

## U.C. Sistemas e Automação

### Trabalho Lab 1

### Projeto de sistema de controlo de cancelas baseado em Arduino

Armando Jorge Sousa – [asousa@fe.up.pt](mailto:asousa@fe.up.pt)

Luís Almeida – [lda@fe.up.pt](mailto:lda@fe.up.pt)

Paulo Costa – [paco@fe.up.pt](mailto:paco@fe.up.pt)

# 1. Apresentação do Trabalho Prático

Este trabalho introdutório da UC de Sistemas e Automação servirá para, utilizando uma linguagem próxima da linguagem C, projetar e implementar o sistema de controlo de duas garagens individuais independentes. Será utilizado o ambiente de desenvolvimento (IDE) sob Windows usando um pequeno computador – um microcontrolador (uC) numa placa Arduino para resolver o caderno de encargos.

Os uC são dispositivos poderosos e complexos que serão alvo de estudo aprofundado mais tarde no curso. Este trabalho pretende apenas abordar conceitos fundamentais de sistemas e automação.

Este microcontrolador deve ser programado em linguagem similar ao C.

O PC será utilizado para editar o código, na IDE, que depois será transferido e executado no uC.

No início da aula será fornecida uma placa com o uC Arduino, já montado ou tal como na figura 1a.

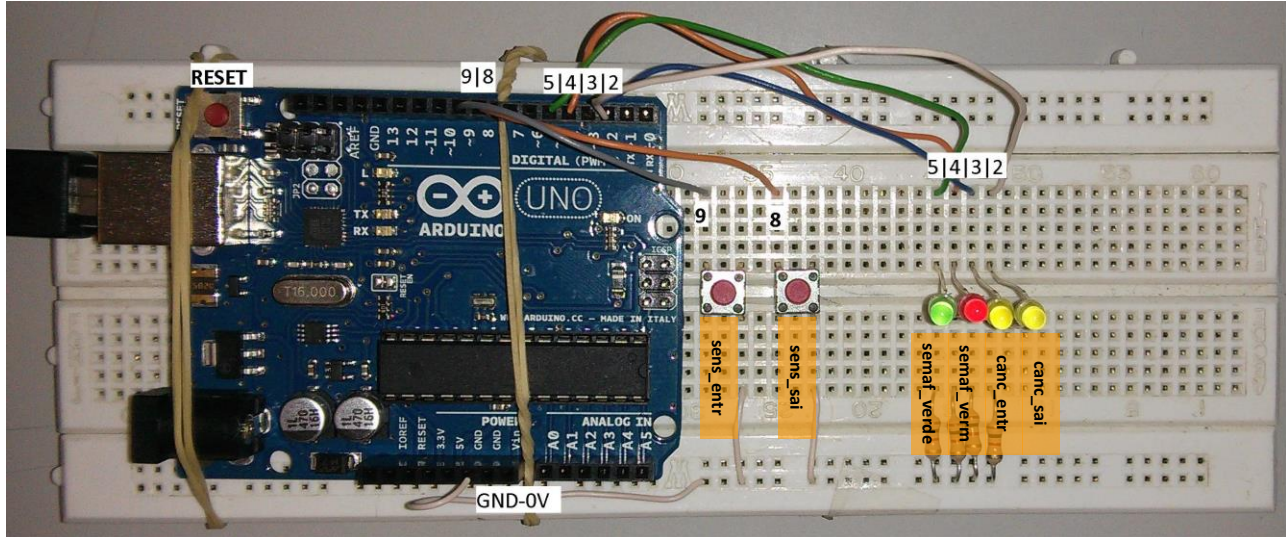


Figura 1a – Placa de montagem com Arduino

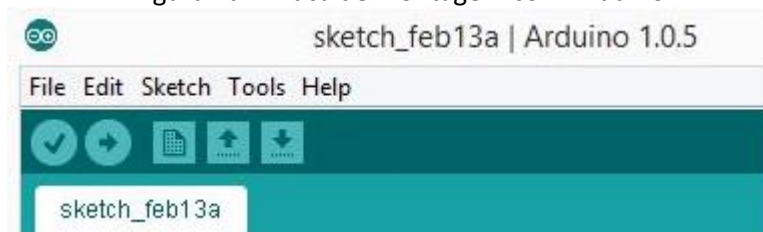


Figura 1b – IDE Arduino

O uC pode ler entradas e comandar saídas, isto é, ler tensão nos pinos de entrada e impor dentro de determinadas limitações tensões nos pinos de saída.

Leia todo o guião antes de iniciar o seu trabalho e no final da aula submeta os elementos produzidos.

