



universidade
de aveiro

Projeto em Engenharia de Automação
Departamento de Engenharia Mecânica

Class Quiz

Prof. Vítor Santos
Prof. Pedro Fonseca

Autor:
António Teixeira (72293)

Índice

Índice.....	3
Índice de Figuras.....	5
Índice de Tabelas.....	5
Resumo.....	7
1. Introdução.....	1
2. Objetivos.....	3
3. Planificação do Projeto.....	5
3.1. Casos de uso.....	5
3.2. Cronograma.....	6
4. Tipos de comunicação.....	9
5. Desenvolvimento do Hardware.....	11
5.1. Hardware dos terminais.....	11
5.2. Hardware da Estação base.....	12
6. Layout da caixa dos Terminais.....	15
7. Desenvolvimento do Software.....	17
7.1. Software dos terminais.....	17
7.2. Software da estação base.....	20
8. Teste e resultados obtidos.....	21
9. Conclusão.....	23
10. Referências bibliográficas.....	25

Índice de Figuras

Figura 1- Cronograma do projeto	7
Figura 2- Diagrama de blocos dos terminais	11
Figura 3- Esquema elétrico dos terminais	12
Figura 4- Diagrama de blocos da estação base	12
Figura 5- Esquema elétrico da estação base	13
Figura 6- Base da caixa dos terminais	15
Figura 7- Parte superior da caixa dos terminais	16
Figura 8- fluxograma do software dos terminais.....	18
Figura 9- Onda quadrada de 1KHz	21
Figura 10- Onda quadrada de 100Hz.....	21
Figura 11- Aquisição de dados do recetor.....	22

Índice de Tabelas

Tabela 1 - Cronograma do projeto	6
Tabela 2- Portas utilizadas no Arduíno Uno	17
Tabela 3- Portas utilizadas pelo Arduíno Mega 2560	20

Resumo

Este trabalho consiste em programar e projetar terminais portáteis conectados por *wireless* a uma estação base.

Os terminais portáteis são programados por um Arduino Uno. Estes terminais são capazes de reconhecimento de alunos através de um leitor de cartões MIFARE SL031.

A estação base é programada por um Arduino Mega2560. A estação base está ligada a interface humana (PC), para o docente conseguir observar as respostas dadas pelos alunos.

Na comunicação Wireless foi necessário um estudo, sobre vantagens e desvantagens dos modems existentes no mercado.

No presente relatório serão descritas e apresentadas todas as considerações necessárias para garantir o funcionamento correto dos mesmos.

1. *Introdução*

No sistema de avaliação pedagógica existe uma necessidade de criar uma avaliação contínua da participação dos estudantes, onde se obterá resultados imediatos e automáticos.

O processo para obter resultados imediatos e automáticos é a elaboração de um exame de escolha múltipla numa plataforma visual. As respostas dos alunos são dadas através de terminais portáteis conectadas por *wireless* a um recetor, que tem como função armazenar as respostas, e enviá-las para um software capaz de avaliar as respostas automaticamente.

A proposta deste trabalho é exatamente uma solução ao processo apresentado anteriormente. O trabalho consiste no desenvolvimento de hardware e software, capaz de realizar resultados imediatos e automáticos da avaliação de alunos. O trabalho dos terminais portáteis desenvolvidos tem por base o Arduíno Uno, e o trabalho da estação-base tem por base o Arduíno Mega 2560.

2. *Objetivos*

Os principais objetivos que este projeto apresenta são os seguintes:

- O dimensionamento e construção dos terminais individuais;
- O circuito eletrónico dos terminais individuais;
- O leitor de cartões a utilizar para o reconhecimento do aluno;
- O software utilizado para o funcionamento dos terminais individuais;
- O software utilizado para o reconhecimento do aluno;
- O modo de comunicação entre os terminais individuais e a estação-base;
- O software para o modo de comunicação;
- O software da estação-base recetora;

3. Planificação do Projeto

3.1. Casos de uso

Este subcapítulo fala de casos de uso que são implementados neste projeto, e possíveis casos que possam ser estudados em teses ou até mesmo noutro projeto com o mesmo tema. O primeiro caso de uso pensado para este projeto apresenta a seguintes tarefas:

- Leitura de cartões, esta tarefa tem como função a restrição de pessoas a utilizar os terminais, ou seja, só utiliza os terminais quem tiver o cartão da universidade consigo;
- Comunicação entre um terminal e a estação base, nesta tarefa o principal objetivo é utilizar um tipo de comunicação e implementar uma programação para testar a comunicação.

O segundo caso de uso apresenta a seguintes tarefas:

- Tal como o primeiro caso, este também tem uma leitura de cartões;
- Comunicação entre vários terminais e a estação base, nesta tarefa o objetivo é verificar o tipo de diálogo adequado a este projeto, e também verificar a qualidade do sinal enviado pelos transmissores.

O terceiro caso de uso apresenta a seguintes tarefas:

- Associar a leitura de cartões a uma base de dados. A base de dados associa os dados de um aluno aos bytes lidos pelo leitor de cartões;
- Tal como no segundo caso, este também tem a comunicação entre vários terminais e a estação base.

Como se pode observar estes 3 casos de uso necessitam de uma investigação sobre vários tipos de comunicação, para ver qual o tipo mais apropriado para cada caso. Também se verifica que este projeto é completo na área de automação industrial, porque é necessário um conhecimento de programação de microcontroladores, para controlar o leitor de cartões e a comunicação entre o emissor e recetor.

