



Universidade do Minho
Escola de Engenharia

Ana Eduarda da Costa Gonçalves

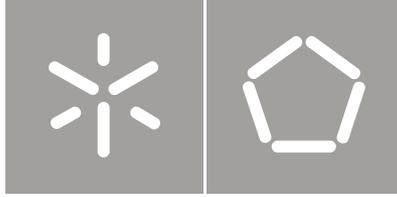
Sistema de Tele-Localização de objetos
utilizando transponder RFID Low-Cost

Sistema de Tele-Localização de objetos
utilizando transponder RFID Low-Cost

Ana Eduarda da Costa Gonçalves

UMinho | 2014

Junho de 2014



Universidade do Minho
Escola de Engenharia

Ana Eduarda da Costa Gonçalves

Sistema de Tele-Localização de objetos
utilizando transponder RFID Low-Cost

Dissertação de Mestrado
Ciclo de Estudos Integrados Conducentes ao
Grau de Mestre em Engenharia de Comunicações

Trabalho efetuado sob a orientação do
Professor Doutor Luís Botelho Ribeiro

DECLARAÇÃO

Nome: Ana Eduarda da Costa Gonçalves

Correio electrónico: anaduarda@gmail.com

Tlm.: 915252297

Número do Bilhete de Identidade: 13779562

Título da dissertação:

Sistema de Tele-Localização de objetos utilizando transponder RFID Low-Cost

Ano de conclusão: 2014

Orientador:

Luís Botelho Ribeiro

Designação do Mestrado:

Ciclo de Estudos Integrados Conducentes ao Grau de Mestre em Engenharia de Comunicações

É AUTORIZADA A REPRODUÇÃO INTEGRAL DESTA DISSERTAÇÃO APENAS PARA EFEITOS DE INVESTIGAÇÃO, MEDIANTE DECLARAÇÃO ESCRITA DO INTERESSADO, QUE A TAL SE COMPROMETE.

Guimarães, ___/___/_____

Assinatura: _____

Dedicatória

Quero dedicar esta tese aos meus pais e irmãos que sempre acreditaram em mim e que acharam que eu poderia fazer um bom trabalho de forma a ter um futuro promissor.

Agradecimentos

Quero agradecer aos meus pais por me darem força para continuar mesmo em dias menos bons e aos meus irmãos por me obrigarem a acreditar mais em mim.

Devo um especial obrigado ao meu orientador que tentou sempre fazer de tudo para me apoiar em tudo o que precisei.

O meu próximo agradecimento vai para os técnicos das oficinas que sempre me acompanharam durante todo o processo do meu protótipo ajudando em tudo o que precisei e também em alegrar o meu dia quando o necessitava.

Agradeço por último aos meus amigos por todo o carinho que me deram e pelo seu apoio.

Sistema de Tele-Localização de Objetos utilizando *transponder* RFID Low-Cost

RESUMO

Esta dissertação situa-se no campo das tecnologias RFID, apresentando uma revisão do seu estado atual. Reporta-se, igualmente, o projeto e construção de um sistema RFID destinado a prevenir roubos de lojas e a permitir a localização de um determinado objeto pela sua etiqueta TAG-it. A solução desenvolvida pretendia-se económica e de fácil utilização.

Numa fase inicial foi necessário um estudo do funcionamento das diversas tecnologias existentes para localização de objetos procurando-se o *standard* RFID mais adequado aos objetivos enunciados.

Dentro do modelo RFID existem várias opções, quer a nível da frequência, quer a nível das ferramentas existentes. Foi optado o protocolo ISO 15693 que trabalha a altas frequências, apresentando um alcance de 1 metro e capacidade para processar até 50 *tags* por segundo. O sistema foi projetado, montado e testado numa escala limitada.

A aplicação montada tem por base dois cenários. No primeiro faz-se inventários periódicos de quaisquer *tags* disponíveis dentro de um determinado perímetro. Caso uma *tag* saia de dentro desse perímetro é acionado um alarme desse objeto o que pode ajudar a prevenir determinados roubos.

O segundo cenário consiste em encontrar um determinado objeto, através do endereçamento individual de uma tag ou de um determinado grupo a que esta pertence. Neste caso o leitor tem de ser deslocado de forma a explorar o território definido para a busca.

Este cenário foi construído à custa de máscaras de endereçamento para evitar colisões na função de inventário.

Finalmente, foram realizados vários testes em laboratório que validaram o conceito subjacente a ambos os cenários, faltando apenas um teste de campo numa firma de advocacia que, por falta de tempo, não foi possível concretizar.

Objects Tele-location System using low-cost RFID transponder

Abstract

This dissertation is based on RFID technology fields, presenting a review of its actual state. It refers, equally, the project and construction of a RFID system destined to prevent thefts from stores and allowing the location of a determinate object by its label Tag-it. The solution developed was intended to be low cost and of easy utilization.

In an initial phase it was necessary a study of the functionality of its several existing technologies for the location of the objects looking the RFID *standard* most appropriate to the objects enunciated.

Within the RFID model there are several options, either the level of frequency or the level of the existing tools. I choose the protocol ISO 15693 working at high frequencies, presenting a range of one meter and capacity to process up to 50 tags per second. The system was projected, assembled and tested on a limited scale.

The assembled application is based on two sceneries. On the first one I made periodic inventories of any tags available within a determined perimeter an alarm triggered of that object what may help to prevent certain thefts.

The second scenery is based in finding a determined object through its addressing of one individual tag or through its addressing of one individual tag or through its addressing of one determined group to which this belongs.

In this case the reader must be dislocated to explore the territory defined to the search.

This scenery was built by using addressing masks to avoid collisions on the function of the inventory.

Finally, there were performed several tests in laboratory that validated the underlying concept to both sceneries, missing just one test of the field in a law firm that, for lack of time, it was not possible to achieve.

Índice

1. Introdução.....	1
1.1. Enquadramento	1
1.2. Objetivos e resultados esperados	2
1.3. Planeamento das tarefas desenvolvidas	4
1.4. Contributos científicos e técnicos do trabalho:.....	8
1.5. Organização do documento	9
2. Estado da Arte	11
2.1. Sistemas de tele-localização	11
2.2. <i>Standard</i> RFID.....	12
2.2.1. Elementos constituintes de um sistema RFID	14
2.2.2. Vantagens do RFID	17
2.2.3. Desvantagens do RFID	17
2.3. Tecnologias disponíveis.....	19
2.3.1. Transmissores universais ISM FSK.....	19
2.3.2. Alpha RF Transceiver.....	21
2.3.3. <i>Tag-it</i>	24
2.3.4. TRF7960.....	27
3. Especificação da solução RFID a desenvolver.....	39
3.1. Descrição dos cenários possíveis.....	39
3.1.1. Cenário 1.....	39
3.1.2. Cenário 2.....	40
3.1.3. Cenário 3.....	40
3.2. Funcionalidades do sistema	41
3.3. Sistema central	41

4. Projeto RFID	43
4.1. <i>Hardware</i>	43
4.1.1. Leitor.....	43
4.1.2. Microcontrolador adotado e principais configurações	50
4.1.3. Antena.....	53
4.2. <i>Software</i>	54
4.2.1. Algoritmos	54
4.2.2. Inventário e tratamento de colisões no protocolo ISO 15693.....	66
4.2.3. Funcionamento das máscaras no protocolo ISO 15693.....	68
4.2.4. Fluxogramas de acesso ao leitor	69
4.2.5. Interface com o utilizador	71
4.2.6. Sistema de alarmes	75
4.2.7. Saída dos dados.....	77
5. Testes e avaliação do protótipo implementado	79
5.1. Leitor contruído	79
5.2. Outros Leitores	86
5.3. Comparação entre o leitor contruído e o outro testado.....	89
6. Considerações finais e perspectivas futuras	91
6.1. Conclusão.....	91
Referências bibliográficas.....	93
Anexos:	97
Anexo 1- Placas	97
Anexo 2-Fluxogramas.....	101
Anexo 3- Outros.....	109

Índice de Figuras

Figura 1-1 Evolução das tecnologias [33].....	1
Figura 1-2 Evolução RFID [34].....	2
Figura 1-3- Diagrama de Gant correspondente ao planeamento acima	5
Figura 2-1- Sistema de Tele-localização Active Badge [50]	11
Figura 2-2- Sistema de ultrassons da Cricket [32].....	12
Figura 2-3-Elementos constituintes de um sistema RFID	14
Figura 2-4-Variação das <i>tags</i> passivas conforme a frequência [37]	15
Figura 2-5-Comparação entre as frequências disponíveis a fatores externos. [36]	16
Figura 2-6 -Pinos dos integrados em modo <i>Microcontroller</i> à esquerda e <i>EEPROM</i> à direita [7] ...	19
Figura 2-7-Pinos do integrado em modo <i>Standalone</i> à esquerda e <i>Microcontroller</i> à direita [5]	20
Figura 2-8-Pinos do integrado em modo REVC e <i>later</i> [4].....	21
Figura 2-9-Alpha RF Transceiver [3]	21
Figura 2-10-Diagrama com a explicação da transmissão com o <i>Alpha RF Transceiver</i>	22
Figura 2-11-Diagrama com a explicação da receção com o Alpha RF Transceiver.....	23
Figura 2-12-Efeito do metal pelas diferentes <i>tags</i> [22].....	24
Figura 2-13-Efeito das <i>tags</i> quando um segundo <i>transponder</i> é colocado (teste do código SID) [22]	25
Figura 2-14- <i>Transponder</i> RI-I02-0110A-00 [9].....	25
Figura 2-15-Constituição da <i>tag</i> [45]	26
Figura 2-16- <i>Tag-it</i> HF-I codificação UID [23].....	27
Figura 2-17-TRF7960 [10].....	27
Figura 2-18-Sistema de leitura completo para a leitura da <i>tag</i> [11].....	28
Figura 2-19-Comunicação em paralelo com <i>Stop Condition</i> simples [11].....	34
Figura 2-20-Comunicação em paralelo com <i>Stop Condition</i> contínuo [11]	34
Figura 2-21-Comportamento da comunicação sem o <i>Slave</i> ativo [11].....	37
Figura 2-22-Comportamento da comunicação com o <i>slave</i> ativo (Modo de escrita) [11]	37
Figura 3-1-Inventário geral	39
Figura 3-2-Inventário endereçado à máscara	40

Figura 3-3-Inventário endereçado à máscara com dispositivo móvel.....	40
Figura 3-4- Cenário futuro de um sistema de histórico das <i>tags</i>	41
Figura 3-5- Cenário futuro de um sistema de validação do utilizador.....	42
Figura 4-1-Esquema funcional do TRF796x [11].....	44
Figura 4-2-Esquema de montagem do leitor em modo série [11].....	45
Figura 4-3-Esquema de montagem do leitor implementado	46
Figura 4-4-Face de cima do PCB final.....	47
Figura 4-5-Face de baixo do PCB final	47
Figura 4-6-Board de duas faces final	48
Figura 4-7-Bobines realizadas	49
Figura 4-8- Leitor final	49
Figura 4-9-Imagem dos três microcontroladores selecionados: a) Msp430F2370 [14], b) AT89C51IC2 [25], c) PIC18F4550 [13].....	50
Figura 4-10-Microcontrolador da ATMEL com a interface criada	51
Figura 4-11-Imagem da antena criada bem como a <i>tag</i> exemplo	53
Figura 4-12-Diagrama com a <i>main</i> principal.....	54
Figura 4-13- Diagrama do comportamento da função <i>FindTags</i>	55
Figura 4-14-Diagrama da função <i>Inventory Request</i>	61
Figura 4-15-Diagrama da função <i>InterruptHandlerReader</i>	63
Figura 4-16-Operação de escrita [11]	69
Figura 4-17-Operação de leitura [11].....	69
Figura 4-18-Comunicação da interface SPI (registo de estado IRQ lido) [11].....	70
Figura 4-19-Cabo utilizado na interface com computadores portáteis [38].....	71
Figura 4-20-Terminal Hercules utilizado.....	72
Figura 4-21- Cabo utilizado na interface com telemóveis [39]	73
Figura 4-22-Terminal Slick USB 2 Serial [43].....	73
Figura 4-23-Esquema de montagem do sistema de alarmes	75
Figura 4-24-Sistema de Alarme implementado	76
Figura 4-25-Display de 7 segmentos [44].....	77
Figura 5-1-Saída do SysCLK com o EN= 0 e o EN2=1	79
Figura 5-2-Saída do SysCLK com o EN=1 e o EN2=1	80

Figura 5-3-Saida do SysCLK com o EN=1 e o EN2=1 após a configuração do Sys_CLK_Control	81
Figura 5-4-Análise do campo <i>Radio Field</i>	82
Figura 5-5-Leitura do comando <i>Reset</i> enviado pelo MOSI.....	83
Figura 5-6-Teste ao Inventário.....	84
Figura 5-7-Resultado do endereçamento direto com a Tag dentro do campo RF	85
Figura 5-8-Resultado do endereçamento direto com a Tag fora do campo RF	85
Figura 5-9-Leitor RFID 125KHz - SEN11425P [26]	86
Figura 5-10- <i>Tags</i> compradas para funcionar com o leitor de 125KHz.....	86
Figura 5-11-Parte traseira do leitor e fios soldados para alimentação	87
Figura 5-12- Bits recebidos com 5ms por divisão	87
Figura 5-13-Bits recebidos com 1ms por divisão	88
Figura 5-14-Bits recebidos com 250µs por divisão	88
Figura 5-15- Bits recebidos com 100µs por divisão	89
Figura A-1-Placa de face simples	97
Figura A-2-Primeiro PCB de face simples criado	98
Figura A-3-Face de cima do PCB	98
Figura A-4-Face de baixo do PCB.....	99
Figura A-5- PCB com as duas faces	99
Figura A-6- Diagrama do comportamento da função <i>FindTags</i> (cont1)	101
Figura A-7- Diagrama do comportamento da função <i>FindTags</i> (cont2)	102
Figura A-8- Diagrama do comportamento da função <i>FindTags</i> (cont3)	103
Figura A-9-Diagrama com a explicação do mecanismo anticolisões para o ISO 14443A.....	104
Figura A-10-Diagrama com a explicação do mecanismo anticolisões para o ISO 14443B.....	105
Figura A-11-Diagrama da função <i>TIInventory Request</i>	106

Índice de tabelas

Tabela 1-1-Planeamento geral da dissertação	4
Tabela 1-2-Tabela referente à fase 4.....	6
Tabela 1-3-Tabela referente à fase 4.1	6
Tabela 1-4-Descrição da fase 4.2	7
Tabela 1-5-Descrição da fase 4.3	7
Tabela 2-1- Pinos na interface paralelo e série no modo de conexão ou direto [11]	32
Tabela 2-2-Trama no modo de endereçamento contínuo [11].....	33
Tabela 2-3-Trama no modo de endereçamento não contínuo [11]	34
Tabela 2-4-Trama no modo de comando [11]	34
Tabela 4-1-Diferenças entre o TRF7960 e o TRF7961 [11].....	43
Tabela 4-2- Comparação entre os microcontroladores	50
Tabela 4-3-Distribuição dos bits das <i>Address</i> e <i>Command Words</i> [11]	51
Tabela 4-4- Exemplo da formação de uma palavra	52
Tabela 4-5-Códigos dos comandos [11]	52
Tabela 4-6- ChipStatus Control [11].....	56
Tabela 4-7-IsoControl [11]	57
Tabela 4-8-Seleção dos modos RFID [11].....	57
Tabela 4-9-Modulador e Sys_CLK Control [11].....	59
Tabela 4-10- Registo de estado do IRQ [11]	64
Tabela 4-11- Estado da FIFO [11]	65
Tabela 4-12-Primeira fase do processamento do inventário	66
Tabela 4-13-Segunda fase do processamento do inventário	67
Tabela 4-14-Primeira fase do processamento com máscara	68
Tabela 4-15-Comandos possíveis de interação com o utilizador.....	74
Tabela 4-16-Significado das mensagens do display	77
Tabela A-1- Recursos.....	109
Tabela A-2-Registos de comunicação série-T2CON.....	111

Siglas e Acrónimos:

AFE	Active Front End
AFI	Application Family Identifier
AGC	Automatic Gain Control
ASK	Amplitude-Shift Keying
AM	Amplitude Modulation
COM	Component Object Model
CRC	Cyclic Redundancy Check
DSFID	Data Storage Field Identifier
EEPROM	Electrically-Erasable Programmable Read-Only Memory
EN	Enable
EOF	End of File
ETSI	European Telecommunications Standards Institute
FCC	Federal Communications Commission
FDX	Full-Duplex Systems
FIFO	First In First Out
FTDI	Future Technology Devices International
FSK	Frequency-Shift Keying
GPS	Global Positioning System
HDX	Half-Duplex Systems
high-Q	Hight Quality
HF	Hight Frequency
IC	Integrated Circuit
ID	Identification
IFF	Identify Friend or Foe

IRQ	Interrupt Request
ISM	Industrial, Scientific and Medical
LF	Low Frequency
LSB	Least Significant Bit
MCU	Microcontroller
MISO	Master In Slave Out
MOSI	Master Out Slave In
MSB	Most Significant Bit
NFC	Near Field Communication
OOK	On Off Keying
PCB	Printed Circuit Board
PET	PolyEthyleneTherephtalate
PLL	Phase Lock Loop
PM	Phase Modulation
RFID	Radio Frequency Identification
RSSI	Received Signal Strength Indicator
RX	Receiver
SOF	Start of Frame
SPI	Serial Peripheral Interface
SS	Slave Select
TI	Texas Instruments
TX	Transmitter
UART	Universal Asynchronous Receiver/Transmitter
UID	Unique Identifier
UHF	Ultra High Frequency
USB	Universal Serial Bus

VCD	Vicinity Coupling Device
VICC	Vicinity Integrated Circuit Card

1. Introdução

1.1. Enquadramento

O enorme crescimento das tecnologias *wireless*, a evolução dos dispositivos móveis e da internet fez com que surgisse um forte interesse em sistemas de tele-localização.

Nos últimos anos este problema foi resolvido graças a sistemas de triangulação, análise de cenários e técnicas de proximidade.

Um sistema de localização bem conhecido é o GPS que inclui um sistema de navegação constituído por 24 satélites colocados em órbita.

Infelizmente este sistema necessita de uma enorme quantidade de satélites e não funciona dentro de casas pelo que foi necessário evoluir para novas soluções.

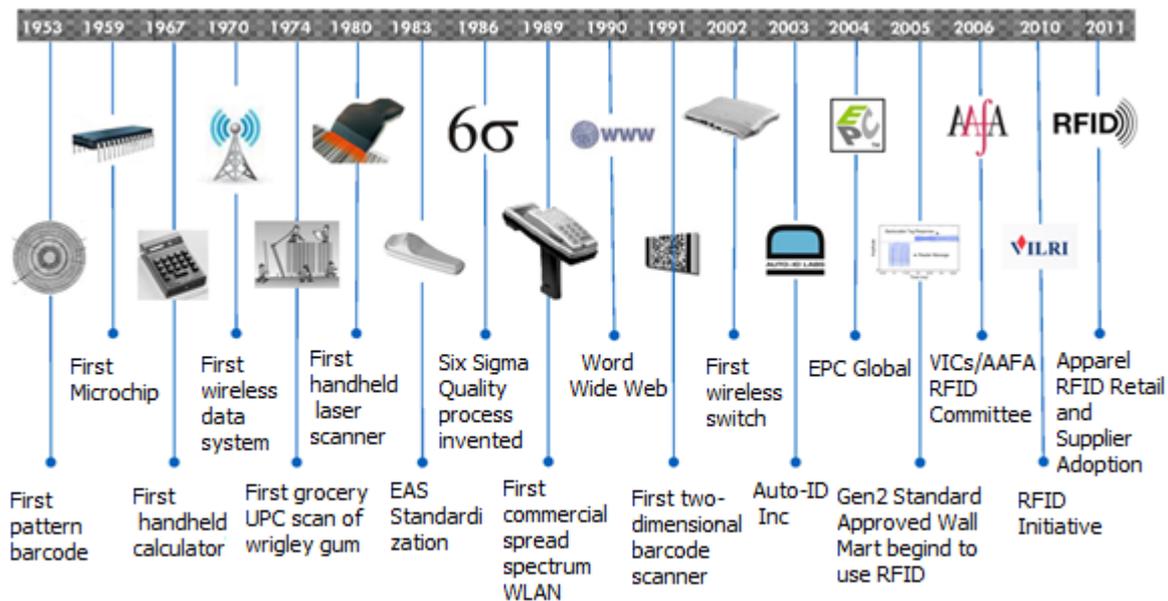


Figura 1-1 Evolução das tecnologias [33]

Uma das soluções possíveis para a localização de objetos dentro de casas é o RFID. Este surgiu na segunda guerra mundial, em 1935, onde todos os aviões usavam um sistema de radares. Foi desenvolvido pelo escocês Robert Alexander Watson-Wat para detetar ataques de aviões inimigos. Porém os radares não diferenciavam os aviões de fogo amigo do inimigo. Os alemães descobriram também que se os seus pilotos realizassem uma manobra de 360° ao longo do eixo de simetria quando estes estivessem a retornar à base, o sinal de rádio refletido para a base iria ser modificado.

Em Inglaterra o Sr Watson-Wat desenvolveu o primeiro identificador ativo capaz de distinguir amigo de inimigo IFF (*Identify Friend or Foe*). Hoje em dia ainda é utilizado este mesmo princípio.

Esta área continuou a ser investigada nas décadas de 50 e 60 onde cientistas dos Estados Unidos, Europa e Japão divulgaram diversas utilidades para esta tecnologia.

Começou a ser comercializada no início para sistemas antifurto onde através de etiquetas se poderia identificar o roubo de um objeto. Infelizmente a maioria das etiquetas utilizadas não possuíam identificação própria, não identificando o objeto em si, mas sim se esta passa por um determinado local ou não.

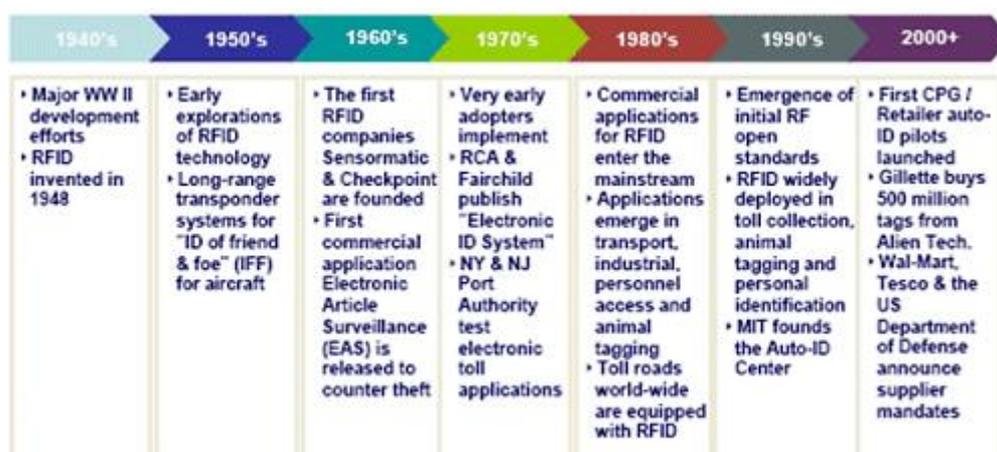


Figura 1-2 Evolução RFID [34]

O objetivo desta dissertação é, utilizando as características de um sistema RFID, desenvolver de uma maneira económica e rentável um sistema de tele-localização de objetos destinado à deteção/prevenção de furto de objetos deixados em campo aberto.

1.2. Objetivos e resultados esperados

Este trabalho tem como principal objetivo a construção de um protótipo que permita aos intervenientes saber sempre o estado dos seus objetos. Se um dispositivo deixar de responder este poderá ter sido alvo de um furto o que resultará num alarme para o utilizador, através de um som de aviso aos interessados.

Neste trabalho serão também realizados dois tipos de sistemas para dois tipos de aplicações diferentes. O primeiro sistema consiste em fazer uma espécie de histórico do inventário de forma a haver acesso a todas as etiquetas e, caso uma não responda, fazer soar o alarme. O segundo cenário consiste em fazer com que apenas o determinado objeto ou documento pretendido nos responda de forma a conseguir ser localizado.

A procura de documentos tem como principal utilidade ajudar uma firma de advogados, por exemplo a encontrar um determinado processo judicial que necessite.

1.3. Planeamento das tarefas desenvolvidas

Neste capítulo é feita uma descrição do planeamento original para a escrita desta dissertação. Todos os recursos utilizados durante o projeto encontram-se descritos no anexo 3 folha 1.

Tabela 1-1-Planeamento geral da dissertação

Task Name	Duration	Start	Finish
Dissertação	246 days	Mon 25-02-13	Fri 31-01-14
1- Revisão bibliográfica sobre o tema	10 days	Mon 25-02-13	Mon 11-03-13
2- Planeamento do trabalho	6 days	Mon 11-03-13	Tue 19-03-13
3- Analise das diversas ferramentas a adotar e respetiva escolha	14 days	Tue 19-03-13	Mon 08-04-13
4- Construção do protótipo	173,88 days	Mon 08-04-13	Thu 05-12-13
4.1- <i>Tags</i>	11 days	Mon 08-04-13	Tue 23-04-13
4.2- Leitor	67 days	Tue 23-04-13	Thu 25-07-13
4.3- Microcontrolador	95,88 days	Thu 25-07-13	Thu 05-12-13
5- Testes de campo	2,5 days	Fri 06-12-13	Fri 20-12-13
6- Correção de possíveis erros encontrados na fase acima	30 days	Sat 21-12-13	Fri 31-01-14
7- Escrita da Dissertação	304,84 days	Tue 02-06-09	Fri 31-01-14

Na tabela acima foi descrito todo o planeamento do projeto. Como se pode constatar o projeto foi feito de forma a ter uma duração de 11 meses e tem em atenção quaisquer acontecimentos inesperados que possam acontecer.

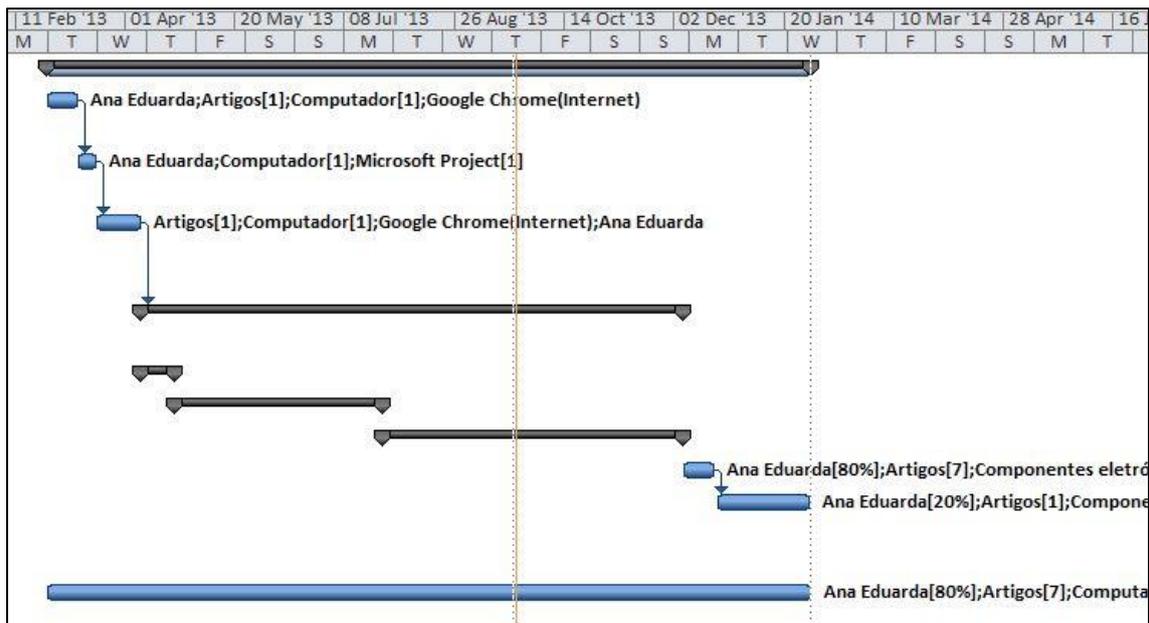


Figura 1-3- Diagrama de Gant correspondente ao planeamento acima

Tal como se pode ver pelo diagrama de Gant o projeto foi planeado de forma a garantir que todo o tempo seja aproveitado.

De seguida será feita uma breve descrição de cada fase descrita acima.

1. Revisão bibliográfica:

Esta primeira etapa visa a pesquisa bibliográfica exaustiva sobre o tema, de forma a conseguir perceber bem o que é um sistema RFID e de que forma pode ser utilizado.

2. Planeamento do trabalho

O planeamento do trabalho tem como principal função a definição de todas as tarefas a fazer e respetiva calendarização. Através desta fase é possível coordenar todo o trabalho e acelerar em algumas fases se tal for necessário.

3. Analisar as diversas tecnologias disponíveis e realizar a respectiva escolha dos componentes:

Nesta etapa do trabalho pretende-se analisar de forma exaustiva todos os componentes que podem ser utilizados neste tipo de sistema e escolher, face à especificação e minimização dos custos, o que melhor se enquadra num cenário a larga escala.

4. Construção do protótipo

Nesta fase irão ser definidos os requisitos de cada componente explorando as diversas formas de implementação e a respectiva construção dos mesmos protótipos.

Tabela 1-2-Tabela referente à fase 4

Task Name	Duration	Start	Finish
4- Construção do protótipo	173,88 days	Mon 08-04-13	Thu 05-12-13
4.1- <i>Tags</i>	11 days	Mon 08-04-13	Tue 23-04-13
4.2- Leitor	67 days	Tue 23-04-13	Thu 25-07-13
4.3- Microcontrolador	95,88 days	Thu 25-07-13	Thu 05-12-13

Tal como se pode ver pela tabela esta fase é dividida em três, sendo cada fase referente a cada componente.

Tabela 1-3-Tabela referente à fase 4.1

Task Name	Duration	Start	Finish
4.1- <i>Tags</i>	11 days	Mon 08-04-13	Tue 23-04-13
4.1.1- Encomenda das <i>tags</i>	3 days	Mon 08-04-13	Thu 11-04-13
4.1.2- Leitura exaustiva sobre as mesmas	8 days	Thu 11-04-13	Tue 23-04-13

Na tabela acima estão ilustradas as tarefas que serão realizadas na componente *tags*. Tal como se pode ver esta fase é bastante rápida e a mais curta devido ao fato de não ser necessário nenhuma configuração interna na *tag*.

Tabela 1-4-Descrição da fase 4.2

Task Name	Duration	Start	Finish
4.2- Leitor	67 days	Tue 23-04-13	Thu 25-07-13
4.2.1- Escolha do <i>software</i> para a elaboração do PCB para o leitor	4 days	Tue 23-04-13	Mon 29-04-13
4.2.2- Esquematização do leitor a construir	8 days	Mon 29-04-13	Thu 09-05-13
4.2.3- Construção da <i>board</i> para o PCB.	10 days	Thu 09-05-13	Thu 23-05-13
4.2.4- Construção do PCB.	15 days	Thu 23-05-13	Thu 13-06-13
4.2.5- Colocação dos diversos componentes no PCB.	15 days	Thu 13-06-13	Thu 04-07-13
4.2.6- Detecção e correção de erros que ocorreram durante esta fase	15 days	Thu 04-07-13	Thu 25-07-13

Tabela 1-5-Descrição da fase 4.3

Task Name	Duration	Start	Finish
4.3- Microcontrolador	95,88 days	Thu 25-07-13	Thu 05-12-13
4.3.1- Leitura e escolha do microcontrolador mais indicado	6 days	Thu 25-07-13	Fri 02-08-13
4.3.2- Elaboração de algoritmos de forma a clarificar o código que irá ser implementado para controlar o microcontrolador e consequentemente o leitor	15 days	Fri 02-08-13	Fri 23-08-13
4.3.3- Estudo de códigos de exemplo fornecidos para este microcontrolador	8 days	Fri 23-08-13	Wed 04-09-13

4.3.4- Adaptação do código para o objetivo do projeto	30 days	Fri 13-09-13	Thu 24-10-13
4.3.5- Detecção e correção de erros que ocorreram durante esta fase	15 days	Fri 25-10-13	Thu 14-11-13
4.3.6- Ligação do leitor ao MCU programado	15 days	Fri 15-11-13	Thu 05-12-13

Estas duas últimas fases são bastante mais trabalhosas devido ao facto de serem dois componentes fundamentais no projeto que não se encontram programados de acordo com os requisitos do projeto.

