito o disparo do número apresentado no botão. À semelhança do modo semiautomático, é possível mudar o canal. Os modos semiautomático e manual têm a possibilidade de disparar até 16384 números (256 canais x 64 números) de forma independente.

No modo automático é possível inserir os tempos de disparo de um espetáculo por cada canal + número de forma a que depois na altura do espetáculo, com apenas um clique no botão "*Fire*" seja possível o disparo coordenado de todos os artigos pirotécnicos, de forma automática de acordo com os tempos inseridos. Este modo suporta até quinze canais, o que possibilita o disparo de até 960 números de forma independente (15 x 64). Só é possível ter até quinze canais devido a limitações da memória RAM do microcontrolador. Em relação à inserção dos tempos, o emissor possui uma página dedicada, na qual é possível inserir e armazenar os tempos de disparo por cada conjunto canal + número. Estes tempos são armazenados num cartão SD, de forma a preservá-los mesmo que se desligue o emissor. A resolução mínima entre disparos é 0.2 s e a máxima é de 9999.8 s. Ou seja, é possível disparar todos os 64 números de um determinado recetor em apenas 12.6 s.

Durante o espetáculo é possível colocar o tempo em pausa e avançar para o próximo número.

Os modos de disparo e a gravação de tempos no cartão SD são protegidos por uma chave física de dois estados, localizada no painel do emissor. Caso a chave esteja na posição Zero, o acesso aos modos de disparo e gravação de tempos é negado, impedindo assim o manuseio por utilizadores não autorizados. Para proteção de toques acidentais, é apresentado um *slider* de proteção no acesso a todos os modos de disparo.

O sistema possui um modo de *SCAN*, que permite conhecer, a partir do emissor, informações sobre os recetores que estiverem devidamente configurados. No HMI do emissor são apresentadas informações dos recetores como a qualidade da comunicação, bateria, humidade, temperatura, canal configurado no recetor e o estado dos terminais (se possui terminal com conector ou não).

O módulo recetor tem a possibilidade de conectar até quatro régua de terminais (cada uma com dezasseis terminais). A ligação do recetor a cada régua é feita a partir de um cabo DB25.

O módulo recetor desenvolvido possui dois modos de trabalho: “modo seguro” e “modo fogo”. Estes modos são controlados através de uma chave física de dois estados incorporada no painel do recetor.

O “modo fogo” coloca o HMI em modo *sleep* (desliga o ecrã), liga um LED vermelho do painel, de forma a indicar o modo ativo, e fornece uma tensão de 18 V às régua nele ligado. Neste modo o recetor apenas trata de pacotes com ordens de disparo.

O emissor só consegue disparar um número do recetor caso o recetor esteja no “modo fogo”, caso o ID do emissor (remetente da mensagem) seja igual ao ID do emissor definido no recetor, caso o canal de disparo seja igual ao definido no recetor e caso a mensagem de 16 bytes descriptada esteja entre 0-255.

O “modo seguro” responde a pedidos vindos do emissor, ignorando, contudo, ordens de disparo. No HMI são apresentados os dados relativos à comunicação, como o ID do emissor ao qual ele irá responder, o ID local, e o canal.

O recetor só responde a pedido de informações do modo *SCAN* do emissor, caso esteja no “modo seguro”, caso o ID do emissor (remetente da mensagem) seja igual ao ID do emissor definido no recetor, caso o ID de destino do pacote seja igual ao ID do recetor e caso a mensagem de 16 bytes descriptada esteja entre 0-255.

Cada régua é composta por dezasseis terminais (aos quais são ligados os ignitores); caso o recetor esteja ligado, ao pressionar num botão “Test” da régua, acende-se um LED verde a indicar que a régua está energizada, e acendem também, cada um dos LED’s de cada terminal, caso possuam um ignitor conectado.

Em relação ao disparo, em cada terminal da régua a tensão e corrente máxima é de 17,5 V e 7,3 A.

Em relação à comunicação em campo aberto, é possível controlar os recetores, ou seja, realizar disparos pirotécnicos, até uma distância máxima de cerca de 6,27 Km.

O sistema pode beneficiar de melhorias em vários aspetos. Algumas são mais relevantes para o seu correto funcionamento, outras são vistas como “pequenas melhorias”.

Em relação às mais relevantes, é necessário realizar um estudo mais aprofundado acerca do comportamento das baterias (células 18650), de forma a desenvolver uma maneira de monitorizar e apresentar em ambos os módulos (emissor e recetor) a percentagem de bateria. Atualmente, a gestão das baterias é realizada por uma BMS genérica, o que faz com que elas se descarreguem até tensões perto da tensão de corte. Será necessário desenvolver um circuito de corte de alimentação que atue sempre que a tensão em cada célula seja inferior a 3 V (de forma a não esgotar completamente a sua capacidade). Este corte na alimentação fará com que se evite a degradação das células, caso estas fiquem paradas sem energia durante bastante tempo. Além disto, a ideia será, no futuro, apresentar uma mensagem a indicar a falta de bateria no ecrã quando cada célula tiver menos de 3 V.

Outro ponto fundamental é apresentar um valor concreto, em percentagem, da qualidade da transmissão sem fios em ambos os sentidos. Isto fará com que o sistema se torne mais fiável.

Em relação às melhorias não fundamentais estas incluem:

- Introduzir o sistema de disparo numa empresa pirotécnica de forma que esta o possa testar e dar a sua opinião, de modo a aperfeiçoá-lo e corrigir possíveis problemas.
- Estudo de todos os componentes utilizados nos módulos, de forma a conhecer o intervalo de temperatura para o qual cada um opera de forma correta. Com isto, através dos dados fornecidos pelo sensor de temperatura e humidade, aplicado em ambos os módulos, é possível apresentar ao utilizador uma mensagem de erro, caso algum componente esteja

a ser exposto a uma temperatura ou humidade fora do recomendado pelo fabricante. Com esta alteração, o sistema torna-se mais “inteligente” e autoprotege-se.

- Apresentar um resumo de tempos, na página de inserção de tempos do modo automático.
- Alterar a cor de todos os botões de disparo já pressionados no modo de disparo manual. Esta alteração fará com que o utilizador tenha noção dos números que já foram disparados.
- Estudo de outros cabos DB25 de diferentes tamanhos, de forma a manter sempre a mesma corrente de disparo, quando se altera o tamanho do cabo.
- Colocar opção de interface em inglês.

## Referências

- Ai-Thinker. (2017). *Ra-02 LoRa Module*. Obtido em 28 de Abril de 2020, de LCSC: [https://datasheet.lcsc.com/szlcsc/1811151537\\_Ai-Thinker-Ra-02\\_C90039.pdf](https://datasheet.lcsc.com/szlcsc/1811151537_Ai-Thinker-Ra-02_C90039.pdf)
- Antena Minho. (09 de Maio de 2020). *Covid-19: Festas canceladas provocam prejuízos de 2ME às empresas da pirotecnia*. Obtido em 8 de Setembro de 2020, de Antena Minho: <https://www.antenaminho.pt/noticias/covid-19-festas-canceladas-provocam-prejuizos-de-2me-as-empresas-da-pirotecnia/6977>
- ApogeeWeb. (2 de Março de 2018). *How Optocoupler Works and the Optocoupler Basics*. Obtido em 5 de Novembro de 2020, de ApogeeWeb: <https://www.apogeeWeb.net/article/69.html>
- Aqeel, A. (03 de 02 de 2018). *Introduction to MOSFET*. Obtido em 28 de Setembro de 2020, de The Engineering Projects: <https://www.theengineeringprojects.com/2018/02/introduction-to-mosfet.html>
- Bor, M., Vidler, J., & Utz Roeding. (15-17 de Fevereiro de 2016). LoRa for the Internet of Things. *International Conference on Embedded Wireless Systems and Networks*, (p. 362). Graz, Austria. Obtido em 16 de Setembro de 2020, de [https://www.researchgate.net/publication/297731094\\_LoRa\\_for\\_the\\_Internet\\_of\\_Things#fullTextFileContent](https://www.researchgate.net/publication/297731094_LoRa_for_the_Internet_of_Things#fullTextFileContent)
- Botnroll. (s.d.). *STEP-DOWN REGULÁVEL LM2596 10W*. Obtido em 20 de Outubro de 2020, de Botnroll: <https://www.botnroll.com/pt/conversores-dcdc/937-lm2596-step-down.html>
- CircuitRocks. (s.d.). *DC DC Step Down Mini 360 Power Supply Module MP2307DN*. Obtido em 6 de Julho de 2020, de CircuitRocks: <https://circuit.rocks/product:2314>
- COBRA Firing Systems. (s.d. a). *18R2 Controller*. Obtido em 22 de Outubro de 2020, de COBRA Firing Systems: [https://www.cobrafiringsystems.com/contact\\_us.html](https://www.cobrafiringsystems.com/contact_us.html)
- COBRA Firing Systems. (s.d. b). *Modules*. Obtido em 23 de Outubro de 2020, de COBRA Firing Systems: <https://www.cobrafiringsystems.com/modules.html>
- Components101. (23 de Novembro de 2017). *2N3904 - Transistor NPN*. Obtido de Components101: <https://components101.com/2n3904-pinout-datasheet>
- Dejan. (s.d.). Obtido em 16 de Junho de 2020, de DHT11 & DHT22 Sensors Temperature and Humidity Tutorial using Arduino: <http://howtomechatronics.com/tutorials/arduino/dht11-dht22-sensors-temperature-and-humidity-tutorial-using-arduino/>
- Desertcart. (s.d.). *WINGONEER XL4016E1 DC 4-40V to DC 1.25-36V 8A Buck Converter Voltage Regulator 36V 24V 12V to 5V High Power Efficiency Step Down Converter Power Supply*. Obtido em 17 de Setembro de 2020, de Desertcart: <https://www.desertcart.be/products/48439108-wingoneer-xl-4016-e-1-dc-4-40-v-to-dc-1-25-36-v-8-a-buck-converter-voltage-regulator-36-v-24-v-12-v-to-5-v-high-power-efficiency-step-down-converter-power-supply>
- Douro Pirotecnia. (s.d.). *Sobre nós*. Obtido em 20 de Outubro de 2020, de Douro Pirotecnia: <https://douropirotecnia.pt/sobre/>
- Engineering. (s.d.). *Battery Management Systems*. Obtido em 28 de Março de 2020, de Engineering: <https://www.engineering.com/productshowcase/batterymanagementsystems.aspx>