O terceiro caso já é um projeto mais avançado, porque apresenta uma relação entre a programação do microcontrolador e a programação da base de dados, onde na programação de base de dados, para a programação da base de dados pode-se utilizar o software MySQL.

Em todos os casos a Estação base necessita de uma interface humana. Para interface pode-se utilizar softwares já existentes, como por exemplo Putty, Terminal e Arduíno, ou pode-se criar uma interface utilizando os softwares Visual Basic ou Matlab.

3.2. Cronograma

A planificação do projeto é uma fase muito importante, porque apresenta todas as tarefas que o grupo vai fazer e o tempo que essas tarefas deram a ser realizadas. Este cronograma pretence ao primeiro caso de uso, pois foi o caso que foi feito neste projeto. Sabendo as tarefas a ser realizadas é necessário definir o tempo para cada tarefa. O tempo é limitado pois é necessário ter as fases concluídas no prazo de entrega definido pelo regente da cadeira. O tempo de cada tarefa pode ser visualizado na tabela 1.

Tabela 1 - Cronograma do projeto

Cronograma Projeto em automação			
Tarefa	Início	Fim	Duração (dias)
Planeamento do trabalho	22/mar	14/abr	24
Elaboração de um roteiro tecnológico	22/mar	1/mai	41
Desenho do circuito elétrico dos terminais	5/mai	5/mai	1
Desenho do circuito elétrico da estação base	5/mai	5/mai	1
Elaboração de uma placa PCB do circuito dos terminais	10/mai	11/mai	2
Montagem do circuito da estação base	10/mai	11/mai	2
Desenho 3D dos terminais	7/jun	8/jun	2
Desenvolvimento do software para emissão e receção de dados	15/mai	4/jun	21
Desenvolvimento do software para reconhecimento dos alunos	15/mai	3/jun	20
Desenvolvimento do software para configurar os botões	4/jun	7/jun	4
Teste ao Software completo	20/jun	22/jun	3
Desenvolvimento do relatório	13/jun	11/jul	29
Entrega do relatório final	12/jul	12/jul	1
Apresentação do trabalho	14/jul	14/jul	1

Com as tarefas e sua duração definidas, o próximo passo foi a realização de um gráfico de *gant* este gráfico demonstra como estão relacionadas as tarefas entre si. A figura 1 representa o gráfico:

Class Quis – PEA – Universidade de Aveiro

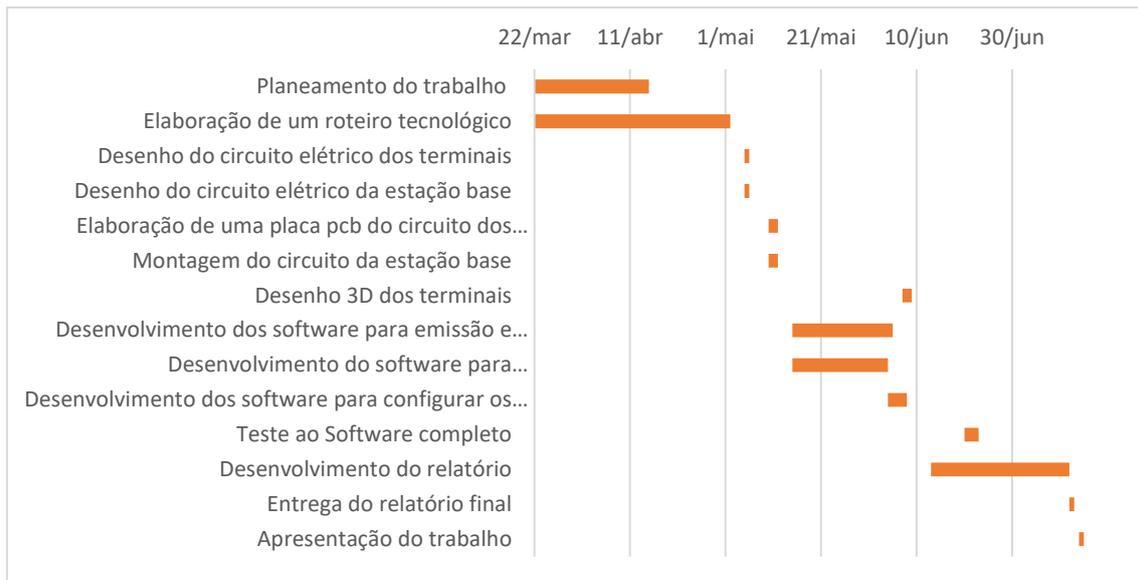


Figura 1- Cronograma do projeto

4. Tipos de comunicação

Neste capítulo foi necessária uma pesquisa sobre os vários tipos de comunicação existentes, e que fossem importantes para o projeto.