5. Testes de campo:

Nesta etapa será feito um conjunto de testes onde será verificado o desempenho do sistema na deteção/prevenção de furto de objetos deixados em campo aberto.

6. Correção de possíveis erros encontrados na fase acima:

Esta fase terá por base uma análise de todos os problemas encontrados e a possível correção dos mesmos.

7. Escrita da Dissertação:

Na última fase será redigida a dissertação que irá conjugar e descrever todo o trabalho desenvolvido.

1.4. Contributos científicos e técnicos do trabalho:

Os contributos científicos e técnicos deste trabalho são os seguintes:

- Tentar detetar possíveis roubos em lojas antes de estes ocorrerem, através do fato de um objeto se afastar demasiado de um determinado campo.
- Identificar onde se encontra um determinado objeto dentro de uma determinada área: perda de carteira, etc.
- Encontrar arquivos importantes numa firma de advocacia.

- Tentar simplificar a tecnologia RFID para o utilizador e o seu manuseamento.
- Este sistema pode ser inovador pois pode-se interagir de uma forma dinâmica com um protótipo para guardar objetos importantes e para ajudar a encontrar objetos desaparecidos.

1.5. Organização do documento

Neste subcapítulo será explicado de uma forma breve a estrutura de cada um dos capítulos.

No primeiro capítulo é feito um enquadramento do tema e é explicada a necessidade deste projeto. De seguida são apresentados os objetivos do trabalho e os resultados que são esperados de cada objetivo. É especificado também todo o planeamento original do trabalho bem como os contributos científicos que o projeto tem.

O capítulo 2 é dedicado ao estado da arte onde se realça os Sistemas de tele-localização de objetos, mais exatamente sobre o RFID. São apresentados alguns conceitos sobre os componentes desse sistema, e são explicados os diversos *standards* e são mencionadas as vantagens e desvantagens do mesmo. O capítulo 2 tem também um subcapítulo onde se pode ver algumas das ferramentas que se podem adotar no RFID. Neste subcapítulo são enumeradas as ferramentas disponíveis: Transmissores universais ISM FSK, o Alpha RF Tranceiver, a *Tag-it* da *Texas Instruments* e o TRF7960, sendo feita uma breve descrição de cada uma e das suas principais características.

O capítulo 3 define quais são as características e funcionalidades que o protótipo tem, bem como os diferentes cenários a que este responde. Refere também as características do sistema central.

No capítulo 4 é descrito como foi implementado o protótipo e quais as técnicas e ferramentas utilizadas, destacando as suas principais vantagens e limitações. Descreve-se também as regras de interação entre os diversos componentes e explica-se como são implementadas as funções mais importantes deste protótipo.

O capítulo 5 é reservado aos testes do protótipo e às avaliações sobre o mesmo. Será neste capítulo que se pode ver o correto funcionamento do sistema e a sua comparação com um sistema similar já existente.

Por último, o capítulo 7 faz uma síntese de todo o trabalho realizado, refletindo um pouco sobre o mesmo. Termina com algumas ideias de como este trabalho pode ser evoluído para novos projetos.

2. Estado da Arte

Neste capítulo serão realçados alguns conceitos teóricos fundamentais para a realização do projeto.

2.1. Sistemas de tele-localização

Tal como já foi referido os sistemas RFID surgiram da necessidade de um sistema de localização dentro de um espaço físico. Além dos sistemas RFID surgiram outros sistemas de tele-localização como: os infravermelhos, 802.11, *ultrasonic*.

Infravermelhos – É um sistema desenvolvido pelo laboratório de pesquisa Olivett [50]. Utiliza a tecnologia de difusão dos infravermelhos para saber a localização de determinados objetos. Tem como principais limitações o fato de precisar de um ponto específico e ter um alcance de transmissão do sinal muito curto.

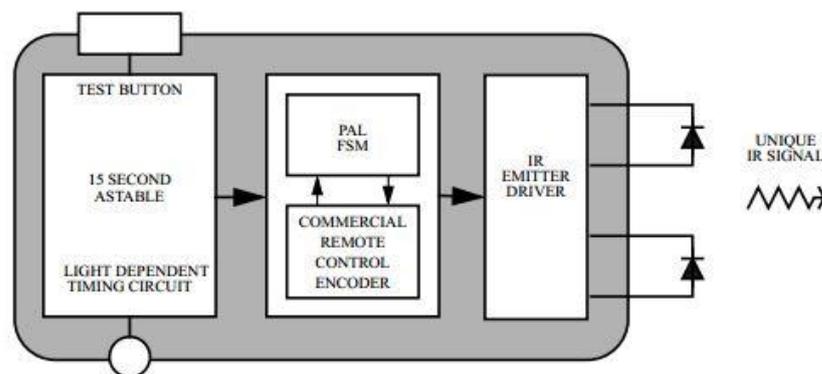


Figura 2-1- Sistema de Tele-localização Active Badge [50]

IEEE 802.11 - RADAR'S [51]- É um sistema baseado em radiofrequência para localizar utilizadores dentro de edifícios. Este aproveita o *standard* 802.11 para medir a força do sinal em múltiplas estações. Estas estações encontram-se posicionadas em toda a área de forma a conseguir obter uma medição correta. A maior qualidade deste sistema é o fato de que é muito fácil de configurar e precisar de poucas estações base. Outra qualidade que este possui é usar a mesma infraestrutura que fornece a internet *wireless* do edifício. Este sistema apenas sabe com 50% de probabilidade a localização exata de um objeto que se encontra a 3 metros dele.

Ultrasonic- Sistema baseado em ultrassons que funciona com a técnica de medição do tempo de voo para dizer a localização de um objeto. Duas entidades que usam este sistema são a *Cricket Location Support System* e a *Active Bat Location system*. A vantagem deste sistema é a sua precisão.

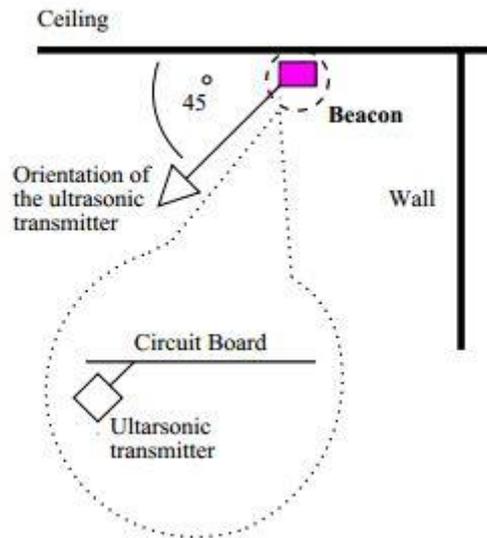


Figura 2-2- Sistema de ultrassons da Cricket [32]

RFID - O RFID é uma tecnologia que usa a sensibilidade de localização baseada na análise da força do sinal para encontrar objetos. Os objetos estão localizados com sensores denominados de *tags* que respondem a um determinado estímulo. A força do sinal de rádio é usado como um sensor de medida para estimar a distância até à *tag*. No subcapítulo abaixo será descrito mais aprofundadamente esta tecnologia.

2.2. Standard RFID

O RFID é um método de identificação de um determinado objeto animado ou inanimado através de ondas rádio. Os primeiros estudos que surgiram nesta área foram em 1940 para identificar aviões militares, de forma a diferenciar os aviões aliados dos inimigos.

Atualmente, a tecnologia RFID também pode ser utilizada para localização de determinados objetos.

Hoje em dia os militares ainda utilizam esta tecnologia bem assim como as companhias aéreas, os sensores das chaves de um carro, os passaportes, os cartões multibanco, os cartões de estudante, etc.

Ao contrário do que muitas pessoas pensam as etiquetas, na maioria das lojas, não são baseadas no sistema RFID, pois estas apenas detetam se um determinado objeto sai de uma determinada área, não diferenciando o objeto.

Nas tecnologias RFID é necessária a existência de uma determinada *tag* que vai estar ligada ao objeto que se quer identificar e um recetor que irá receber esta informação e processá-la. Esta *tag* terá de enviar a informação periodicamente, de forma a avisar o seu UID e conseqüentemente poder ser identificado o local onde esta se encontra.

Mais recentemente, nos testes realizados no terreno com a tecnologia RFID conclui-se que, mesmo com as *tags* ativas de maior capacidade, só se conseguia operar até uma distância máxima correspondente a um campo de baseball.

Atualmente já existem fechaduras eletrônicas baseadas em RFID.

O RFID usa diversos *standards* entre os quais:

- ISO 14223- consiste no *standard* utilizado para identificar animais
- ISO 14443 – *standard* para altas frequências, usado para passaportes por exemplo.

Está dividido em 4 partes:

1. Características Físicas
 2. Potência de radiofrequência e a interface do sinal
 3. Inicialização e anticolisão
 4. Protocolo de transmissão
- ISO 15693- *standard* utilizado para altas frequências, 13.56 Mhz, usado para cartões de crédito. Este *standard* está dividido em 3 partes:
 1. Características físicas
 2. Inicialização e interface do ar
 3. Anticolisão e protocolo de transmissão
 - ISO 18000- *standard* de informação, isto é trata da identificação por radiofrequência de itens que estejam armazenados. Encontra-se dividido em 6 partes:
 1. Define a formação de todas as interfaces de ar e as suas definições
 2. Define os parâmetros da interface ar para a comunicação nos 135khz. Existe o tipo FDX a 125 kHz e o tipo HDX a 134.2 Khz
 3. Define os parâmetros de interface ar para a comunicação nos 13.56 Mhz.
 4. Define os parâmetros de interface ar para a comunicação nos 2.45 GHz. Tem dois modos de operação das *tags*: *Tags* passivas cujo interrogador pergunta primeiro e *tags* com bateria cujas *tags* respondem primeiro
 5. Define os parâmetros de interface ar para a comunicação dos 860 Mhz até 960 Mhz. Tem 3 tipos, o tipo A e B que só diferem no algoritmo de anticolisão usado e o tipo C que é conhecido como EPCglobal Call 1 Gen 2.

6. Define os parâmetros de interface ar para a comunicação até aos 433Mhz

- ISO 18092/ 21481- Usadas para comunicação NFC
- ASTM D7434/ D735/D7580- *Standard* de testes de performance para o RFID

O *standard* adotado foi o ISO 15693 devido às características deste tipo de sistema e à frequência a que responde. No subcapítulo abaixo serão explicadas estas mesma características em comparação com outros *standards* que obedecem a outras frequências.

2.2.1. Elementos constituintes de um sistema RFID

Um sistema de identificação por radiofrequência tem por base os seguintes componentes:

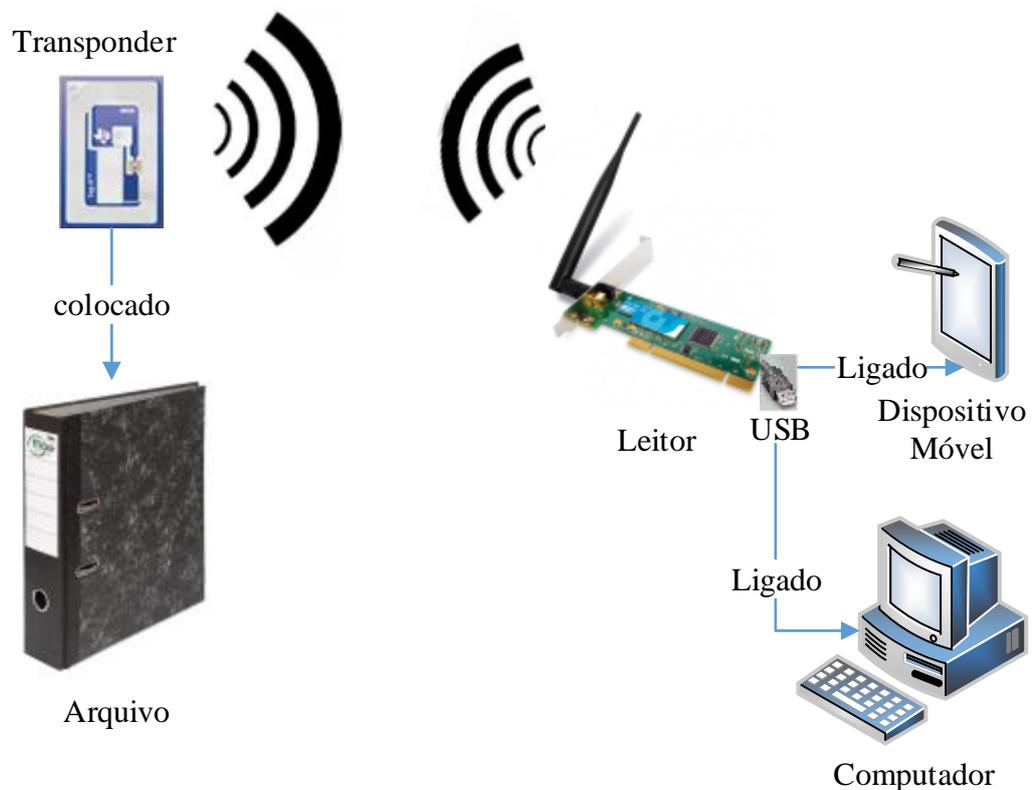


Figura 2-3-Elementos constituintes de um sistema RFID

2.2.1.1. Leitor

O leitor pode fazer operações de escrita e de leitura sobre o *transponder*, sendo mais utilizado apenas a leitura para saber o local em que este se encontra.

Para comunicar com o *transponder* este utiliza uma antena que funciona à mesma frequência que ele.

2.2.1.2. Transponder

O *transponder* numa fase inicial encontra-se inativado possuindo apenas um código correspondente ao seu identificador único (UID). Cabe ao leitor enviar um sinal para ativar o *transponder*. Quando é ativado este começa a emitir o seu UID para o leitor podendo armazenar informação extra na sua memória, caso o leitor o solicite.

Existem 2 tipos de *transponders*: o *transponder* passivo onde a *tag* responde ao leitor apenas quando é solicitada e o *transponder* ativo onde as *tags* enviam por elas mesmas a informação sem ser necessário algum estímulo.

As *tags* utilizadas também variam em relação à frequência utilizada, sendo que as *tags* passivas podem variar de acordo com a seguinte figura.

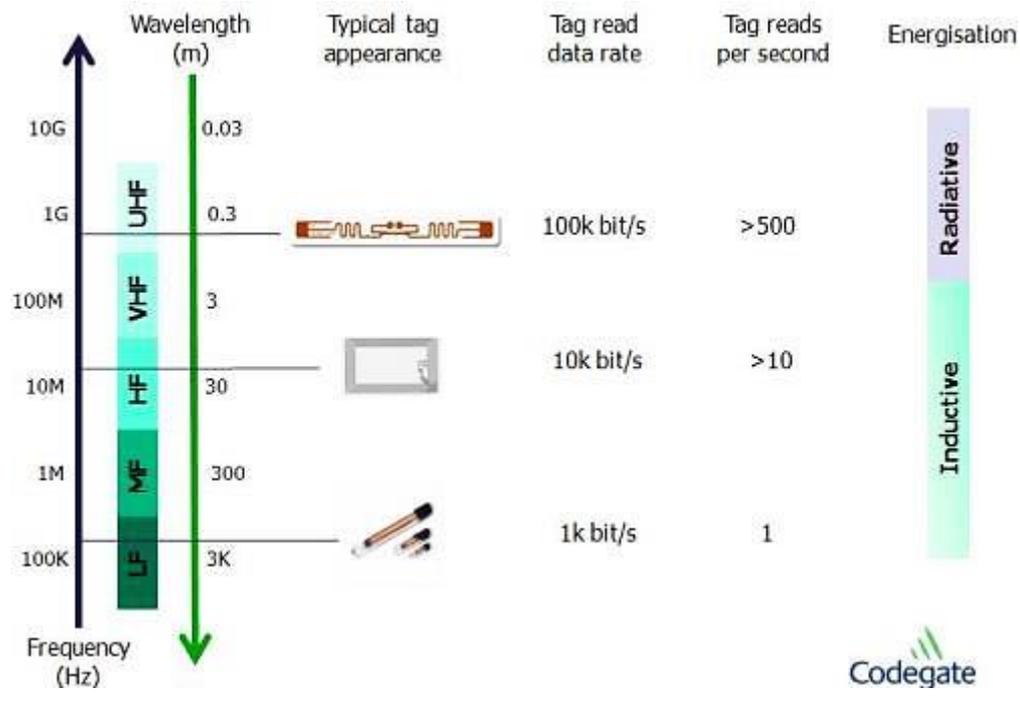


Figura 2-4-Variação das *tags* passivas conforme a frequência [37]

2.2.1.3. Frequências disponíveis

As frequências neste sistema podem ser:

- Baixa frequência (LF)- 10Khz até 500Khz- As *tags* utilizadas nesta frequência são usadas em ambientes industriais. A desvantagem nesta comunicação é que as *tags* lidas são relativamente lentas.
- Frequência ultra alta (UHF)- 900Mhz até 5800MHz- Nesta frequência são utilizadas *tags* ativas. Estas *tags* possuem uma fonte de energia que pode durar de dias a anos.

Esta fonte faz com que esta *tag* possa transmitir a longas distâncias. A desvantagem destas *tags* é o custo que pode ser de 10 a 100 vezes mais do que uma *tag* passiva.

- Alta frequência (HF)- entre os 13Mhz e 27 Mhz- As *tags* aqui não têm nenhuma fonte de potência. O comprimento de onda é muito longa, pelo que as *tags* nesta frequência, usam múltiplos ciclos para formar uma boa antena. Estas *tags* são usadas em sistemas de pagamento de cartão de crédito e em bibliotecas. Têm um campo de alcance limitado e a velocidade de comunicação ainda é relativamente baixa.

	< 135 kHz	13.56 MHz	UHF	2.45 GHz
Effect	Magnetic or Inductive coupling		Radio transmission or backscatter	
Water, humidity				
Metallic environment				
Data transfer rate				
Field characteristics				

Figura 2-5-Comparação entre as frequências disponíveis a fatores externos. [36]

Legenda:

- Bom desempenho - 
- Desempenho médio - 
- Mau desempenho - 

A frequência adotada foi a HF, correspondente à gama dos 13.56Mhz que consegue descartar algumas desvantagens do LF, como o tamanho das suas antenas e reduzir as interferências induzidas por fatores externos. Não se optou pelas UHF pois eram mais complexas. Na figura 2.5 está apresentada uma avaliação dos fatores externos relativamente às frequências que se poderiam utilizar.

2.2.2. Vantagens do RFID

O RFID tem como principais vantagens:

- Identificar possíveis mercadorias para controlo de inventário.
- Identificar e rastrear pessoas e objetos.
- Reduzir o número de perdas, roubos e falsificações.
- Diminuir os riscos operacionais existentes.
- Aumentar a produtividade do armazém, através da identificação dos objetos enquanto se movem pelo armazém.
- Aumentar a produtividade no ponto de venda. O material que sai da loja é registado automaticamente sem ter a necessidade de um agente humano.
- Gestão de inventário.
- Leitura de uma etiqueta sem ser necessário a proximidade do leitor para captar os dados.
- Todas as *tags* RF podem ser lidas mesmo em fatores ambientais extremos como a neve, fogo, gelo.
- As *tags* são extremamente rápidas. Em alguns casos as *tags* podem ser lidas em menos de 100 milissegundos.
- O modo RFID é bastante barato.

2.2.3. Desvantagens do RFID

O RFID tem as seguintes desvantagens:

- O elevado custo da tecnologia RFID quando comparada aos sistemas de código de barras.
- O preço final dos produtos que sobe imenso, devido ao facto da estrutura necessitar de antenas, leitores, etc.
- O ataque à privacidade dos consumidores uma vez que as etiquetas dos produtos são monitorizadas.
- O alcance das antenas pode ser reduzido se forem utilizados materiais condutivos, uma vez que a operação é baseada em campos magnéticos.
- As frequências utilizadas devem ter um certo padrão para que possam ser lidas por toda a indústria de maneira uniforme.
- Risco de ocorrer o *Reader collision*. Este ocorre quando os sinais de dois ou mais leitores se sobrepõem. Como a *tag* não consegue responder a dois leitores ao mesmo tempo ocorre este problema.

- *Tag collision*- ocorre quando as *tags* estão muito juntas. Este problema pode ser resolvido se for mandado a cada *tag* responder na sua vez.

2.3. Tecnologias disponíveis

Neste subcapítulo discutir-se-ão as principais ferramentas disponíveis para montar um sistema RFID e as suas principais características.

2.3.1. Transmissores universais ISM FSK

Como transmissores universais estavam disponíveis os seguintes componentes:

2.3.1.1- IA4221

É um chip de baixa potência cujo transmissor de multicanais FSK é usado em aplicações que requeiram o uso das normas FCC ou ETSI conforme as bandas 433, 868 e 915 MHz. Este oferece uma potência de saída muito alta e melhora o ruído da fase. Possui também um mecanismo de PLL, que permite usar multicanais em qualquer banda. Tem ativação automática para aplicações de baixo consumo de potência. Possui dois modos de trabalho que são o modo *Microcontroller* e o modo *EEPROM*.

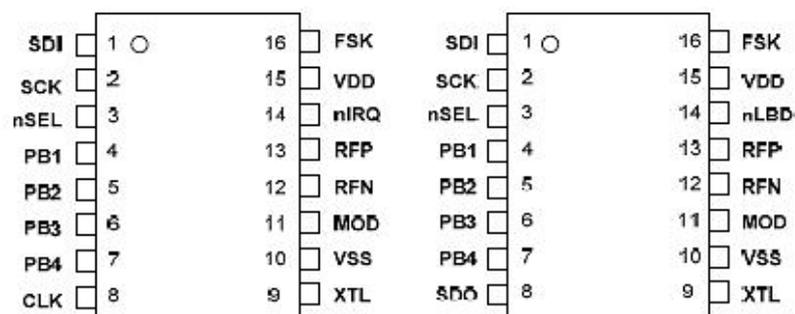


Figura 2-6 -Pinos dos integrados em modo *Microcontroller* à esquerda e *EEPROM* à direita [7]

Quando se está a trabalhar com aplicações simples o chip digital providencia o transmissor, que está diretamente ligado à interface, não sendo necessário nestes casos um microcontrolador externo. Quando, por outro lado, lidamos com aplicações mais complexas devemos utilizar o modo *Microcontroller*, para ter um microcontrolador externo que nos irá ajudar a armazenar informação e a executar operações mais complexas.

O integrado possui um temporizador de *wake-up* com um consumo de corrente bastante baixo e recalibração automática a cada 30 segundos.

Este é usado em conjunto com o IA4320 que será descrito abaixo.

2.3.1.2- IA4220

Funciona da mesma forma que o IA4221, tendo por isso a mesma estrutura, embora funcione com as larguras de banda de 315,433,868 e 915 MHz. No sistema de *wake-up* este tem a particularidade de utilizar um cristal mas que não reinicia a cada 30 segundos, reiniciando apenas no início.

2.3.1.3- IA4320

É usado em conjunto com o IA4220/21 funcionando às mesmas larguras de banda que estes. Tem dois modos de funcionamento: o modo *Microcontroller* e o modo *Standalone*. Este último modo permite receber todos os dados e controlar as quatro saídas digitais baseadas na entrada de dados sem um microcontrolador.



Figura 2-7-Pinos do integrado em modo *Standalone* à esquerda e *Microcontroller* à direita [5]

2.3.1.4- IA4420

Funciona nas mesmas larguras de banda que o IA4220. Este reduz o carregamento do microcontrolador com os recursos de processamento de dados digitais: filtragem de dados, recuperação de relógio, reconhecimento de dados padrão, integração de registo de dados FIFO e TX. Para aplicações de baixa potência o IA4420 suporta uma operação de *duty-cycle* baseada no temporizador interno *wake-up*.

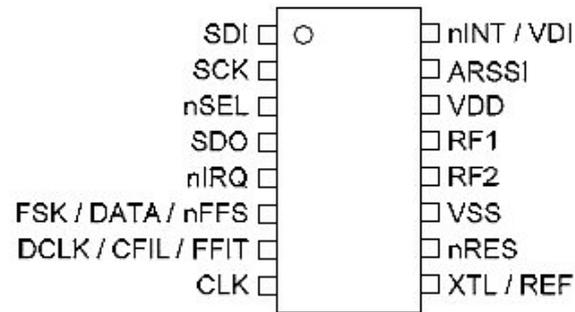


Figura 2-8-Pinos do integrado em modo REVC e *later* [4]

2.3.1.5- IA4421

Funciona nas mesmas larguras de banda que o IA4221, mas possui as mesmas características de funcionamento que o IA4420, possuindo a mesma estrutura.

2.3.2. Alpha RF Transceiver

Este *transceiver* é um dispositivo bastante acessível de alta performance em modulações rádio, possuindo um baixo consumo.

Tem um consumo de tensão entre 2.2 e 3.8 V e um cristal de 10Mhz com um sistema PLL.



Figura 2-9-Alpha RF Transceiver [3]

A corrente em *standby* deste dispositivo é inferior a 0.3uA, possuindo 16 bits para receber os dados.

O Tx (ligação para escrever mensagens) tem uma frequência programável de 15 a 240Khz e o RX (ligação para ler mensagens) tem a frequência programável de 67 a 400KHz.

O módulo pode acordar intermitentemente e fornecer controlo direto para o microcontrolador o que o faz perfeito para aplicações de bateria.

Estes módulos podem ser usados em aplicações de monitorização remota, segurança, localização e inventário de determinados objetos.

Embora os módulos Alpha possuam o custo de 13 euros o par, estes são bastante efetivos em performance de módulos de rádio, podendo transmitir e receber até 115Kbps, a uma distância máxima de 300m.

Como opera entre 2 a 5V o módulo da monitorização e a bateria podem adormecer quando se trata de aplicações de baixo consumo.

A transmissão e receção neste módulo funcionam de acordo com os seguintes diagramas:

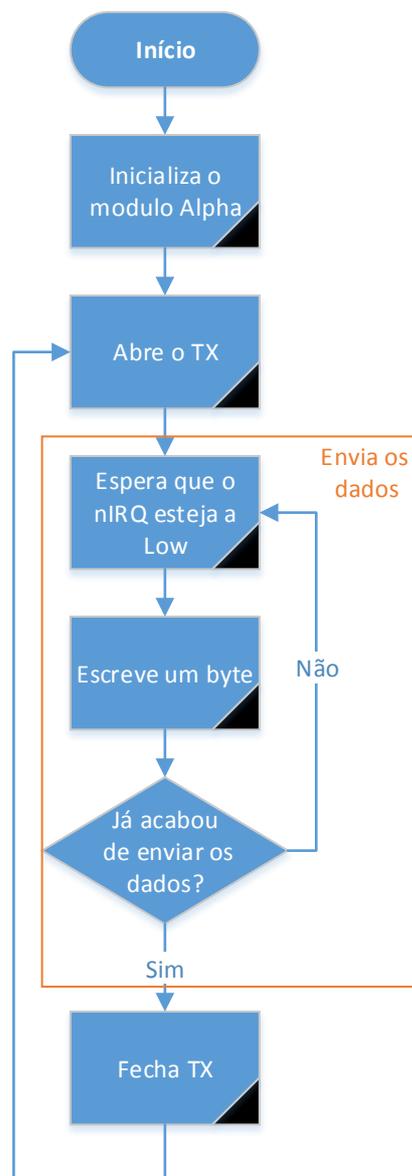


Figura 2-10-Diagrama com a explicação da transmissão com o *Alpha RF Transceiver*

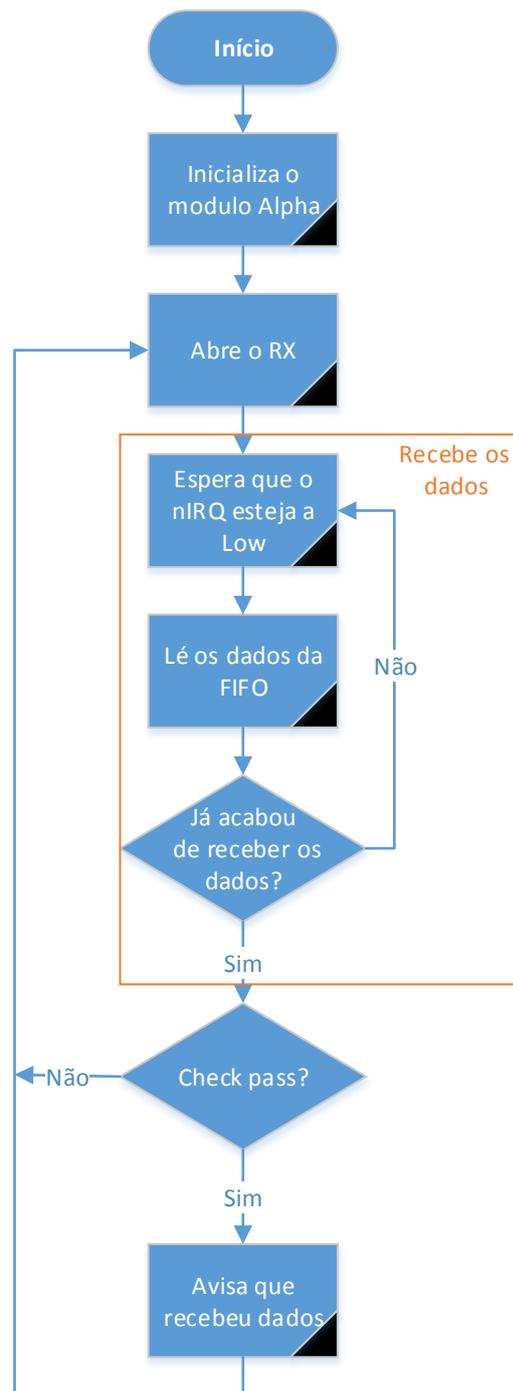


Figura 2-11-Diagrama com a explicação da receção com o Alpha RF Transceiver

2.3.3. Tag-it

A *tag-it* da *Texas Instruments* é um *transponder* que funciona à frequência de 13.56MHz a uma distância máxima de 1 metro.

Estas *tags* são muito utilizadas para a identificação de um produto e para armazenamento de um determinado artigo ou objeto.

Poderiam ter sido escolhidas duas *tags* que funcionam pelas condições descritas acima, a RI-I01-0110A-00 e a RI-I02-0110A-00.

A decisão da utilização do *transponder* foi feita à custa de dois pontos:

- A performance do *transponder* face a um metal
- Os danos causados quando um segundo *transponder* é colocado.

A performance de um *transponder* face a um metal depende da distância entre o *transponder* e o metal, bem como o tamanho e o tipo do mesmo. Na figura abaixo está- ilustrado o comportamento dos dois *transponders* face à distância e ao metal:

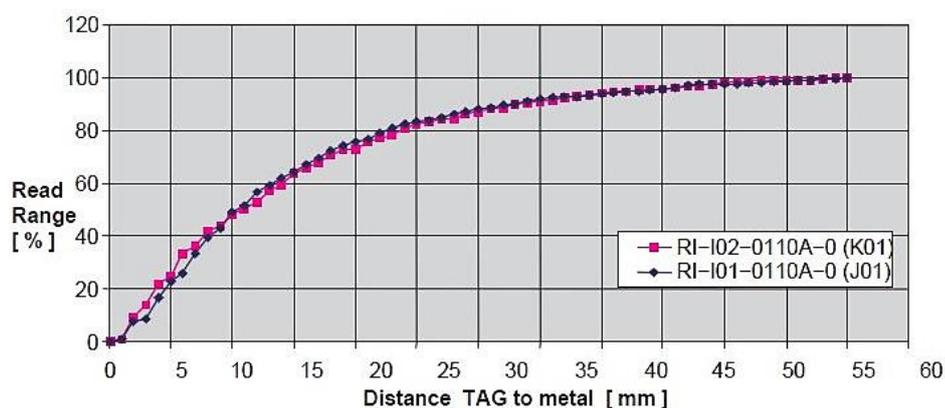


Figura 2-12-Efeito do metal pelas diferentes *tags* [22]

O eixo vertical simboliza o alcance da leitura relativamente ao leitor enquanto o eixo horizontal simboliza a distância da *tag* ao metal em teste.