- Galaxis Showtechnik. (s.d. a). *PFC Advanced Black Edition*. Obtido em 14 de Outubro de 2020, de Galaxis Showtechnik: <https://www.firing-system.com/project/pfc-advanced-black-edition/>
- Galaxis Showtechnik. (s.d. b). *PFE Advanced 10 Outputs*. Obtido em 12 de Setembro de 2020, de Galaxis Showtechnik: <https://www.firing-system.com/project/pfe-advanced-10-outputs/>
- Galaxis Showtechnik. (s.d. c). *Matrix Module 50/100*. Obtido em 23 de Setembro de 2020, de Galaxis Showtechnik: <https://www.firing-system.com/project/matrix-module/>
- Grote. (s.d.). *Fuse & Circuit Protection*. Obtido em 12 de Maio de 2020, de Grote: <https://www.grote.com/technical-tips/fuse-circuit-protection-technical-tips/>
- International Rectifier. (s.d.). *International Rectifier IRF9530*. Obtido em 25 de Setembro de 2020, de International Rectifier: <http://www.irf.com/product-info/datasheets/data/irf9530.pdf>
- Liuyang happiness firing system factory. (s.d.). *Stage fountain & single shot firing system*. Obtido em 20 de Setembro de 2020, de Liuyang happiness firing system factory: <https://www.firingsystem.net/product/product-91-20.html>
- Luso. (12 de Março de 2019). *Empresas de pirotecnia defendem que Governo deve acabar com as proibições "cegas"*. Obtido em 5 de Agosto de 2020, de Diário de Notícias: [https://www.dn.pt/lusa/empresas-de-pirotecnia-defendem-que-governo-deve-acabar-com-as-proibicoes-cegas-10670734.html?fbclid=IwAR2COUMF2wGxU1Gga3VHhGBDQrysDtzif4ATA\\_VmrdsaHqDOIoE2n3PMpOA](https://www.dn.pt/lusa/empresas-de-pirotecnia-defendem-que-governo-deve-acabar-com-as-proibicoes-cegas-10670734.html?fbclid=IwAR2COUMF2wGxU1Gga3VHhGBDQrysDtzif4ATA_VmrdsaHqDOIoE2n3PMpOA)
- Meta Pyro. (s.d.). *Technical specifications*. Obtido em 15 de Outubro de 2020, de Meta Pyro: <http://www.meta-pyro.cz/technical-specifications>
- Murta, G. (17 de Agosto de 2018). <https://blog.eletrogate.com/guia-completo-do-arduino-mega/>. Obtido em 10 de Abril de 2020, de Eletrogate: <https://blog.eletrogate.com/guia-completo-do-arduino-mega/>
- Nextion. (s.d.). <https://nextion.tech/>. Obtido em 4 de Junho de 2020, de NEXTION INTRODUCTION: <https://nextion.tech/>
- ON Semiconductor. (s.d.). *LM2576*. Obtido em 7 de Setembro de 2020, de mouser: [https://pt.mouser.com/datasheet/2/308/LM2576\\_D-1810688.pdf](https://pt.mouser.com/datasheet/2/308/LM2576_D-1810688.pdf)
- P. Vignat, J. M. (2015). A technology shift for a fireworks controller.
- Poole, I. (s.d.). *Radio Link Budget: details & formula*. Obtido em 9 de Setembro de 2020, de Electronics Notes: <https://www.electronics-notes.com/articles/antennas-propagation/propagation-overview/radio-link-budget-formula-calculator.php>
- Pyro Firing System. (s.d.). *Fireworks firing system, 500M remote range transmitter, new wire control function, hot sale fireworks firing system, 96 cues expandable receivers*. Obtido em 10 de Setembro de 2020, de Pyro Firing System: [http://pyrofiringsystem.net/fireworks-firing-system500m-remote-range-transmitternew-wire-control-functionhot-sale-fireworks-firing-system96-cues-expandable-receivers\\_p0893.html](http://pyrofiringsystem.net/fireworks-firing-system500m-remote-range-transmitternew-wire-control-functionhot-sale-fireworks-firing-system96-cues-expandable-receivers_p0893.html)
- Pyrotechic Innovations. (s.d.). *Electrical Firing*. Obtido em 6 de Agosto de 2020, de Pyrotechic Innovations: <http://pyroinnovations.com/electricfire.html>
- S.C. Mukhopadhyay, G. S. (2008). Wireless Remote Controlled Solid-State Fireworks Detonation System. *Conference Record - IEEE Instrumentation and Measurement Technology Conference*.



- S.T.A. (s.d.). *Comparando Baterias Recarregáveis*. Obtido em 15 de Março de 2020, de S.T.A.: <https://www.sta-eletronica.com.br/artigos/baterias-recarregaveis/baterias-recarregaveis-em-geral/comparando-baterias-recarregaveis>
- SEMTECH. (s.d.). *SX1276/77/78/79*. Obtido em 17 de Junho de 2020, de SEMTECH: [https://semtech.my.salesforce.com/sfc/p/#E0000000JelG/a/2R0000001Rc1/QnUuV9TviODKUgt\\_rpBlPz.EZA\\_PNK7Rpi8HA5..Sbo](https://semtech.my.salesforce.com/sfc/p/#E0000000JelG/a/2R0000001Rc1/QnUuV9TviODKUgt_rpBlPz.EZA_PNK7Rpi8HA5..Sbo)
- SHARP. (30 de Setembro de 2003). *PC817X Series*. Obtido de Farnell: <https://www.farnell.com/datasheets/73758.pdf>
- SparkFun. (s.d.). *SparkFun Analog/Digital MUX Breakout - CD74HC4067*. Obtido em 3 de Novembro de 2020, de SparkFun: <https://www.sparkfun.com/products/9056>
- Sparkfun. (s.d.). *SparkFun Logic Level Converter - Bi-Directional*. Obtido em 23 de Agosto de 2020, de Sparkfun: [https://www.sparkfun.com/products/12009?\\_ga=2.100771845.500589515.1606682874-1494949694.1604665560](https://www.sparkfun.com/products/12009?_ga=2.100771845.500589515.1606682874-1494949694.1604665560)
- SparkFun. (s.d.). *SparkFun Logic Level Converter - Bi-Directional*. Obtido em 7 de Setembro de 2020, de SparkFun: [https://www.sparkfun.com/products/12009?\\_ga=2.100771845.500589515.1606682874-1494949694.1604665560](https://www.sparkfun.com/products/12009?_ga=2.100771845.500589515.1606682874-1494949694.1604665560)
- StarFire. (s.d.). *Basic Specifications*. Obtido em 20 de Junho de 2020, de StarFireSales: <http://starfiresales.com/specs.shtml>
- Texas Instruments. (Junho de 2020). *LM117, LM317-N Wide Temperature Three-Pin Adjustable Regulator*. Obtido de Texas Instruments: [https://www.ti.com/lit/ds/snvs774q/snvs774q.pdf?ts=1604105657762&ref\\_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FLM317-N%253FHQS%253DTI-null-null-alldatasheets-df-pf-SEP-ww%2526DCM%253Dyes%2526clid%253DCN\\_xlInP3ewCFfYFuQYddMAP6g](https://www.ti.com/lit/ds/snvs774q/snvs774q.pdf?ts=1604105657762&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FLM317-N%253FHQS%253DTI-null-null-alldatasheets-df-pf-SEP-ww%2526DCM%253Dyes%2526clid%253DCN_xlInP3ewCFfYFuQYddMAP6g)
- Velleman. (2 de Fevereiro de 2017). *VMA304*. Obtido em 12 de Agosto de 2020, de Velleman: [https://www.velleman.eu/downloads/29/vma304\\_a4v01.pdf](https://www.velleman.eu/downloads/29/vma304_a4v01.pdf)