- Módulo Infravermelho
Comunicação infravermelho transmite informações entre dispositivos utilizando radiação infravermelho. Este tipo de comunicação só funciona exclusivamente em linha de vista, ou seja, os terminais e a estação-base tem de estar em contacto visual um com o outro para que esta comunicação funcione. A comunicação infravermelha é muito utilizada controles remotos de curto alcance. Para que esta comunicação seja utilizada, é necessário um emissor de sinais, e um recetor que vai capturar e interpretar os sinais transmitido [5].
- Módulo Wi-Fi
Wi-Fi é uma comunicação de baixa potência que é utilizada por vários dispositivos eletrónicos. Esta comunicação utiliza um router como estação-base conectada a uma rede, e por sua vez os terminais podem-se conectar a essa rede. Para os terminais conseguirem conectar, estes necessitam de ter no seu dispositivo eletrónico um recetor de sinal Wi-Fi. Os terminais na comunicação com estação-base utilizam onda de radiofrequência. A questão de segurança e privacidade não é problema na comunicação Wi-Fi, porque são utilizadas senhas que fazem que o acesso seja restrito aos terminais vão ser utilizados, essas senhas são utilizadas no de reconhecimento dos dados que estão a ser transmitidos entre os terminais e a estação-base. Na comunicação Wi-Fi pode-se utilizar dispositivos eletrónicos denominados repetidores, que servem para expandir o alcance da comunicação [5].
- Módulo Bluetooth
A comunicação Bluetooth é utilizada para comunicar entre vários dispositivos eletrónicos, é uma comunicação de baixa potência e também de curto alcance, esta comunicação tem um alcance máximo de 15 metros. Os terminais na comunicação com estação-base utilizam onda de radiofrequência. Antes de iniciar a transmissão de dados, é necessário que os terminais que irão se conectar passem por um processo de reconhecimento [5].
- Módulo WIMAX
WIMAX é uma comunicação muito parecida com a comunicação Wi-Fi, embora esta apresente um melhor desempenho, permitindo velocidades maiores que 1Gbit/s. O WIMAX tem um problema a linearidade do sinal

Class Quis – PEA – Universidade de Aveiro

durante a transmissão de dados dentro de um canal de rádio entre os terminais e a estação-base. Nesta comunicação a transmissão de dados terá vários tipos de modulação, essa modulação depende da posição do sinal dentro do canal. Esta comunicação só utiliza as modulações de sinal com elevado desempenho, com a finalidade de possibilitar a otimizar a taxa de transferência da transmissão de dados [5].

- Módulo Zigbee

Zigbee é uma comunicação de baixa potência, baixa taxa de transmissão de dados e baixo custo de implementação, que é utilizada por vários dispositivos eletrónicos. Os terminais na comunicação com estação-base utilizam onda de radiofrequência. Esta comunicação apresenta um menor consumo, por um alcance de 100 metros, e a comunicação pode ser repetida entre unidades existentes na rede até chegar á estação-base. A comunicação Zigbee pode ter taxas de 250 Kbps que alcançam 70 metros. Esta comunicação possibilita que todas as unidades recebam dados, para configurar o nome de utilizador e o endereço da rede, e por fim os endereços recebam dados numa determinada secção [5].

- Módulo Radiofrequência

O módulo Radiofrequência permite a comunicação entre vários dispositivos. Esta comunicação permite a transmissão de sinais digitais. Para conseguir a comunicação por radiofrequência é necessário utilizar emissores de transmissão de ondas rádio conectados a uma antena recetora que esteja sintonizada na mesma frequência que os emissores. A radiofrequência é uma comunicação de baixo custo, de baixa potência e tem um alcance de 200 metros [5].

Escolha do tipo de comunicação

Como este projeto inicial é um teste as comunicações sem fios, utilizou-se um módulo de radiofrequência de baixo custo. Este módulo apenas é um teste para verificar quantos terminais podemos ligar ao recetor, a velocidade e distâncias de comunicação.

5. Desenvolvimento do Hardware

5.1. Hardware dos terminais

Para organizar e auxiliar a elaboração do hardware do projeto foi necessário a elaboração de um diagrama de blocos para os terminais e outro diagrama de blocos para a estação base. Na figura 2 representa-se o diagrama de blocos dos terminais.

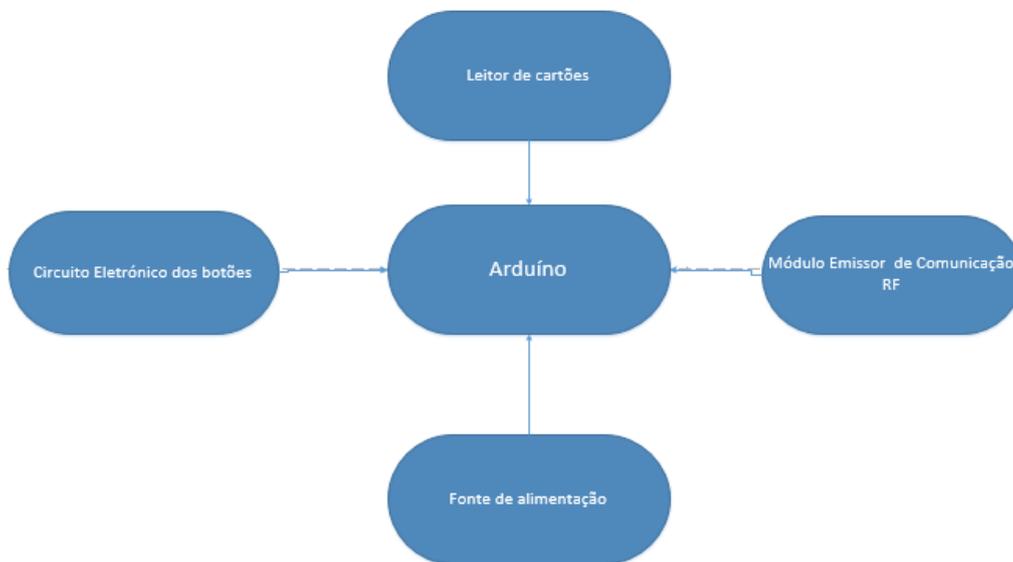


Figura 2- Diagrama de blocos dos terminais

Como podemos verificar na figura anterior o Arduíno Uno vai estar a controlar o leitor de cartões SL031, um circuito elétrico constituído por 5 botões que na programação significam as alíneas A até E das respostas, e também vai controlar um transmissor RF de modelo MX-FS-03V. A alimentação do Arduíno é feita através de uma bateria de 9V, esta alimentação foi verificada na *datasheet* do Arduíno Uno ^[1]. Por fim este circuito vai ser montado numa placa PCB que vai estar dentro de uma caixa desenhada em *Solid Edge*.