Como na imagem acima não se conseguiu verificar com grande impacto qual a melhor etiqueta face ao metal, avaliaram-se os efeitos sofridos de um *transponder* quando um segundo é colocado. A resposta dos dois *transponders* a estas condições encontram-se ilustrados na figura abaixo:

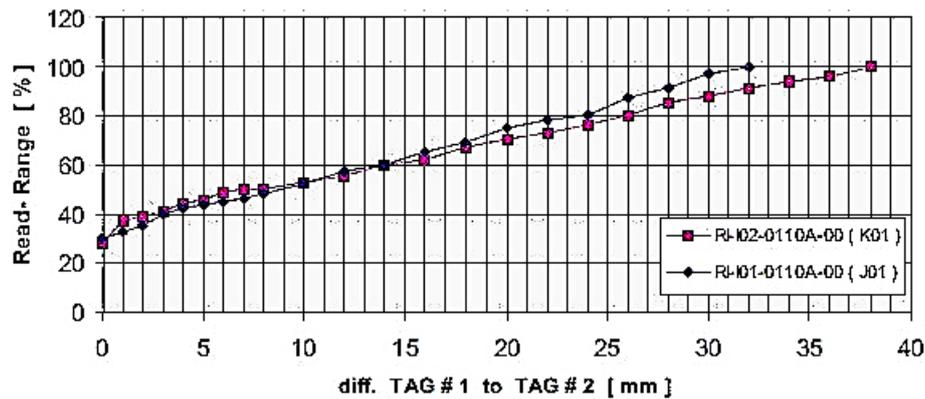


Figura 2-13-Efeito das *tags* quando um segundo *transponder* é colocado (teste do código SID) [22]

O eixo vertical simboliza o alcance da leitura relativamente ao leitor, enquanto o eixo horizontal simboliza a distância da *tag* ao metal em teste.

Tal como se pode observar pela figura a *tag* RI-I01 é bastante mais afetada que a *tag* RI-I02.

Como a melhor performance de leitura é alcançada pela RI-I02-0110A-00 resolveu-se optar pela utilização da mesma.

A imagem do *transponder* RI-I02-0110A-00 está ilustrada na figura abaixo:



Figura 2-14-*Transponder* RI-I02-0110A-00 [9]

Esta *tag* é usada em conjunto com a ISO/IES 15693 e a ISO/IEC 1800-3.

A *tag* funciona de modo passivo, isto é não responde ao leitor sem um *request* do mesmo, quer seja em modo endereçado ou em modo não endereçado.

Possui um circuito de ressonância montado sobre uma folha de PET com um *flip-chip* montado no *microchip*. A TI usa o condensador de forma a individualizar cada dispositivo para uma certa frequência de ressonância.

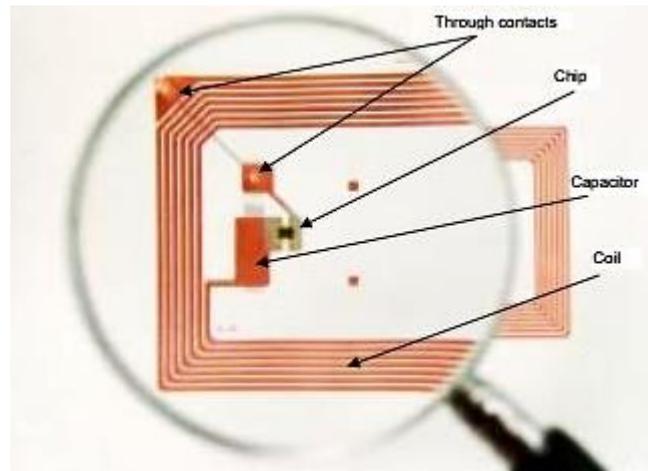


Figura 2-15-Constituição da tag [45]

A tag é utilizada em sistemas de baixa potência e *full duplex* à frequência de 13.56Mhz.

Esta tag tem de responder a um leitor.

Pode-se usar modulação ASK com uma sub-portadora ou FSK para duas sub-portadoras.

O *transponder* vai responder sempre no modo em que é interrogado pelo leitor.

Cada *transponder* pode ser utilizado de forma individual pois cada um tem um UID único.

Também foi estabelecido na norma da ISO 15693 um mecanismo que resolve as colisões quando são usadas tags múltiplas.

Além de todas as funcionalidades dadas pela ISO 15693 esta Tag tem como funcionalidades extra:

- O facto de se poder utilizar um segundo bit de bloqueio por bloco, designado por “*Factory Lock*”. Este *lock* permite que cada bloco possa ser bloqueado durante a produção.
- O sistema da TI permite a combinação dos comandos de inventário dados pela ISO 15693 com outros comandos.

A tag tem uma dimensão de 38mm por 22.5 mm e possui para a comunicação com o leitor uma pequena antena incorporada.

O acesso de memória ao utilizador é de 2048 bits e está organizada em 64 blocos usando uma tecnologia não volátil, EEPROM.

Cada bloco é programado separadamente quer pelo fabricante, quer pelo utilizador e pode ser protegido contra escrita. Quando os dados estiverem protegidos não poderão ser alterados sem a introdução da *password* correta.

Estas *tags* possuem um código UID (*Unique Identification*) e usam o terceiro *byte* mais significativo para identificar o ID (fig. 19). O UID está armazenado em dois blocos (64 bits).

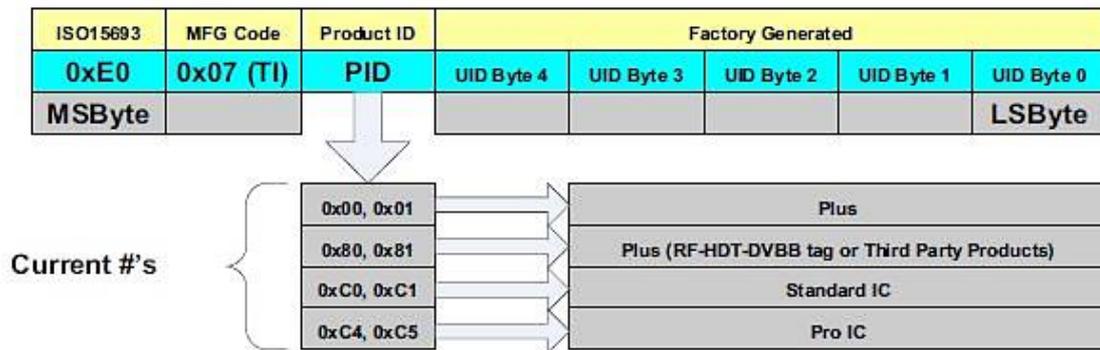


Figura 2-16-*Tag-it* HF-I codificação UID [23]

Desses dois blocos um bloco vai ser utilizado para armazenar o DSFID (Data Storage ID) e outro será usado para armazenar o AFI (Aplication Family ID).

É também usado um grande espaço de memória para armazenar a referência IC e a informação da memória física.

2.3.4. TRF7960

O integrado TRF7960/61 é o leitor utilizado em conjunto com esta mesma *tag*. Este responde à mesma frequência da *tag* 13.56Mhz.

O leitor é configurado através da seleção do protocolo desejado pelos registros de controlo, possuindo dois modos de funcionamento: o modo paralelo e o modo série.

O integrado TRF7960 é um transmissor capaz de modelar em ASK ou OOK e um recetor que desmodula em AM ou PM.



Figura 2-17-TRF7960 [10]

O sistema de leitura completo é constituído pelo TRF7960 e o microcontrolador MSP430.

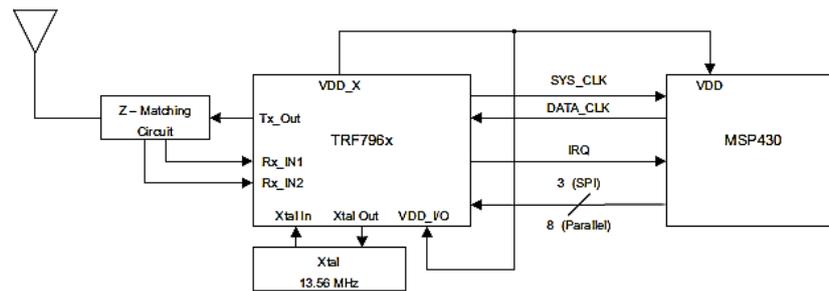


Figura 2-18-Sistema de leitura completo para a leitura da tag [11]

As funções de transmissão e recepção de dados usam *encoders* e *decoders* internos com um registo FIFO de 12 bytes.

O sistema de recepção dos dados permite desmodulação AM e PM, através da arquitetura de *dual-input* e um aumento automático do ganho. O tamanho do sinal recebido da modulação AM e PM está acessível pelo registo RSSI.

Este sistema inclui também um sistema de *framing*, CRC e paridade para detetar possíveis erros.

O TRF7960 fornece ao microcontrolador dados de 1.8 a 5.5V e também um relógio de sincronização de dados provisório.

2.3.4.1- Modos de energia

O TRF7960 tem 7 modos de energia que são controlados pelos pinos EN e EN2 e 3 bits no registo do *chip status control*(00h).

O controlador principal é o EN que pode ser habilitado com qualquer valor desde 1.8V ao valor que se encontrar no VIN. Se o EN for habilitado então todos os reguladores do leitor estão ativados com a frequência de oscilação definida no SYS_CLK. O SYS_CLK é o relógio externo para o microcontrolador.

O controlador auxiliar EN2 tem duas funções:

1. Se for ligado o EN2 diretamente ao V_{IN} é garantida uma energia regular ao VDD_X e um sinal de relógio auxiliar de 60KHz, para a saída SYS_CLK. Este modo é utilizado quando se quer que o Microcontrolador, que controla o leitor, esteja a ser alimentado pelo VDD_X e a receber o sinal de relógio pela saída SYS_CLK do leitor. Isto permite que a alimentação e o relógio estejam sempre disponíveis durante o tempo em que este esteja desligado.

2. A segunda função é habilitar o mecanismo de arranque do leitor quando este está desligado ($EN=0$, $EN2=0$). Neste caso o leitor está a ser controlado por um MCU ou um outro sistema que não possui fornecimento de energia quando está desligado.

Quando o EN é ativado, ou quando é feita uma mudança de estado de 0 para 1 e é confirmado que o EN é igual a 1, os reguladores de energia são ativados e o oscilador de 13.56MHz é ativado. Neste caso o SYS_CLK é modificado para a frequência de oscilação do cristal. Quando isto acontece o leitor está pronto para comunicar e fazer as suas respetivas tarefas.

O modo de operação é também controlado pelo MCU que vai escrever nos bits indicados do *chip status control* e escolher o modo de operação.

- Modo *standby*- Se puser o bit 7 a 1 consegue-se operar com o mínimo consumo de energia possível, o que no neste caso é bastante importante.
- Modo ativo com a desabilitação da secção RF - o segundo modo de baixo consumo é ter o bit 5 e bit 1 a 0.
- O modo ativo com apenas a secção de receção do RF ativa - colocando o bit 1 a 1.
- O modo ativo com todas as secções do RF ativas- é conseguido colocando o bit 5 a 1 e é o módulo mais usado nas aplicações de leitura e escrita.

2.3.4.2- Receção dos dados

A receção é feita com base em dois recetores, o RX_IN1 no pino 8 e o RX_IN2 no pino 9.

Estas duas entradas encontram-se multiplexadas em dois canais de receção: o canal principal e o canal auxiliar controlados pelo B3 no *chip status*.

A função por defeito do canal auxiliar é de medir o RSSI na modulação do sinal. Este canal também possui uma deteção de radiofrequência, ganho, filtragem com AGC e blocos RSSI.

Por defeito o RX_IN1 encontra-se conectado ao canal principal e o RX_IN2 ligado ao canal auxiliar. Quando é detetada uma resposta da *tag* pelo RSSI os valores das duas entradas irão ser medidos e armazenados no registo *RSSI Level*.

A amplitude de radiofrequência para valores superiores a 3.3V no RX_IN1 e no RX_IN2 deve ser aproximadamente 3V_{PP}. Como neste caso se está a trabalhar a 5V este ponto tem de ser garantido.

O RSSI faz *Reset* enquanto está a ler de forma a conseguir um *update* do valor para cada nova *tag*.

Parte digital

Na parte digital o sinal é codificado, é transformado em dados série e os dados de relógio são extraídos.

O registo principal na parte digital é o ISO *control*. Ao escrever neste registo o utilizador seleciona o protocolo que quer utilizar, tal como será mencionado à frente.

Framing

Neste processo irão ser retirados os bits especiais tais como: o início de trama, final de trama, etc. Os bits de paridade e do CRC serão validados e após a correta validação removidos. O resultado final é enviado para o micro, que irá enviar para a pilha de 12 bytes.

A *framing* também deteta a colisão entre bits. Quando é detetada uma colisão é mandado um pedido de interrupção para o IRQ e o registo de estado.

Para ser controlado o *time-out* de uma determinada *tag* a recepção tem ao seu dispor dois timers:

- O *RX wait-time*, controlado pelo valor do registo *RX wait time*. Este tem como função definir o tempo depois da transmissão, cujos *decoders* de receção não estão ativos, o que faz com que não ocorra deteções erradas resultantes da operação de transmissão.
- *RX no-response*, que é controlado pelo registo *RX no response wait time*. Este tem como função medir o tempo do início do *slot* da sequência de anticolisão até ao início da resposta da *tag*. Se uma *tag* responder dentro do tempo ativamos essa *tag*.

2.3.4.3-Transmissão dos dados

O transmissor consiste num oscilador de 13.56Mhz, um protocolo de processamento digital e um estado de saída RF.

O cristal oscilador de 13.56Mhz gera diretamente a frequência RF para o estado de saída, o sinal de relógio para a parte digital e para o Sys_Clk, que é usado pelo micro externo.

O tamanho do pulso de modulação é definido pelo protocolo selecionado no registo de controlo ISO. Com uma antena *high-Q* a modulação do pulso é normalmente prolongada e a *tag* deteta um pulso maior do que o que era pretendido. Nestes casos, o tamanho da modulação do pulso pode ser corrigido através do uso do registo de tamanho do pulso TX. Se este registo só tiver zeros então o tamanho do pulso é controlado pelo protocolo selecionado, senão o tamanho do pulso é igual ao valor do registo. Isto significa que o alcance do ajustamento pode ser entre 73.7ns e 29.8µs.

A transmissão no modo digital é muito similar com a receção neste modo sendo que, em primeiro lugar, a pilha deve ser limpa e só depois é que o micro manda o leitor escrever todos os seus dados. Se os dados forem demasiados o micro vai emitir um aviso, de que a pilha está cheia, no IRQ e passa para o próximo pacote da pilha. Os bits adicionais irão ser guardados no bit 0, sendo o número de bits indicados nos bits de B1-B3.

A transmissão dos dados é inicializada com um comando de seleção, descritos na tabela 2.4. O microcontrolador comanda o leitor para que este faça um comando de escrita continua a partir do registo 1Dh. Os dados que são escritos no registo 1Dh correspondem ao tamanho do Tx no byte 1, *nibbles* altos e baixos, enquanto no registo 1Eh são escritos o byte 2 do tamanho do TX, *nibbles* baixos e quebrados. Este tem como responsabilidade determinar quando o leitor manda o byte de fim de ficheiro. Depois de alocar o tamanho do TX a FIFO de dados é carregada para o registo 1Fh com o armazenamento dos bytes de 0 a 11. A transmissão de dados começa automaticamente depois do primeiro byte ter sido escrito na FIFO. O tamanho do TX em bytes e a FIFO podem ser carregados com um comando de escrita contínua, devido ao facto do endereço de armazenamento ser contínuo.

O Tx usa dois registos:

- Tx Timer H-Byte, onde os bits B7 e B6 são usados para definir as condições do *trigger*. Os restantes 6 bits são os bits dos endereços mais significativos.
- Tx timer L-byte, onde os seus bits possuem os endereços menos significativos.

2.3.4.4- Modo direto

O modo direto é programado através do bit 6 do registo do ISO, o que permite ao utilizador configurar o leitor de duas maneiras:

- Modo direto 0- permite ao utilizador utilizar apenas as funções *front-end*, o que contorna a implementação do protocolo no leitor. Neste modo o utilizador tem acesso à modulação de transmissão pelo MOD (pino 14) e ao sinal da sub-portadora no I/O_6 (pino 23).
- Modo direto 1- é usado para que o utilizador possa implementar um protocolo cuja codificação seja igual à que foi implementada pelo leitor, mas com um formato de *framing* diferente. Neste modo a saída do recetor não possui o sinal da sub-portadora, mas sim a *stream* de bits em série descodificada e os sinais de relógio. Os dados série encontram-se disponíveis no I/O_6 (pino 23) e o bit de relógio no I/O_25 (pino 22).

O modo direto começa mal o bit 6 do registo ISO fique a 1. O comando de escrita não deve ser terminado com uma condição de paragem, *Stop bit*, porque esta condição de paragem termina o modo direto e limpa o bit 6. No modo direto não se pode utilizar a comunicação em paralelo, sendo os dois pinos mais utilizados o I/O_6 e o I/O_5.

As configurações que estão disponíveis para o modo direto são:

- Modo 0, onde o leitor usa apenas AFE e o protocolo de manuseamento é ignorado.
- Modo 1, em que não é feito o *framing*, mas está presente o SOF e o EOF. Isto permite ao utilizador seleccionar o nível de *framing*, baseado num *standard* ISO.
- Modo 2, onde os dados estão no formato *ISO-Standard*. Neste modo o SOF, EOF e o *error checking* são removidos, para que o microprocessador receba apenas os bytes de dados pela FIFO de 12 bytes.

2.3.4.5- Interface de comunicação do leitor

A interface de comunicação com o leitor pode ser configurada de duas formas:

- 8 pinos de interface paralela e um Data_CLK.
- Interface série periférica.

Estes modos são exclusivos, sendo que em cada aplicação só se pode escolher um deles.

Se escolhermos a interface SPI não serão utilizados os pinos I/O_2, I/O_1 e I/O_0, pelo que estes devem estar ligados de acordo com a tabela 2.1. Quando o leitor é iniciado verifica-se o estado destes três pinos e entra-se no modo correspondente do SPI.

Tabela 2-1- Pinos na interface paralelo e série no modo de conexão ou direto [11]

Pino	Paralelo	Paralelo-Direto	SPI com SS	SPI sem SS
DATA_CLK	DATA_CLK	DATA_CLK	DATA_CLK do master	
I/O_7	A/D[7]		MOSI-entrada de dados no leitor – saída do micro	
I/O_6	A/D[6]	Modo direto, saída dos dados	MISO-saída de dados do leitor - entrada do micro	
I/O_5	A/D[5]	Modo direto, strobe- saída de relógio	Só é usado para informação quando os dados são colocados fora do chip. Normalmente este pino não é usado no SPI.	

<i>I/O_4</i>	A/D[4]		SS- <i>slave select</i> - ativo a 0	
<i>I/O_3</i>	A/D[3]			
<i>I/O_2</i>	A/D[2]		VDD	
<i>I/O_1</i>	A/D[1]		VDD	VSS
<i>I/O_0</i>	A/D[0]		VSS	
<i>IRQ</i>	Interrupção IRQ	Interrupção IRQ	Interrupção IRQ	

O leitor é sempre o *slave*, enquanto que o micro é sempre o *master*. O MCU inicia todas as comunicações com o leitor, sendo este também usado para comunicar com as camadas de nível mais alto. O leitor tem o pino IRQ para chamar a atenção do MCU sempre que detetar uma resposta de uma *tag* num campo próximo.

A comunicação é iniciada com uma condição de início seguida por uma palavra de endereço ou comando.

O bit mais significativo, bit 7, vai determinar se a palavra utilizada é um comando ou um endereço.

Quando é enviada essa palavra é que se podem enviar os dados, sendo que no modo *continuous-Address*, colocando o modo contínuo a 1, os primeiros dados que seguem o endereço são de escrita, ou leitura, para, ou de, o respetivo endereço. Quando se quer enviar dados adicionais o endereço é incrementado por um. O modo contínuo pode ser usado para escrever um bloco de registos de controlo de uma simples *stream* sem mudar o endereço. No modo de endereçamento não contínuo é esperado apenas uma palavra de dados após o endereço.

O modo de endereçamento é usado para escrever e ler configurações de registos, ou da pilha.

Quando se quer escrever mais de 12 bytes na FIFO o modo de endereçamento contínuo é posto a 1.

O modo de comando é usado para se introduzir um comando, resultante de uma ação do leitor: inicialização, transmissão, permitir a leitura, etc.

A comunicação entre o MCU e o leitor deve ser feita a partir das seguintes tramas:

Tabela 2-2-Trama no modo de endereçamento contínuo [11]

<i>Start bit</i>	Endereço X	Dado (X)	Dado (X+1)	Dado (X+2)	Dado (X+3)	Dado (X+4)	...	Dado (X+n)	<i>Stop bit contínuo</i>
------------------	---------------	-------------	---------------	---------------	---------------	---------------	-----	---------------	--------------------------

Tabela 2-3-Trama no modo de endereçamento não contínuo [11]

<i>Start bit</i>	Endereço X	Dado (X)	Endereço Y	Dado (Y)	...	Endereço (Z)	Dado (Z)	<i>Stop bit single</i>
------------------	---------------	-------------	---------------	-------------	-----	-----------------	-------------	------------------------

Tabela 2-4-Trama no modo de comando [11]

<i>Start bit</i>	Comando X	(Dados ou comandos opcionais)	<i>Stop bit</i>
------------------	--------------	-------------------------------	-----------------

2.3.4.6-Interface paralela

Quando se usa a interface paralela a condição de início de transmissão de uma mensagem é feita quando o pino I/O_7 fica a 1, enquanto o *clock* se encontra ativo. Na figura abaixo pode-se visualizar este mesmo comportamento.

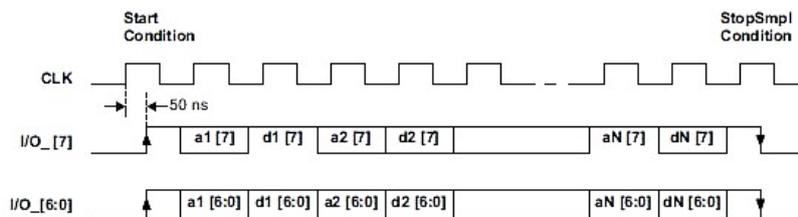


Figura 2-19-Comunicação em paralelo com *Stop Condition* simples [11]

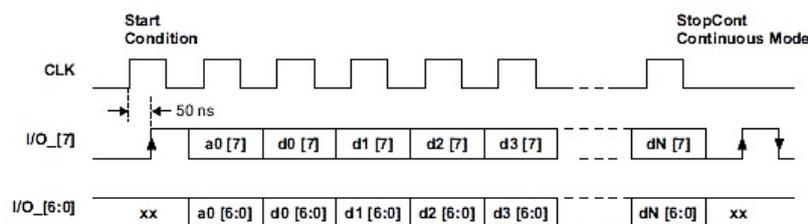


Figura 2-20-Comunicação em paralelo com *Stop Condition* contínuo [11]

Nas figuras acima está ilustrado o comportamento da transmissão de uma mensagem quando existe um *Stop* simples e quando existe um *Stop* contínuo.

A diferença entre estas duas é visível a nível do *clock* pois, enquanto que no *StopSmpl* o *clock* tem de estar a 1 para que possa haver um *stop* da mensagem, no *StopCont* o *stop* é feito quando o *clock* está a 0. Este *StopCont* é feito para que seja efetuado um *Reset* total da interface paralela e para ser preparada uma nova comunicação.

2.3.4.6.1- Receção de dados em paralelo

A receção dos dados em paralelo é feita quando é detetado um início de uma trama através do SOF. Quando é detetado este início são elaborados os seguintes passos:

- O bit 6 do registo IRQ é ativado e é enviado um pedido de interrupção ao MCU. Este fenómeno ocorre no final da receção dos dados e apenas se estes mesmos dados forem menores ou iguais a 8 bytes.
- O MCU vai receber o pedido de interrupção e vai verificar a razão da mesma através da leitura do registo de estado IRQ. No final o MCU irá ler os dados da FIFO.
- Se o pacote for maior que 8 bytes, a interrupção é mandada antes do final da operação de receber os dados isto é, quando o nono byte estiver a ser lido para a FIFO. O MCU neste caso deve voltar novamente a ler o registo de estado IRQ. Quando a FIFO está 75% cheia é marcado no registo B5 do IRQ este mesmo acontecimento e o MCU deve responder lendo a FIFO, de forma a arranjar algum espaço para os próximos dados.
- Quando a operação de receção dos dados estiver terminada é mandada uma nova interrupção para o MCU, que deverá verificar quantas palavras estão ainda presentes na FIFO antes do mesmo terminar a leitura.
- Se o leitor detetar um erro ao receber os dados é acionada a *flag* correspondente no registo de estado IRQ.

2.3.4.6.2- Transmissão de dados em paralelo

A transmissão dos dados deve ser feita de acordo com os seguintes passos:

- Antes de ser iniciada a transmissão dos dados em paralelo a FIFO deve ser limpa, através do uso de um comando *Reset*.
- A transmissão de dados é iniciada com um dos comandos de código: *Software Initialization, Transmission with CRC*, etc.
- Depois deste comando o microcontrolador é então responsável por mandar o leitor escrever o tamanho do seu TX e os dados que irão ser guardados na FIFO. A transmissão dos dados começa automaticamente depois do primeiro byte ter sido escrito na FIFO.
- No início da transmissão o bit 7 do registo de estado IRQ é colocado a 1.
- Se os dados a serem transmitidos forem inferiores a 4 bytes a interrupção é mandada apenas no final da transmissão,

- Se o número for maior ou igual a 5 a interrupção é gerada. Este fenómeno também ocorre se o número de bytes for 3.
- O microcontrolador deve ter sempre em atenção o registo de estado IRQ e o registo de estado da FIFO. Caso seja necessário este deverá ler os dados adicionais da FIFO.
- No final da transmissão é mandada uma interrupção para informar o micro que a tarefa foi concluída.

2.3.4.7- Interface série

Se se pretender trabalhar com a interface série ter-se-á de utilizar os pinos I/O_0 I/O_1 e I/O_2. O leitor quando é ativado irá ver o estado destes mesmos pinos e caso estes não se encontrem todos na mesma configuração, todos a 1 ou todos a 0, o leitor entra numa das possibilidades do modo série: SPI sem *Slave Select* ou SPI com *Slave Select*.

A interface série trabalha da mesma forma que a interface paralela no que diz respeito à FIFO, exceto na seguinte condição:

- Quando o MCU está a receber o IRQ do leitor este vai ler o registo IRQ e determinar como irá servir o mesmo. Posteriormente, o MCU tem de fazer uma leitura fictícia, para poder limpar o registo de estado do IRQ do leitor, processo este denominado por *Dummy Reader*.

O leitor fictício tem de estar no modo SPI, porque o registo de estado IRQ necessita de um *clock* adicional para poder limpar o registo. Este procedimento não é requerido no modo paralelo, porque o *clock* adicional está incluído na condição de *stop*.

O procedimento para a realização de um leitor fictício é a seguinte:

- Inicializar o leitor fictício
 - Quando se está a usar o *Slave Select* é colocado o bit SS desativado.
 - Quando não se está a usar o SS a condição de início ocorre quando o SCLK está ativo.
- Envio da palavra de endereço para o registo de estado IRQ, quando o leitor e o modo de endereço contínuo são colocados a 1.
- Leitura de 1 byte do IRQ.
- Leitura fictícia de 1 byte do registo 0Fh.
- Paragem do leitor fictício
 - Quando se está a usar o *Slave Select* o SS é colocado a 1.

- Quando não se está a usar o SS, o acontecimento de *stop* é feito quando o SCLK está ativo.

2.3.4.7.1- Modo série sem o *slave* ativo.

Quando se utiliza este modo é necessário usar delimitadores para o início da transmissão e o final da mesma, *Start Condition* e *Stop Condition*. Entre estes delimitadores irão ser transferidos o endereço, os dados e as palavras de comando. Estas palavras devem ser de 8 bits sendo o MSB transmitido em primeiro lugar.

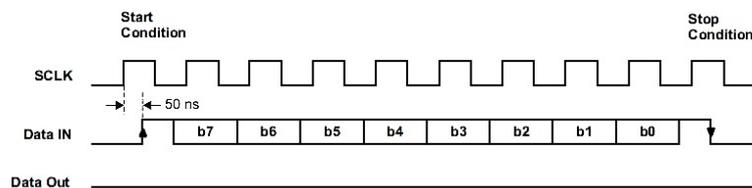


Figura 2-21-Comportamento da comunicação sem o *Slave* ativo [11]

Neste modo a chegada dos dados com o SCLK ativo faz com que aconteça um *Reset* à interface série para que o dispositivo fique preparado para receber os dados. Os dados apenas poderão ser modificados quando o SCLK está desativo, sendo o leitor quem controla este procedimento. A comunicação termina quando se vê uma condição de *stop*, ou quando ocorre uma falha de dados durante o período em que o SCLK está ativo.

2.3.4.7.2- Modo série com o *Slave* ativo.

Quando o sinal do *slave* se encontra ativo é realizado o *Reset* na interface.

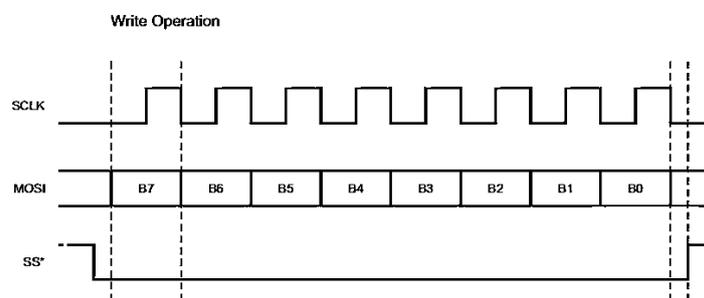


Figura 2-22-Comportamento da comunicação com o *slave* ativo (Modo de escrita) [11]

Os dados série MOSI mudam no próximo estado e são validados quando o leitor está ativado, como é possível ver pela figura acima. A comunicação termina quando o *slave* é desativado.

As palavras têm de ser de 8 bits e o MSB tem de ser transmitido primeiro.

2.3.4.8- FIFO

A FIFO é um registo de 12 bytes que está alocada no endereço 1Fh e vai de 0 a 11. Esta tem associados a ela 3 contadores e 3 registos de estado. O primeiro contador é um contador de 4 bits FIFO, que mantém o tráfego do número dos bytes dentro da mesma. O segundo contador com 12 bits indica o número de bytes que foram transmitidos na *frame* de dados. Estes contadores são usados de forma a garantir que o leitor saiba quando ocorre o final dos dados.

O FIFO *counter* é um contador responsável por contar o número de bytes que foram guardados na pilha. A FIFO pode ter mais que 12 bytes, porque o micro é avisado que existem mais dados. Para tal irão ser lidos alguns dados, que posteriormente irão ser apagados da pilha, de forma a dar espaço ao leitor para novos dados.

A FIFO irá ser responsável por armazenar o UID das *tags* lidas.

3. Especificação da solução RFID a desenvolver

3.1. Descrição dos cenários possíveis

A ideia adotada para o desenvolvimento do trabalho foi através do mecanismo de *tags* com o integrado TRF7960. Abaixo encontra-se ilustrado como se pode trabalhar com estes dois componentes.

3.1.1. Cenário 1



Figura 3-1-Inventário geral

Este cenário representa o cenário de um possível roubo de objetos. Neste caso representado por roubo de arquivos. Aqui será utilizado um leitor passivo que apenas recebe sinais de rádio das *tags* e caso não receba informação de uma delas fará soar um alarme.

Os *transponders* irão funcionar apenas com a energia fornecida pelo sinal de ativação, o que evita a utilização de uma bateria o que iria limitar o número de utilizações.