## Apêndice A - Pseudocódigo referente ao programa dos módulos Emissor e Recetor

### Pseudocódigo referente ao programa do ATmega2560 do Emissor

```
// A função loop() é encarregue de percorrer o código constantemente (em loop). Nesta
função, caso algum evento associado aos objetos do HMI (botão por exemplo) seja recebido
(por série), é chamada a função associada ao evento do objeto.
loop()
{
// Leitura de todos os objetos relativos ao HMI. Caso seja recebido um evento, é chamada
a função associada ao objeto (botão por exemplo).
}

// Evento do clique do botão "+" (aumentar canal) da página "Modo Automático" (onde se
inserem e modificam os tempos relativos a cada conjunto canal + número).
void AutomaticoBtnCanalPlus()
{
// Lê os valores do tempo HMI e armazena-os em duas matrizes de tempo (globais).
// Apresenta no HMI o canal seguinte e atualiza o tempo relativo ao conjunto canal + número
armazenado nas duas matrizes de tempo, caso o canal atual não seja igual a 14.
}

// Função associada ao evento de clique no botão "-" (diminuir canal) da página "Modo
Automático" (onde se inserem e modificam os tempos relativos a cada conjunto canal +
número).
void AutomaticoBtnCanalMinus()
{
// Lê os valores do tempo HMI e armazena-os em duas matrizes de tempo.
// Apresenta no HMI o canal anterior e atualiza tempo relativo ao conjunto canal + número
armazenado nas duas matrizes de tempo, caso este não seja igual a zero.
}

// Função associada ao evento de clique no botão "+" (aumentar número) da página "Modo
Automático" (onde se inserem e modificam os tempos relativos a cada conjunto canal +
número).
void AutomaticoBtnNumerolPlus()
{
// Lê os valores do tempo HMI e armazena-os em duas matrizes de tempo.
// Apresenta no HMI o número seguinte e atualiza o tempo relativo ao conjunto canal +
número armazenado nas duas matrizes de tempo, caso não seja igual a 64.
}

// Função associada ao evento de clique no botão "-" (diminuir número) da página "Modo
Automático" (onde se inserem e modificam os tempos relativos a cada conjunto canal +
número).
void AutomaticoBtnNumeroMinus()
{
// Lê os valores do tempo HMI e armazena-os em duas matrizes de tempo.
// Apresenta no HMI o número anterior caso este não seja igual a zero e o atualiza o tempo
relativo ao conjunto canal + número armazenado nas duas matrizes de tempo.
}

// Função associada ao evento de clique no botão "Menu Principal" da página 2 "Modo
Automático".
void AutomaricoBtnMenuPrincipal()
{
// Chama a função MenuPrincipal() e retorna ao loop principal.
}

// Função associada ao evento de clique no botão "Modo Automático" da página "Menu
Principal". A função apresenta a página 2 com os dados dos ficheiros de tempo armazenados
no cartão SD.
void BtnModoAutomatico()
{
```

```

// É verificado se o cartão SD está inserido, caso não esteja, apresenta a página 10 com
uma mensagem de erro. Depois de um clique no botão “Para Sair Carregue aqui!”, chama a
função MenuPrincipal() e volta ao loop principal.
// Abre os ficheiros .txt que armazenam os tempos por cada conjunto canal + número
// Guarda esses tempos nas duas matrizes de tempo, uma de segundos e outra de décimas de
segundo.
// Apresenta a página 2 e atualiza e apresenta o tempo relativo ao canal 0 e número 1 de
acordo com os dados no índice [0][1] das duas matrizes de tempo.
}

// Função associada ao evento de clique no botão “Reset Tempos” na página 2 “Modo
Automático”. Esta função coloca todos os valores das matrizes de tempo a zero.
void BtnResetTempos()
{
// Percorre todos os índices das matrizes de tempo e coloca o seu valor a zero.
// Coloca a zero o valor do tempo apresentado na página 2.
}

// Função associada ao evento de clique no botão “Reset Tempos” da página 2 “Modo
Automático”. A função grava os tempos das matrizes de tempo em ficheiros .txt num cartão
SD.
void BtnGravarTemposNoCartao()
{
// Verifica se a chave física do painel está na posição um (nível alto), caso não esteja
apresenta uma mensagem de erro na página 10. Depois de pressionar o botão na página,
apresenta a página 2, atualiza os valores de tempo e volta ao loop principal.
// Guarda o valor do tempo (segundos e décimas de segundo) do canal + número apresentado
nas matrizes de tempo.
// Elimina os dois ficheiros .txt do cartão SD que armazenam os segundos e as decimas de
segundo por cada canal + número.
// Cria os dois ficheiros .txt que armazenam os tempos.
// Percorre as duas matrizes de tempo e guarda os valores em cada um dos ficheiros .txt.
}

// Função associada ao evento de clique no botão “Disparo” na página 2 “Modo Automático”.
void BtnDisparoAutomatico()
{
// Verifica se a chave física está na posição 1 (nível alto). Caso não esteja apresenta a
página 10 com um erro. Depois de pressionar o botão na página apresenta a página 2,
atualiza as variáveis de tempo de acordo com o canal 0 e número 1 e retorna ao loop
principal.
// Inicia o cartão SD e o módulo de rádio LoRa e caso haja um problema, apresenta a página
10 com uma mensagem de erro. Depois de pressionar o botão na página, volta para a página
2 e retorna ao loop principal.
// Apresenta a página 11 com o slider e, enquanto o valor do slider não estiver com o
valor 100, verifica o valor do slider.
// Acede aos ficheiros de tempo .txt do cartão SD, percorre os seus valores e transcreve-
os para as matrizes de tempo.
// Verifica os 2 primeiros índices e o último índice das matrizes de tempo que possuem
tempos diferentes de zero e guarda numa matriz os índices de cada um.
// Caso não exista pelo menos 2 índices com tempos diferentes de zero, apresenta uma
mensagem de erro na página 10. Depois de pressionar o botão na página, apresenta a página
2 com os tempos atualizados (canal 0 e número 1) e retorna ao loop principal.
// Apresenta a página 3 com os valores dos índices e tempo relativo ao primeiro conjunto
canal + número. Apresenta ainda os índices do conjunto canal + número seguinte com tempo
diferente de zero.
// Chama a função ModoAutomatico().
}

// Esta função apresenta a página 2 com tempos relativos ao canal 0 e número 1
void RetornoPaginaAutomatico()
{
// Apresenta a página 2 “Modo Automático” e atualiza o tempo relativo ao canal 0 e número
1 armazenado nas matrizes de tempo.
}