Com o diagrama de blocos terminado, o desenho do circuito no software EAGLE é muito mais fácil de realizar. A figura 3 representa o esquema do circuito dos terminais.

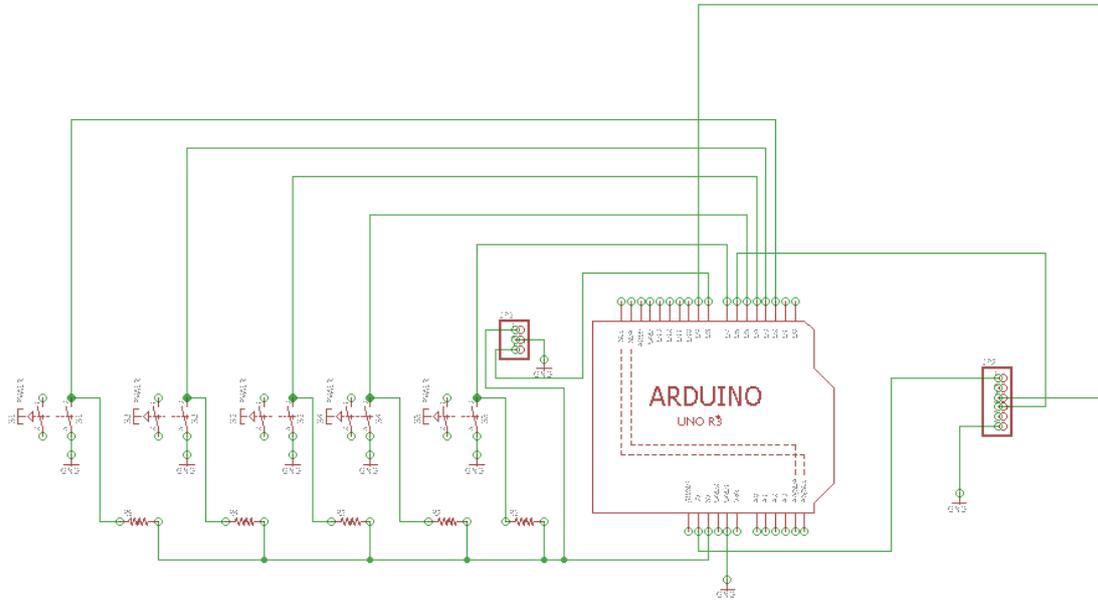


Figura 3- Esquema elétrico dos terminais

Na figura 3 utilizaram-se resistências de 100Ω em série com os botões, esta resistência serve para colocar o interruptor a 5V quando este está aberto, e a 0V quando o interruptor fecha. Verifica-se que o JP2 do circuito é para ligar o leitor de cartões, este leitor é alimentado a 3,3V [2], e como utiliza a comunicação UART só necessitamos de ligar o TX a uma porta analógica do Arduino e o RX a outra porta analógica. O JP1 serve para ligar o transmissor RF, que é alimentado por 5V [3], e o pino que envia dados do transmissor liga a uma porta analógica do Arduino.

5.2. Hardware da Estação base

A elaboração do hardware da estação também começou pela realização de um diagrama de blocos. Na figura 4 representa-se o diagrama de blocos da estação base.



Figura 4- Diagrama de blocos da estação base

Class Quis – PEA – Universidade de Aveiro

Como podemos verificar na figura anterior o Arduino Mega 2560 vai estar a controlar apenas um recetor RF de modelo MX-05V. Alimentação deste circuito é através do cabo que liga o Arduino ao PC. A aplicação de programação do Arduino serve também de interface humana, esta interface serve para verificar as respostas dadas pelos terminais.

Com o diagrama de blocos terminado, o desenho do circuito no software EAGLE é muito mais fácil de realizar. A figura 5 representa o esquema do circuito da estação base.

