

U.C. Sistemas e Automação

Lab04

Sistema de Controlo de Miniatura de Cancela sob Arduino

Armando Jorge Sousa – asousa@fe.up.pt

Paulo Costa – paco@fe.up.pt

1. Apresentação do Trabalho Prático

Este trabalho pretende ser uma introdução ao controlo e interação com sistemas reais (mesmo que miniaturizados) e suas especificidades.

Será ainda introduzida uma pequena funcionalidade de supervisão.

Preparação

Antes da aula, individualmente ou em grupo, é necessário:

- Estudar todo o guião;
- Saber implementar um sistema de máquinas de estado;
- Levar os DTEs a implementar (em papel) para o sistema base;
- Levar os DTEs a implementar (em papel) para o sistema com a funcionalidade de supervisão;
- Preparar a implementação em Arduino_C (por exemplo para o sistema base)
- Preparar as ligações elétricas a fazer (na aula) e configurações de software associadas.

Hardware e kit

O kit laboratorial inclui o do Lab01 (Arduino Uno, botões e LEDs) a que acresce o kit com um motor e uma miniatura de uma cancela.

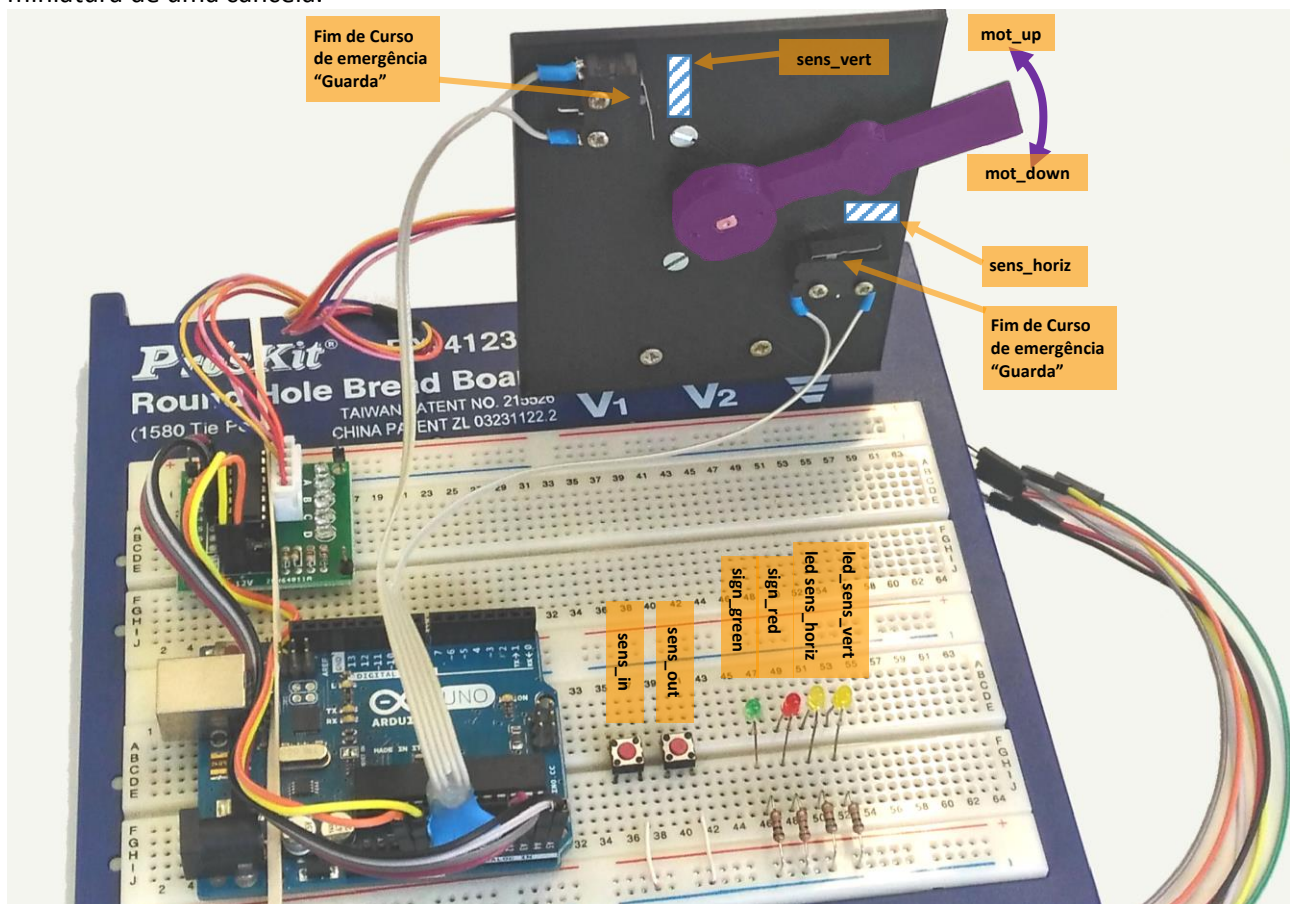


Figura 1 – Placa de montagem com Arduino

Objetivos

- Implementação em Arduino_C de um controlador modelizado como um Sistema de Máquinas de Estados (com temporizadas e com contadores)
- Tomar contacto com dificuldades associadas a sistemas reais
- Praticar ligações elétricas e configuração pino a pino das entradas / saídas sob Arduino
- Tomar consciência da interligação entre ligações elétricas e software