```

```

// Esta função faz a decremento do tempo por cada conjunto canal + número. Quando esse
tempo chegar a zero, envia uma mensagem com ordem de disparo para os recetores.
void ModoAutomatico()
{
// Muda o valor da variável global modo para 5 (é uma variável que é adicionada ao pacote
LoRa, serve para o recetor saber de que tipo de instrução é para executar quando recebe o
pacote).
// Entra dentro do while, e enquanto não for pressionado o botão de disparo "FIRE" ou o
botão do HMI "Menu Anterior", lê os estados de ambos os botões e acende e apaga o LED do
botão de disparo a cada 100ms.
// Se o botão "Menu Anterior" for pressionado, desliga o LED do botão de disparo e chama
a função RetornoPaginaAutomatico(). Após isto, o valor variável modo passa para 0 e depois
retorna ao loop principal.
// Desliga o LED do botão de disparo.
// Coloca invisível o botão "Menu Anterior".
// Coloca visível o botão "||" e o botão ">>|".
// Armazena nas variáveis X e Y os índices do primeiro conjunto canal + número que contenha
o tempo diferente de zero.
// Armazena nas variáveis Z e W os índices do segundo conjunto canal + número que contenha
o tempo diferente de zero.
// Entra dentro de um ciclo while().
    // Apresenta o tempo e o canal + número atual no ecrã de acordo com as variáveis X
    e Y.
    // Apresenta o canal e número seguinte no ecrã de acordo com as variáveis Z e W.
    // Guarda o índice da matriz de tempo que indica o canal atual, na variável global
    canal.
    // Chama a função sendMessage() e é passado o número de disparo como parâmetro de
    entrada.
    // Caso seja o ultimo conjunto canal + número com tempos diferentes de zero, sai
    da função, é chamada a função RetornoPaginaAutomatico(), muda o modo para 0 e depois
    volta ao loop principal.
    // Entra dentro de dois ciclos for(). Um para decrementar os segundos e outro as
    décimas de segundo de acordo com os valores contidos nos índices das duas matrizes
    de tempo.
    // Decrementa o valor do tempo associado ao canal + número atual de 200ms em 200ms
    e, apresenta uma atualização do tempo do tempo no ecrã (HMI).
    // Se o botão ">>|" for pressionado, coloca o tempo a zero, o que faz sair de ambos
    os ciclos for().
    // Se o botão "||" for pressionado, entra dentro de um ciclo while, onde fica a
    espera que o botão seja pressionado novamente.
    // Armazena o valor das variáveis Z e W nas variáveis X e Y.
    // Verifica nas matrizes de tempo e armazena nas variáveis Z e W qual é o próximo
    conjunto canal + número que contenha o tempo associado diferente de zero. (começa
    por verificar os números do canal atual, só depois procura os números do canal
    seguinte).
}

// Função responsável por formar e enviar o pacote LoRa com a variável recebida (encriptada)
e variáveis globais, como canal, ID pacote, ID do recetor e ID do emissor.
A função recebe um valor inteiro (para ser cifrado).
void sendMessage(int valor)
{
// Passa o valor recebido como parâmetro de entrada, de inteiro para string.
// Forma o pacote LoRa, com ID do pacote, ID do recetor, ID do emissor e canal.
// Chama a função TrataEncripta() e a string é lhe passada como parâmetro de entrada.
// Envia o pacote duas vezes seguidas.
// Incrementa o valor do ID do pacote.
}

// Função recebe uma String, transforma-a, encripta e adiciona os dados encriptados ao
pacote LoRa.
void TrataEncpripa(String aux)
{
// Percorre num for() cada índice da string e adiciona o valor de cada posição a um vetor
de bytes.

```

```

// Chama a função para encriptar o vetor de bytes. É lida passada como parâmetro de entrada,
a chave de 16 bytes e o vetor de bytes que se pretende encriptar com a chave.
// Percorre num ciclo for() cada índice do vetor de bytes encriptados e adiciona ao pacote
LoRa.
}

// Função associada ao evento de clique no botão "Semiautomático" na página 1 "Menu
Principal". Nesta função, apenas com um botão físico (FIRE), é feito um disparo
"independente" de cada número. A cada clique é enviada uma mensagem com ordem de disparo
para os recetores. Ao largar o botão o número apresentado no HMI é incrementado
automaticamente.
void SemiAutomatico()
{
// Verifica se a chave física está na posição 1 (nível alto), caso não esteja apresenta a
página 10 com um erro. Depois de pressionar o botão "Para sair carregue aqui!" volta para
a página 2 onde apresenta os tempos relativos ao canal 0 e número 1 e retorna ao loop
principal.
// Inicia o cartão SD e inicia o módulo de rádio LoRa e, caso haja um problema, apresenta
a página 10 com um erro. Depois de pressionar o botão "Para sair carregue aqui" na página
10 chama a função MenuPrincipal() e retorna ao loop principal.
// Apresenta a página 11 com o slide e, enquanto o valor do slider não estiver a 100,
verifica o valor do slider.
// Apresenta a página 4 "Modo Semiautomático" e atualiza o valor do número apresentado
para "1".
// Muda o valor da variável modo para 5 (modo é adicionado ao pacote LoRa, e indica
"disparo").
// Acende o LED do botão de disparo.
// Entra dentro de um ciclo infinito.
    // Enquanto o botão de disparo, não estiver com "nível alto", verifica o estado do
botão de disparo e o estado do botão no HMI "Menu Anterior".
    // Caso o valor do botão "Menu Anterior" tenha "nível alto" chama a função
MenuPrincipal(), muda o valor da variável modo para 0 e volta ao loop principal.
    // Armazena o valor do componente que apresenta o canal no HMI na variável global
canal.
    // Armazena o valor do componente que apresenta o número no HMI na variável número.
    // É chamada a função sendMessage() e é passado como parâmetro de entrada a variável
número.
    // Enquanto o botão de disparo não for largado, o LED acende e apaga de 100ms em
100ms, e é chamada a função sendMessage() onde o número é passado como parâmetro
de entrada.
    // Incrementa o valor do componente que apresenta o número no HMI, caso não seja
igual a 64. Caso seja igual, coloca a componente número do HMI a 1 e incrementa e
atualiza o canal no HMI se o canal não for igual a 255.
}

// Função associada ao evento clique do botão "SCAN" da página 8 "Info Recetores". Esta
função faz a procura e apresentação dos recetores encontrados de acordo com os ID's
inseridos.
void BtnSCAN()
{
// Inicia o cartão SD e o módulo de rádio LoRa, caso haja um problema, apresenta a página
10 com um erro. Depois de pressionar o botão "Para sair carregue aqui" na página 10 chama
a função MenuPrincipal() e retorna ao loop principal.
// Coloca invisível o botão "SCAN".
// Coloca invisível o botão "Menu Principal".
// Muda o valor da variável modo para 10 (a variável modo é acrescentada ao pacote LoRa).
// Faz um get aos componentes numéricos do HMI que contêm o ID mínimo e ID máximo dos
recetores que se pretende procurar.
// Remove do cartão SD o ficheiro dadosRecebidos.txt relativo ao armazenamento de
informações de recetores guardados anteriormente.
// Percorre num ciclo for() o intervalo de todos os valores de entre ID mínimo e ID máximo.
    // Guarda na variável global ID do recetor o valor do ID do recetor atual (do
ciclo).
    // Chama a função sendMessage() e passa-lhe um valor inteiro como parâmetro de
entrada (é feito isto apenas para melhorar a fiabilidade da comunicação, onde

```

```

    encriptação da variável é feita, de forma a poder ser decifrada no recetor e
    verificada se está entre 0 e 255).
    // Espera um tempo para que o recetor responda (200ms).
        // Enquanto não acaba o tempo chama a função onReceive() onde passa como
        parâmetro de entrada a função Lora.parsePacket() da biblioteca LoRa.h onde
        verifica se algum pacote é recebido.
    // Abre o ficheiro dadosRecebidos.txt com os dados dos ID's encontrados (que responderam).
    // Percorre as primeiras 6 linhas do ficheiro dadosRecebidos.txt e guarda os dados num
    vetor, as informações recebidas dos primeiros ID's.
    // Guarda numa variável global o número de páginas que são precisas para apresentar os
    ID's encontrados (uma vez que em cada página só aparecem informações de 6 ID's.
    // Apresenta a página 6 "Dados dos recetores".
    // Percorre a matriz com as informações dos ID's encontrados num for() e, em cada índice,
    transcreve os valores armazenados para cada componentes de texto da página 6.
    // Caso existam mais do que duas páginas, é colocado o botão "Próximos" visível.
    // Coloca a variável modo com o valor 0.
}

// Esta função recebe e trata os pacotes dos recetores recebidos pelo emissor.
void onReceive(int packetSize)
{
    // Verifica se o tamanho do pacote é igual a zero, se for retorna da função.
    // Verifica se o ID do recetor (recebido) é igual ao qual fez o pedido.
    // Lê os vários bytes (ID do emissor, ID do recetor, canal e bateria) recebidos pelo
    recetor e armazena-os em variáveis.
    // Faz a leitura do SNR e RSSI do pacote recebido e armazena em duas variáveis
    // Abre o ficheiro no cartão SD, dadosRecebidos.txt.
    // Armazena numa string separadas por vírgulas, todos as variáveis relativas aos dados
    recebidos.
    // Adiciona a string ao ficheiro dadosRecebidos.txt.
    // Fecha o cartão SD.
}

// Função associada ao evento clique no botão "Próximos" da página 6 "Dados dos recetores".
Faz a apresentação das variáveis dos próximos 6 ID's dos recetores encontrados.
void BtnProximos()
{
    // Incrementa a variável global que indica a página atual (de forma a existir uma orientação
    do microcontrolador).
    // Abre o ficheiro dadosRecebidos.txt no cartão SD.
    // Percorre os dados dos 6 ID's seguintes guardados no ficheiro e armazena-os num vetor
    do tipo string.
    // Percorre cada índice do vetor, e atualiza cada componente de texto do HMI.
    // Coloca visível o botão "Anteriores".
    // Caso não exista mais ID's seguintes o botão "Próximos" é colocado invisível.
}

// Função associada ao evento clique no botão "Próximos" da página 6 (Dados dos recetores).
Faz a apresentação das variáveis dos próximos 6 ID's dos recetores encontrados.
void BtnProximos()
{
    // Decrementa a variável global que indica a página atual (de forma a existir uma
    orientação).
    // Abre o ficheiro dadosRecebidos.txt no cartão SD.
    // Percorre os dados dos 6 ID's anteriores guardados no ficheiro e guarda-os num vetor do
    tipo string.
    // Percorre cada índice do vetor, e atualiza cada componente de texto do HMI
    // Coloca visível o botão "Próximos".
    // Caso não exista mais ID's anteriores o botão "Anterior" é colocado invisível.
}

// As seis funções seguintes, são associadas aos eventos clique sob uma das 6 caixas de
texto da página 6. Basicamente faz um envio de um pacote para os recetores, com o modo 15
(que indica, pedido de informações mais específicas do recetor). Aguarda um tempo e
apresenta a página "Dados do Recetor" com as informações do recetor recebidas.
void TxtCliqueUm()