## Objetivos do trabalho TP1:

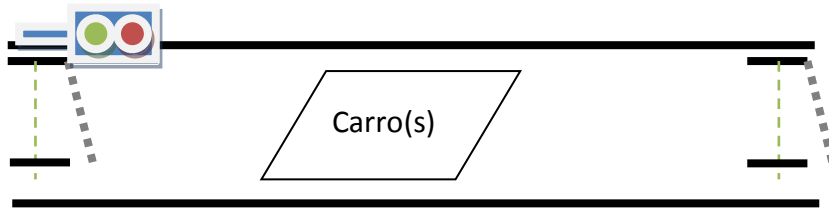
- Introduzir e mostrar (i) controlador, (ii) entradas e (iii) saídas em ambiente “real”
- Programar uC da placa Arduino em linguagem similar à linguagem C
- Tomar contacto com dificuldades de problemas multitarefa e temporizações

# 2. Caderno de encargos

Considere o problema simplificado de um sistema que controla o acesso a um (ou mais) lugar(es) de estacionamento privilegiado(s) mas partilhado por diversas pessoas protegido por cancelas.

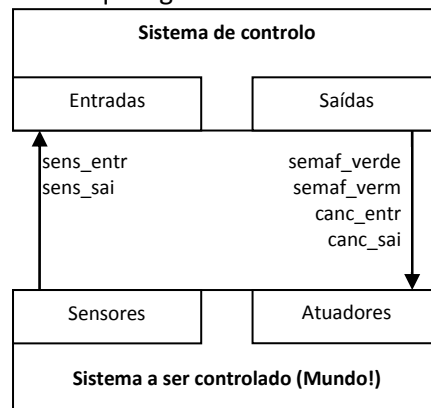
O acesso é conseguido através da passagem numa porta automática de entrada e a saída está também protegida por uma porta automática. Há um sensor de entrada dos carros que indica que um carro pretende entrar e o mesmo para outro sensor de saída, indicando que um carro pretende sair.

Adicionalmente há um semáforo apenas com uma luz verde e outra vermelha. A luz verde indica estacionamento livre e a vermelha estacionamento ocupado.



### 3. Conceitos e problema sob análise

Qualquer sistema de controlo/comando visa comandar um determinado sistema real (sistema a ser controlado). O sistema de comando recebe informação através de entradas onde ligam sensores e altera o sistema a ser controlado através de saídas que ligam a atuadores.



Para resolver o problema proposto temos:

**Atenção:** utiliza-se lógica negada em todos as entradas




- Sensor de Carro à Entrada => Pino\_9 => leitura do pino "sens\_entr"
  - É "0" quando o carro está presente (um carro quer entrar) = botão premido
  - É "1" quando não está nenhum carro presente = botão no estado de repouso
- Sensor de Carro à Saída => Pino\_8 => leitura do pino "sens\_sai"
  - É "0" quando o carro está presente (um carro quer sair)
  - É "1" quando não está nenhum carro presente = botão no estado de repouso
- Atuador da Cancela de Entrada => Pino\_3 => escrita do pino "canc\_entr"
  - Quando este atuador é posto a "1"="HIGH" = o LED liga = a cancela levanta e permite entrada carros
  - Quando é posto a "0"="Low" o LED desliga => a cancela desce (não permite entrada de carros)
- Um Atuador da Cancela de Saída => Pino\_2 => escrita do pino "canc\_sai"
  - Quando este atuador é posto a "1"="HIGH" = o LED liga = a cancela levanta e permite entrada carros
  - Quando está a "0", a cancela desce e não permite a saída de carros

Sequência (genérica):

- O carro apresenta-se à entrada e se parque livre, a cancela levanta
- Depois de a cancela levantar, esperar-se até o carro "desaparecer" do sensor de entrada para descer
- O carro apresenta-se à saída e a cancela levanta
- Depois da cancela levantar, deve esperar até ao carro "desaparecer" do sensor de saída para descer

Adicionalmente, estude o funcionamento do semáforo que tem luz vermelha e luz verde.

## 4. Instalação e teste inicial

- 1.1 Vá ao Moodle da UC e faça *download* do \*.zip do TP1
- 1.2 Crie uma diretoria com o seu nome para o seu trabalho e desempacote aí o ficheiro retirado do Moodle (mantendo as subdiretorias)
- 1.3 Abra o IDE do Arduino e faça open do ficheiro TP1
- 1.4 No IDE clique no 'certo'  ou prima CTRL+R em cima a esquerda (compila o programa) – confirme que o programa não tem erros
- 1.5 De seguida clique na 'seta'  ou prima CTRL+U (descarregar o programa para o uC).
- 1.6 Verifique que todos os Leds acendem durante 1 segundo
- 1.7 No IDE clique na 'lupa'  ou prima CTRL+Shift+M para abrir o “monitor” (consola) que lhe permite ver o que o Arduino escreveu; configure o Baud Rate para 115200 (canto inferior direito)
- 1.8 Clique em qualquer dos botões verifique que todos os LEDs acendem

## 5. Sequência de passos para o trabalho

Para cada um dos seguintes pontos, implemente e teste. Não espere pelo professor, prossiga o seu trabalho, aproveite bem o tempo de aula.