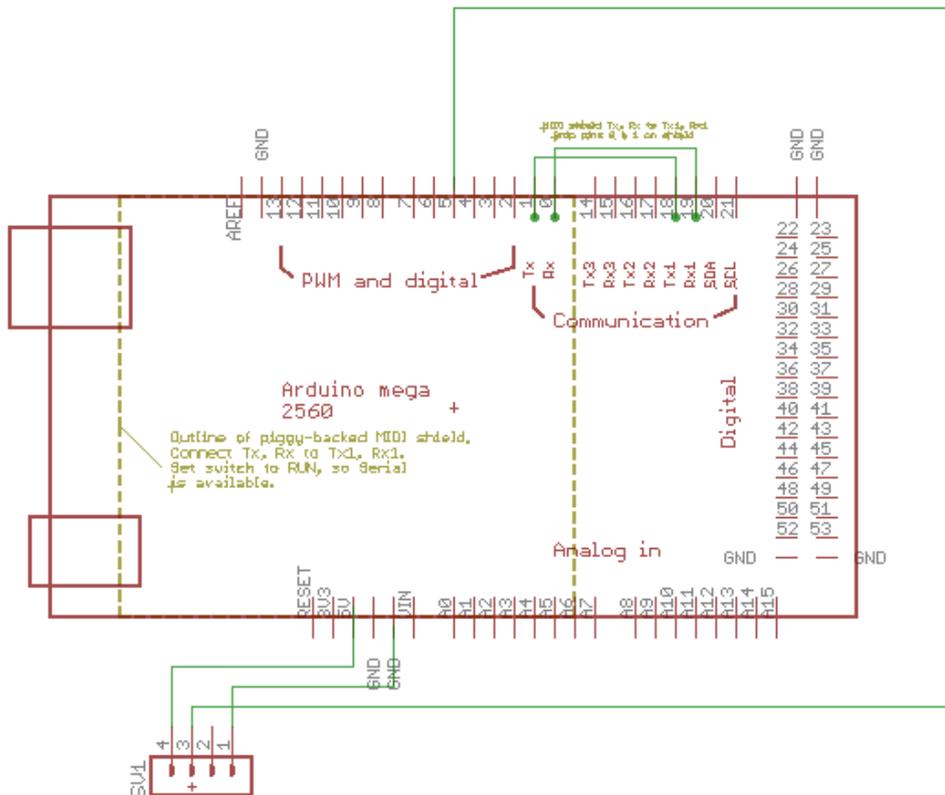


Figura 5- Esquema elétrico da estação base

Na figura 5 o SV1 representa o recetor RF, este é alimentado a 5V e tem dois pinos para a saída de dados [3]. Os dados nesses dois pinos são representados de igual forma, por isso só se liga um dos pinos a uma porta analógica do Arduino.

6. *Layout da caixa dos Terminais*

Depois de o circuito elétrico estar montado numa placa PCB, desenhou-se uma caixa no software Solid Edge. O desenho da caixa foi dividido em duas peças. A primeira peça vai ter um sítio para colocar a bateria do circuito, e por cima da bateria vai ficar o circuito elétrico dos terminais. A figura 6 demonstra o desenho da primeira peça.

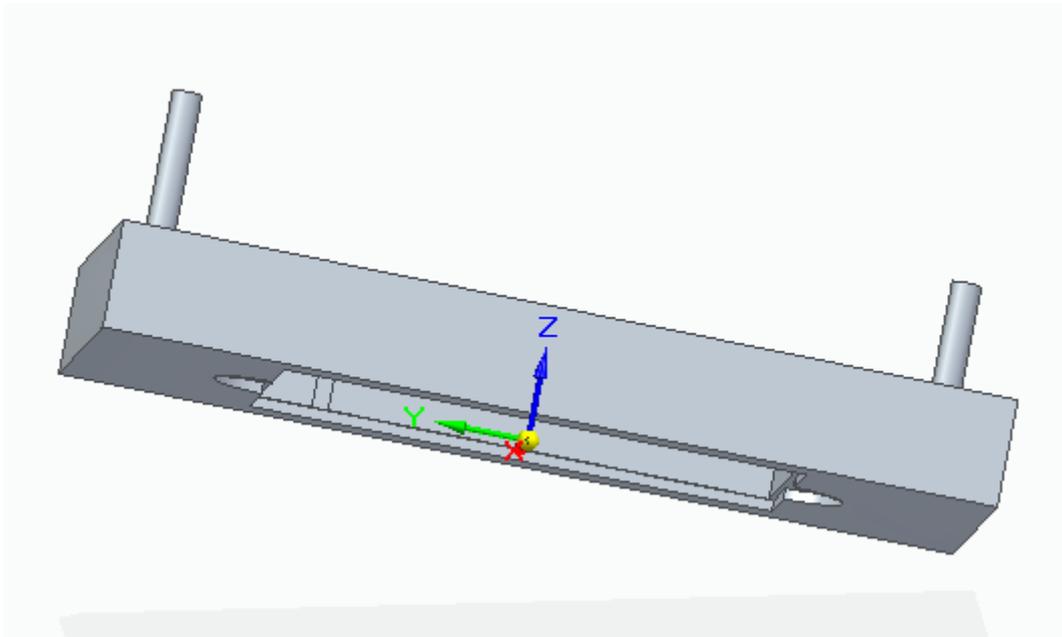


Figura 6- Base da caixa dos terminais

A segunda peça é a parte superior da caixa que serve para cobrir o circuito. Esta peça necessita de furos para se conseguir ter acesso aos botões. Nesta peça fez uma fenda perto do leitor de cartões, de modo a ter as dimensões precisas para inserir o cartão corretamente. Fez-se também um furo em cima da peça para conseguir colocar o fio que serve de antena do transmissor fora da caixa, de maneira a conseguir ter um maior alcance de transmissão de dados. A figura 7 representa a segunda peça.