```

```

{
// Faz um get e guarda num vetor do tipo char, os três primeiros caracteres (onde contém
o ID do recetor) da primeira caixa de texto com informações do recetor com um determinado
ID.
// Caso o primeiro caractere não seja um dígito retorna e volta para o loop principal.
// Concatena para uma string o valor de cada índice do vetor e depois converte para
inteiro.
// Guarda o valor do inteiro na variável global ID do recetor.
// Chama a função TratamentoTesteIndividual() e passa-lhe como parâmetro de entrada o
valor inteiro (ID do recetor).
}

void TxtCliqueDois()
{
// Faz um get e guarda num vetor do tipo char, os três primeiros caracteres (onde contém
o ID do recetor) da primeira caixa de texto com informações do recetor com um determinado
ID.
// Caso o primeiro caractere não seja um dígito retorna e volta para o loop principal.
// Concatena para uma string o valor de cada índice do vetor e depois converte para
inteiro.
// Guarda o valor do inteiro na variável global ID do recetor.
// Chama a função TratamentoTesteIndividual() e passa-lhe como parâmetro de entrada o
valor inteiro (ID do recetor).
}

void TxtCliqueTres()
{
// Faz um get e guarda num vetor do tipo char, os três primeiros caracteres (onde contém
o ID do recetor) da primeira caixa de texto com informações do recetor com um determinado
ID.
// Caso o primeiro caractere não seja um dígito retorna e volta para o loop principal.
// Concatena para uma string o valor de cada índice do vetor e depois converte para
inteiro.
// Guarda o valor do inteiro na variável global ID do recetor.
// Chama a função TratamentoTesteIndividual() e passa-lhe como parâmetro de entrada o
valor inteiro (ID do recetor).
}

void TxtCliqueQuatro()
{
// Faz um get e guarda num vetor do tipo char, os três primeiros caracteres (onde contém
o ID do recetor) da primeira caixa de texto com informações do recetor com um determinado
ID.
// Caso o primeiro caractere não seja um dígito retorna e volta para o loop principal.
// Concatena para uma string o valor de cada índice do vetor e depois converte para
inteiro.
// Guarda o valor do inteiro na variável global ID do recetor.
// Chama a função TratamentoTesteIndividual() e passa-lhe como parâmetro de entrada o
valor inteiro (ID do recetor).
}

void TxtCliqueCinco()
{
// Faz um get e guarda num vetor do tipo char, os três primeiros caracteres (onde contém
o ID do recetor) da primeira caixa de texto com informações do recetor com um determinado
ID.
// Caso o primeiro caractere não seja um dígito retorna e volta para o loop principal.
// Concatena para uma string o valor de cada índice do vetor e depois converte para
inteiro.
// Guarda o valor do inteiro na variável global ID do recetor.
// Chama a função TratamentoTesteIndividual() e passa-lhe como parâmetro de entrada o
valor inteiro (ID do recetor).
}

void TxtCliqueSeis()
{

```

```

// Faz um get e guarda num vetor do tipo char, os três primeiros caracteres (onde contém
o ID do recetor) da primeira caixa de texto com informações do recetor com um determinado
ID.
// Caso o primeiro caractere não seja um dígito retorna e volta para o loop principal.
// Concatena para uma string o valor de cada índice do vetor e depois converte para
inteiro.
// Guarda o valor do inteiro na variável global ID do recetor.
// Chama a função TratamentoTesteIndividual() e passa-lhe como parâmetro de entrada o
valor inteiro (ID do recetor).
}

// Esta função recebe o valor inteiro referente ao ID que se pretende saber informações
mais específicas. Envia a mensagem para o recetor e fica a escuta durante um tempo, pela
resposta do recetor.
void TratamentoTesteIndividual(int IDrecetor)
{
// Coloca vazia a variável global ConectadosEDesconectados (string) que armazena o valor
das saídas que tem continuidade (ignitor ligado).
// Coloca a variável modo com o valor 15 (de forma a que o recetor perceba que é para
responder com informações específicas).
// Chama a função sendMessage() e passa-lhe o valor da variável global ID do pacote atual
(é feito isto apenas para melhorar a fiabilidade da comunicação, onde encriptação da
variável é feita, de forma a poder ser decifrada no recetor e verificada se está entre 0
e 255).
// Aguarda 900ms (tempo para receber uma resposta do recetor) e enquanto não chegar a
900ms, chama a função RecebeDadosUnitarios() onde é passado como parâmetro de entrada a
função Lora.parsePacket() onde verifica se algum pacote é recebido.
// Coloca a variável modo com o valor 0.
}

// Esta função verifica se recebeu algum pacote pelo modulo de rádio e trata os dados
recebidos.
void RecebeDadosUnitarios()
{
// Verifica se recebeu um pacote, através verificação do tamanho do pacote. Caso seja
zero, sai da função.
// Faz a leitura e guarda as variáveis (bytes) do pacote recebido (ID do emissor, ID do
recetor, canal, temperatura, humidade e bateria).
// Faz a leitura e guarda os valores de SNR e RSSI do pacote recebido.
// Faz a leitura dos restantes 64 bytes do pacote recebido, e concatena cada byte numa
string. Cada byte pode conter o valor 0 ou 1 (onde indica se tem continuidade nos terminais
do recetor (réguas)). O primeiro byte corresponde à saída 1 e o último à saída 64.
// Percorre as 64 índices da string num for() e verifica se o valor armazenado no índice
na matriz é 0 ou 1. Se for 0 armazena numa string separado por um hífen o índice (este
indica o número do terminal sem continuidade). Caso seja 1 armazena numa string, o índice
separado por hífen (este indica o número do terminal com continuidade (possui ignitor)).
// Apresenta a página 7 “Dados do recetor” e atualiza os componentes numéricos e os
componentes de texto da página com os dados do recetor recebidos.
}

// Função associada ao evento clique no botão “Verificar novamente” da página 7 “Dados do
recetor”. Esta função faz um pedido novamente de informações mais específicas ao recetor
(de forma a atualizar as variáveis).
void BtnVerificarNovamente()
{
// Faz um get ao componente numérico da página que armazena o valor do ID do recetor e
guarda-o numa variável.
// Chama a função TratamentoTesteIndividual() e passa-lhe o valor da variável como parâmetro
de entrada (ID do recetor).
}

// Função associada ao clique do botão “Modo Manual” da página 1 “Menu Principal”. Percorre
uma lista de objetos (32 botões com número de disparo) e caso um evento seja recebido,
executa a função associada ao evento do objeto.
void BtnManual()
{