**Atenção:** Guarde separadamente as respostas de cada alínea.

Passos para a solução completa:

- a. Inicialmente considere apenas a entrada de um único carro no estacionamento; a cancela de entrada abre sempre que o carro for detetado na entrada; depois do carro entrar, o semáforo vermelho liga (depois do carro ter deixado de ser detetado); o verde está ligado inicialmente; teste e confirme a solução;  
Considere depois a entrada e posterior saída do carro no estacionamento; o semáforo verde liga depois do carro ter deixado de ser detetado na saída; admita inicialmente que não entram e saem carros ao mesmo tempo; teste diversas entradas e saídas em sequência; garanta que quando o lugar está ocupado, a cancela não abre;  
Admita que é possível entrar um carro depois de um outro iniciar a saída;  
No sistema operativo, copie este ficheiro TP1.ino para  
**Lab1A\_GXX\_PrimNomeUltNomeAAA+PrimNomeUltNomeBBB.ino**
- b. Admita agora que o estacionamento tem capacidade para 5 lugares;  
Desenhe uma nova solução para este problema;  
No sistema operativo, copie este ficheiro TP1.ino para  
**Lab1B\_GXX\_PrimNomeUltNomeAAA+PrimNomeUltNomeBBB.ino**
- c. Altere a solução anterior para que ambas as cancelas tenham um tempo mínimo de abertura de 5 segundos;  
**Atenção:** Tire proveito do facto que o tempo de cada ciclo do Arduino é de cerca de 10 ms  
No sistema operativo, copie este ficheiro TP1.ino para  
**Lab1C\_GXX\_PrimNomeUltNomeAAA+PrimNomeUltNomeBBB.ino**

## 6. Final de aula – submissão e questionário

No final da aula submeta os 3 ficheiros \*.ino no Moodle.

Não saia da sala sem responder ao questionário do Moodle (questione o professor acerca da password).

## 7. Código base e exemplos

Neste trabalho há um conjunto de pinos numerados que utilizados para trocar informação (enviar/receber) entre painel (botões e leds) e controlador (uC).

Todos os pinos estão configurados.

O controlador será feito pelos estudantes alterando o código abaixo mostrado.

Note que existe um ciclo infinito e instruções de leitura e escrita das entradas e saídas.

```
// Definições Globais

// Definir pinos atribuídos a cada variável
const int sens_entr    = 9; // Entrada
const int sens_sai     = 8; // Entrada
const int semaf_verde  = 5; // Saída
const int semaf_vermelho= 4; // Saída
const int canc_entr    = 3; // Saída
const int canc_sai     = 2; // Saída

// Corre aquando do reset (uma única vez) - não alterar!
void setup()
{
    Serial.begin(115200);    // Inicializar e definir velocidade da porta série

    // Configurar pinos como Entradas/Saídas
    pinMode(semaf_verde,    OUTPUT);
    pinMode(semaf_vermelho, OUTPUT);
    pinMode(canc_entr,      OUTPUT);
    pinMode(canc_sai,       OUTPUT);
    pinMode(sens_entr,      INPUT_PULLUP);
    pinMode(sens_sai,       INPUT_PULLUP);

    // Teste Inicial - ligar tudo e depois desligar tudo
    digitalWrite(semaf_verde,    HIGH);
    digitalWrite(semaf_vermelho, HIGH);
    digitalWrite(canc_entr,      HIGH);
    digitalWrite(canc_sai,       HIGH);
    delay(1000);                  //Espera 1000ms (1s)
    digitalWrite(semaf_verde,    LOW);
    digitalWrite(semaf_vermelho, LOW);
    digitalWrite(canc_entr,      LOW);
    digitalWrite(canc_sai,       LOW);
}

// Sequência de código que corre em ciclo
void loop()
{
    // Código de teste, a alterar, apenas alterar código dentro da função loop()
    // Se qualquer dos botões carregado => acende todos LEDs, senão apaga-os todos
    if ((digitalRead(sens_entr)==LOW) || (digitalRead(sens_sai)==LOW))
    {
        digitalWrite(semaf_verde,    HIGH);
        digitalWrite(semaf_vermelho, HIGH);
        digitalWrite(canc_entr,      HIGH);
        digitalWrite(canc_sai,       HIGH);
    } else {
        digitalWrite(semaf_verde,    LOW);
        digitalWrite(semaf_vermelho, LOW);
        digitalWrite(canc_entr,      LOW);
        digitalWrite(canc_sai,       LOW);
    }
    // Enviar informação via porta série
    Serial.print (digitalRead(sens_entr));
    Serial.println(digitalRead(sens_sai));
    delay(10);                      // 10 ms
}
```

- Fim do Guião TP1 -