3.1.2. Cenário 2

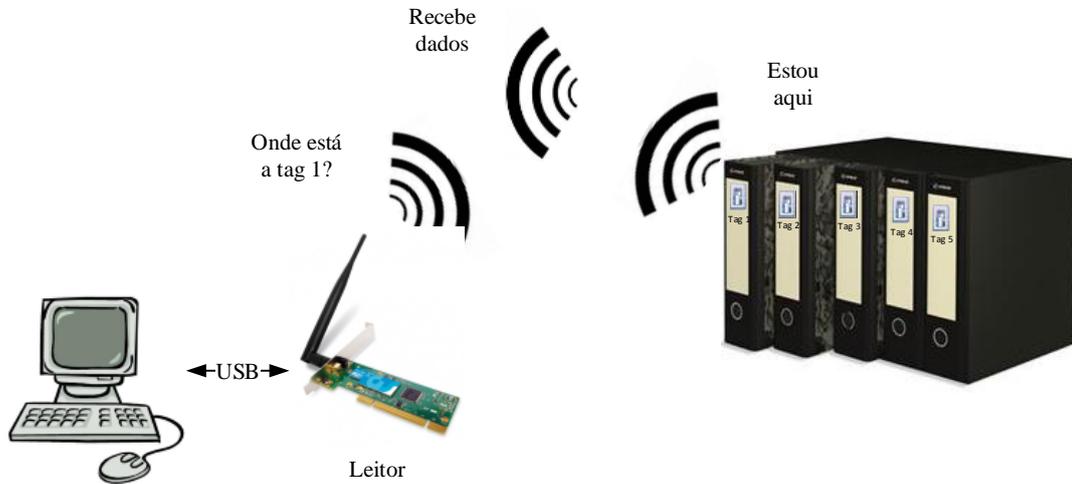


Figura 3-2-Inventário endereçado à máscara

No segundo cenário tenta-se encontrar um determinado objeto solicitado pelo utilizador. Para tal será utilizado um leitor ativo, que irá ativar a *tag* correspondente a uma determinada máscara introduzida pelo utilizador.

3.1.3. Cenário 3

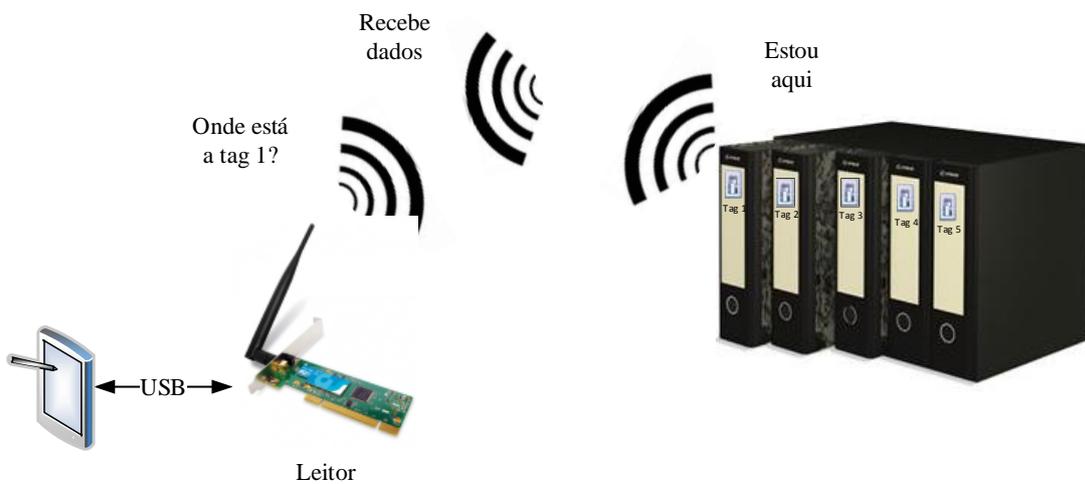


Figura 3-3-Inventário endereçado à máscara com dispositivo móvel

Este cenário é igual ao cenário 2, com a exceção de que em vez de se utilizar um computador, se utiliza um telemóvel *android* para procurar a *tag* em questão.

3.2. Funcionalidades do sistema

Uma próxima abordagem a este trabalho seria incluir o ecrã e o teclado junto com o leitor.

Não foi optada esta implementação pois presumiu-se que cada utilizador estaria mais habituado ao seu próprio telemóvel. Assim seria apenas necessário trabalhar com uma aplicação nova em vez de um novo dispositivo. Além de fazer com que a aplicação seja mais prática e leve do ponto de vista do utilizador.

3.3. Sistema central

O sistema central é responsável por armazenar a informação das *tags*, que se encontram dentro do campo RF na memória EEPROM do 8051. A informação é guardada num *array* de UIDS de *tags*. Sempre que o utilizador quiser programar este *array* de *tags* e alterá-lo para as *tags* existentes no espaço naquele instante, terá de fazer um inventário normal e as *tags* serão escritas.

Uma versão melhorada desta abordagem seria possuir uma base de dados com o histórico da *tag* com aquele UID. As *tags* que não se encontrassem no inventário não seriam apagadas mas sim seria colocada uma *flag* no campo da *tag* a dizer que esse item já não estava disponível.

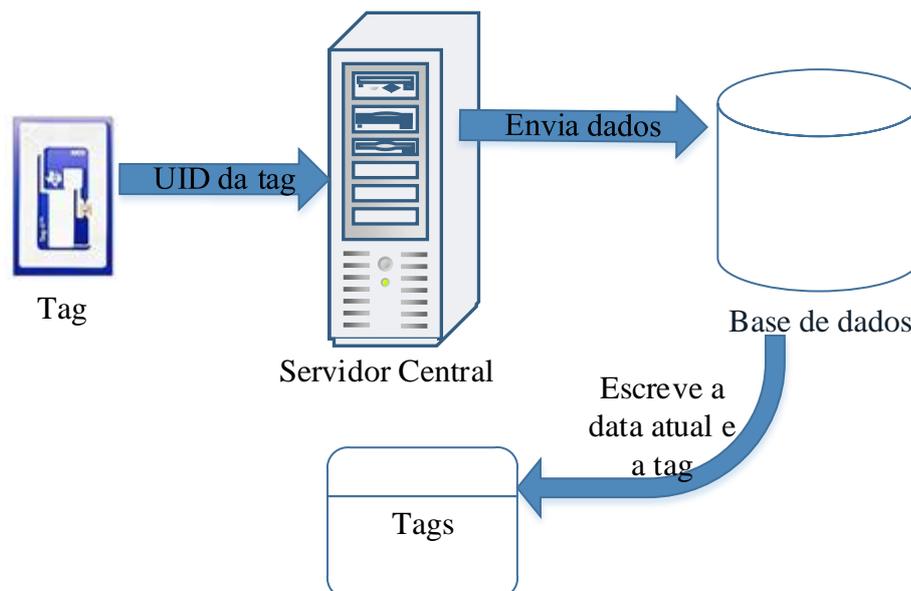


Figura 3-4- Cenário futuro de um sistema de histórico das *tags*

Esta *flag* quando é desativada sem permissão do administrador irá ativar um alarme para o exterior.

Caso o utilizador apenas quisesse encontrar aquela *tag* seria mandada a informação da localização para um nível mais acima. Sendo que a localização estava armazenada no registo RSSI, que mede a força do sinal e consequentemente num registo no microcontrolador.

Numa versão melhorada poderia possuir uma base de dados com a autenticação do utilizador que estaria a lidar com o sistema através do ID do computador, ou do ID do SIM do telemóvel que estaria a utilizar.

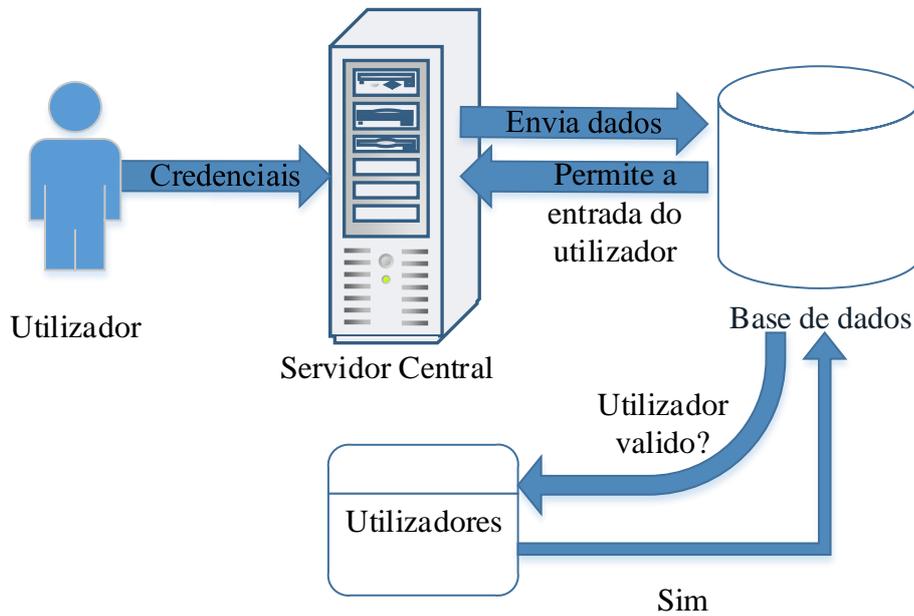


Figura 3-5- Cenário futuro de um sistema de validação do utilizador

Esta autenticação iria ser encriptada para que as trocas de informação com o sistema central fossem o mais seguras possíveis contra ataques de utilizadores externos maliciosos.

4. Projeto RFID

Neste capítulo será descrita a implementação de um sistema RFID idealizado. Este sistema teve por base duas componentes:

- A componente física que foi projetada, montada e colocada em funcionamento, denominada por *hardware*.
- A componente de *software* responsável por programar a componente anterior.

4.1. Hardware

A parte de *Hardware* ou componente física que foi projetada é constituída por 3 elementos fundamentais: O leitor, o microcontrolador e a antena construída para interagir com as *tags*.

4.1.1. Leitor

O leitor podia ser feito à custa de um de dois integrados: o TRF7960 e o TRF7961. A decisão entre eles foi feita com base na seguinte tabela:

Tabela 4-1-Diferenças entre o TRF7960 e o TRF7961 [11]

DISPOSITIVO	PROTOCOLOS					
	ISO1443A/B				ISO 15693	Tag-it TM
	106 kbps	212 kbps	424 kbps	848 kbps	ISO18000-3	
TRF7960	X	X	X	X	X	X
TRF7961					X	X

Ao olhar para a tabela constata-se que qualquer um dos dispositivos era adequado para a aplicação, mas como o TRF7960 se adequa a mais protocolos optou-se por este.

Na implementação do trabalho foi considerado como primeiro foco o esquema funcional do TRF7960 e a decisão das configurações que se poderiam utilizar.

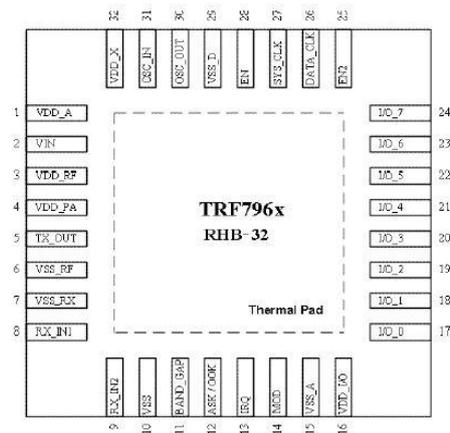


Figura 4-1-Esquema funcional do TRF796x [11]

As configurações adotadas para os pinos das alimentações foram as seguintes:

- V_{in} -5V
- V_{DD_A} -3.5V
- V_{DD_X} -3.4V
- V_{DD_RF} -4.3 a 5V
- V_{DD_PA} -liga-se ao V_{DD_RF}

- V_{SS} -0V
- V_{SS_A} -0V
- V_{SS_D} -0V
- V_{SS_RX} -0V

- $V_{DD_I/O}$ - está ligado ao V_{in} porque foi escolhida a opção de por o V_{in} a 5 V
- SYS_CLK - alimenta o micro com 13.56Mhz quando o EN está a 1
- $Data_CLK$ - entrada relógio para comunicação com o micro

Como foi decidido montar o leitor em modo SPI com SS os pinos I/O₀ a I/O₄ devem estar ligados da seguinte forma:

- I/O₀-ligado ao V_{SS}
- I/O₁-ligado ao V_{DD}
- I/O₂-ligado ao V_{DD}
- I/O₄-é o *Slave Select* (ativo a *low*)

O leitor será montado no modo série em vez do paralelo devido ao facto de, numa primeira aplicação ter poucos dados a entrar ao mesmo tempo. O esquema de montagem em que o protótipo foi baseado foi o seguinte:

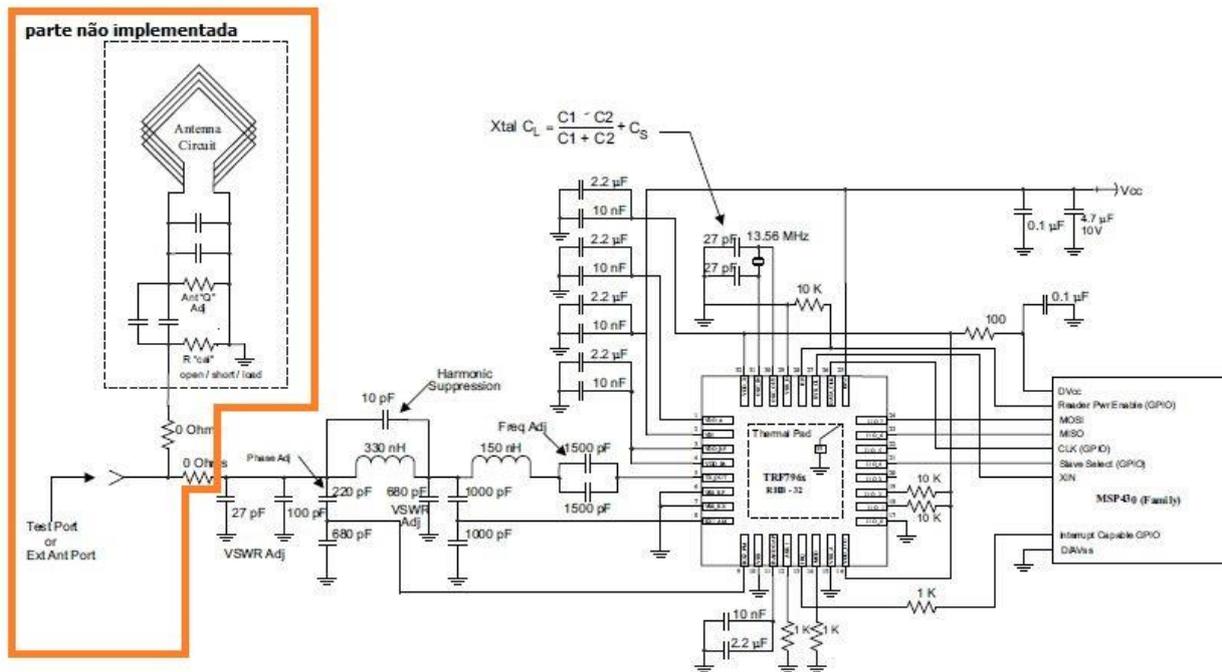


Figura 4-2-Esquema de montagem do leitor em modo série [11]

As modificações feitas a este esquema foram as seguintes:

- O microcontrolador escolhido foi outro, devido a fatores de preço e de utilização de material que já existiam.
- Não foi implementada a parte a laranja que corresponde ao esquema da antena.

A primeira alteração foi feita adicionando ao esquema da placa um conjunto de ligadores, que irão posteriormente ser ligados ao microcontrolador.

Devido às propriedades do integrado TRF7960 foi necessário a criação de um PCB que funciona como um dissipador de calor do mesmo.

Numa primeira fase foi construído o seguinte esquema:

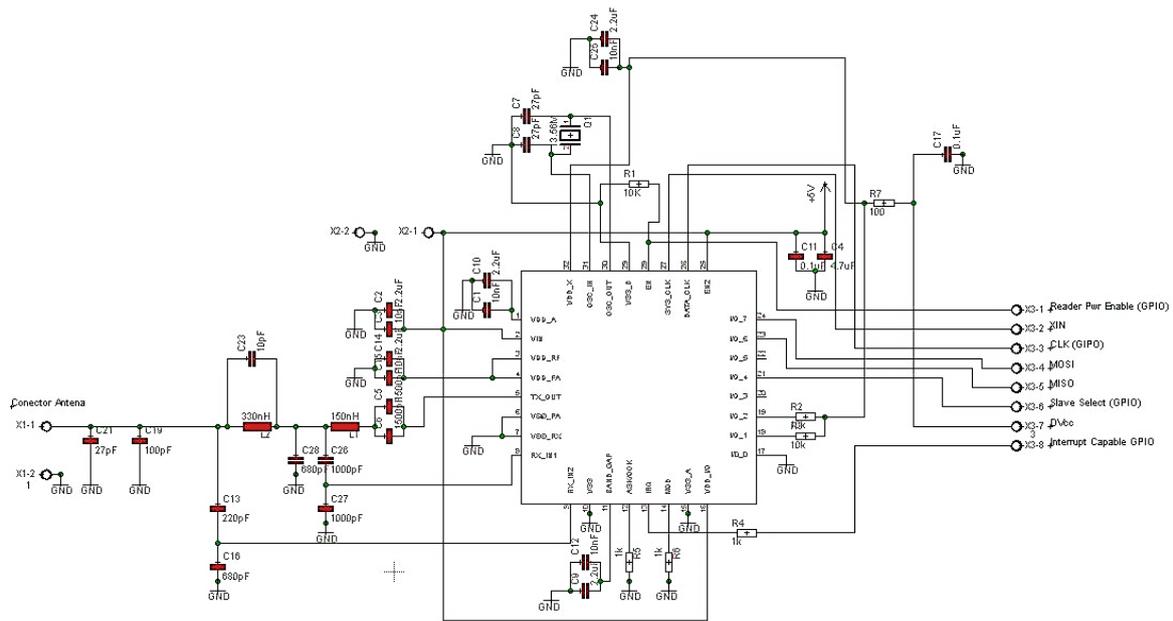


Figura 4-3-Esquema de montagem do leitor implementado

Tal como se pode observar pela figura acima foram colados 2 ligadores para fornecer o VCC e a massa ao circuito, 8 ligadores para ligar aos respetivos pinos do microcontrolador e 2 últimos ligadores para ligar à antena.

As diversas *boards* criadas encontram-se em anexo sendo aqui apresentada apenas a *board* final.

Nas figuras abaixo encontram-se ilustradas cada uma das faces da *board*, bem como as duas sobrepostas e a imagem final do PCB montado.

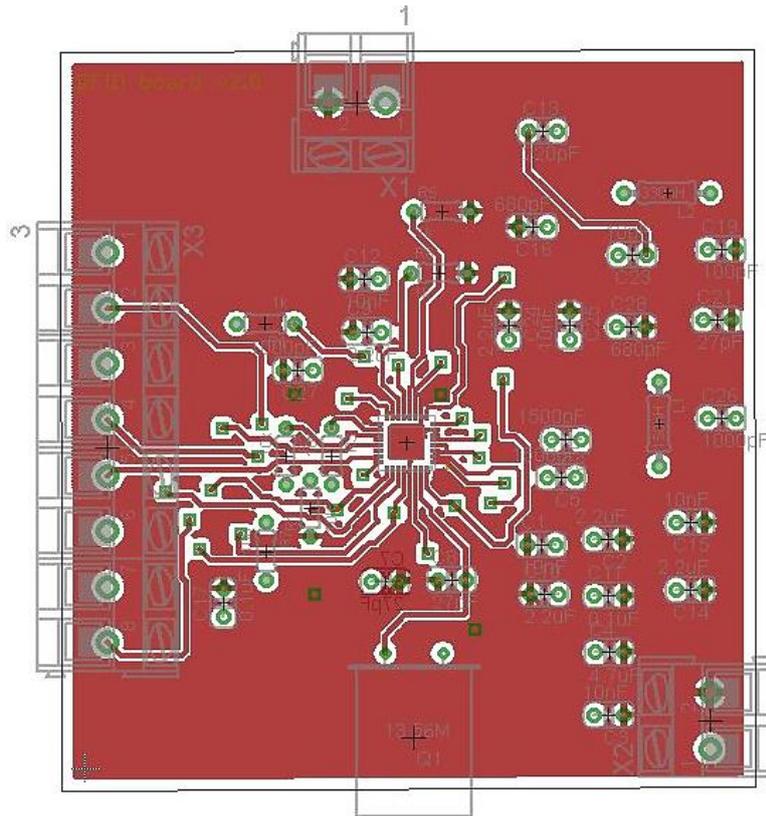


Figura 4-4-Face de cima do PCB final

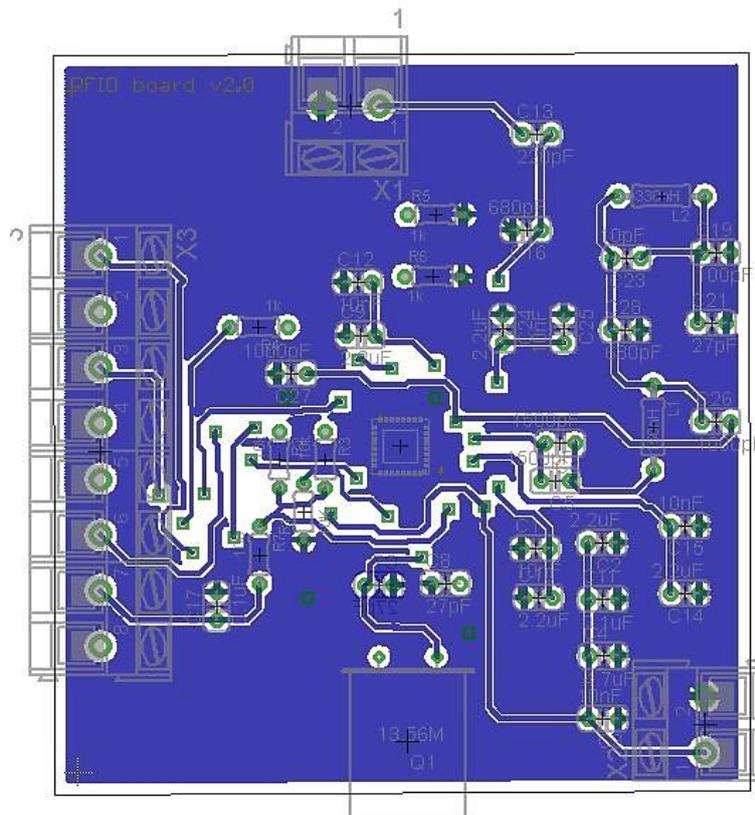


Figura 4-5-Face de baixo do PCB final

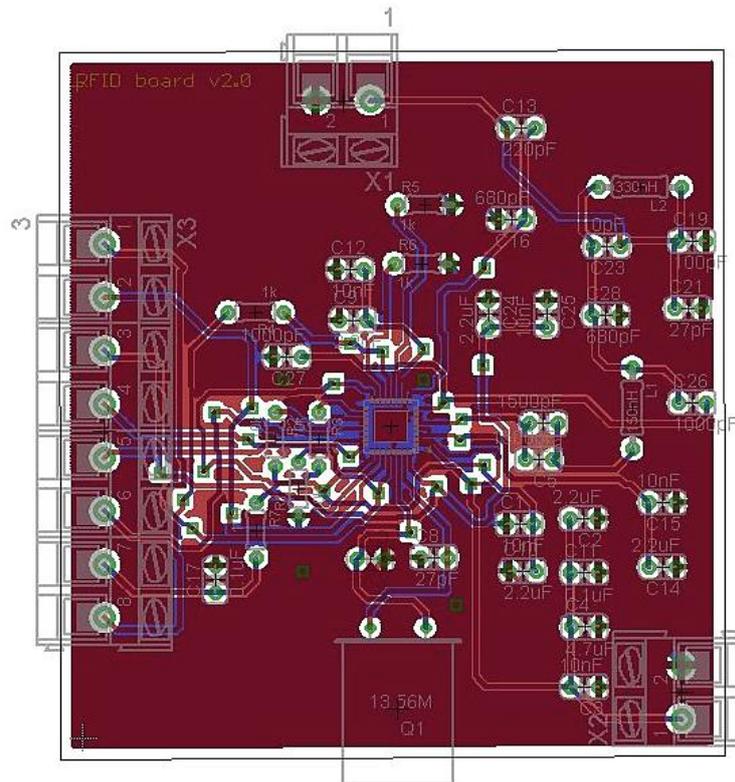


Figura 4-6-Board de duas faces final

Para desenhar o PCB poderiam ser utilizados dois programas, o *Eagle* e o *Pads*.

Foi optado o *Eagle* pois embora no *Pads* possa ser tudo configurado pelo utilizador, inclusive criar os nossos próprios componentes, o *Eagle* possui uma biblioteca mais vasta de componentes e faz as coisas de uma forma mais automática e simples para um principiante.

O leitor construído consegue ler no máximo 50 *tags* por segundo e funciona como um escravo do microcontrolador.

Montagem da placa

No desenvolvimento do leitor foi necessário a conceção de 2 bobines, pois os valores das mesmas eram demasiado pequenos (330 nH e 150 nH) para serem compradas nos sítios mais habituais. O valor destas foi gerido e calibrado para que conseguissem ter uma frequência de 60Khz quando o EN está a 0. Este teste está ilustrado na secção 6, que diz respeito aos testes realizados.

Na figura abaixo temos ilustrado as duas bobines fabricadas bem como o leitor final.



Figura 4-7-Bobines realizadas

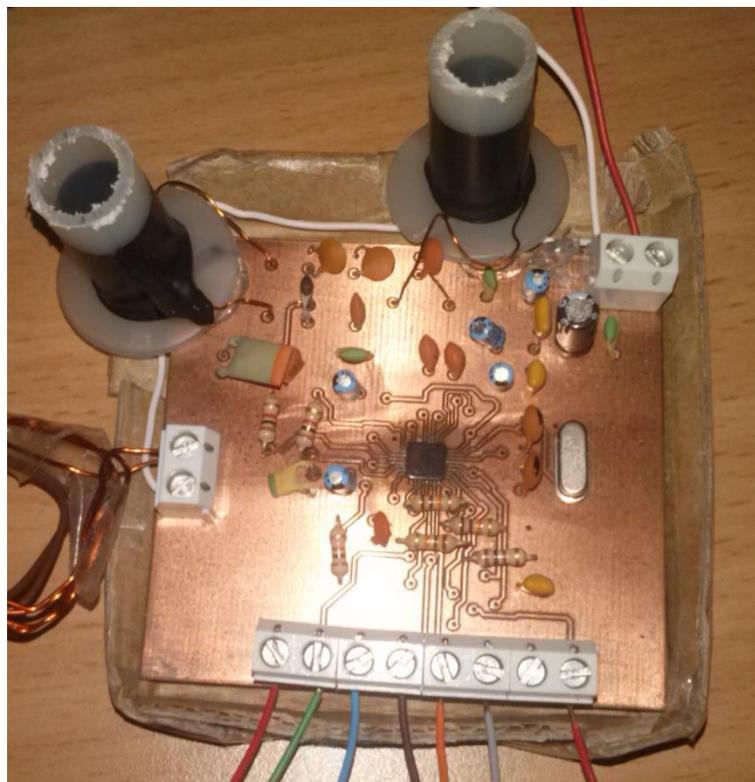


Figura 4-8- Leitor final

Por defeito o Trf7960 funciona no modo automático. Neste modo os reguladores são ativados sempre que o EN fica a *Hight*. Os reguladores internos são também reconfigurados cada vez que o bit de seleção automática do regulador é posto a *Hight*.

4.1.2. Microcontrolador adotado e principais configurações

A escolha do microcontrolador teve em conta três opções. O MSP430 que é indicado pela TI para estes processos, o 8051 da ATMEL que estava em *stock* e cuja interface já estava criada e o PIC 18F.



Figura 4-9-Imagem dos três microcontroladores selecionados: a) Msp430F2370 [14], b) AT89C51IC2 [25], c) PIC18F4550 [13]

A decisão entre estes três microcontroladores foi feita com base na seguinte tabela:

Tabela 4-2- Comparação entre os microcontroladores

Referência Microcontrolador	Tipo de comunicação	Preço na Farnell
MSP430	RS232	22,95 euros
8051ATMEL	RS232 e atualmente USB	25 euros
PIC 18F	USB	35,12 euros

Como o microcontrolador MSP430 possui uma interface RS232 este foi logo descartado. A escolha entre o 8051 da ATMEL e o PIC foi feita por base no fator preço, pelo que foi decidido o 8051.

Microcontrolador AT89C51RD2

O microcontrolador é um dispositivo que é responsável por comandar sistemas inteligentes através de um conjunto de instruções definidas para cada microcontrolador.

Este microcontrolador em particular é um microcontrolador de 8 bits que foi introduzido em 1977. É conhecido pela sua fácil programação em *assembly*, com um conjunto de instruções bastante simples de manipular e pelo seu baixo custo. Tem como principal foco os circuitos lógicos na área da automação industrial, rádios, telemóveis e micro-ondas. Neste microcontrolador pode-se utilizar memória interna e externa ao dispositivo.

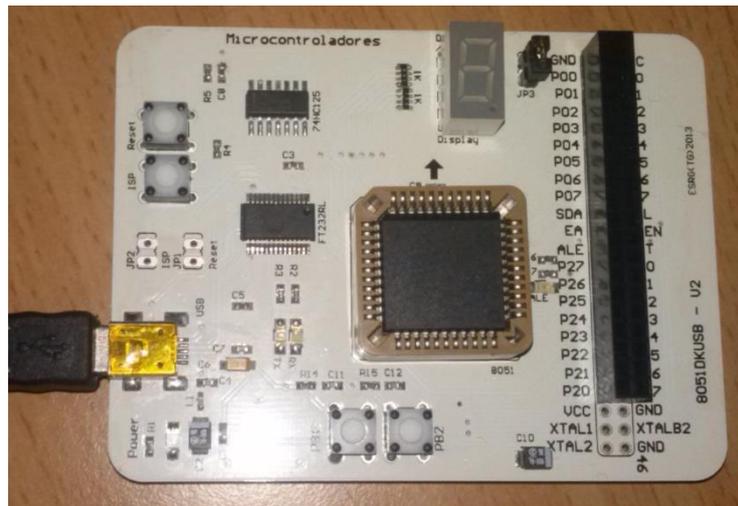


Figura 4-10-Microcontrolador da ATMEL com a interface criada

De forma a conseguirmos desenvolver um programa para o microcontrolador foi usado o compilador/simulador *KeilµVision*. Esta ferramenta auxilia no desenvolvimento de código *assembly* e na geração final de um ficheiro *.HEX* que pode ser descarregado para a memória do microcontrolador.

A programação do microcontrolador teve por base os diagramas temporais ilustrados nas figuras 4.17 a 4.19 .

O formato dos dados das tramas foi baseado nas tabelas 2.2 a 2.4.

As *Address Words* e as *Command Words* foram feitas de acordo com a seguinte tabela:

Tabela 4-3-Distribuição dos bits das *Address* e *Command Words* [11]

Bit	Descrição	Função do bit	<i>Address</i>	<i>Command</i>
Bit 7	Bit de controlo	0= <i>Address</i> , 1= <i>Command</i>	0	1
Bit 6	Leitura/Escrita	1-leitura, 0-escrita	R/W	0
Bit 5	Modo de endereçamento contínuo	1=modo contínuo	R/W	0
Bit 4	<i>Address/Comand</i> bit 4		Adr 4	Cmd 4
Bit 3	<i>Address/Comand</i> bit 3		Adr 3	Cmd 3
Bit 2	<i>Address/Comand</i> bit 2		Adr 2	Cmd 2
Bit 1	<i>Address/Comand</i> bit 1		Adr 1	Cmd 1
Bit 0	<i>Address/Comand</i> bit 0		Adr 0	Cmd 0

De acordo com a tabela acima pode-se ver que se se quiser ler o endereço do IRQ, cujo endereço é 0Ch, ter-se-ia de enviar a seguinte palavra:

Tabela 4-4- Exemplo da formação de uma palavra

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Valor em hexadecimal:
0	1	0	0	1	1	0	0	4CH

O Bit 7 teria de estar a 0 porque é um endereço, o bit 6 a 1 porque é uma leitura, o bit 5 a 0 porque não serão enviados mais dados e os bits de 4 a 0 teriam o endereço 0CH.

A palavra final a ser enviada seria 4CH.

Se se quisesse, por outro lado mandar um comando ter-se-ia de escolher um dos comandos disponíveis.

Tabela 4-5-Códigos dos comandos [11]

Código do Comando	Descrição do Comando
00H	Inativo
03H	Inicialização do <i>software</i>
04H	Inicializar o RFCollision
05H	Resposta do RFCollision
06H	Resposta do RFCollision0
0FH	<i>Reset</i>
10H	Transmite sem CRC
11H	Transmite com CRC
12H	Transmissão atrasada sem CRC
13H	Transmissão atrasada com CRC
14H	Transmite o próximo <i>slot</i>
15H	Fecha a sequência de <i>slots</i>
16H	Para os decodificadores
17H	Corre os decodificadores / Habilita o recetor
18H	Testa o RF interno
19H	Testa o RF externo
1AH	Ajuste do ganho do recetor

Se fosse escolhido o envio do comando *Reset* o bit 7 teria de estar a 1 e depois seria colocado o código do respetivo comando. Neste caso ficaria 8FH.