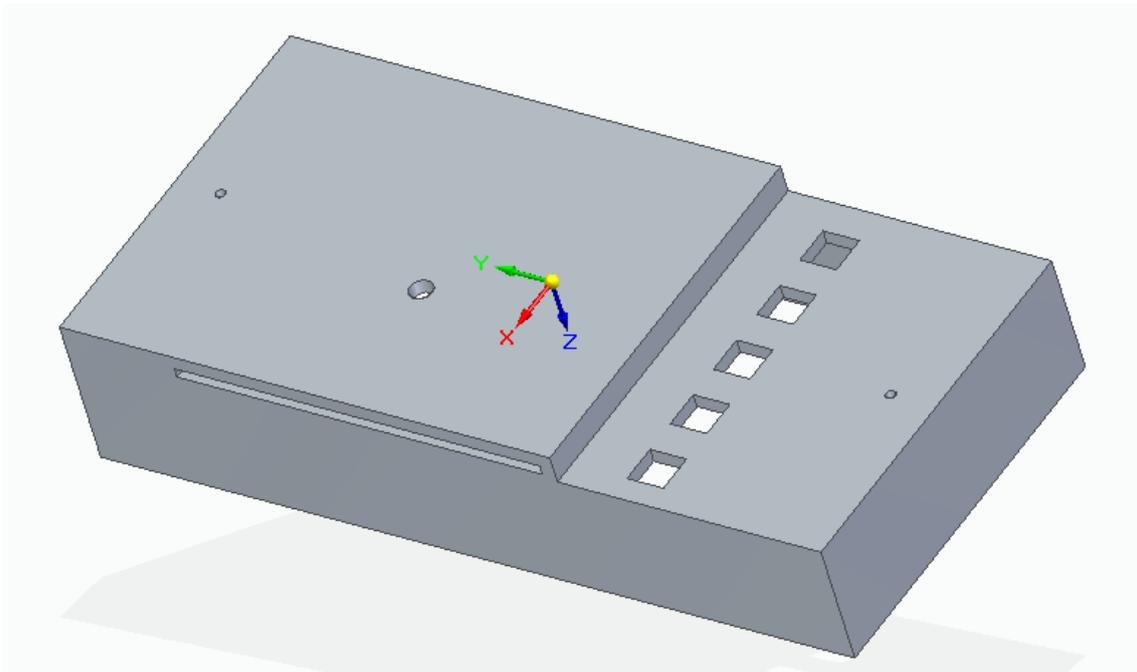


Figura 7- Parte superior da caixa dos terminais

Depois de o desenho estar concluído, guarda-se o desenho como ficheiro stl, e enviar esse ficheiro para o software da impressora 3D, e assim construir uma caixa com as dimensões reais para a colocação do circuito.

7. *Desenvolvimento* *do* *Software*

7.1. Software dos terminais

A programação do software deste projeto teve duas fases principais, a fase de programação dos terminais e a fase de programação da estação base. Começou-se por desenvolver a programação do software para os terminais. A primeira tarefa desta programação foi verificar quais as portas utilizadas no Arduino Uno. As portas usadas no Arduino Uno estão na tabela 2.

Tabela 2- Portas utilizadas no Arduino Uno

Porta	Descrição
RX0	Aquisição de dados enviados pelo pino TX do leitor de cartões
6	Emissão de dados para o pin RX do leitor de cartões
2	Colocar alínea A como resposta
3	Colocar alínea B como resposta
4	Colocar alínea C como resposta
5	Colocar alínea D como resposta
7	Colocar alínea E como resposta
8	Aquisição de dados do emissor RF

Depois de configurar as portas do Arduino, foi necessário fazer um fluxograma do software dos terminais, para facilitar o método de programação, e saber quais as tarefas a programar. A figura 8 representa o fluxograma do software dos terminais.

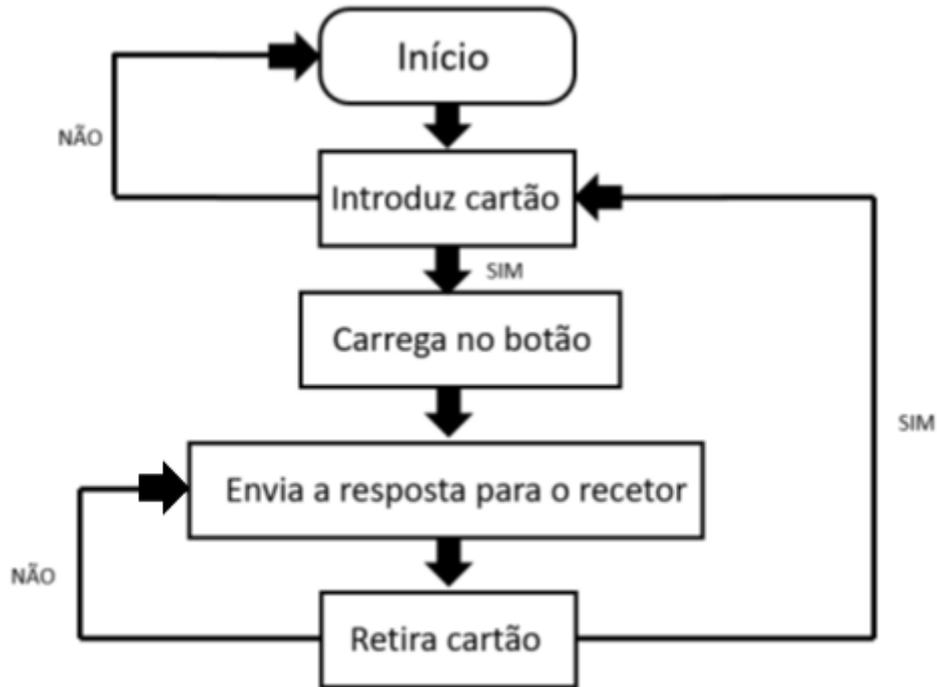


Figura 8- fluxograma do software dos terminais

Como se pode ver na figura 8 a primeira fase de programação foi o leitor de cartões, e como já foi referido no relatório o leitor de cartões utiliza a comunicação UART para enviar dados para o Arduino. Definiu-se a seguinte biblioteca na programação da UART:

```
#include <SoftwareSerial.h>
```

Para configurar as portas da Rx e TX mencionadas na tabela 2, utilizou-se o seguinte código:

```
SoftwareSerial rfid = SoftwareSerial(9, 6);
```

A porta RX serve para o leitor de cartões comunicar com o Arduino. Para decodificar a informação que o leitor de cartões envia para o Arduino é importante saber o significado do *Preamble*, *Len*, *Command*, *Status*, *Data* e *Checksum*, pois representam os bytes de um cartão [2].