```



```

// Verifica se a chave física está na posição 1 (nível alto) e, caso não esteja apresenta
a página 10 com um erro. Depois de pressionar o botão “Para sair carregue aqui” na página
10 chama a função MenuPrincipal() e retorna ao loop principal.
// Inicia o cartão SD e o módulo de rádio LoRa, caso haja um problema, apresenta a página
10 com uma mensagem de erro. Depois de pressionar o botão “Para sair carregue aqui” na
página 10, chama a função MenuPrincipal() e retorna ao loop principal.
// Apresenta a página 11 com o slide e, enquanto o valor do slider não estiver com o valor
100, verifica o valor do slider.
// Apresenta a página 9 “Modo Manual”.
// Coloca a variável modo com o valor 5.
// Entra dentro de um ciclo while.
    // chama a função nexLoop() e passa-lhe como parâmetro de entrada a lista dos 32
    objetos (botões) a serem analisados.
    // Verifica o estado do botão “Menu Principal” e, caso seja 1, sai fora do ciclo.
// Apresenta a página 1 “Menu Principal”.
// Altera o valor da variável modo para 0.
// Coloca a variável global TrocaDeNumeros com o valor 0 (variável pode tomar valor 0 e
32 e é somada sempre ao valor do botão com o número de disparo. Serve para o programa
“saber” se o clique no botão 1 de disparo corresponde ao 1 ou 33.
}

// Função associada ao evento clique no botão “MUDAR 33-64”. Esta função troca o valor de
cada botão de disparo do modo manual e atualiza o texto do botão que foi pressionado para
“MUDAR 1-32”.
void BtnMudarValor()
{
// Verifica a variável global TrocaDeNumeros.
// Caso seja igual a 0, troca o texto individualmente apresentado pelos 32 botões, em vez
apresentar do 1 ao 32, passa a apresentar do 33 ao 64. Coloca a variável global
TrocaDeNumeros com o valor 32.
// Caso seja igual a 32, troca o texto individualmente apresentado pelos 32 botões e, em
vez de apresentar do 32 ao 64, passa a apresentar do 1 ao 32. Coloca a variável global
TrocaDeNumeros com o valor 0.
}

// Função associada ao evento clique de um dos 32 botões de disparo da página “Modo
Manual”. Esta função irá receber consoante o botão, um parâmetro de entrada de 1 a 32.
Depois é efetuado um envio de um pacote com ordem de disparo para os recetores.
void BtnDisparoManual(int numeroBotao)
{
// Faz um get ao componente que apresenta o canal no HMI e guarda-o numa variável.
// Guarda o valor dessa variável na variável global canal.
// Chama a função sendMessage() e passa-lhe como parâmetro de entrada, o valor recebido
somado pelo valor da variável TrocaDeNumeros (de forma a que se distinga por exemplo o
botão 1 é 1 ou 32).
}

// Função associada ao evento clique do botão “Configurações” da página 1 “Menu Principal”.
Esta função apresenta a página “Configurações” e atualiza os componentes que apresentam a
luminosidade e ID do emissor.
void BtmConfiguracoes()
{
// Apresenta a página 5 “Configurações”.
// Atualiza os componentes do HMI que apresentam a Luminosidade e ID do Emissor com as
variáveis globais do microcontrolador. O valor das variáveis globais é adquirido no
arranque do microcontrolador, através dos endereços de memória EEPROM. Verifica através
de uma função da biblioteca dht.h o valor da temperatura e humidade, e apresenta nos
componentes numéricos da página.
}

// Função associada ao evento clique do botão “Menu Principal” da página 5 “Configurações”.
Esta função verifica e guarda os novos valores dos componentes luminosidade e ID do
emissor.
void BtnMenuPrincipalC()
{
// Guarda o valor das variáveis globais em duas variáveis auxiliares.

```

```

// Faz um get ao componente numérico que apresentam o ID do emissor e luminosidade e guarda
o seu valor nas variáveis globais Luminosidade e ID do emissor.
// Caso novo valor da Luminosidade seja diferente do anterior (armazenado numa variável
auxiliar) guarda-o num endereço de memória da EEPROM.
// Caso novo valor do ID do emissor seja diferente do anterior (armazenado numa variável
auxiliar) guarda-o num endereço de memória da EEPROM.
// Chama a função MenuPrincipal().
}

// Função analisa o valor da tensão bateria, apresenta a página “Menu Principal” e atualiza
o valor da tensão da bateria apresentado.
void MenuPrincipal()
{
// Analisa e guarda numa variável o valor da porta analógica no qual um divisor de tensão
está conectado (de forma a reduzir a tensão da bateria, na entrada do microcontrolador).
// O valor da variável é convertido para o valor real, através da multiplicação do valor
por 12.6 (uma vez que a razão do divisor de tensão utilizado é de 12.6 para 4.45)
// Divide a variável por 911 (4.45 V numa entrada analógica de 10 bits e 5 V de resolução,
corresponde a 911).
// Apresenta a página 1 “Menu Principal”
// Atualiza o valor do componente numérico que apresenta a tensão da bateria com o valor
da variável analógica que foi tratada.
}

```

### Pseudocódigo referente ao programa do ATmega2560 do Recetor

```

// A função loop() é encarregue de percorrer o código constantemente (em loop). Nesta
função, caso algum evento associado aos objetos do HMI (botão por exemplo) seja recebido,
é chamada a função associada ao evento. O estado da chave física também é monitorizada e,
caso esteja na posição 1 o modo fogo (modoFogo) é ativo. São também verificados todos os
pacotes recebidos.
void loop()
{
// Leitura de todos os objetos relativos ao HMI. Caso seja gerado um evento num deles, é
chamada a função associada.
// É chamada a função onReceive() onde é passado como parâmetro de entrada a função
LoRa.parsePacket().
// Analisa o estado da chave física de proteção.
// Caso o estado da chave seja igual a um.
    // É a variável global modoFogo é colocada verdadeira.
    // É colocada a saída que ativa o LED vermelho do painel com o nível alto (5V).
    // É colocado o HMI no modo “sleep” onde desliga o LCD e entra em modo de economia.
        // Entra dentro de um ciclo while e, enquanto a chave estiver com o valor
        1, chama constantemente a função função onReceive() onde é passada como
        parâmetro de entrada a função LoRa.parsePacket().
    // É a variável global modoFogo é colocada falsa.
    // É colocada a saída que ativa o LED vermelho do painel com o nível baixo (0V).
    // É retirado o modo “sleep” do HMI.
}

// Esta função verifica se recebeu um pacote. Caso receba, analisa cada um dos bytes do
pacote de modo a verificar se o pacote tinha como destino o recetor em questão. Caso seja
executa a função associada de acordo com a variável modo recebida.
void onReceive(int packetSize)
{
// Se o tamanho do pacote for diferente de 21, sai da função e volta ao loop principal.
// Lê as variáveis (bytes) recebidas e guarda-as nas variáveis globais (IDpacoteAtual, ID
do recetor, ID do emissor).
// Se o IDpacoteAnterior for igual do IDpacoteAtual ou ID do emissor for diferente do ID
do emissor predefinido no dispositivo.
    // Sai da função e volta ao loop principal.
// Lê as variáveis (bytes) recebidas e guarda-as (canal e modo).
// Lê os restantes bytes recebidos e armazena-os num vetor de bytes.
// Chama a função DecifraDados() onde é passado como parâmetro de entrada uma chave de
16 bytes e o vetor de bytes recebido. A função descripta o vetor.
// Percorre o vetor descriptado num for().
}