Os códigos dos comandos que se encontram a azul, na tabela acima, não se encontram descritos na *datasheet*. Estes foram descobertos no código elaborado pela Texas aquando a programação do seu microcontrolador.

4.1.3. Antena

No desenvolvimento da antena teve-se como principal foco as *tags* utilizadas. Para tal fez-se uma réplica da antena usada pela *tag*. Isto garantiu que ambos funcionassem à mesma frequência, 13.56Mhz, sem perder tempo desnecessário.

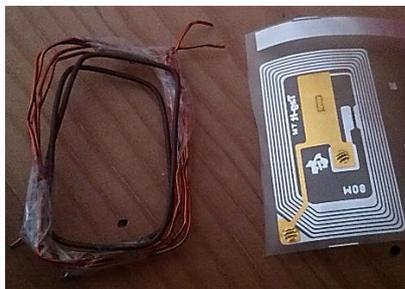


Figura 4-11-Imagem da antena criada bem como a *tag* exemplo

A antena da *tag* bem como a antena criada é do tipo de anel de corrente, sendo constituída por bobines curtas. Neste tipo de antenas o campo eletromagnético diminui drasticamente à medida que a distância aumenta.

O campo radiado pela antena é uma onda progressiva onde se encontram os campos elétricos e magnéticos juntos na fase, ficando a energia dividida entre estes dois campos.

Quando se comunica com o *transponder* deve-se ter em atenção os seguintes aspetos:

- A resistência à radiação.
- A eficiência da radiação por parte da antena.
- Densidade da potência.

4.2. Software

A componente de *software* foi onde foi programado o microcontrolador, para que o leitor fizesse as funções pretendidas. Neste capítulo será explicado de uma forma sucinta como foi feita a implementação em *software* das funções principais.

4.2.1. Algoritmos

Para que o dispositivo funcionasse como pretendido foi necessário a programação dos respetivos componentes. Esta programação teve por base os seguintes algoritmos:

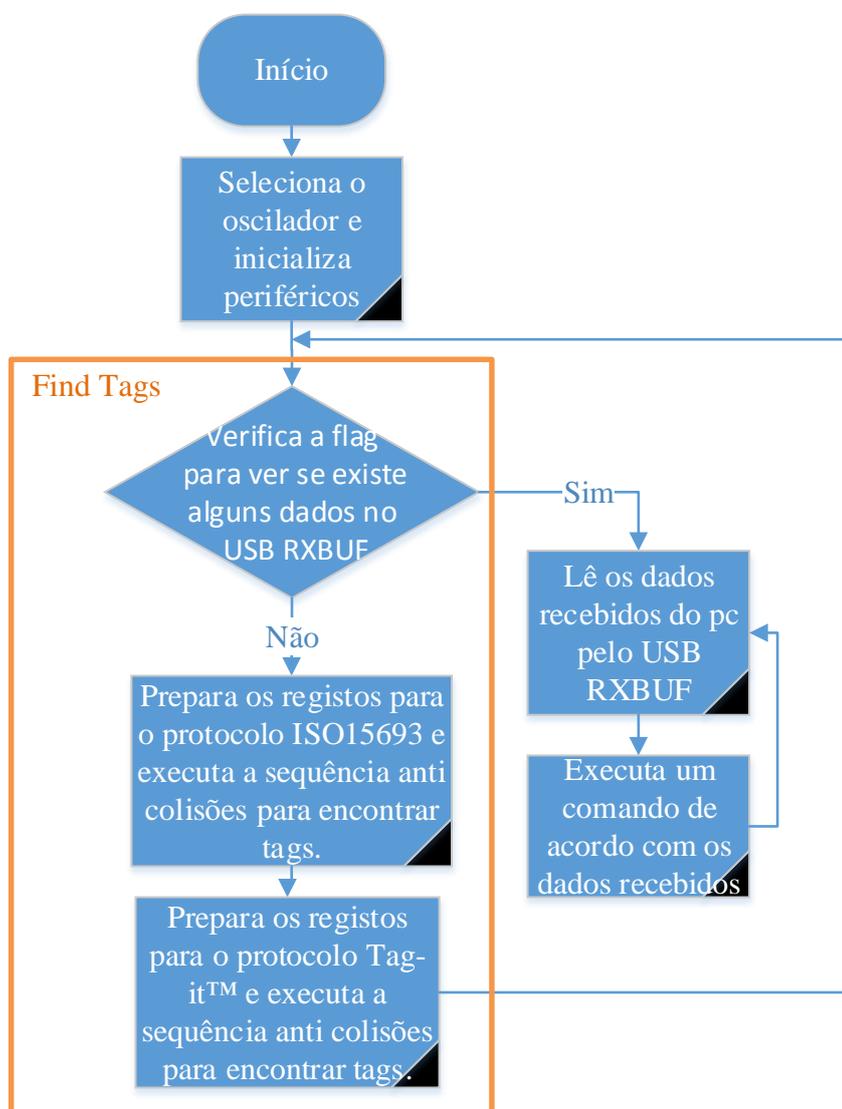


Figura 4-12-Diagrama com a *main* principal

Como se pode ver na imagem acima o programa principal consiste numa chamada recursiva do mesmo, onde se encontra a procurar *tags* e a tratar colisões.

Na imagem abaixo será detalhada a função *FindTags*, assinalada na figura 4.12 a laranja.

FindTags

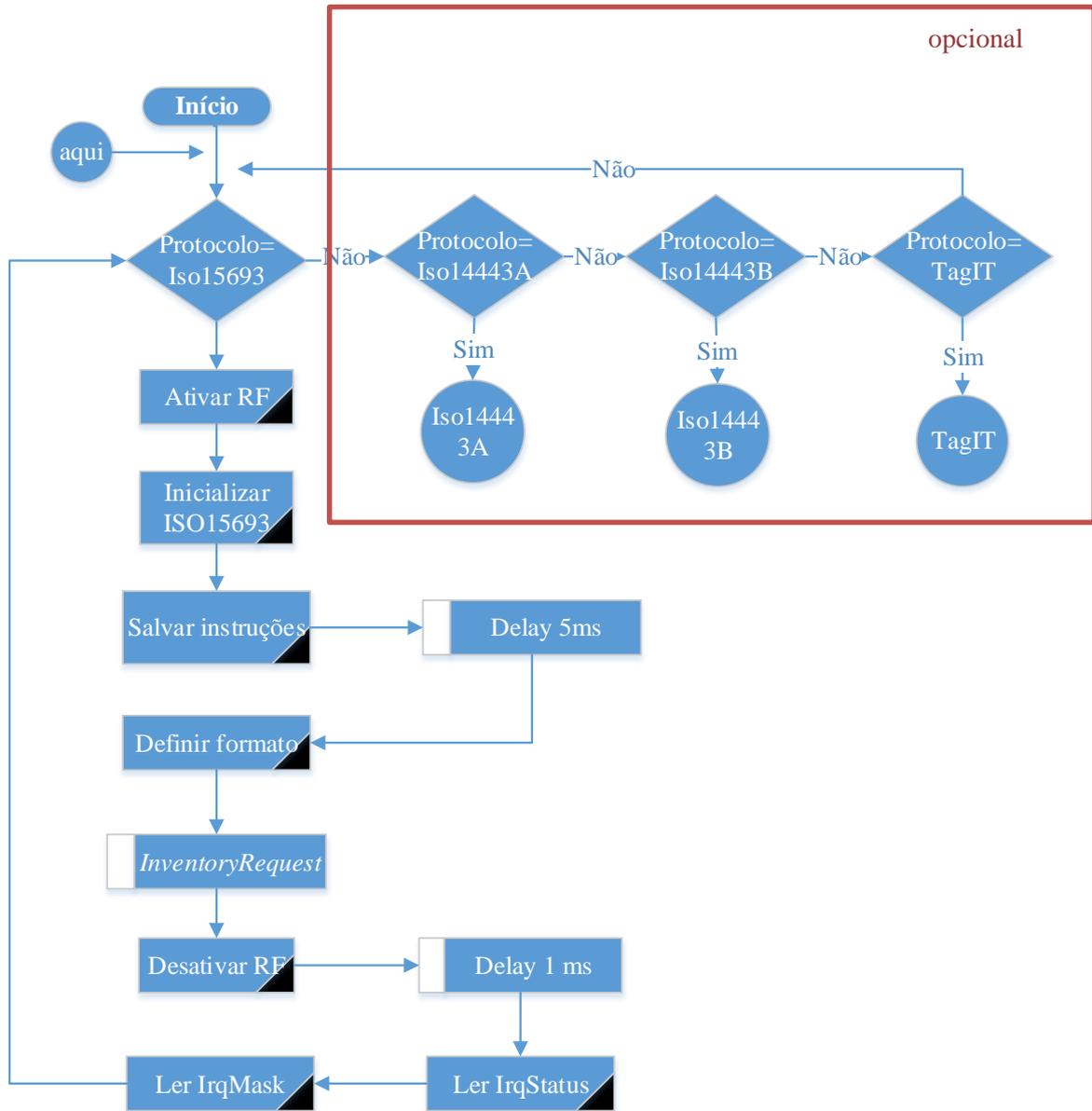


Figura 4-13- Diagrama do comportamento da função *FindTags*

A continuação deste diagrama encontra-se ilustrado no Anexo 2.

Quando no diagrama se diz para mandar um comando para ligar o *Radio Field*, corresponde a enviar uma mensagem de escrita para o registo *ChipStateControl* com o valor 21H.

O registo *ChipStatusControl* permite manipular o modo em que será trabalhado: RF ligado ou desligado, o AGC e o tipo de modulação AM/PM. Por defeito o registo tem o valor 01H o que significa que está operado para trabalhar a 5V.

Tabela 4-6- ChipStatus Control [11]

Bit	Id Bit	Função	Descrição
B7	Stby	1- Modo <i>standy</i> 0- Modo ativo	O modo <i>standy</i> mantém o regulador e o oscilador a correr.
B6	Direct	1- Recebe o sinal da sub-portadora 0- Recebe o sinal decodificado do decodificador	O controlo da modulação é direto através da entrada MDO. A sub-portadora recebida encontra-se no I/O_6.
B5	rf_on	1- RF ativo 0- RF desligado	Quando o bit B5 é colocado a 1 o escudo RF é ativado.
B4	rf_pwf	1- potência de saída <i>half</i> 0- potência de saída <i>full</i>	1- condutor de RF a 8Ω 0- condutor de RF a 4Ω
B3	pm_on	1- RX_IN2 0- RX_IN2	1- Seleciona a entrada do sinal em PM 0- Seleciona a entrada do sinal em AM
B2	agc_on	1- AGC ativo 0- AGC desativado	Seleção do AGC
B1	rec_on	1- Habilitação do recetor para medição do campo externo	O recetor e o oscilador são habilitados para medição do campo externo.
B0	vrs5_3	1- operação a 5V (V_{in}) 0- operação a 3V (V_{in})	Seleciona o alcance do V_{DD_RF} : se $V_{in}=5V$ o alcance vai variar de 4.3V a 5V; se $V_{in}=3V$ o alcance vai variar de 4.2.7V a 3.4V

Quando se diz para escrever neste registo 21H, significa que se está a mandar colocar o bit 5 a 1, ou seja está-se a ligar o RF. No final apenas se tem de o desligar colocando este mesmo bit a 0.

A potência de transmissão configurada no bit 4, quando está configurada para 5V varia entre:

- Baixa potência - 100mW (20dBm), com a resistência de transmissão de 8Ω.
- Alta potência - 200mW (23dBm), com a resistência de transmissão de 4Ω.

Quando se informa que se irá trabalhar no protocolo ISO 15693 significa que irá ser enviado para o registo *IsoControl* o valor 02H.

Por defeito este registo está configurado para este mesmo valor.

Tabela 4-7-IsoControl [11]

Bit	Id Bit	Função	Descrição
B7	rx_crc_n	Recebe sem CRC	1- sem RX CRC 0- RX CRC
B6	dir_mode	Tipo de modo direto	1- a saída é uma sub-portadora de dados 1- a saída é de bit stream(I/O_6) e bit clock (I/O_5) do decodificador selecionado pelos bits ISO.
B5	rfid	Modo RFID	Deve estar sempre a 0.
B4	iso_4	Modo RFID	Ver tabela 14
B3	iso_3		
B2	iso_2		
B1	iso_1		
B0	iso_0		

Tabela 4-8-Seleção dos modos RFID [11]

Iso_4	Iso_3	Iso_2	Iso_1	Iso_0	Protocolo	Observações
0	0	0	0	0	ISO 15693 com baixa taxa de bits	6.62 kbps
					uma sub-portadora	1 saída de 4
0	0	0	0	1	ISO 15693 com baixa taxa de bits	6.62 kbps
					uma sub-portadora	1 saída de 256
0	0	0	1	0	ISO 15693 com alta taxa de bits	26.48 kbps
					uma sub-portadora	1 saída de 4

0	0	0	1	1	ISO 15693 com alta taxa de bits	26.48 kbps	
					uma sub-portadora	1 saída de 256	
0	0	1	0	0	ISO 15693 com baixa taxa de bits	6.67 kbps	
					sub-portadora dupla	1 saída de 4	
0	0	1	0	1	ISO 15693 com baixa taxa de bits	6.67 kbps	
					sub-portadora dupla	1 saída de 256	
0	0	1	1	0	ISO 15693 com alta taxa de bits	26.69 kbps	
					sub-portadora dupla	1 saída de 4	
0	0	1	1	1	ISO 15693 com alta taxa de bits	26.69 kbps	
					sub-portadora dupla	1 saída de 256	
0	1	0	0	0	ISO 14443A com taxa de bits	106 kbps	Taxa de bits do RX quando a taxa de bits do TX é diferente do RX (reg03)
0	1	0	0	1	ISO 14443A com alta taxa de bits	212 kbps	
0	1	0	1	0	ISO 14443A com alta taxa de bits	424 kbps	
0	1	0	1	1	ISO 14443A com alta taxa de bits	848 kbps	
0	1	1	0	0	ISO 14443B com taxa de bits	106 kbps	Taxa de bits do RX quando a taxa de bits do TX é diferente do RX (reg03)
					ISO 14443B com alta taxa de bits	212 kbps	
					ISO 14443B com alta taxa de bits	424 kbps	

0	1	1	1	1	ISO 14443B com alta taxa de bits	848 kbps	
1	0	0	1	1	Tag-it		

Como foi escolhido o protocolo ISO 15693 de alta taxa de bits deve mandar-se o valor 02H para o registo da IsoControl. Se se pretender usar o protocolo ISO 14443A deve mandar-se o valor 08H e assim sucessivamente.

No leitor existem dois controladores principais o EN e o EN2 referidos nas seções acima.

Quando estes dois controladores EN e EN2 estão a 0 e a 1 respetivamente a frequência do SYS_CLK é de 60 kHz.

Quando o leitor está a funcionar em modo normal, com o EN a 1, o SYS_CLK pode ser programado pelos bits B4 e B5 do modulador e pelo registo de controlo do SYS_CLK, endereço 09H. As frequências de relógio podem ser de 13.56Mhz, 6.78Mhz ou 3.39Mhz.

Tabela 4-9-Modulador e Sys_CLK Control [11]

Bit	Id Bit	Função	Descrição
B7	Não é usado		
B6	en_ook_p	Quando este bit é 1 habilita a seleção externa da modulação ASK ou OSK	Este bit só é valido quando o registo de controlo ISO está configurado para modo direto
B5	Clo1	A frequência de saída do SysCLK está no modo MSB	Clo1 Clo0 Estado de saída do Cl_Sys 0 0 Desabilitado
B4	Clo0	A frequência de saída do SysCLK está no modo LSB	0 1 3.3Mhz 1 0 6.78Mhz 1 1 13.56Mhz
B3	en_ana	Quando este bit é 1 habilita a saída do ASK ou OOK	Usado para medições e testes
B2	Pm2	MSB da modulação central	Pm2 Pm1 Pm0 Modo e % 0 0 0 ASK 10%

B1	Pm1	Modulação central	0	0	1	OOK 100%
B0	Pm0	LSB da modulação central	0	1	0	ASK 7%
			0	1	1	ASK 8.5%
			1	0	0	ASK 13%
			1	0	1	ASK 16%
			1	1	0	ASK 22%
			1	1	1	ASK 30%

A modulação de transmissão é controlada pelo registo de controlo ISO. Caso se queira pôr em modo direto as operações passam a ser controladas por controladores externos através do bit 6. A modulação ASK é controlada pelos bits B0, B1 e B2 no modulador e pelo registo de controlo do SYS_CLK. O seu alcance pode ser de 7%, 30% ou 100% (OOK).

InventoryRequest

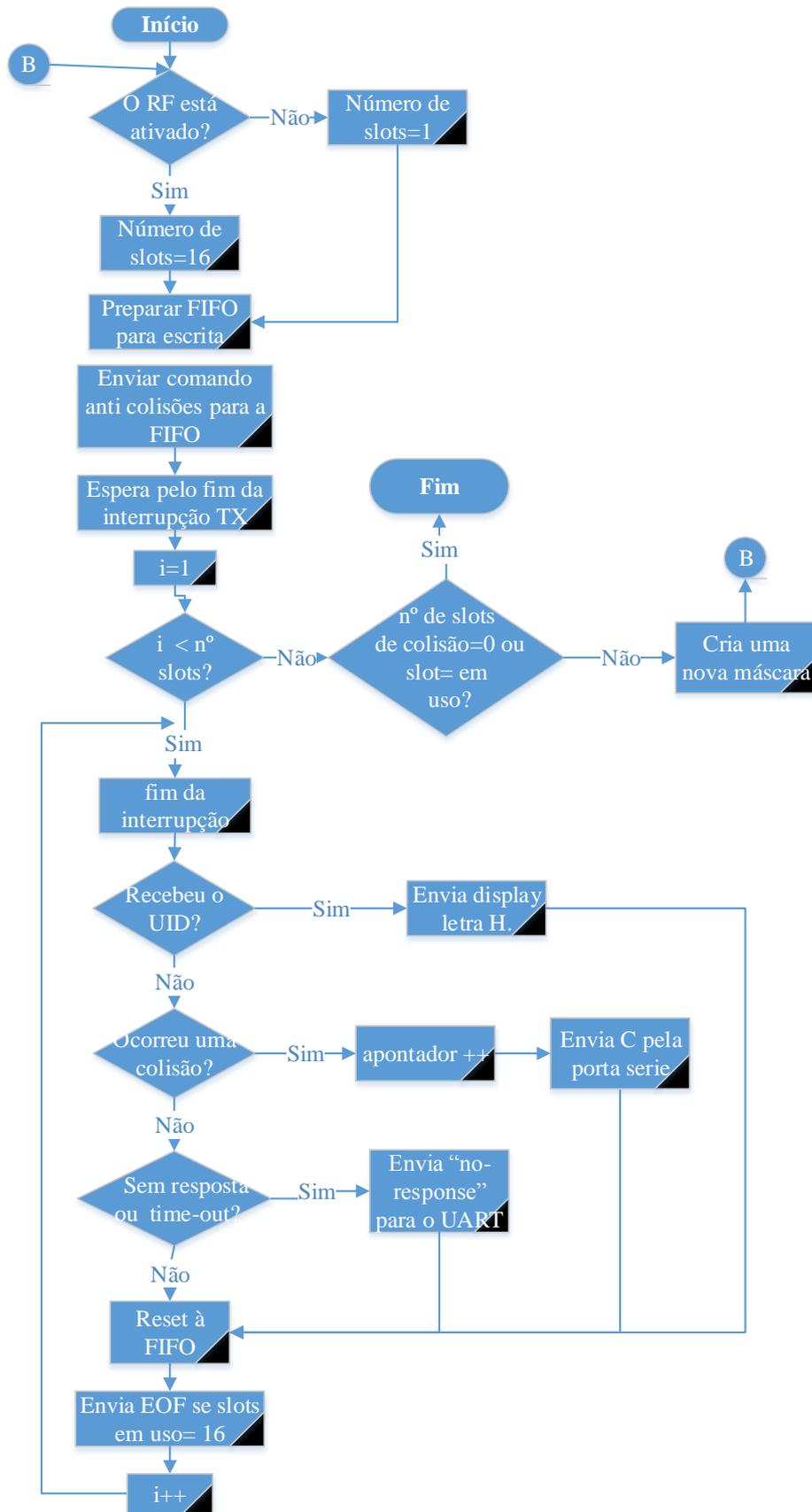


Figura 4-14-Diagrama da função Inventory Request

O diagrama ilustra o comportamento da função *InventoryRequest*, cuja função é fazer uma pesquisa das *tags* que se encontram nas proximidades e avisar o utilizador da presença das mesmas. Esta função é usada quando está a ser utilizado o protocolo ISO 15693.

Os registos fundamentais que se usam nesta função são:

- *Chip Status Control* - Para verificar se o campo RF se encontra ativo.
- *RX Wait Time* - Define o tempo entre o fim da interrupção do TX quando o RX está a ser desconsiderado. Este registo por defeito está configurado para 1FH que corresponde ao tempo utilizado no protocolo ISO 15693.
- *Collision Position and Interrupt Mask Register* – De forma a habilitar as possíveis interrupções para a FIFO, CRC, Paridade, erro de *frame* ou EOF e erros de colisão.
- *RSSI levels* - Para saber a força do sinal recebido pelos canais e a amplitude RF durante o período em que o campo RF está inativo.

Os comandos utilizados nesta função são:

- *Stop Decoders*- Faz *Reset* à FIFO no final do TX e para o recetor
- *Run Decoders*- O Recetor inicia.
- *Transmit Next Slot*- Quando o leitor recebe este comando ele transmite o *slot* definido pelo protocolo selecionado.

InterruptHandlerReader

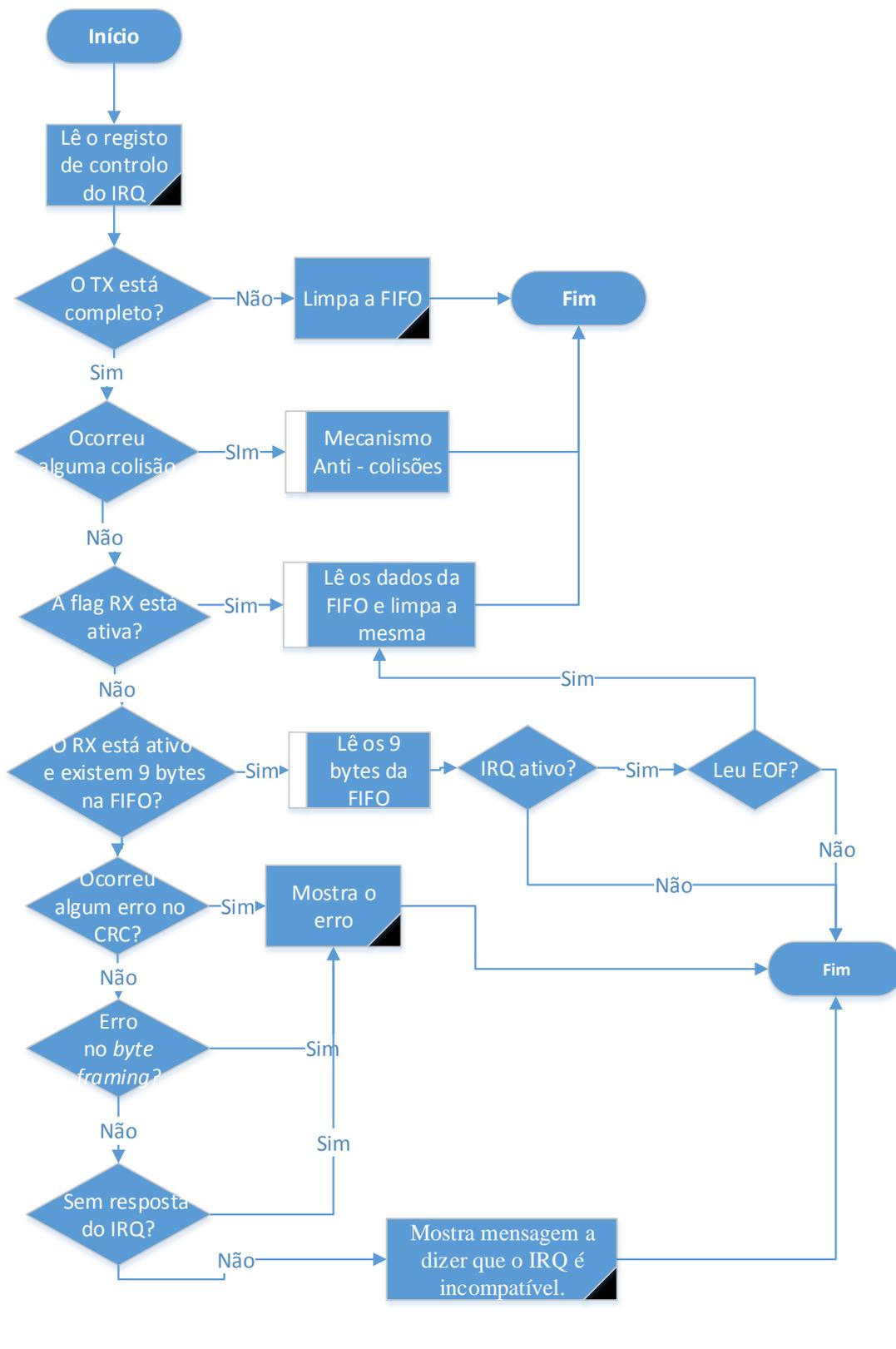


Figura 4-15-Diagrama da função *InterruptHandlerReader*

Este procedimento é chamado quando ocorre uma interrupção ou quando o TX termina.

O registo IRQ lido é constituído da seguinte forma:

Tabela 4-10- Registo de estado do IRQ [11]

Bit	Id Bit	Função	Descrição
B7	Irq_tx	Irq ativado no final do TX.	Sinaliza que o TX está em progresso. A <i>flag</i> é ativa no início do TX mas o <i>interrupt request</i> é enviado quando o TX terminar.
B6	Irq_srx	Irq ativado no início do RX.	Sinaliza que recebeu um início de trama do RX e que o mesmo está em progresso. A <i>flag</i> é ativada quando o RX é ativado mas o <i>interrupt request</i> só é enviado quando o RX terminar.
B5	Irq_fifo	Sinaliza que a FIFO é menor que 1/3 ou maior que 2/3.	Sinaliza que a FIFO está a <i>high</i> ou <i>low</i> , menos de 4 ou mais que 8.
B4	Irq_err1	Erro de CRC.	Indica que recebemos um erro de CRC.
B3	Irq_err2	Erro de paridade	Indica que recebemos um erro de paridade
B2	Irq_err3	Erro de Final de ficheiro ou <i>Byte framing</i>	Indica que recebemos um erro de <i>framing</i>
B1	Irq_col	Erro de colisão	Usado no protocolo IS014443A e ISO 15693 com uma sub-portadora simples
B0	Irq_noresp	Sem resposta	Sinaliza ao Micro que o comando para transmitir o próximo <i>slot</i> pode ser enviado.

Como se pode ver pela tabela este registo será responsável por dizer se se encontrou alguma *tag* nas proximidades ou que erro pode ter ocorrido.

Quando detetar alguma *tag* é necessário ver o estado da FIFO, onde estão armazenados os UIDS das *tags* detetadas.

Tabela 4-11- Estado da FIFO [11]

Bit	Id Bit	Função	Descrição
B7	RFU	Coloca a <i>Low</i>	Bit reservado para uso futuro (RFU)
B6	Fhil	O nível da FIFO está a <i>High</i>	Indica que 9 bytes já se encontram na FIFO (para o RX).
B5	Flol	O nível da FIFO está a <i>Low</i>	Indica que apenas 3 bytes estão na FIFO (para o TX).
B4	Fove	Erro de <i>overflow</i> na FIFO	Indica que foram escritos demasiados dados na FIFO.
B3	Fb3	Bytes na FIFO fb[3]	Estes bits dizem quantos bytes se encontram na FIFO sem serem lidos. Mostra N-1 número de bytes. Se estiverem 8 bytes na FIFO este número é 7.
B2	Fb2	Bytes na FIFO fb[2]	
B1	Fb1	Bytes na FIFO fb[1]	
B0	Fb0	Bytes na FIFO fb[]	

O estado da FIFO é responsável por informar o número de bytes que se encontram na FIFO de forma a conseguirmos ler a pilha que está no endereço 1FH. Esta pilha possui 12 bytes como já foi falado anteriormente e será onde se encontrarão alocadas as informações sobre as nossas *tags*.

Outro registo utilizado é o *Tx_Length Byte2* que diz os números mais baixos a serem transferidos pela FIFO. Este registo possui também a informação do número de bytes corrompidos.

4.2.2. Inventário e tratamento de colisões no protocolo ISO 15693

O procedimento de inventário é feito da seguinte forma:

- O VCD manda uma máscara e o número de *slots* para o *inventory request*.
- O VICC da *Tag* compara os bits menos significativos do seu UID com o *slot counter* mais o valor da máscara, caso corresponda irá responder ao leitor
- Se o VICC responder apenas uma vez então não ocorreu nenhuma colisão e o VCD recebeu o UID completo.
- Se o leitor detetar uma colisão ele toma nota do número de *slot* em que isso ocorreu e manda um EOF para mudar para o próximo *slot*.
- O VICC incrementa o *slot counter* em cada receção de EOF
- Este procedimento é repetido para todos os *16 slots*.
- No final dos *16 slots* o *slot pointer* será examinado. Se este valor for diferente de 0 significa que ocorreu uma colisão em mais do que um *slot*. Uma nova máscara é calculada e o *inventory request* é recomeçado com essa nova máscara. Este processo irá continuar até que não existam mais colisões.

No primeiro passo do inventário é enviada uma máscara a 0 e um tamanho 0 pelo que o VICC irá apenas comparar os bits menos significativos do UID com o *slot counter*. Nesta primeira fase todas as *tags* irão responder num dos *16 slots*.

Tabela 4-12-Primeira fase do processamento do inventário

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
			Encontra a <i>tag</i> x123					Encontra a <i>tag</i> x28, x38 Colisão							

Caso mais do que uma *tag* responda no mesmo *slot* o leitor ativará a *flag* de colisão no registo de estado da interrupção. O microcontrolador toma nota do *slot* em que esta colisão ocorreu.

Em todos os outros *slots* o VCD irá esperar um determinado tempo pela resposta das *tags*, caso não receba nada no tempo estipulado enviará um EOF para o leitor enviar o próximo *slot*.

Na segunda fase o valor da máscara será igual ao valor da máscara anterior, o que corresponde ao número do *slot* em que ocorreu a interrupção, mais o *slot counter*. O tamanho por sua vez é incrementado por 4 bytes. Ao ser aumentada a máscara faz-se com que diminua o risco de colisões.

Tabela 4-13-Segunda fase do processamento do inventário

08	18	28	38	48	58	68	78	88	98	A8	B8	C8	D8	E8	F8
		Encontra a tag x28	Encontra a tag x38												

Caso ocorra novamente uma colisão é feita uma terceira ronda em que a máscara será igual à máscara anterior mais o *slot counter*. O tamanho será incrementado novamente por 4 bytes. Este procedimento é feito até que não haja colisões.

O algoritmo utilizado foi baseado no documento *Implementation of the ISO 15693 Protocol in the TI TRF796x* [42], pois adequou-se às necessidades pretendidas pelo trabalho.

Algoritmo anticolisões: [42]

1. Verificar o bit 5 da *flag* no *inventory request*. Se esta estiver ativa o número de *slots* é 1, senão é 16. Caso seja 16 habilita a interrupção sem resposta
2. Inicializa o tamanho da máscara e o valor da mesma a 0.
3. Inicializa o *slot pointer* a 0.
4. É feito o *inventory request* com o tamanho da máscara e o seu valor.
5. Espera pelo fim da transmissão.
6. Espera pela próxima interrupção que pode ocorrer:
 - a. No fim do TX.
 - b. Em caso de colisão.
 - c. Sem resposta.
7. Limpa a FIFO.
8. Se o número de *slots* for 16 transmite EOF. Se for 1 sai do procedimento.
9. Repete os passos 5 e 6 para os 16 *slots*. No final desabilita a interrupção de resposta.
10. Examina o *slot number*. Se não for 0 calcula uma nova máscara. Se for 0 sai.
 - a. Incrementa o tamanho da máscara por 4 bytes.
 - b. Calcula uma nova máscara sendo que esta é igual ao *slot number* (onde ocorreu a colisão) mais a máscara antiga.
11. Vai para o passo 4, mas com uma nova máscara e tamanho.