- *Preamble* é o primeiro byte enviado, esse byte é sempre igual a 0xBD.
- *Len* representa o comprimento da mensagem enviada.
- *Command* é um byte que representa o código de comando. Se o byte for igual 0x01, significa que está a ser selecionado um cartão MIFARE.
- *Status* representa o estado de comando. Se o byte for igual 0x00, significa que a operação foi bem-sucedida.
- *Data* é um byte que representa o tipo de cartão que está a ser lido. O byte data do cartão da universidade de Aveiro é igual a 0x04, significa que é

Class Quis – PEA – Universidade de Aveiro

um cartão MIFARE 4K, com 4 bytes de números de series exclusivos do cartão MIFARE.

- *Checksum* é um XOR de todos os bytes anteriores.

Um ponto importante na programação do leitor de cartões é que este só envia informação se for solicitado pelo Arduino. O pedido do Arduino é feito através da porta Tx. A mensagem enviada pelo Arduino tem a seguinte codificação [2]:

- O primeiro byte enviado é sempre igual a 0xBA.
- O segundo byte representa o comprimento da mensagem enviada.
- O terceiro byte representa o número de comando, e este é igual a 0x01.
- O ultimo byte é um XOR de todos os bytes anteriores.

A segunda fase da programação é ativação dos botões. Começou-se por definir as portas que estão ligadas ao Arduino. Por fim lê-se o valor de entrada, e cria-se uma condição para verificar se o interruptor está aberto. Se a condição estiver correta ele imprime a letra correspondente ao botão. A programação para colocar alínea A como resposta é a seguinte:

```
val1 = digitalRead(inPin1); // ler o valor de entrada

if (val1 != HIGH) // verificar se a entrada é HIGH (interruptor livre)
{
    send("A"); // dados a enviar
    while (!digitalRead(inPin1));
    delay(50);
}
```

A ultima fase da programação é enviar dados utilizando o transmissor RF. Na programação utilizou-se a seguinte biblioteca:

```
#include <VirtualWire.h>
```

De seguida foi necessário definir a porta que vai ligar ao pino de dados do emissor, e também definir a velocidade de transmissão de dados. A programação para definir a porta e a velocidade é a seguinte:

```
vw_set_tx_pin(8);
vw_setup(2000);
```

Por fim programa-se uma função para enviar os caracteres da mensagem. A programação dessa função é a seguinte:

```
void send (char *message) {
    vw_send((uint8_t*)message, strlen(message));
    vw_wait_tx(); }
```

7.2. Software da estação base

A programação da estação base também começou pela verificação das portas utilizadas pelo Arduino. As portas usadas no Arduino Mega 2560 estão na tabela 3.

Tabela 3- Portas utilizadas pelo Arduino Mega 2560

Porta	Descrição
5	Aquisição de dados do recetor RF

O software da estação base apenas necessita de programar a aquisição de dados que o recetor RF recebe. Na programação de receção de dados é necessário utilizar a seguinte biblioteca:

```
#include <VirtualWire.h>
```

O início desta programação é definir a porta que vai estar ligada ao pino de dados do recetor e também definir a velocidade de receção de dados. A programação para definir a porta e a velocidade é a seguinte:

```
vw_set_rx_pin(5);
vw_setup(2000);
vw_rx_start();
```

De seguida é necessário criar bytes para receber os dados. Depois criar uma função para verificar os bytes recebidos, e por fim transformar esses bytes em caracteres iguais aos enviados pelo emissor. A programação de criação e transformação de bytes é a seguinte:

```
uint8_t message [VW_MAX_MESSAGE_LEN];
uint8_t msgLength = VW_MAX_MESSAGE_LEN;
if (vw_get_message(message, &msgLength)) {
    Serial.print("Recebido: ");
    for(int i=0; i < msgLength; i++) {
        Serial.write(message[i]); }
    Serial.println(); }
```

Depois de colocar a programação no Arduino, abre-se a interface humana que o software do Arduino tem, e consegue-se verificar as respostas dadas pelos terminais.

8. Teste e resultados obtidos

O teste existente neste projeto foi o envio de duas ondas quadradas a frequências diferentes através dos transmissores RF, e verificar a onda que o recetor RF recebe. A figura 9 mostra a onda quadrada de 1KHz enviado por um transmissor.

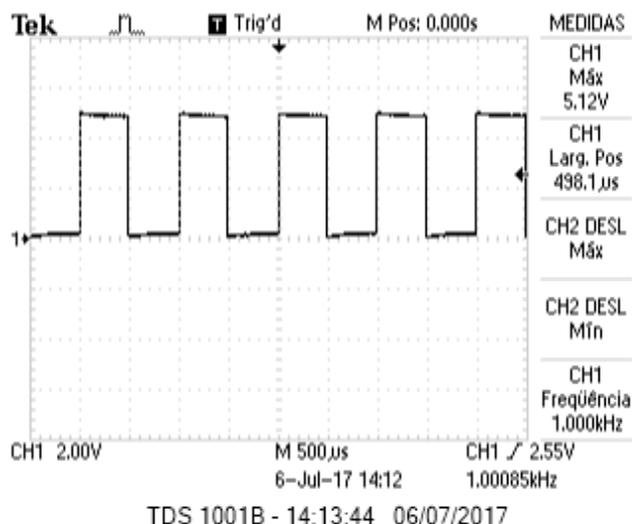


Figura 9- Onda quadrada de 1KHz

A figura 10 mostra a onda quadrada de 100 Hz enviado pelo o outro transmissor.