```

```

        // Converte os dados para char e adiciona a uma String.
        // Se o caractere não é um dígito ou um espaço vazio, sai da função e retorna ao
        loop principal.
// Converte a String para um inteiro.
// Se o inteiro não contiver um valor entre 0 e 255, sai da função e retorna ao loop
principal.
// Armazena na variável IDpacoteAnterior, o valor do IDpacoteAtual.
// Se o modo for igual a 5, o canal predefinido no dispositivo for igual ao recebido no
pacote e o modoFogo for verdadeiro
    // Chama a função Disparo() e passa-lhe o inteiro que foi decifrado como parâmetro
de entrada.
// Se o modo for igual a 10, o ID recetor predefinido no dispositivo for igual ao ID
recetor recebido e o modoFogo for falso.
    // Chama a função EstadoBateria().
    // Forma um pacote lora com algumas variáveis do dispositivo recetor (ID do emissor
(local), ID do recetor (local), canal (Local) e bateria (estado).
    // Envia o pacote e retorna ao loop principal.
// Se o modo for igual a 15, o ID do recetor predefinido no dispositivo for igual ao ID
do recetor recebido e o modo fogo for falso.
    // Chama a função analise.
    // Faz a leitura dos valores de temperatura e humidade e guarda-os em duas variáveis.
    // Chama a função EstadoBateria().
    // Forma um pacote lora com algumas variáveis do dispositivo recetor (ID do emissor
(local), ID do recetor (local), canal (Local), Temperatura, Humidade, bateria
(estado) e os dados (de uma string) relativos à continuidade de cada um dos 64
terminais das réguas ligadas ao recetor.
    // Envia o pacote e retorna ao loop principal.
}

// Função disparo, recebe como parâmetro de entrada um byte que indica o número de disparo.
Faz o controlo dos demultiplexers (demux) de forma a seleccionar o terminal pretendido, e
coloca um nível alto na entrada do demux, o que faz aparecer esse nível alto no terminal
seleccionado por um curto período de tempo.
void Disparo(byte numeroDis)
{
// Um switch/case recebe o número de disparo.
    // Cada uma das condições (cases) possui o controlo da saída de um demux através
de 4 pinos de seleção. Existe num total 4 demux (4 réguas), contudo a seleção de
portas em cada demux é feita individualmente (existem no total 16 pinos apenas
dedicados a controlo dos 4 demux).
// Se o numeroDis (número de disparo) estiver entre 1 e 16.
    // Coloca nível baixo no pino de enable do Demux correspondente (o que faz com que
entre em funcionamento.
    // Coloca na entrada do Demux correspondente nível alto (que por sua vez coloca
esse nível na saída seleccionada no switch case.
    // Espera 50ms (tempo mais do que suficiente para o ignitor ser ativo).
    // Retira o nível alto da entrada do Demux.
    // Coloca nível alto no pino enable do Demux.
// Se for de 17 a 32, 33 a 48 ou de 49 a 64 faz o processo idêntico ao último "Se".
}

// Função Analise. Esta função percorre cada uma das 16 entradas de cada um dos 4
multiplexer (mux) e armazena numa String os dados relativos a cada entrada do multiplexer.
O estado das entradas de cada mux indica se existe continuidade (se possui ignitor) ou não
(através do 0 - sem continuidade e 1 - com continuidade).
void Analise()
{
// Percorre num ciclo for() as 64 entradas dos mux.
    // Em cada iteração do ciclo, são feitas combinações com os pinos de seleção de
entradas do mux.
    // Ativa o mux correspondente (colocando um nível baixo).
    // Analisa o estado da saída do mux (a saída do mux representa o sinal da entrada
seleccionada).
    // Concatena o estado numa string.
    // Desativa o mux correspondente (colocado um nível alto).
}

```

```

// Função associada ao evento clique sob a página 1 (que apresenta o nome da aplicação).
Nesta função é feita a apresentação da página 1 (página principal) e atualiza os dados
apresentados na página com os dados guardados em variáveis globais.
void BtnPaginaApresentacao()
{
// Apresenta a página 1.
// Atualiza os valores dos componentes do HMI com o valor do canal, ID do emissor, ID do
recetor (estas variáveis são globais e o seu valor provém de endereços de memória EEPROM).
// Chama a função EstadoBateria().
// Atualiza o valor do componente no HMI que apresenta a tensão da bateria.
}

// Função associada ao evento de clique do botão “Configurações” da página 1. Esta função
apresenta a página 5 “Configurações” e atualiza os dados dos componentes presentes na
página.
void BtnConfigura()
{
// Apresenta a página 5 “Configurações”.
// Atualiza o valor dos componentes que apresentam a luminosidade, com o valor da variável
global Luminosidade.
}

// Função associada ao evento de clique no botão “Alterar” o canal na página 1. Esta função
apresenta a página 2 “Alterar Canal” e atualiza o valor do componente da página.
void BtnAlteraCanal()
{
// Apresenta a página 2 “Alterar Canal”.
// Atualiza o valor do componente que apresenta o canal, com o valor da variável global
canal. O valor desta variável provem no arranque do programa de um endereço de memória
EEPROM do microcontrolador.
}

// Função associada ao evento de clique no botão “Alterar” o ID do recetor na página 1.
Esta função apresenta a página 4 “Alterar ID recetor” e atualiza o valor do componente da
página.
void BtnAlteraIDrecetor()
{
// Apresenta a página 4 “Alterar ID recetor”.
// Atualiza o valor do componente que apresenta o ID do recetor, com o valor da variável
global ID do recetor. O valor desta variável provem no arranque do programa de um endereço
de memória EEPROM do microcontrolador.
}

// Função associada ao evento de clique no botão “Alterar” o ID do emissor na página 1.
Esta função apresenta a página 3 “Alterar ID emissor” e atualiza o valor do componente da
página.
void BtnAlteraIDemissor()
{
// Apresenta a página 3 “Alterar ID emissor”.
// Atualiza o valor do componente que apresenta o ID do emissor, com o valor da variável
global ID emissor. O valor desta variável provem no arranque do programa de um endereço
de memória EEPROM do microcontrolador.
}

// Função associada ao evento de clique no botão “Sair e guardar” na página 2 “Alterar
Canal”. Esta função guarda o valor do componente canal, na variável canal.
void BtnSairPcanal()
{
// Guarda numa variável auxiliar o valor da variável global canal.
// Faz um get ao componente do HMI que apresenta o canal, e guarda-o na variável global
canal.
// Se a variável global que armazena o canal for diferente da variável auxiliar.
// Guarda num endereço de memória EEPROM, o valor do canal.
// Chama a função MenuPrincipal().
}

```

```

// Atualiza o valor do componente que apresenta o canal, com o valor da variável global
canal.
}

// Função associada ao evento de clique no botão “Sair e guardar” na página 4 “Alterar ID
emissor”. Esta função guarda o valor do componente ID emissor do HMI, na variável ID
emissor.
void BtnSairPcanal()
{
// Guarda numa variável auxiliar o valor da variável global ID do emissor.
// Faz um get ao componente do HMI que apresenta o ID emissor, e guarda-o na variável
global ID emissor.
// Se a variável global que armazena o canal for diferente da variável auxiliar.
// Guarda num endereço de memória EEPROM, o valor do ID do emissor.
// Chama a função MenuPrincipal().
// Atualiza o valor do componente que apresenta o ID do emissor, com o valor da variável
global ID do emissor.
}

// Função associada ao evento de clique no botão “Sair e guardar” na página 3 “Alterar ID
do recetor”. Esta função guarda o valor do componente ID do recetor do HMI, na variável ID
do recetor.
void BtnSairPcanal()
{
// Guarda numa variável auxiliar o valor da variável global ID recetor.
// Faz um get ao componente do HMI que apresenta o ID do recetor, e guarda-o na variável
global ID do recetor.
// Se a variável global que armazena o canal for diferente da variável auxiliar.
// Guarda num endereço de memória EEPROM, o valor do ID do recetor.
// Chama a função MenuPrincipal().
// Atualiza o valor do componente que apresenta o ID do recetor, com o valor da variável
global ID do recetor.
}

// Função associada ao evento de clique no botão “Sair e guardar” na página 5
“Configurações”. Esta função guarda o valor do componente que apresenta luminosidade do
HMI, na variável Luminosidade.
void BtnSairPcanal()
{
// Guarda numa variável auxiliar o valor da variável global luminosidade.
// Faz um get ao componente do HMI que apresenta o valor da luminosidade, e guarda-o na
variável global Luminosidade.
// Se a variável global que armazena o canal for diferente da variável auxiliar.
// Guarda num endereço de memória EEPROM, o valor da variável Luminosidade.
// Chama a função MenuPrincipal().
// Atualiza o valor do componente que apresenta o valor da luminosidade, com o valor da
variável global Luminosidade.
}

// Esta função analisa a tensão da bateria e atualiza o valor numa variável.
void EstadoBateria()
{
// Analisa e guarda na variável bateria o valor do estado da entrada analógica no qual um
divisor de tensão está conectado.
// O valor da variável global bateria é convertida para o valor real, através da
multiplicação do valor por 25.2 (uma vez que a razão do divisor de tensão utilizado é de
25.2 para 4.78) e depois dividido por 979 (4.78V numa entrada analógica de 10 bits de
resolução corresponde a 979).
}

// Função analisa o valor da tensão bateria, apresenta a página “Menu Principal” e atualiza
o valor que apresenta a tensão da bateria no ecrã.
void MenuPrincipal()
{
// Chama a função EstadoBateria().
// Apresenta a página 1 “Página Principal”.
}

```

```
// Atualiza o valor do componente numérico que apresenta a tensão da bateria com o valor
da variável global bateria.
}

// Esta função chama a função da biblioteca AESLib.h que decifra dados. Ela recebe como
parâmetros de entrada uma chave de 16 bytes e um vetor de 16 bytes a ser decifrado.
void DecifraDados()
{
// Chama a função aes128_dec_single() onde é lhe passada como parâmetro de entrada, uma
chave de 16 bytes e um vetor de 16 bytes a ser decifrado.
}
```