12. Decrementa o *slot pointer* por 1.

13. Vai para o passo 10.

O UID recebido pelo micro vai estar armazenado num registo *buffer* pela ordem do bit menos significativo. Isto é se uma *tag* com o UID E007000006D6AC1C responder esta ficará armazenada da seguinte forma:

```
buffer[0]=1C
buffer[1]=AC
buffer[2]=D6
buffer[3]=06
buffer[4]=00
buffer[5]=00
buffer[6]=07
buffer[7]=E0
```

4.2.3. Funcionamento das máscaras no protocolo ISO 15693

O funcionamento da máscara é igual ao inventário geral, com a exceção de que em vez de mandar a máscara 0 na primeira fase, manda-se a máscara introduzida pelo utilizador. O tamanho definido será igual ao número de bytes introduzidos pelo utilizador. Se este introduzir 1 byte a máscara é de 4, se introduzir 2 bytes a máscara é de 8 e assim sucessivamente.

Se o utilizador introduzir apenas parte do UID isto é se este apenas introduzir que os últimos bytes da *tag* são x28 é feito um inventário onde a máscara a procurar das *tags* corresponde aquela terminação. Este vai procurar o valor da máscara introduzida mais o valor do *Slot Counter*.

Tabela 4-14-Primeira fase do processamento com máscara

028	128	228	328	428	528	628	728	828	928	A28	B28	C28	D28	E28	F28
			Encontra a <i>tag</i> x4328					Encontra a <i>tag</i> x5828							

Caso mais do que uma *tag* responda no mesmo *slot* é realizado o mesmo procedimento que na função de inventário onde o leitor ativa a *flag* de colisão no registo de estado da interrupção. O microcontrolador irá então tomar nota do *slot* onde a colisão ocorreu e fará uma segunda ronda cuja máscara será igual ao valor anterior mais o *slot counter*. O tamanho será incrementado novamente por 4 bytes até que não existam mais colisões.

Se o utilizador introduzir a máscara toda, isto é se ele introduzir o UID da *tag* completo, a máscara a procurar irá corresponder apenas à máscara introduzida sem o valor do *slot counter*. Não sendo necessário ciclos extra de tratamento de colisões ou de incrementação do *slot counter*.

4.2.4. Fluxogramas de acesso ao leitor

O código realizado para o microcontrolador teve em conta os diagramas temporais de escrita e leitura nos endereços.

Quando o utilizador pretende escrever num registo deve utilizar o diagrama temporal ilustrado na figura abaixo.

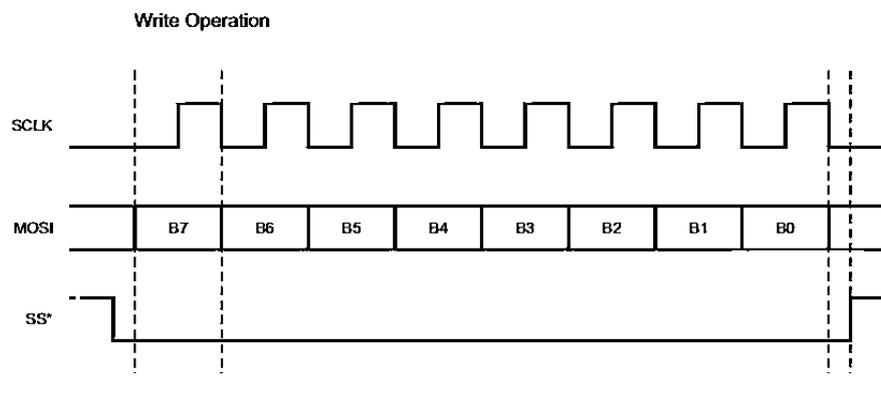


Figura 4-16-Operação de escrita [11]

Como se pode ver por esse diagrama só se pode começar a transmitir quando o *Slave Select* se encontra a 0, sendo os respetivos bits escritos por cada símbolo de relógio.

No modo de escrita a transmissão dos dados encontra-se no modo *SCLK Falling Edge* e a validação MOSI está no modo *SCLK Rising Edge*.

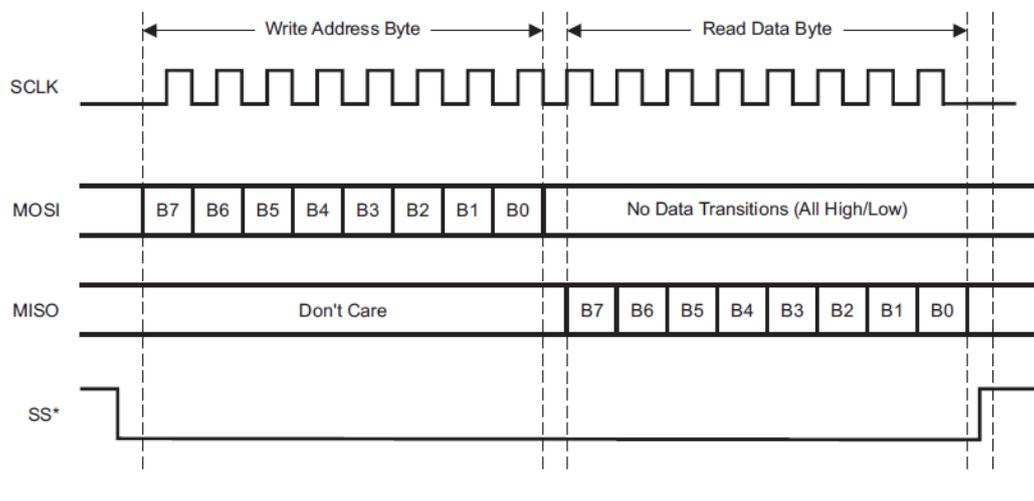


Figura 4-17-Operação de leitura [11]

No modo de leitura por outro lado a transmissão dos dados está no modo *SCLK Rising Edge* e a validação MISO está no modo *SCLK Falling Edge*.

Quando se quer ler um determinado registo é preciso primeiro escrever no registo MOSI o que se quer ler para proceder à leitura propriamente dita.

Tal como se pode visualizar pela figura acima a comunicação passa do estado de escrita para leitura quando o último bit de comando B0 é validado.

A leitura do registo IRQ é feita de forma diferente à descrita acima. Os passos a seguir são os seguintes:

- Escrever no comando 6CH que se vai ler o registo IRQ em modo contínuo.
 - O registo IRQ corresponde ao endereço 0Ch, como se pretende ler é preciso ativar o bit de leitura 60H. O resultado desta operação é 6CH mencionado acima.
- Gerar mais 8 sinais de relógio, como se se estivesse a ler os dados do registo 0DH, mas ignorando os dados MISO.

Na figura abaixo temos ilustrado o comportamento descrito acima quando se lê o registo IRQ.

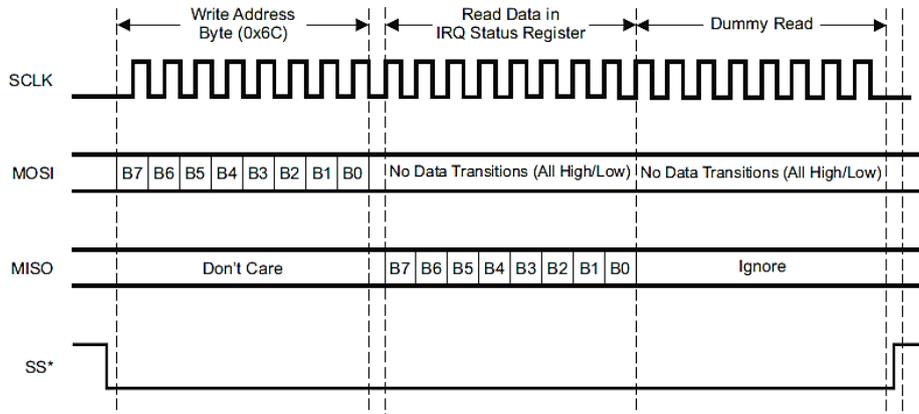


Figura 4-18-Comunicação da interface SPI (registo de estado IRQ lido) [11]

4.2.5. Interface com o utilizador

A interação com o utilizador será realizada através da interface USB



Figura 4-19-Cabo utilizado na interface com computadores portáteis [38]

Este cabo liga do microcontrolador com a interface mini USB e ao computador com a interface USB normal.

Quando se liga este dispositivo deve-se ir a um terminal previamente instalado onde será feita a interação com o protótipo através de algumas letras designadas na tabela 4.14.

O terminal utilizado foi o Hercules do grupo HW-group.

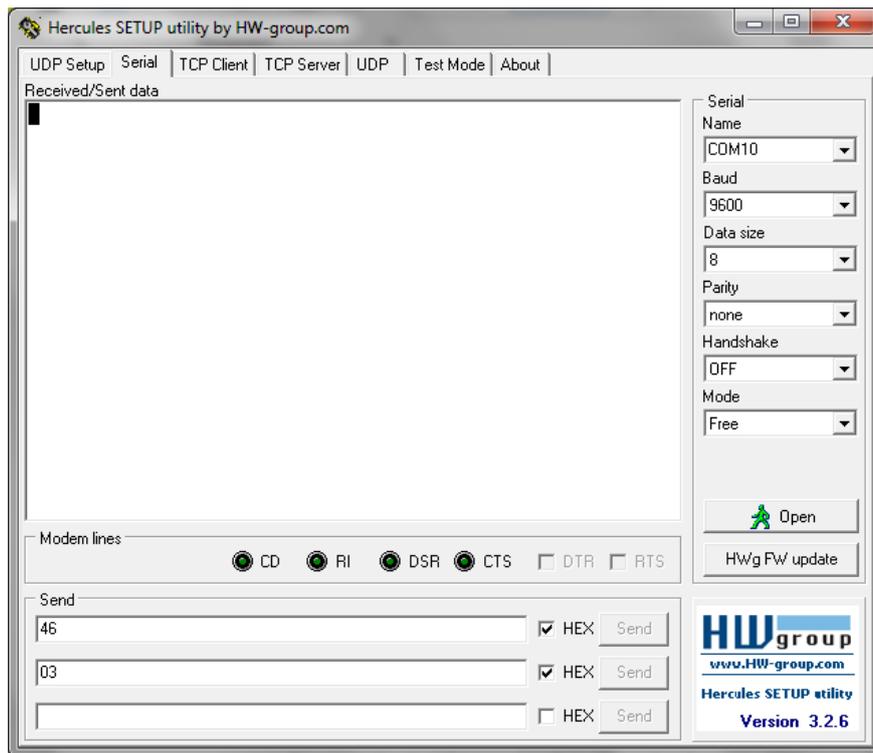


Figura 4-20-Terminal Hercules utilizado

Antes de se ligar o dispositivo ao computador é necessário instalar o driver CDM20814_Setup.exe, que está disponível para download no site da FTDI. Este driver deve ser instalado em modo de administrador.

Para saber a porta em que está configurada o terminal deve-se ir ao gestor de dispositivos, ou então verificar a porta que está configurada quando se liga o cabo USB.

Depois de saber a porta basta colocar a COM correta no programa e deixar as definições que estão por defeito, *Baudrate* a 9600, 8 dados, sem paridade.

Basta clicar na janela e pressionar as teclas do computador que se encontram na tabela 4.14 e pode-se interagir com o protótipo



Figura 4-21- Cabo utilizado na interface com telemóveis [39]

Este cabo liga do microcontrolador com a interface mini USB ao telemóvel com a interface micro USB.

Infelizmente este cabo não se encontra à venda nas lojas tradicionais em Portugal, sendo necessário encomendar *online*.

O telemóvel Android também tem uma aplicação que funciona da mesma forma que o Hercules usado no computador denominado de Slick USB 2 Serial.

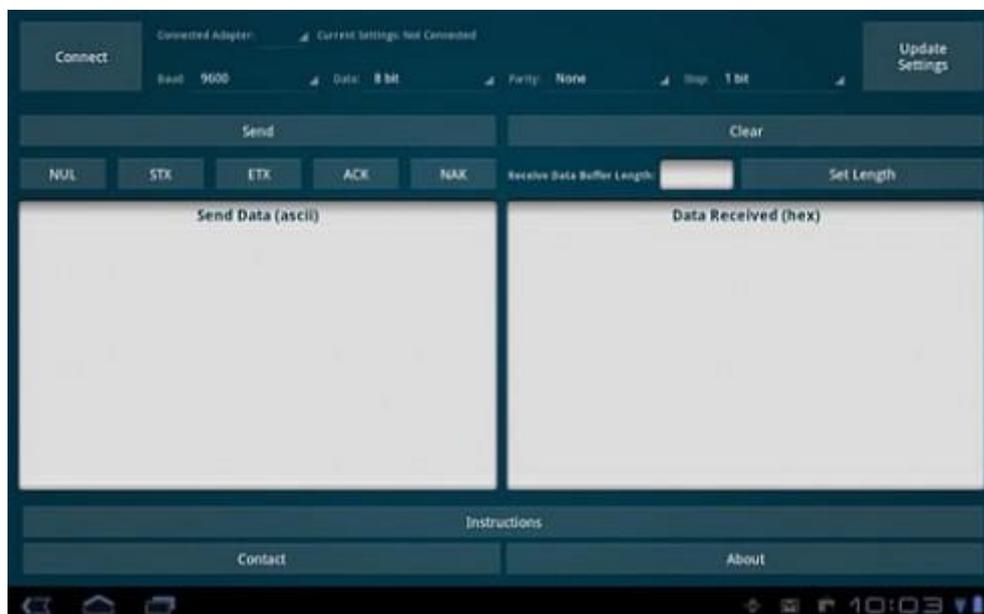


Figura 4-22-Terminal Slick USB 2 Serial [43]

Esta aplicação é utilizada para Androides com um sistema operativo superior a 3.1. É completamente gratuito, de forma que todos podem instalar a aplicação.

O conjunto de comandos que o utilizador pode fazer encontra-se descrito na tabela abaixo.

Tabela 4-15-Comandos possíveis de interação com o utilizador

Tecla pressionada	Significado	Descrição
R	<i>RESET</i>	Este comando faz um <i>Reset</i> à FIFO e aos registos
C	Escrever no Cristal	Este comando escreve no registo do Sys_CLK de forma a mudar o valor da frequência do Cristal
F	<i>Find Tags</i>	Função de inventário geral
M[código UID < 14]S Ex: M12S	Encontra grupo de <i>tags</i> . Ex: Encontra grupo que termina com o UID 12	Através dos valores introduzidos pelo utilizador este encontra o grupo de <i>tags</i> correspondentes aquele UID parcial que foi pedido. Sendo o carater S o símbolo de fim do endereço parcial
M[código UID =14] Ex: MD015A012B107E0	Encontra <i>Tag</i> específica Ex: Encontra a tag E007B112A015D0	Utilizando o UID da <i>tag</i> específica, este encontra a <i>tag</i> correspondente.

Caso seja pressionada uma tecla diferente das permitidas é enviada uma mensagem de erro ao utilizador.

4.2.6. Sistema de alarmes

O sistema de alarmes implementado consistiu num *buzzer* cujo esquema foi o seguinte:

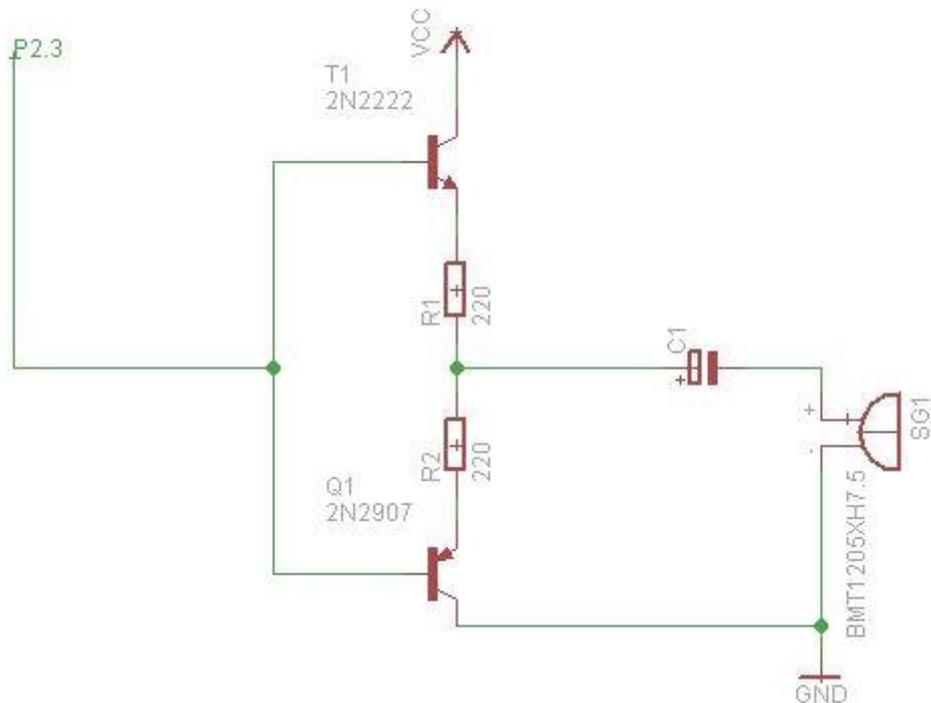


Figura 4-23-Esquema de montagem do sistema de alarmes

O esquema de alarmes está ligado a um Pino do Microcontrolador responsável por ativar o circuito e emitir um som de alarme quando um objeto se afasta do campo. Sempre que é detetado um possível roubo o timer 0 é ativado e é feita uma mudança de estado do pino P2.3 sempre que ocorre um *overflow*. Esta oscilação no pino faz com que seja produzido um som intermitente de alarme.

A imagem abaixo representa a montagem deste mesmo dispositivo.

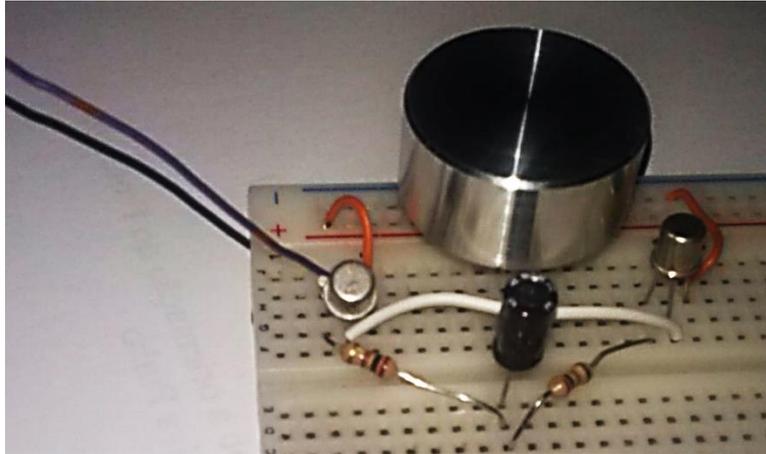


Figura 4-24-Sistema de Alarme implementado

4.2.7. Saída dos dados

De forma a se conseguir saber o que está a acontecer com o protótipo foi utilizado o display de 7 segmentos para avisar em caso de eventos de alarme. Estes eventos estão descritos na tabela 4.16.

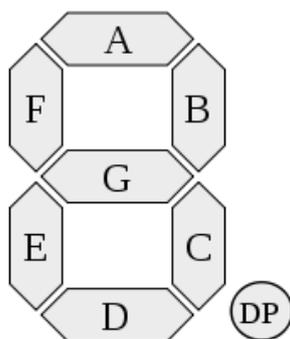


Figura 4-25-Display de 7 segmentos [44]

Tabela 4-16-Significado das mensagens do display

Valor	Significado	Código em hexadecimal
E.	Erro.	30H
1	Pressionou a tecla 1 como parte do UID a procurar.	CFH
2	Pressionou a tecla 2 como parte do UID a procurar.	54H
3	Pressionou a tecla 3 como parte do UID a procurar.	46H
4	Pressionou a tecla 4 como parte do UID a procurar.	87H
5	Pressionou a tecla 5 como parte do UID a procurar.	26H
6	Pressionou a tecla 6 como parte do UID a procurar.	24H
7	Pressionou a tecla 7 como parte do UID a procurar.	4FH
8	Pressionou a tecla 8 como parte do UID a procurar.	04H

9	Pressionou a tecla 9 como parte do UID a procurar.	06H
0	Pressionou a tecla 0 como parte do UID a procurar.	0CH
A	Pressionou a tecla A como parte do UID a procurar.	05H
B.	Pressionou a tecla B como parte do UID a procurar.	00H
C	Pressionou a tecla C como parte do UID a procurar.	3CH
D.	Pressionou a tecla D como parte do UID a procurar.	08H
E	Pressionou a tecla E como parte do UID a procurar.	34H
F	Pressionou a tecla F como parte do UID a procurar.	35H
H.	Encontrou a <i>tag</i> .	81H
H	Não encontrou a <i>tag</i> .	85H

5. Testes e avaliação do protótipo implementado

Neste capítulo serão apresentados os testes do protótipo desenvolvido e de seguida será feita uma avaliação da viabilidade da sua produção. Este protótipo será também comparado relativamente a outro leitor existente, de forma a saber quais as suas hipóteses de implementação no mercado.

5.1. Leitor contruído

Teste de funcionamento do leitor

Nas imagens abaixo está demonstrado o teste inicial, para saber se o leitor se encontrava a funcionar corretamente. Este teste foi feito tendo em atenção a frequência de oscilação do cristal.

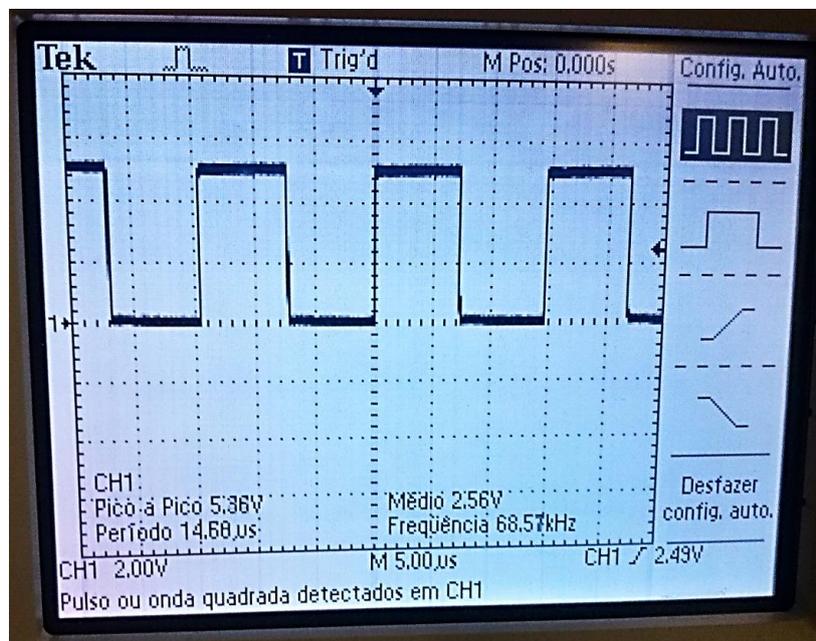


Figura 5-1-Saída do SysCLK com o EN= 0 e o EN2=1

Como se pode verificar pelo diagrama o resultado do SysCLK, quando o EN possui o valor 0, foi dentro do esperado, aproximadamente 60KHz.

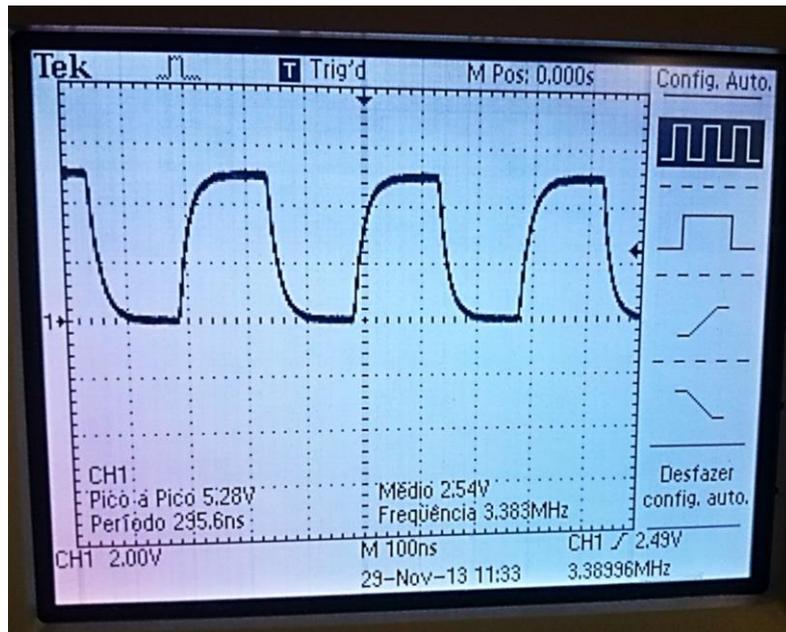


Figura 5-2-Saida do SysCLK com o EN=1 e o EN2=1

Quando temos o EN a 1 a saída do SysCLK está dentro dos valores de defeito do Modulador, que para o valor 11H diz que o SysCLK responde a 3.3Mhz.

Como se quer que o modulador responda à frequência das *tags* é necessário no início do programa mandar uma mensagem de configuração do SysCLK para que, este trabalhe à frequência de 13.56MHZ.

Teste do cristal

Este teste foi feito para ver se o cristal já estava a responder à frequência das *tags*.

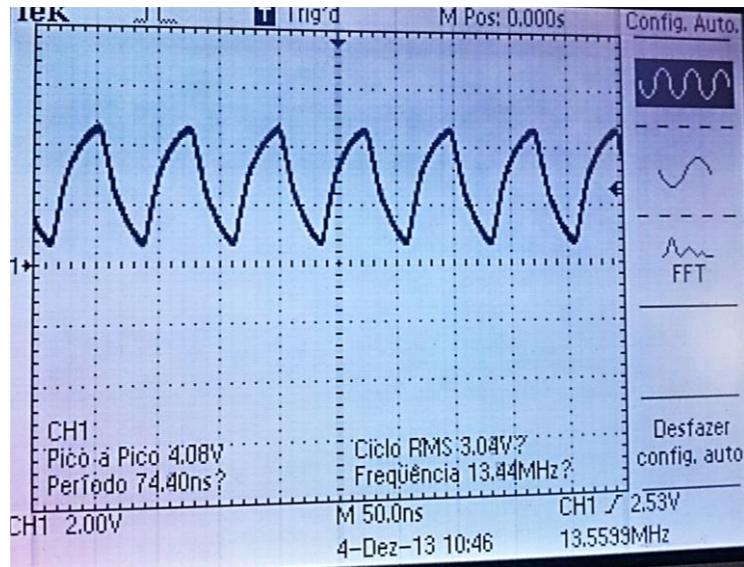


Figura 5-3-Saida do SysCLK com o EN=1 e o EN2=1 após a configuração do Sys_CLK_Control

Na imagem acima está apresentado o resultado do SysCLK quando é configurado o modulador para o valor 31H. Quando este está configurado com esse valor, o cristal passa a funcionar à frequência de 13.56Mhz.

Teste de funcionamento do campo RF

Este teste foi feito às ondas rádio de forma a conseguir ver o comportamento do campo.

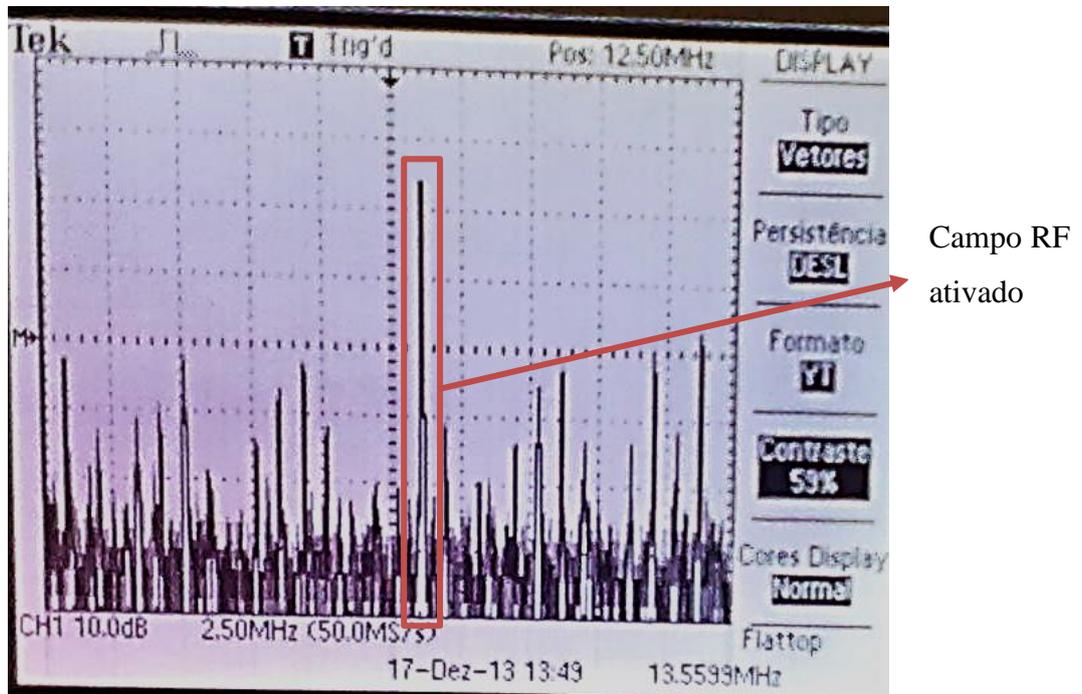


Figura 5-4-Análise do campo *Radio Field*

Na imagem acima está ilustrado o comportamento do campo *Radio Field* quando este é ativado a 13.56Mhz. Este campo é responsável por ativar as *tags* e fornecer-lhes a energia necessária para que estas respondam quando forem chamadas.

Teste de funcionamento dos comandos

O teste de funcionamento dos comandos foi feito para se ter a certeza que o microcontrolador estava a comunicar com o leitor.

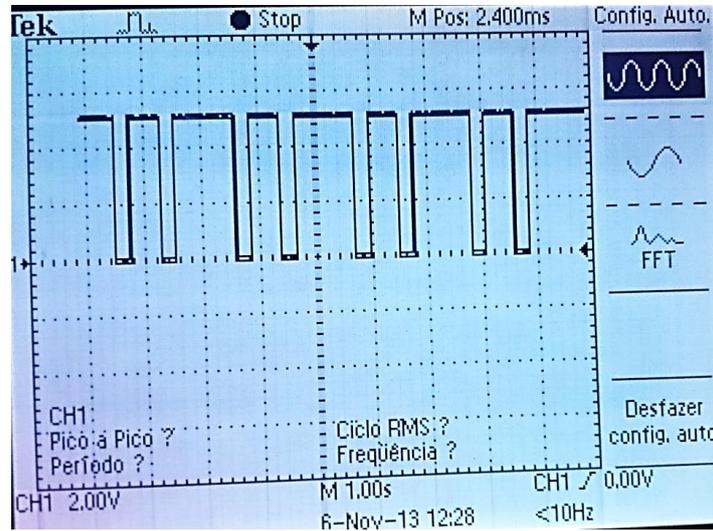


Figura 5-5-Leitura do comando *Reset* enviado pelo MOSI

Na imagem acima está ilustrado os bits enviados do leitor para o micro quando este manda uma mensagem de escrita, neste caso um comando *Reset* ao sistema.

Teste de funcionamento da função de inventário

O teste de funcionamento da função inventário tem por base mostrar que se conseguiu detetar as *tags* que estão dentro do campo RF.