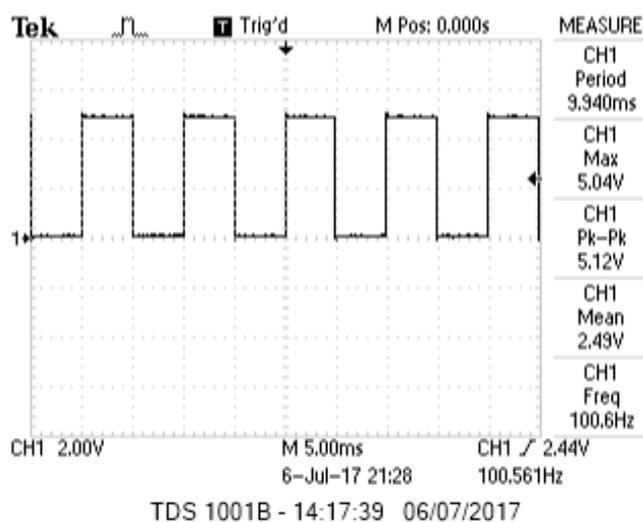


Figura 10- Onda quadrada de 100Hz

Neste teste o transmissor que envia a onda de 1KHz estava mais próximo do recetor. A figura 11 ilustra a aquisição de dados feita pelo recetor.

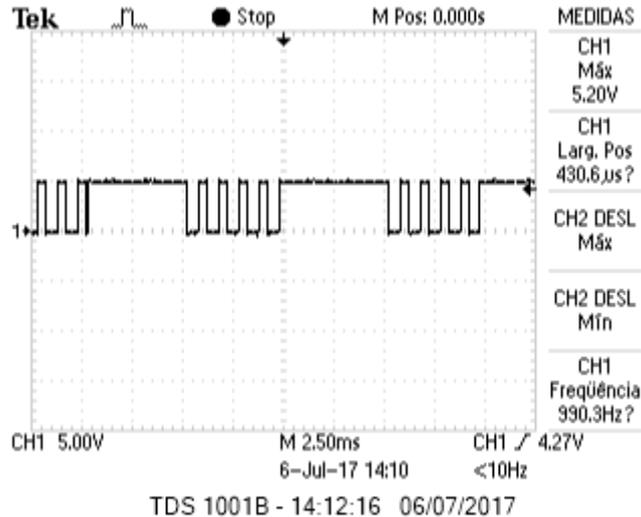


Figura 11- Aquisição de dados do recetor

Como se verifica na figura 11, o recetor RF faz uma soma lógica das ondas que os transmissores enviam. O transmissor com a frequência de 100 Hz quando está com o valor lógico 1, o recetor está a receber informação dele, quando este vai a zero, o recetor começa a receber dados do transmissor de 1000 Hz. Para que a comunicação entre o recetor e vários emissores esteja correta e não exista perda de informação de nenhum transmissor, é necessário que os transmissores tenham um tipo de diálogo *half duplex* ou *full duplex*. Um transmissor com os tipos de diálogo anteriores consegue receber informação de outros transmissores, e se algum estiver a enviar dados para o recetor, ele espera até a troca de dados esteja completada, para que ele possa comunicar com o recetor.

Ao alimentar o recetor RF verifica-se que ele recebe interferências exteriores ao projeto, essas interferências afetam a comunicação entre o recetor e o transmissor. Muitos dados enviados pelo transmissor não são decodificados pelo recetor, devido as interferências relatadas anteriormente.

9. Conclusão

No decorrer deste trabalho foram abordados conceitos de programação de Arduíno, de desenvolvimento de circuitos elétricos no software Eagle, de tipos de comunicação de dados e de modulação 3D no software Solid Edge.

No tipo de comunicação comprovou-se que os transmissores RF não são os mais indicados para a comunicação, por causa do seu tipo de diálogo.

Os recetores RF estudados apresentam muitas interferências nas saídas dos pinos de dados, estas interferências prejudicam a comunicação.

Os módulos de radiofrequência apresentam um bom alcance para ser utilizado nas salas de aulas da universidade de Aveiro.

Os objetivos pretendidos no primeiro caso de uso foram alcançados.

Como melhoramentos ao projeto é a criação de uma interface humana mais sofisticada, e também terminar todas as tarefas mencionadas no segundo e terceiro caso de uso. O layout dos terminais pode ser melhorado, utilizando um microcontrolador e uma bateria com dimensões mais pequenas, para apresentar um terminal com um design moderno e comercial. O recetor RF utilizado num projeto futuro tem que ser mais sofisticado e tem de ser imune a interferências exteriores ao projeto.

10. Referências bibliográficas

- [1] Arduino.(2017).Arduino Uno Rev3. [Em linha]. Disponível em <https://store.arduino.cc/usa/arduino-uno-rev3>. [consultado em 18-05-2017].
- [2] RFID MODULE. SL031 User Manual: Mifare Reader / Writer. 2015.
- [3] ELECTROFUN.(2017).Módulo RF Transmissor e Recetor 433MHz para Arduino e Raspberry PI. [Em linha]. Disponível em <https://www.electrofun.pt/modulo-rf-transmissor-receptor-433mhz>. [consultado em 20-05-2017].
- [4] Arduino.(2017).Arduino Mega. [Em linha]. Disponível em <https://www.arduino.cc/en/Main/arduinoBoardMega>. [consultado em 21-06-2017].
- [5] Comunicação sem fios. In: Wikipédia: a enciclopédia livre. Disponível em: https://pt.wikipedia.org/wiki/Comunicações_sem_fio Acesso em: 23 março 2017