**Apêndice B - Características do HMI Nextion 7" Enhanced Series<sup>52</sup>**

<b>Características</b>	<b>Descrição</b>
<b>Referência</b>	NX8048K070 _011R
<b>Fabricante</b>	Nextion
<b>Tamanho do Display</b>	7"
<b>Resolução do Display</b>	800x480 pixel
<b>Área Visual</b>	154,08 mm(L)×85,92 mm(W)
<b>Cor</b>	64K 65536 color, 16 bit 565, 5R-6G-5B
<b>Brilho ajustável</b>	0 ~ 230 nits, com intervalo de ajuste de 1% 096958473625
<b>Capacidade multi-touch</b>	Não
<b>Tipo de touch screen</b>	Resistivo
<b>Controlador dedicado</b>	Sim
<b>Tensão de alimentação</b>	5 V DC
<b>Corrente de alimentação</b>	2 A
<b>Corrente de pico (Brilho 100%)</b>	510 mA
<b>Corrente de mínima</b>	15 mA
<b>Número de pinos</b>	4
<b>Dimensões</b>	181(L)×108(W)×9,3(H)
<b>Peso</b>	598 g
<b>Velocidade de operação</b>	≤115200 bps
<b>Memória Flash</b>	32 MB
<b>Memória RAM</b>	8192 Bytes
<b>Tipos de interface</b>	TTL Serial
<b>Página do fabricante</b>	<a href="https://nextion.itead.cc/">https://nextion.itead.cc/</a>

---

<sup>52</sup> <https://nextion.tech/datasheets/nx8048k070/>

**Apêndice C - Características do HMI Nextion 2.8" Basic Series<sup>53</sup>**

<b>Características</b>	<b>Descrição</b>
<b>Referência</b>	NX3224T028
<b>Fabricante</b>	Nextion
<b>Tamanho do Display</b>	2,8"
<b>Resolução do Display</b>	320×240 pixel
<b>Área Visual</b>	57,6 mm(L)×43,2 mm(W)
<b>Cor</b>	64K 65536 color, 16 bit 565, 5R-6G-5B
<b>Brilho ajustável</b>	0 ~ 180 nit, com intervalo de ajuste de 1%
<b>Capacidade multi-touch</b>	Não
<b>Tipo de touch screen</b>	Resistivo
<b>Controlador dedicado</b>	Sim
<b>Tensão de alimentação</b>	5 V DC
<b>Corrente de alimentação</b>	500 mA
<b>Corrente de pico (Brilho 100%)</b>	65 mA
<b>Corrente mínima</b>	15 mA
<b>Número de pinos</b>	4
<b>Dimensões</b>	85(L)×49,8(W)×5,8(H)
<b>Peso</b>	29.5 g
<b>Velocidade de operação</b>	≤115200 bps
<b>Memória Flash</b>	4 MB
<b>Memória RAM</b>	3584 Bytes
<b>Tipos de interface</b>	TTL Serial
<b>Página do fabricante</b>	<a href="https://nextion.itead.cc/">https://nextion.itead.cc/</a>

---

<sup>53</sup> <https://nextion.tech/datasheets/nx3224t028/>



#### Apêndice D - Características do Arduino Mega 2560 Pro mini<sup>54</sup>

<b>Microcontrolador</b>	ATmega2560
<b>Tensão operacional</b>	5 V DC
<b>Tensão de entrada (recomendado)</b>	7-12 V DC
<b>Tensão de entrada (limite)</b>	6-20 V DC
<b>Portas Digitais I/O</b>	54 (15 oferecem saída PWM ( <i>Pulse-Width Modulation</i> ))
<b>Pinos de entrada analógicos</b>	16
<b>Velocidade do clock</b>	16 MHz
<b>Memória Flash</b>	256 kB
<b>Comprimento</b>	55mm
<b>Largura</b>	38mm
<b>Peso</b>	17g

---

<sup>54</sup>[https://www.amazon.es/gp/product/B07MQ1J9MR/ref=ppx\\_yo\\_dt\\_b\\_asin\\_title\\_o05\\_s00?ie=UTF8&psc=1](https://www.amazon.es/gp/product/B07MQ1J9MR/ref=ppx_yo_dt_b_asin_title_o05_s00?ie=UTF8&psc=1)

**Apêndice E - Tabela de designação e descrição de cada componente dos esquemas elétricos do módulo emissor (Figura 99 e Figura 100)**

Designação do componente no esquema elétrico	Componente
ARD1	Arduino 2560 Pro Mini (Microcontrolador ATmega2560)
STH1	DHT11 (Sensor de Temperatura e Humidade)
R1	Resistência 10 K $\Omega$
R2	Resistência 10 K $\Omega$
TP1	Bloco de 8 Terminais
SD1	Módulo para Cartões SD (Velleman VMA304)
CNV1	Conversor de Nível Logico de 5 V para 3,3V Bidirecional (Sparkfun BOB-12009)
CNV2	Conversor de Nível Logico de 5 V para 3,3 V Bidirecional (Sparkfun BOB-12009)
RAD1	Módulo de Rádio (Ai-Thinker RA-02)
SD3A1	Conversor Buck 3 A (MP2307DN)
BAT1	Conjunto de 3 células INR18650-35E
R3	Resistência 33 k $\Omega$
R4	Resistência 18 k $\Omega$
BT1	Botão ON/OFF
FU1	Fusível de proteção 600 mA
BT2	Botão de pressão
CH1	Chave de 2 estados do painel
D1	Led Vermelho (do botão BT2)
NX7	HMI NEXTION NX8048K070-011
BMS3S	BMS 3S
R19	Resistência 5 k $\Omega$

**Apêndice F - Tabela de designação e descrição de cada componente dos esquemas elétricos do módulo recetor e régua (Figura 100, Figura 101 e Figura 103)**

Designação do componente no esquema elétrico	Componente
CDB25	Conector Fêmea DB25
FU2	Fusível 1 A
FU3	Fusível 8 A
SD8A	Conversor Buck 18 V 8 A (Módulo XH-M401)
SD3A2	Conversor Buck 5 V 3 A (Módulo LM2596)
R10	Resistência 10 k $\Omega$
R11	Resistência 10 k $\Omega$
R13	Resistência 47 k $\Omega$
R12	Resistência 11 k $\Omega$
R7	Resistência 270 $\Omega$
ARD2	Arduino 2560 Pro Mini (Microcontrolador ATmega2560)
DC1	Conector DC fêmea (5,5 mm x 2,1 mm)
BT3	Botão ON/OFF
BAT2	Conjunto 6 células Panasonic NCR18650B
BMS6S	BMS 6S (SEM DADOS DO FABRICANTE)
TR1	Transistor NPN 2N3904
R6	Resistência 10 k $\Omega$
R5	Resistência 10 k $\Omega$
STH2	Sensor DHT11 (temperatura e humidade)
CN3	Conversor de Nível Lógico 3,3-5 V bidirecional (Sparkfun BOB-12009)
CN4	Conversor de Nível Lógico 3,3-5 V bidirecional (Sparkfun BOB-12009)
RAD2	Módulo de Rádio (Ai-Thinker RA-02)
R8	Resistência 10 k $\Omega$
R9	Resistência 10 k $\Omega$

Designação do componente no esquema elétrico	Componente
TP2	Bloco de 8 terminais
TP3	Bloco de 8 terminais
TP4	Bloco de 7 terminais
TP5	Bloco de 4 terminais
TP6	Bloco de 8 terminais
TP7	Bloco de 8 terminais
D2	LED vermelho 5mm
CH2	Chave 2 estados
NX2.8	HMI Nextion NX3224T028
R19	Resistência 1 $\Omega$ (100 W)
DMX	Demultiplexer Analógico de 16 Canais (CD74HC4067)
MX	Multiplexer Analógico de 16 Canais (CD74HC4067)
T1	Bloco de 2 terminais
M1	MOSFET P IRF9530
R14	Resistência 10 k $\Omega$
TR2	Transistor 2N3904
OPT	Acoplador Ótico PC817
R17	Resistência 3,3 k $\Omega$
R15	Resistência 10 k $\Omega$
R16	Resistência 270 k $\Omega$
D4	LED Vermelho 5 mm
D3	LED Verde 5 mm
BT5	Botão de pressão
R18	Resistência 270 $\Omega$
R20	Resistência 5 k $\Omega$
D1	Díodo 10A10