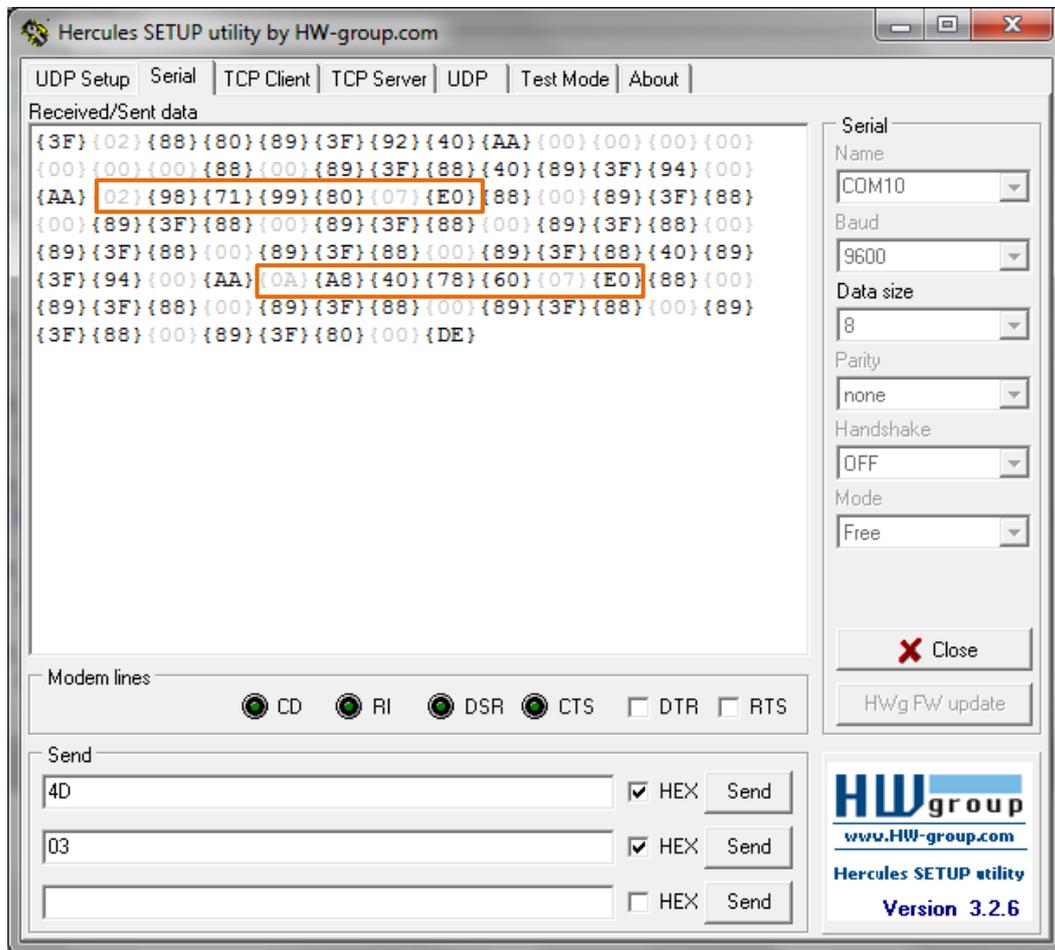


Figura 5-6-Teste ao Inventário

Como se pode verificar na imagem foram encontradas duas *tags* que se encontravam dentro do campo RF. O UID que se encontra a laranja encontra-se de trás para a frente, pois tal como já foi explicado anteriormente, a tag envia a sua informação dessa forma. O UID das *tags* encontradas foram os seguintes:

- E0078099719802 - esta *tag* é encontrada no terceiro ciclo porque termina em 2
- E007607840A80A - esta *tag* é encontrada no ciclo 11 porque termina em A.

Teste de funcionamento do endereçamento específico de uma tag

O teste de funcionamento por um endereçamento específico foi feito com base no UID E0078099719802.

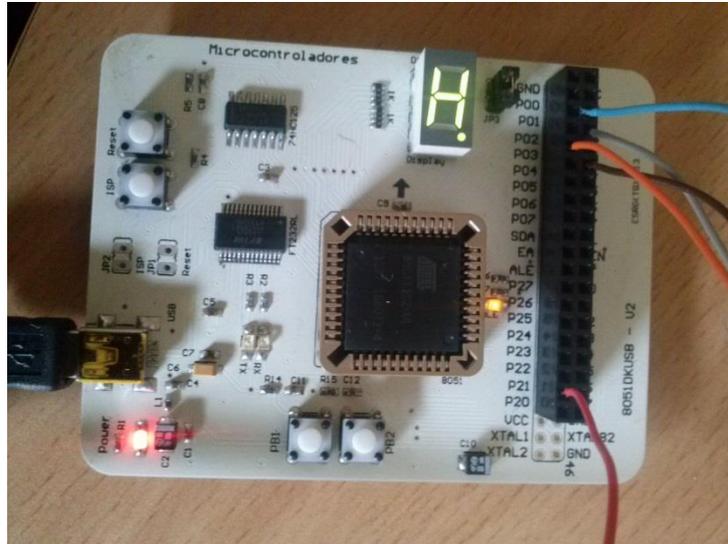


Figura 5-7-Resultado do endereçamento direto com a Tag dentro do campo RF

A figura mostra o resultado correspondente ao momento em que a tag se encontrava dentro do campo RF. Como se pode ver o resultado no display H. mostra que conseguiu encontrar a tag pretendida.

No teste abaixo afastou-se a tag em questão a uma distância superior ao seu campo RF, mantendo as outras tags dentro desse perímetro. Quando se fez este teste foi obtido o seguinte resultado:

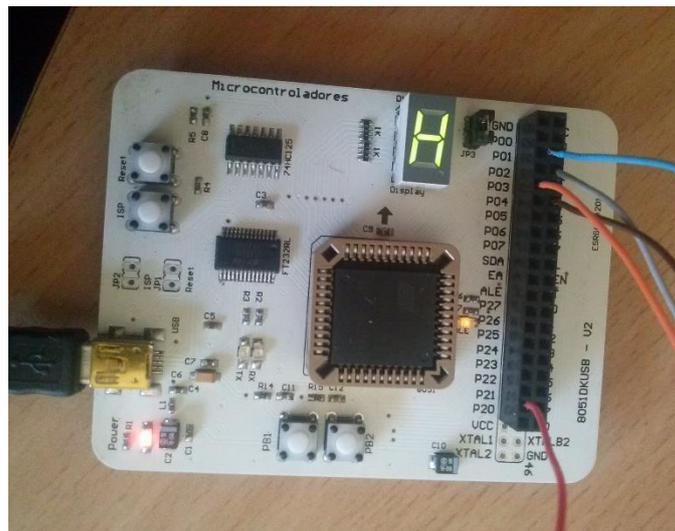


Figura 5-8-Resultado do endereçamento direto com a Tag fora do campo RF

Como se pode ver pela imagem o display mostra o resultado H logo não conseguiu detetar a *tag* pretendida.

5.2. Outros Leitores

Outro leitor testado foi o leitor RFID 125KHz (SEN11425P), comprado na *botnroll*.

Este leitor é um módulo utilizado para ler cartões RFID UEM4100 com dois modos de saída UART e Wiegand. A distância máxima de deteção dos cartões é de 7 cm.

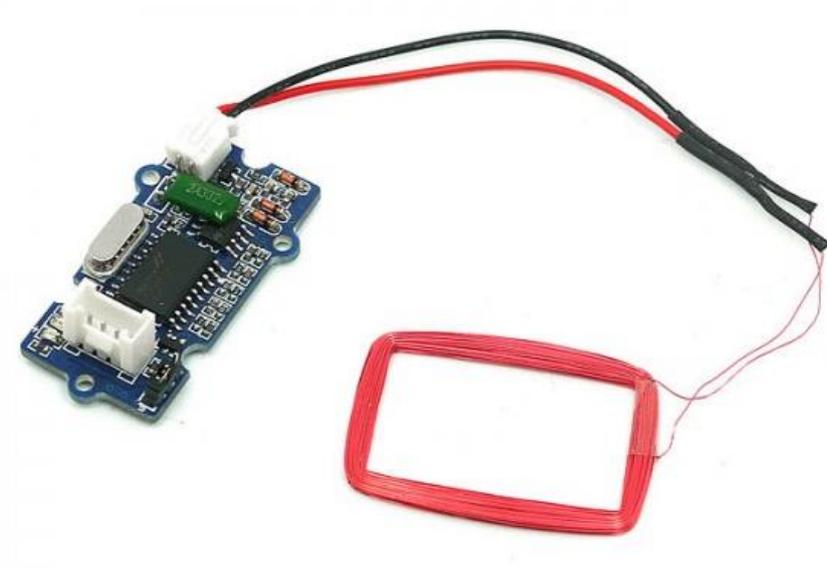


Figura 5-9-Leitor RFID 125KHz - SEN11425P [26]



Figura 5-10-*Tags* compradas para funcionar com o leitor de 125KHz

Para se poder testar o leitor foram incorporados 2 fios soldados na parte traseira do dispositivo, de forma a que fosse possível alimentar o leitor corretamente



Figura 5-11-Parte traseira do leitor e fios soldados para alimentação

Na imagem abaixo encontra-se ilustrado o comportamento do TX face à aproximação das tags.

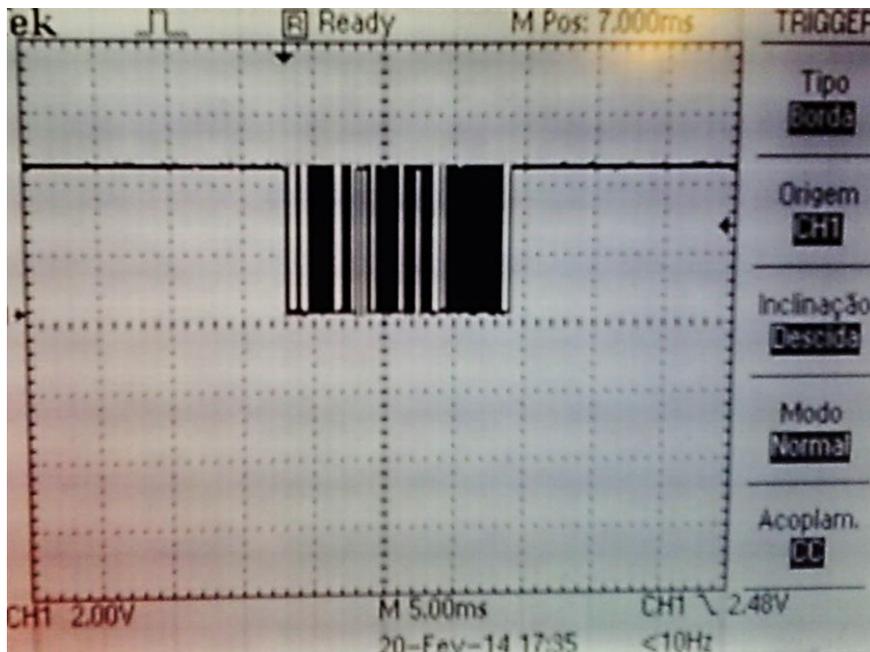


Figura 5-12- Bits recebidos com 5ms por divisão

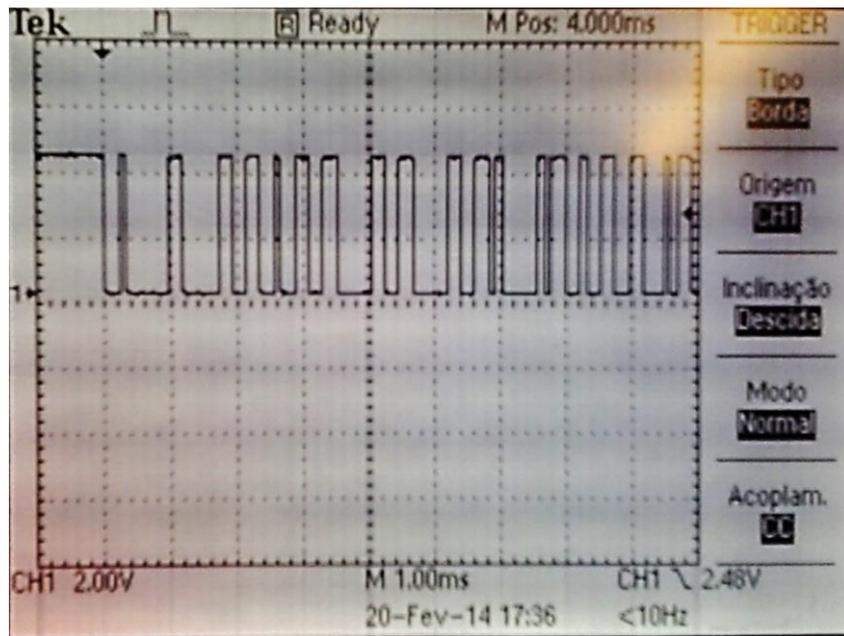


Figura 5-13-Bits recebidos com 1ms por divisão

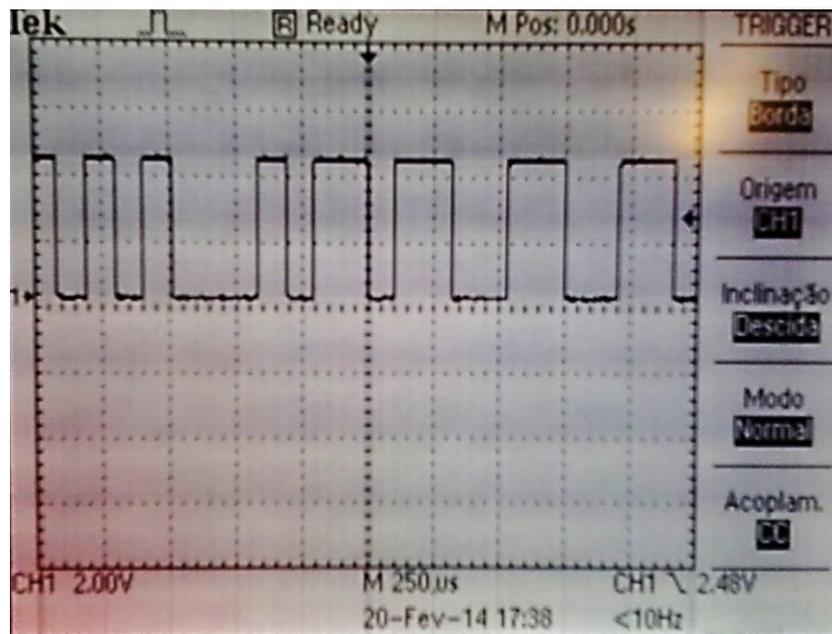


Figura 5-14-Bits recebidos com 250µs por divisão

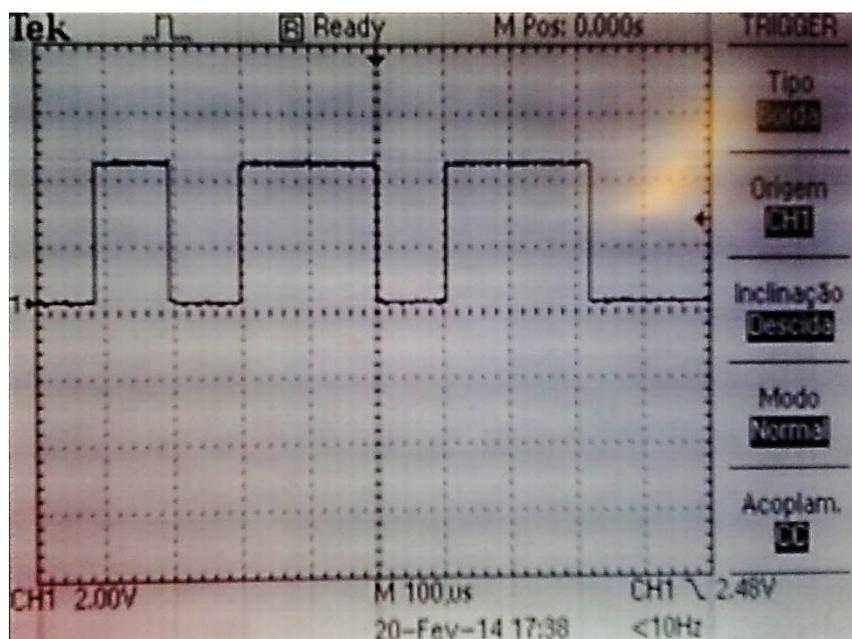


Figura 5-15- Bits recebidos com 100 μ s por divisão

Nas imagens acima está ilustrado, embora não de uma forma muito clara, os bits recebidos pelo leitor. Estes bits foram enviados pela *tag* e correspondem ao seu UID.

Pelas imagens acima pode-se constatar que o *Baudrate* deste leitor é aproximadamente 9600.

A trama recebida é aproximadamente de 160 bits de acordo com uma contagem estimativa que foi feita com base na figura 5.12 em relação ao tempo do bit.

5.3. Comparação entre o leitor contruído e o outro testado

O leitor testado em comparação com o que foi feito mostrou-se ser bastante mais simples, embora de curto alcance. A distância máxima possível entre o leitor e as *tags* foi de apenas 4 cm, enquanto o protótipo montado conseguiu uma distância máxima de 80 cm aproximadamente.

A interface com o utilizador é ligeiramente mais complicada no leitor da *botnroll*, pois este apenas acende um led de forma muito rápida, não possuindo a interface para se conseguir saber de forma simples o UID da *tag* lida.

O leitor construído possui uma interface que elucida mais o utilizador sobre os acontecimentos de todo o processo podendo, este mesmo, interagir com o leitor.

O leitor da *botnroll* tem apenas a capacidade de detetar *tags* na vizinhança, não tendo a possibilidade de encontrar uma *tag* específica.

No final o leitor da *botnrol*, tendo uma interface bastante rudimentar teve um custo de 14.75€ mais 5€ pelas 2 *tags* que perfaz um total de 19.75€, enquanto o construído teve um custo de 16€ mais o custo do microcontrolador de 25€, o que perfaz um total de 41€.

Embora o leitor construído seja mais caro, este é muito mais eficaz na procura de *tags* e tem também capacidades extra de endereçamento por grupo e *tags* específicas. Possuindo além disso um sistema de alarme, o que faz com que este seja consideravelmente melhor que o anterior.

6. Considerações finais e perspectivas futuras

6.1. Conclusão

O enorme crescimento das tecnologias *wireless*, a evolução dos dispositivos móveis e da internet foram os principais motores para o interesse na investigação dos sistemas de tele-localização.

No início foi usado o sistema GPS evoluindo mais tarde para sistemas de triangulação e análise de cenários. Infelizmente o GPS depende de uma enorme quantidade de satélites e não funciona dentro de casas. Em virtude desta situação começaram a utilizar o RFID que já tinha sido inventado e utilizado durante a segunda guerra mundial para detetar ataques de aviões inimigos. O primeiro sistema que se pode dizer que realmente identificava os aviões inimigos de amigos surgiu em Inglaterra pelo Sr. Watson-Wat que desenvolveu o identificador IFF. O RFID continuou a ser investigado nas décadas de 50 e 60 e foi comercializado inicialmente para sistemas antifurto.

O objetivo desta dissertação foi, utilizando as características do sistema RFID, desenvolver de uma maneira económica e rentável, um sistema de tele-localização de objetos. O mesmo é aplicado à deteção/prevenção de possíveis furtos de objetos, que foram colocados em campo aberto e, num outro cenário, encontrar um determinado grupo de objetos ou um objeto específico.

O trabalho realizado durante esta dissertação foi o seguinte:

- Foi montado e idealizado um PCB que minimiza o tamanho do leitor.
- Foi criado um leitor funcional.
- Foi feita a correta programação do Leitor através de um novo MCU, sendo que todo o código foi criado de raiz através dos diagramas temporais do TRF7960.
- O cenário 1, que descreve o inventário de um determinado local, funciona corretamente tendo pequenos problemas de colisão, ocasionalmente, derivados a interferências que andem no ar.
- O cenário 2 está melhor implementado que o primeiro pois, como este parte de uma máscara, o risco de colisões diminui consideravelmente.

A parte dos grupos infelizmente não está a funcionar de acordo com a ideia inicial, pois o protótipo inicia a função pelos últimos bytes, enquanto que, na ideia original, pretendia-se que este comesse pelos bytes iniciais, de forma a poder saber quais as *tags* daquele fabricante por exemplo.

Infelizmente não foi possível acabar a dissertação no tempo proposto inicialmente, sofrendo um atraso de um mês, para implementar a identificação com máscaras das *tags*.

Também não foi possível fazer o teste com o telemóvel pois o cabo USB não se encontrava disponível.

A linguagem utilizada foi de baixo nível não sendo muito aconselhada para futuras implementações. O microcontrolador adotado por sua vez possui também pouca memória.

Como trabalho futuro poderiam ser melhorados os seguintes aspetos:

- Realizar uma antena com os requisitos do documento [30].
- Implementar um sistema central mais robusto com base de dados e um sistema de identificação do utilizador tal como foi descrito no subcapítulo 3.3.
- Implementar um ecrã no próprio dispositivo para que ele seja totalmente independente como foi mencionado no subcapítulo 3.2.
- Colocar uma bateria no protótipo para não ser necessário ao computador fornecer energia ao dispositivo.
- Adaptar o código para um microcontrolador de mais capacidade de armazenamento de dados e com um nível de linguagem de programação de alto nível.
- Partilhar informações entre utilizadores para que um utilizador possa localizar um objeto nosso e avisar-nos.

Referências bibliográficas

1. *The RFID NETWORK*. Disponível em: <http://rfid.net/> [consultado em: 21/02/2013]
2. *Fórum NFC*. Disponível em <http://www.nfc-forum.org/aboutnfc/> [consultado em 15/02/2013]
3. *ALPHA-TRX433S and ALPHA-TRX915S*, Data Sheet, RF Solutions, (<http://www.farnell.com/datasheets/1442999.pdf>)
4. *IA4420 Universal ISM Band FSK Transceiver*, Data Sheet, Silicon Labs (<http://pdf1.alldatasheet.com/datasheet-pdf/view/134654/ETC1/IA4420.html>)
5. *IA4320 Universal ISM Band FSK Transceiver*, Data Sheet, Integration (<http://www.datasheetdir.com/IA4320+download>)
6. *IA4220 Universal ISM Band FSK Transceiver*, Data Sheet, Silicon Labs (<http://www.datasheetdir.com/IA4220+download>)
7. *IA4221 Universal ISM Band FSK Transceiver*, Data Sheet, Silicon Labs (<http://www.digchip.com/datasheets/parts/datasheet/000/IA4221.php>)
8. *IA4421 Universal ISM Band FSK Transceiver*, Data Sheet, Integration (<http://www.cdiweb.com/datasheets/integration/IA4421-DS.pdf>)
9. *RI-I03-112A-03 -Tag-it™ HF-I PLUS TRANSPONDER INLAYS MINIATURE RECTANGLE*, Data Sheet, (Outubro de 2001, Revisto em: Abril 2010). Texas Instruments (<http://www.ti.com/lit/ds/symlink/ri-i03-112a-03.pdf>)
10. Página oficial da Texas sobre oTRF7960. Disponível em: <http://www.ti.com/product/trf7960> [consultado em: 07/03/2013]
11. *TRF7960 and TRF7961, MULTI-STANDARD FULLY INTEGRATED 13.56-MHZ RFID ANALOG FRONT END AND DATA-FRAMING READER SYSTEM*. (Agosto de 2006, Revisto em Agosto de 2010). Data Sheet. *Texas Instruments* (<http://www.ti.com/lit/ds/slou186f/slou186f.pdf>)
12. Imagem de arquivos usados no esquema. Disponível em: <http://www.staples.pt/Imagem.ashx?file=4750&alt=250&larg=250>
13. Imagem do pic18f4550. Disponível em: <http://cienciafeecafe.files.wordpress.com/2010/08/pic18f4550.jpg>
14. Imagem do Microcontrolador MSP430F2370. Disponível em: http://t3.gstatic.com/images?q=tbn:ANd9GcRa0NZJQmDXhOrkjecuX_SlGg2JVtIQXIuweLTIbnikF54VBCW0G

15. *Crystal Oscillator Circuit Design, Data Sheet MX.com, inc.* Mixed Signal ICS.
(<http://www.datasheetcatalog.org/datasheet/CML/mXyyzquy.pdf>)
16. Vantagens e desvantagens do RFID. Disponível em:
<http://ogereente.com/logisticando/2006/12/22/rfid-vantagens-e-beneficios/> [consultado em 01/03/2013]
http://www.gta.ufrj.br/grad/07_1/rfid/RFID_arquivos/prosecontras.htm [consultado em 01/03/2013]
17. Usos e Vantagens do RFID. Disponível em:
<http://www.fit-tecnologia.org.br/laboratorios/rfid/usos-e-vantagens> [consultado em 01/03/2013]
18. *TRF7960 Evaluation Module - ISO 15693 Host Commands.* (Abril de 2008), Texas Instruments (<http://www.ti.com/lit/an/sloa141/sloa141.pdf>)
19. Manual do RFID sobre Alta Frequência. Disponível em:
<http://www.ti.com/rfid/docs/manuals/refmanuals/HFInlaysrefGuide.pdf> [consultado em 10/04/2013]
20. *Tag-it™ HF-I Plus Transponder Inlays- Reference Guide.* (Dezembro 2005, Revisto em 2010), Texas Instruments (<http://www.ti.com/lit/ug/scbu004b/scbu004b.pdf>)
21. *Tag-it™ HF-I Plus Transponder Chip/Inlays, Extended Commands and Options - Reference Guide.* (Dezembro 2005, Revisto em 2010), Texas Instruments (<http://www.ti.com/lit/ug/scbu003a/scbu003a.pdf>)
22. *Tag-it™ Environmental Effects on Transponder Inlays, Application Report.* (Março 1999). Texas Instruments (<http://www.ti.com/lit/an/scba018/scba018.pdf>)
23. Joshua Wyatt et al. *Using Texas Instruments Tag-it™ HF-I Transponder Technology for NFC Vicinity Applications.* (Fevereiro de 2012). Texas Instruments (<http://www.ti.com/lit/an/sloa166/sloa166.pdf>)
24. *Tag-it™ Transponder Protocol - Reference Manual.* (Março de 2000), Texas Instruments (http://www.ti.com/rfid/docs/manuals/refmanuals/tag-it_transponder_protocol.pdf)
25. Imagem do Microcontrolador AT89C51IC2. Disponível em: https://encrypted-tbn1.gstatic.com/images?q=tbn:ANd9GcTLIA_h224aCQ2_9DSLjVPA9SCB5g_V288w2NfUCjzYdY8mSqw0Jw
26. Leitor RFID de 125 khz da botnroll. Disponível em:
http://botnroll.com/product.php?id_product=369 [consultado 05/05/2013]
27. Imagem de um arquivo usado nos esquemas. Disponível em:
http://www.novipapel.pt/imagens/produtos/FO20230-pasta_arquivo_forpus.jpg

28. Identificação animal *Destron Fearin*. Disponível em: <http://www.destronfearing.com/>
[consultado em 12/06/2013]
29. *International Card Manufacturers Association*. Disponível em: <http://www.icma.com/>
[consultado em 13/06/2013]
30. John Schillinger et al, *Antenna Matching for the TRF7960 RFID Reader, SLOA135*. (Maio de 2009), Texas Instruments
31. Imagem da explicação do funcionamento dos infravermelhos. Disponível em:
http://www.mecatronicaatual.com.br/cache/sensores_02_files_image_thumb_medium600_300.jpg
32. Nissanka B. Priyantha, Anit Chakraborty, and Hari Balakrishnan. (Agosto de 2000) “*The Cricket Location-Support System*”. MIT Laboratory for Computer Science, Cambridge
33. Imagem da evolução do RFID. Disponível em: <http://www.seeonic.com/wp-content/uploads/2011/10/Evolution-chart-reduced.png>
34. Imagem da história do RFID. Disponível em:
<http://i.msdn.microsoft.com/dynimg/IC53958.gif>
35. Imagem usada num esquema para ilustrar um leitor.
http://images.tcdn.com.br/img/img_prod/35238/2215_1.jpg
36. Imagem da resposta da frequência das *tags* a fatores externos. Disponível em:
http://www.sensorsmag.com/files/sensor/nodes/2010/7891/Figure2_0.gif
37. Explicação das diversas *tags* RFID. Disponível em:
<http://www.codegate.co.uk/rfid/whichrfidtechnology>
38. Imagem do cabo USB com entrada mini USB para normal. Disponível em:
<http://www.deal pops.com/images/CB-USB-AAMM.jpg>
39. Imagem do cabo USB com entrada micro para mini USB. Disponível em:
<http://usbtips.com/wp-content/uploads/2012/09/palm-800w-12.jpg>
40. ShreHarsha Rao. *Firmware Description of the TI TRF796x Evaluation Module (EVM)*. (Março 2009). *Texas Instruments*
41. ShreHarsha Rao. *Using the SPI Interface With TRF7960*. (Abril 2009). *Texas Instruments*
42. ShreHarsha Rao. *Implementation of the ISO 15693 Protocol in the TI TRF796x*. (Abril 2009). *Texas Instruments*
43. Imagem do terminal android Slick-USB 2. Disponível em:
<http://cdn9.staztic.com/app/a/779/779282/slick-usb-2-serial-demo-8-0-s-307x512.jpg>
44. Cabral Jorge, Gomes Tiago. (19 de Setembro de 2012). *Kit8051USB VI.2*. Guimarães (Documento interno DEI-UM)

-
45. *Tag-it™ HF-I Transponder Inlays- Reference Guide*. (Maio de 2002), Texas Instruments (<http://www.slideshare.net/PeterSam67/tirfid-tagit-hfi-transponder-inlays-reference-guide>)
 46. Lionel M. Ni, Yunhao Liu, Yiu Cho Lau, Abhishek P. Patil (Novembro 2004). *Landmarc: Indoor Location Sensing Using Active RFID*. Volume 10, pp 701-710. Netherlands. (<http://link.springer.com/article/10.1023/B:WINE.0000044029.06344.dd#page-1>)
 47. História do RFID. Disponível em: <http://www.idonic.com/index.php?id=337> [consultado em 25/07/2013]
 48. *Standards RFID*. Disponível em: <http://rfid.net/basics/186-iso-rfid-standards-a-complete-list> [consultado em 07/08/2013]
 49. Antunes Mafalda, Fontes José. (26 de Abril 2004) “*Transponders*”. Universidade do Minho, Engenharia Eletrónica e Industrial, Comunicações Industriais, Guimarães
 50. Roy Want et al. “*The Active Badge Location System*”. Olivetti Research Ltd. (ORL), Cambridge, England
 51. Bahl, P., Padmanabhan, V.N. “*Radar: na in-building RF-based user location and tracking system*”(26 Mar 2000), Tel Aviv

Anexos:

Anexo 1- Placas

A primeira *board* criada através deste esquema tinha apenas uma face como se pode ver abaixo.

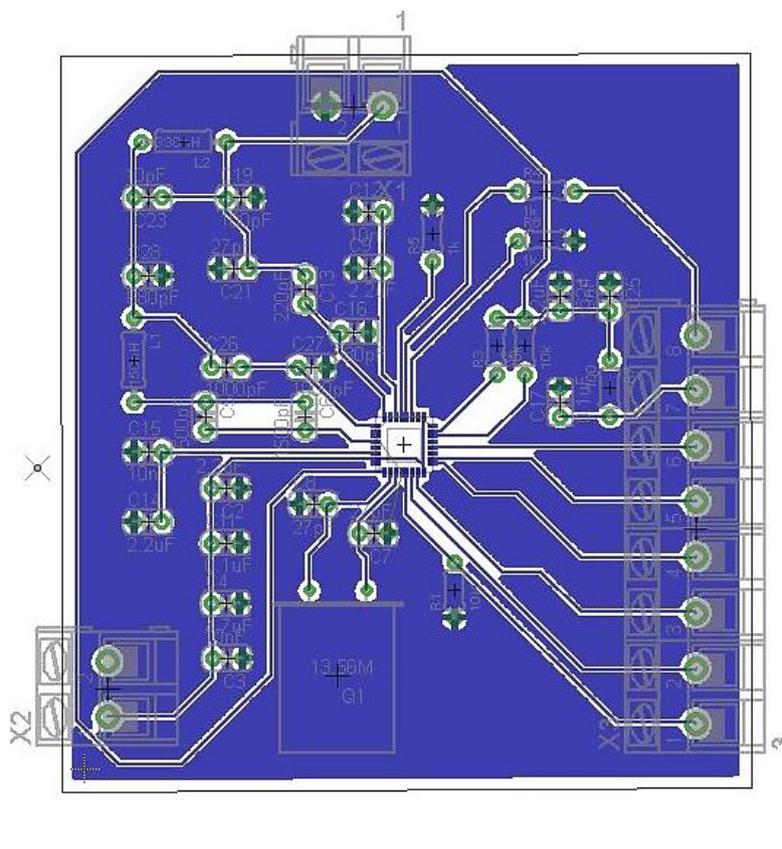


Figura A-1-Placa de face simples

Infelizmente devido às propriedades do TRF7960 viu-se que ao colar o respetivo integrado na placa este iria sobreaquecer todo o circuito e conseqüentemente destruir o correto funcionamento do mesmo. Ainda se tentou isolar a parte que poderia aquecer, mas infelizmente não se conseguiu resolver este problema. O PCB criado encontra-se ilustrado na figura abaixo:

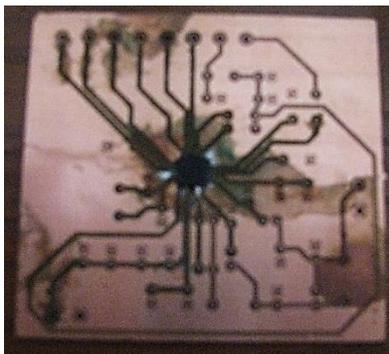


Figura A-2-Primeiro PCB de face simples criado

Tal como se pode ver este PCB encontra-se em mau estado devido às diversas tentativas para o tentar colocar a funcionar.

Devido ao problema do sobreaquecimento do integrado foi feita uma nova placa, desta vez de face dupla. Nas figuras abaixo encontram-se ilustradas cada uma das faces, bem como as duas sobrepostas e a imagem final do PCB montado.

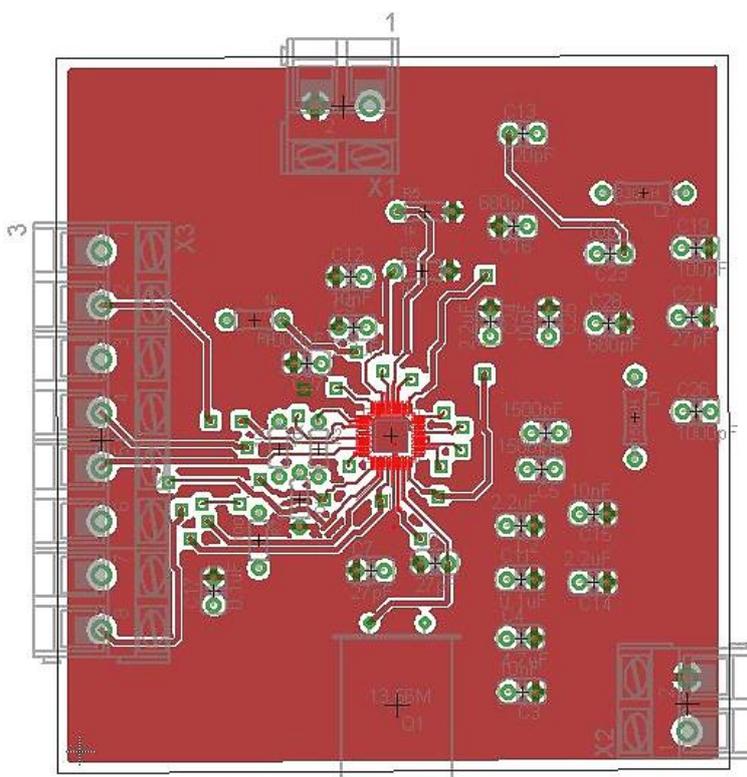


Figura A-3-Face de cima do PCB

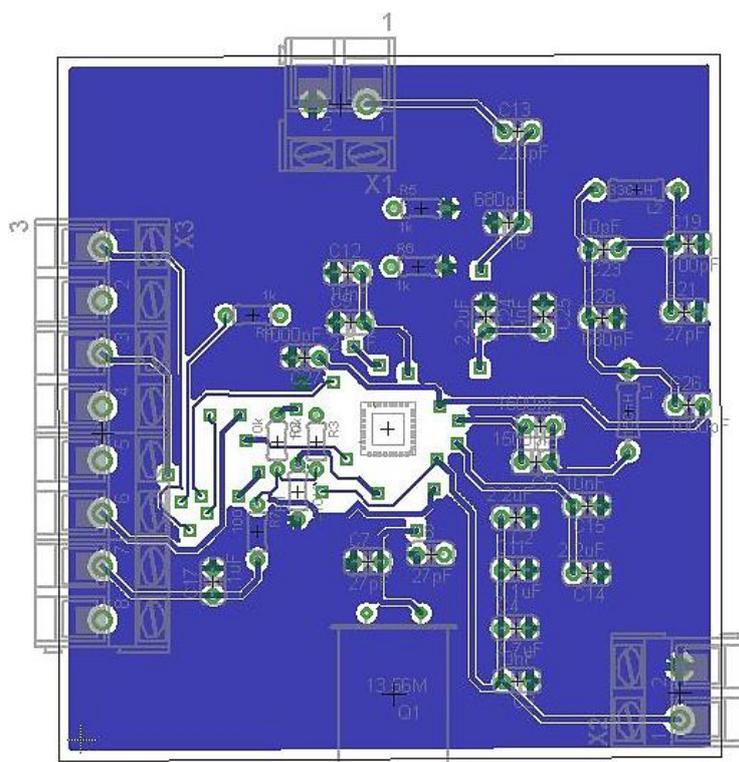


Figura A-4-Face de baixo do PCB

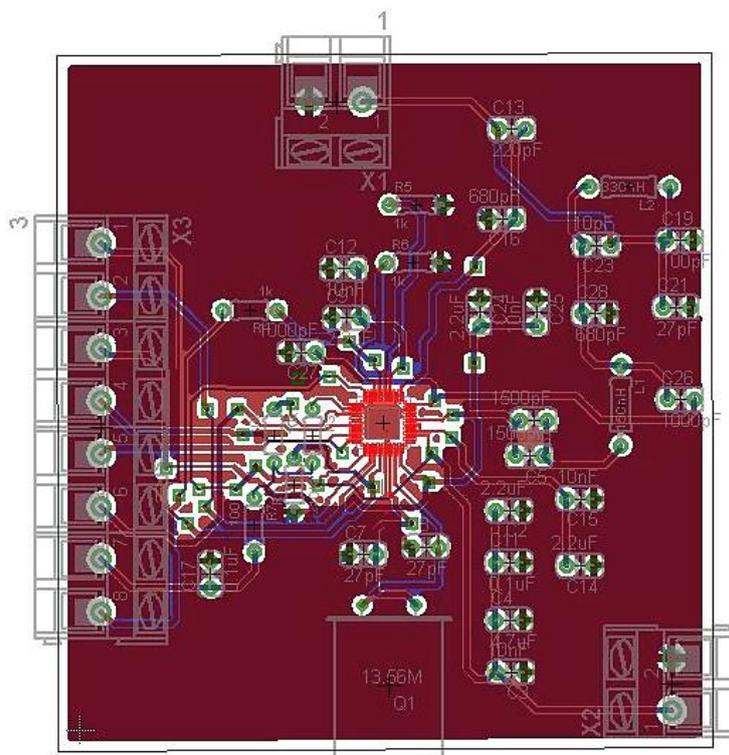
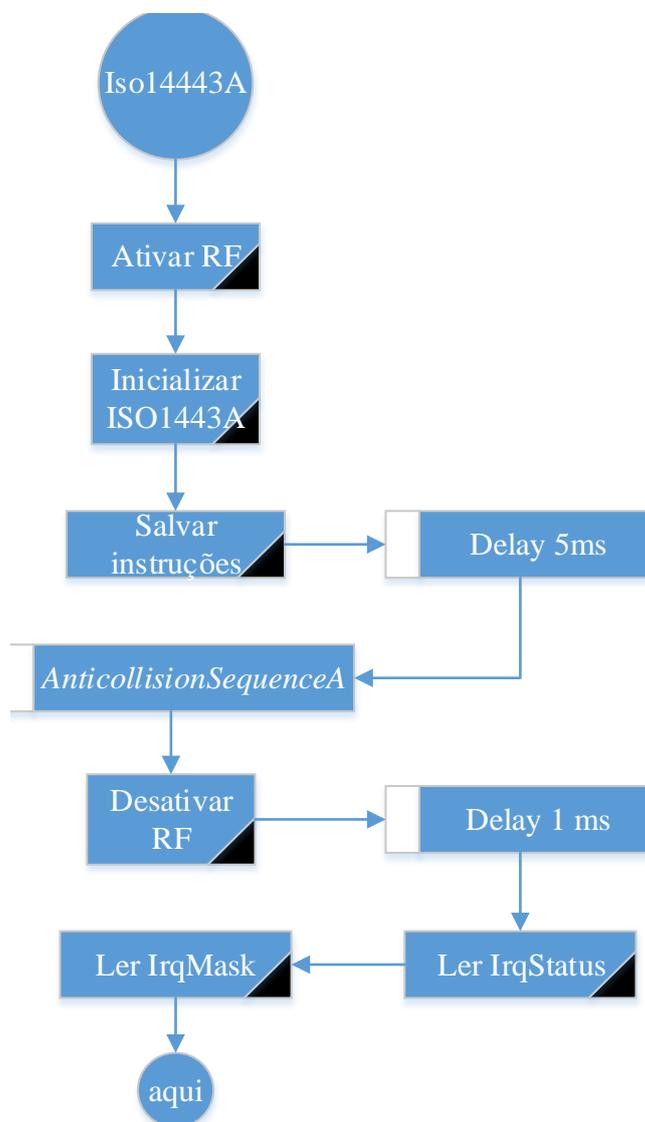


Figura A-5- PCB com as duas faces

Este esquema também não foi o adotado pois o integrado TRF7960 queimou devido a uma má configuração da placa e teve de se fazer uma nova.

Anexo 2-Fluxogramas

Figura A-6- Diagrama do comportamento da função *FindTags* (cont1)

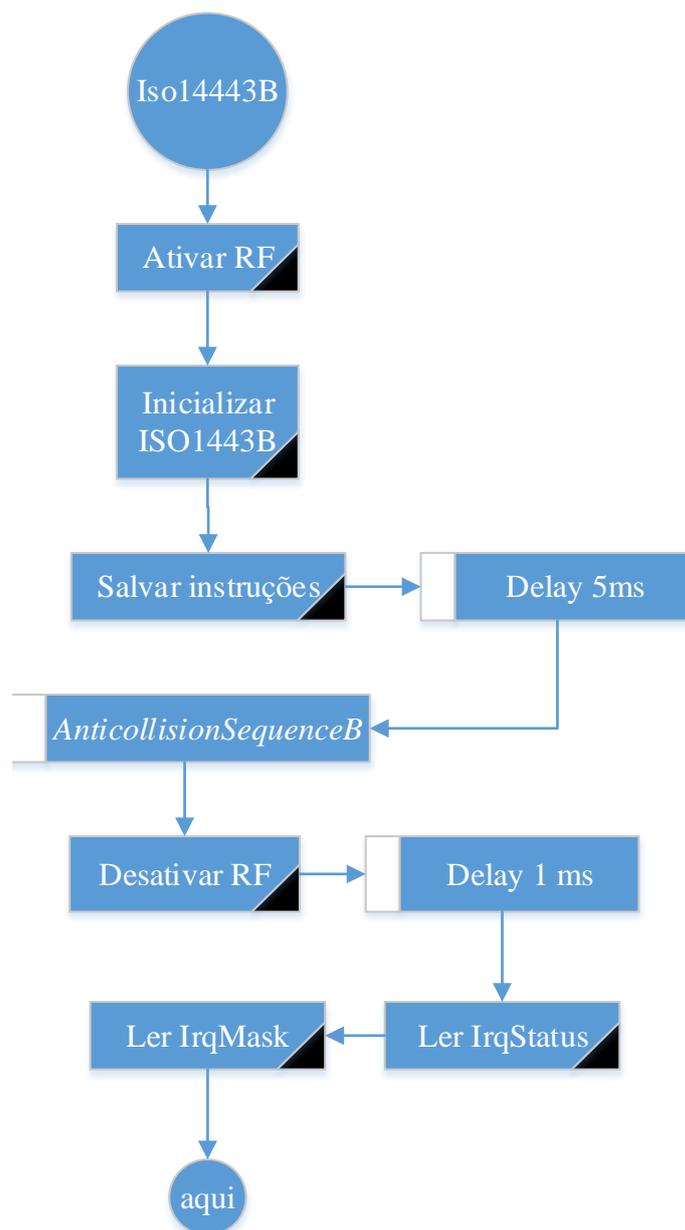
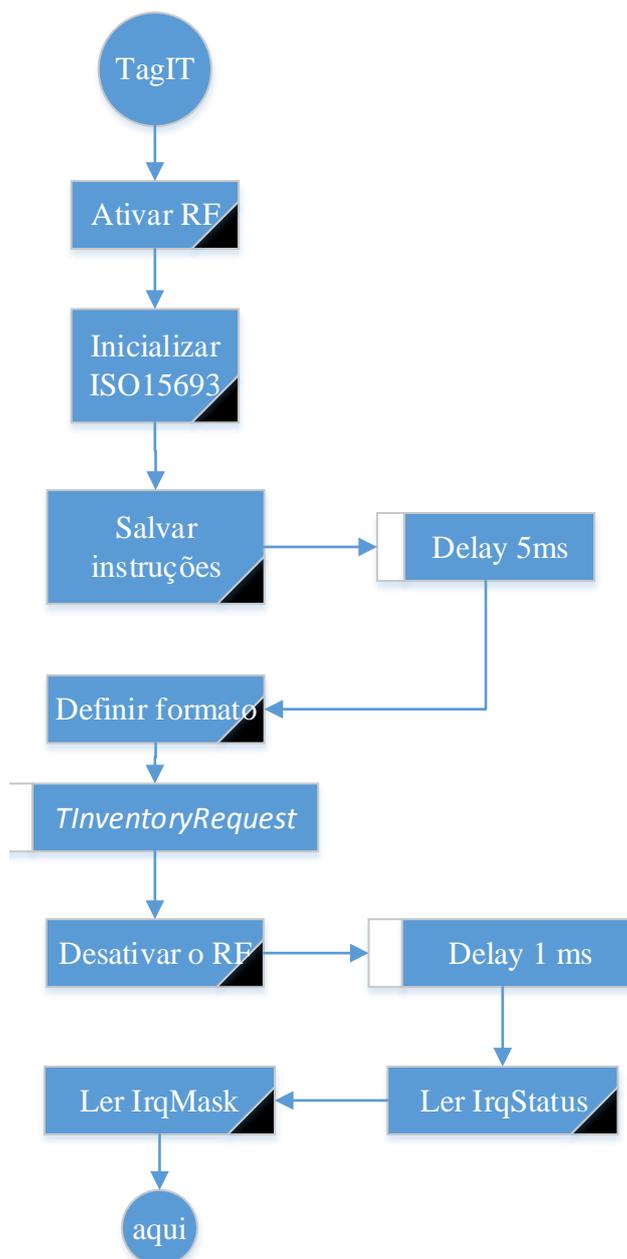


Figura A-7- Diagrama do comportamento da função *FindTags* (cont2)

Figura A-8- Diagrama do comportamento da função *FindTags* (cont3)

AnticollisionSequenceA

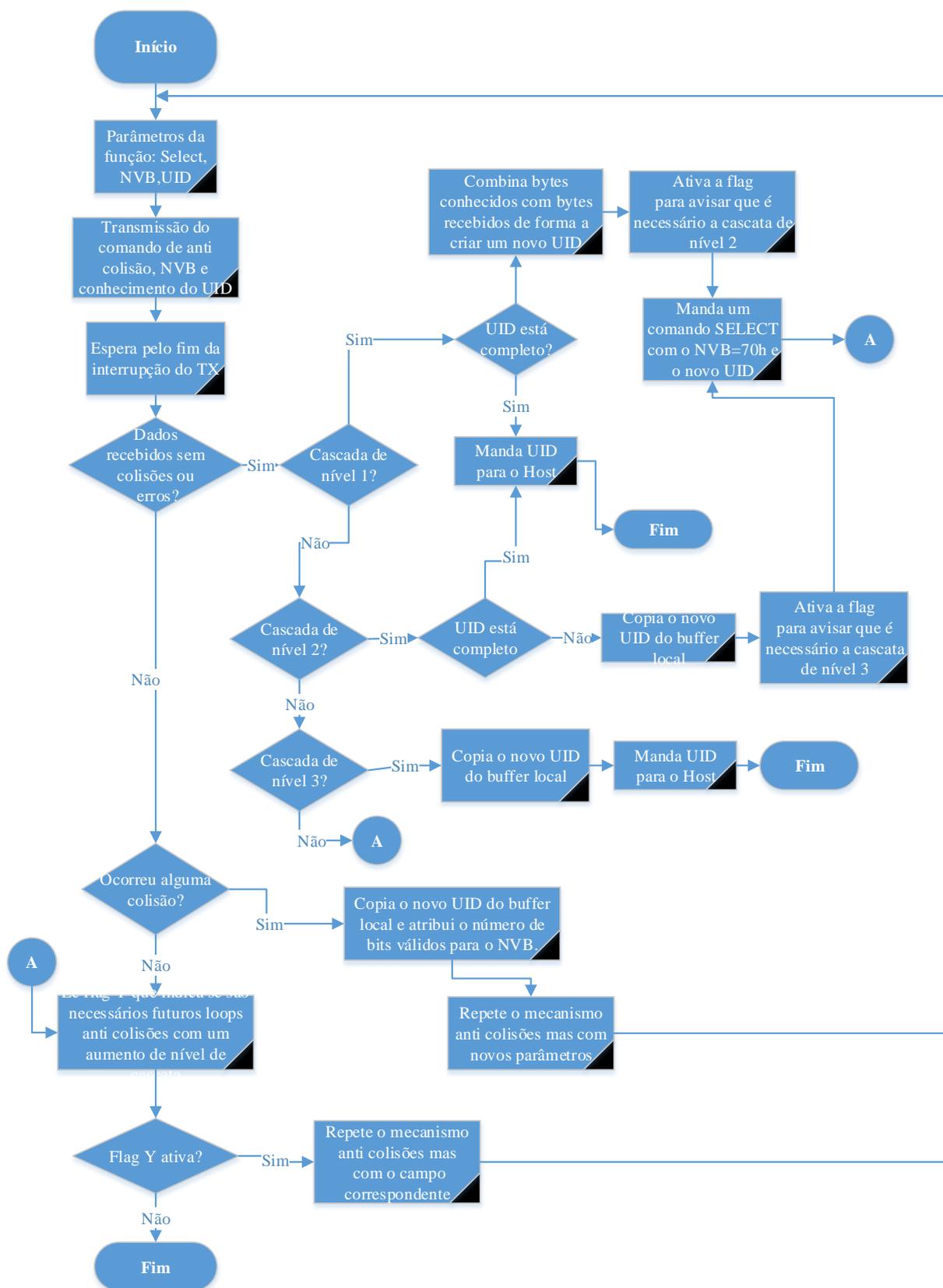


Figura A-9-Diagrama com a explicação do mecanismo anticolisões para o ISO 14443A

Esta função é invocada sempre que é escolhido o protocolo ISO 14443A.

Se tivermos em conta o *standard* ISO 14443B temos de usar o seguinte algoritmo:

AnticollisionSequenceB

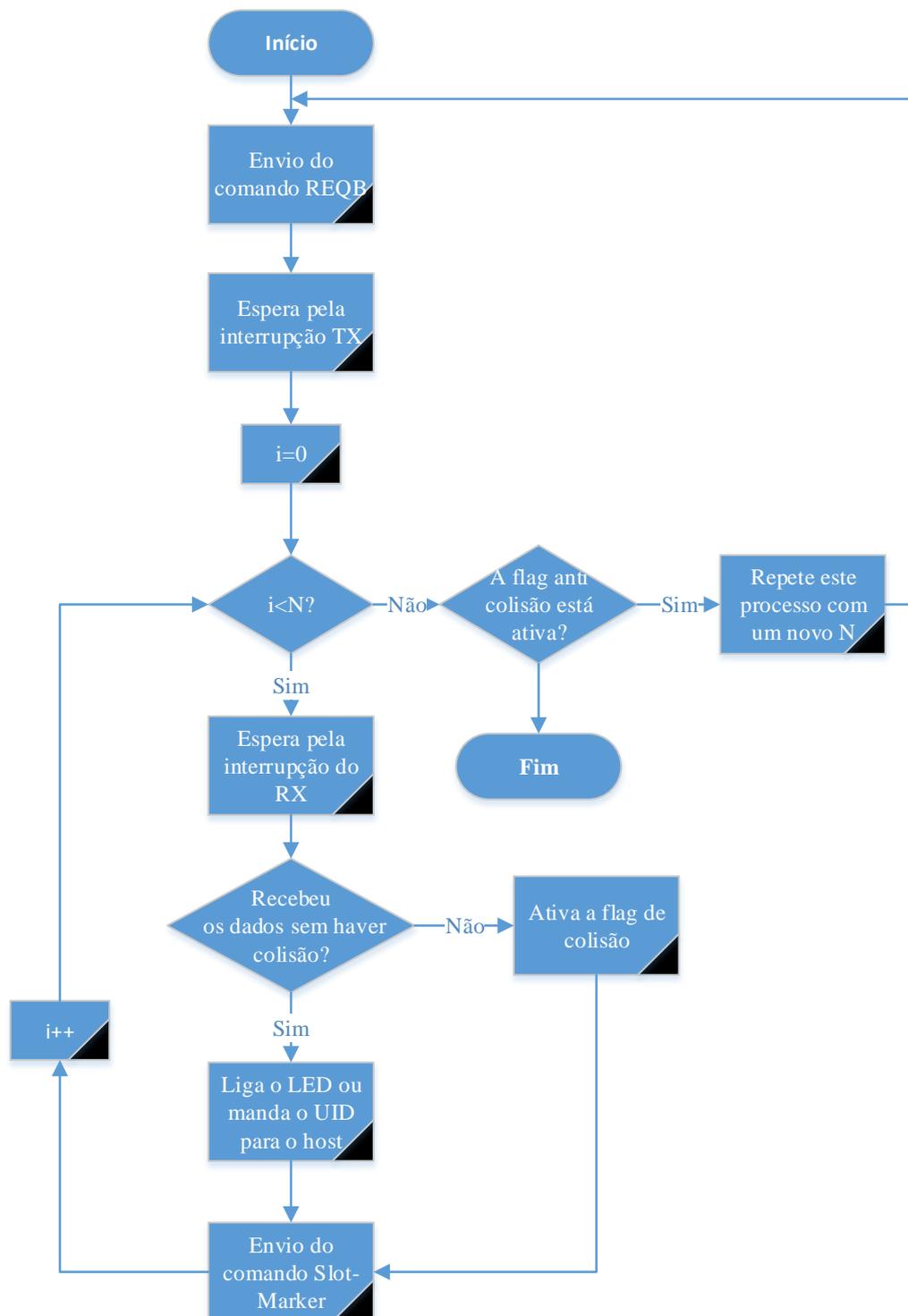


Figura A-10-Diagrama com a explicação do mecanismo anticolisões para o ISO 14443B

Como se pode ver pela imagem o tratamento anticolisões deste *standard* é muito mais simples que o anterior.

TInventoryRequest

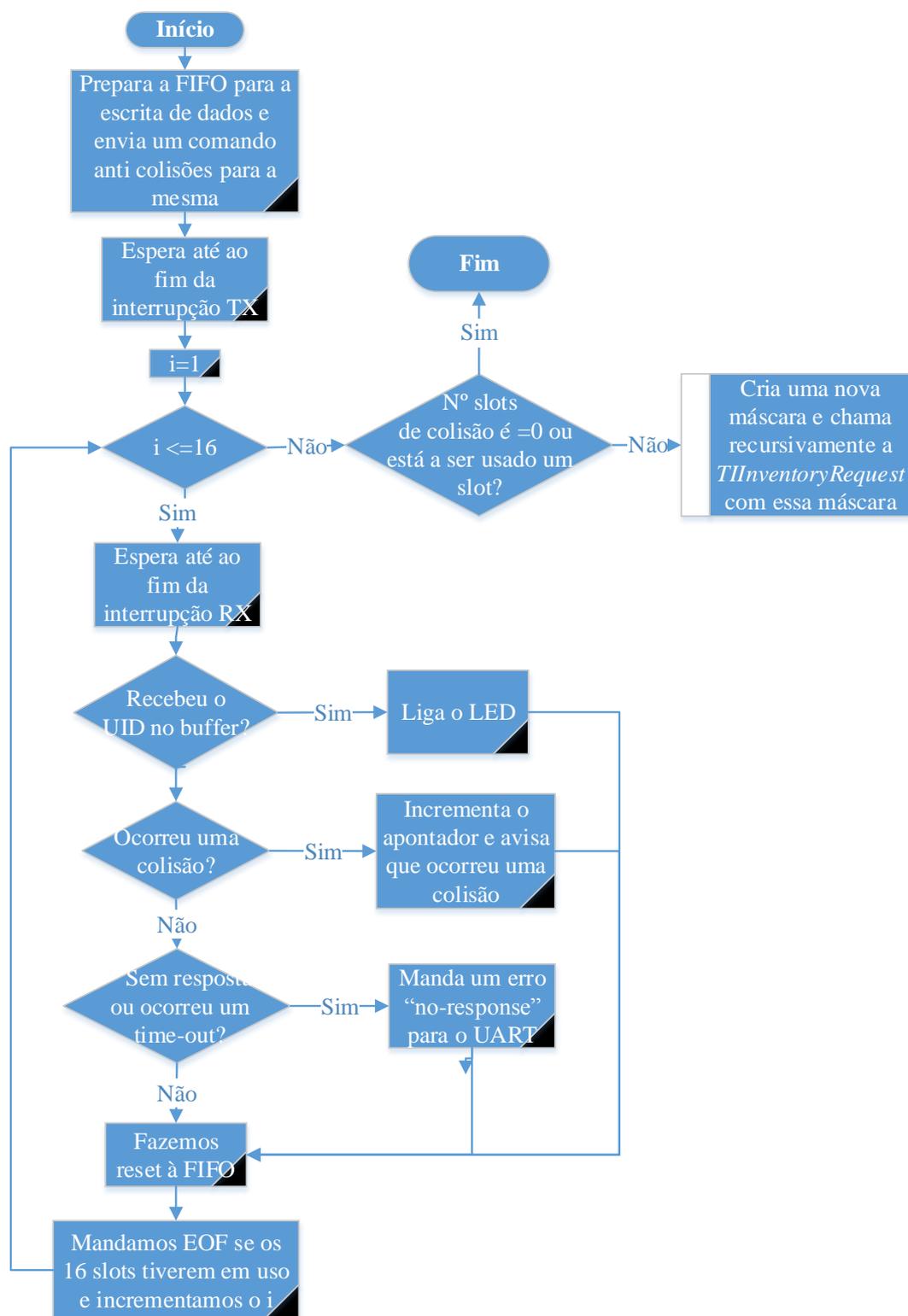


Figura A-11-Diagrama da função TInventory Request

O diagrama acima é feito quando é escolhido o protocolo *Tag-it* no *Find Tags* em vez da função *InventoryRequest* do protocolo ISO 15693

Anexo 3- Outros

Recursos necessários

Tabela A-1- Recursos

Resource Name	Type	Material Label	Initials	Group	Max. Units
Eagle 5.10.0	Material		E	Conceção	
Microsoft Visual C++ 2010 Express	Material		MVC	Conceção	
Visio 2013	Material		V	Conceção	
Componentes eletrónicos (resistências,...)	Material		CE	Conceção	
Ferro de soldar	Material		F	Conceção	
Microsoft Word	Material		MW	Escrita	
Miktex	Material		M	Escrita	
Winedt 7.0	Material		W	Escrita	
Latex	Material		L	Linguagem Programação	
Linguagem C	Material		LC	Linguagem Programação	
Google Chrome (Internet)	Cost		GCH	Pesquisa	
Artigos	Material		A	Pesquisa	
Ana Eduarda	Work		A	Pessoas	100%
Texas Instruments	Work		T	Pessoas	100%
Oficinas UMINHO	Work		OU	Pessoas	100%
Microsoft Project	Material		MP	Planeamento	
Digital Lab	Material		DL	Testes	
Osciloscópio	Material		O	Testes	
Gerador de onda	Material		GO	Testes	

Computador	Material		C	Varios	
------------	----------	--	---	--------	--

Tal como se pode ver pela figura os recursos foram divididos 7 grupos principais:

- Conceção- corresponde ao conjunto de materiais que vão ser utilizados para contruir o protótipo e programá-lo.
- Escrita- corresponde às ferramentas que vão ser utilizadas para escrever a dissertação.
- Linguagem de programação- Corresponde às linguagens que se vai utilizar para construir o protótipo.
- Pesquisa- corresponde às ferramentas utilizadas para fazer pesquisa sobre o tema.
- Pessoas- é o grupo de pessoas que trabalharam no projeto. Neste grupo tenho incluído a *Texas Instruments* que é responsável por enviar diversas amostras para o projeto, as Oficinas onde um grupo de profissionais me ajuda sempre que preciso e por último eu mesma que trabalho em todas as fases do projeto.
- Testes- Material que foi usado para auxílio de testes de funcionamento do protótipo.
- Vários- Material que se insere em mais do que um grupo, o computador foi usado para pesquisa, conceção, escrita, ...

Configuração da comunicação série do microcontrolador

A interação com o micro é feita através da porta série onde o utilizador pode através do teclado dizer qual é a atividade que quer que o leitor faça.

Os registos usados para aceder a porta série foram os seguintes:

-> Registo SCON, #01010000 (50H)

SM0	SM1	SM2	REN	TB8	RB8	TI	RI
-----	-----	-----	-----	-----	-----	----	----

- SM0 e SM1 (Seleciona o modo de operação):
SM0=0 e SM1=1, Modo 1, pois neste modo a porta série efetua a transferência série assíncrona, operando sobre 10 bits (8 bits de dados, encapsulados entre o start bit e o stop bit), é assíncrona porque não queremos o clock pois vou usar Timer's.
- SM2 (Ativação de capacidade de comunicação entre múltiplos processadores nos modos 2 e 3):
SM2=0, como estamos no modo funcionamento 1, se SM2=1 então RI não seria ativado se não fosse encontrado um stop bit válido.

- REN (Ativação da receção):
REN=1, ativa a porta série.
- TB8 (Nono bit transmitido nos modo 2 e 3):
TB8=0, pois estamos no modo 1.
- RB8 (Nono bit recebido nos modos 2 e 3):
RB8=0, no modo 1 se SM2=0, RB8 é o stop bit.
- TI (*Flag* de interrupção da transmissão):
TI=0, esta têm de estar a 0 pois ainda não houve um final de uma transmissão.
- RI (*Flag* de interrupção de receção):
RI=0, esta têm de estar a 0 pois ainda não houve um final de uma receção.

->Registo T2CON, #00110100 (34H)

Tabela A-2-Registos de comunicação série-T2CON

TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2
-----	------	------	------	-------	-----	------	--------

- TF2 (*Flag* de transbordo de contagem do Timer 2):
TF2=0, ativada por *hardware* (colocada a 1) na ocorrência do transbordo no registo de contagem, apenas quando RCLK=0 e TLCK=0, ou seja, não pode estar a 1 ao mesmo tempo que RCLK e TLCK.
- EXF2 (*Flag* externa do Timer 2):
EXF2=0, ativada quando a captura ou carregamento ocorre devido a uma transição negativa T2EX e EXEN2=1.
- RCLK (Relógio de receção do Timer 2):
RCLK=1, como está habilitado indica que o relógio da comunicação série durante a receção deve ser obtido do timer 2.
- TCLK (Relógio de transmissão do Timer 2):
TCLK=1, como está habilitado indica que o relógio da comunicação série durante a transmissão deve ser obtido do timer 2.
- EXEN2 (Habilita entrada externa do Timer 2):
EXEN2=0, quando ativada permite que a captura ou o carregamento dependam de uma entrada externa.
- TR2 (Controla o arranque/paragem da contagem do Timer 2):

TR2=1, Ativa o Timer 2, ou seja, ativa o arranque e a paragem da *contagem* quando colocado a 1 e 0, respetivamente.

- C/T2 (Seleciona modo temporizador ou contador):

C/T2=0, modo temporizador com *contagem* a cada ciclo máquina.

- CP/RL2 (Seleção entre o modo de captura ou de carregamento):

CP/RL2=0, como RCLK=1 e TCLK=1 o estado deste bit é ignorado.