- Introdução a funcionalidades de supervisão

2. Caderno de encargos

Considere o problema simplificado de um sistema que controla o acesso a um parque de estacionamento. O acesso é conseguido através da passagem numa cancela de entrada mas a saída é “livre”, sem cancela. Há um sensor na entrada que indica que um carro pretende entrar.

Há outro sensor para a saída, indicando que um carro está a sair neste momento.

Adicionalmente há um “semáforo” com luzes verde e vermelha.

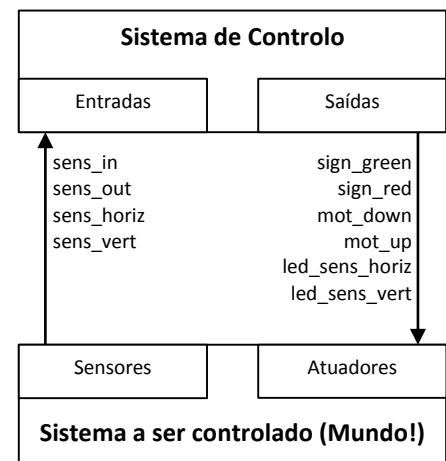
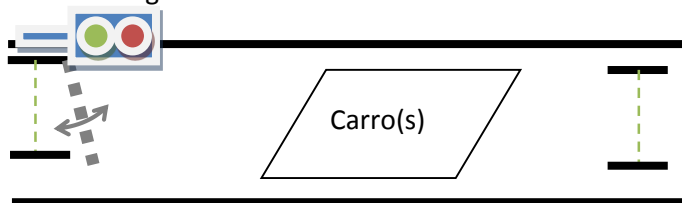
Quando um carro se apresenta para entrar (**sens_in** ativo) o sistema de controlo deve ligar o motor de subida da cancela atuando **mot_up** até chegar à posição de aberta (**sens_vert** ativo). Depois do carro passar, quando **sens_in** ficar inativo, **mot_down** deve ser ligado para atuar o motor de descida da cancela até **sens_horiz** ficar ativo.

O sensor de saída **sens_out** serve para detetar a saída de carros.

Admitindo que o parque não está cheio, o semáforo verde (**sign_green**) deve estar ativo. Nessa situação podem entrar e sair carros em simultâneo, a qualquer ritmo.

Quando o parque ficar cheio, o semáforo deve ficar vermelho (**sign_red**) e o controlador deve impedir a entrada de novos carros. Os carros podem sair sempre.

Suponha ainda uma funcionalidade de supervisão: se um carro “avariar” na saída (**sens_out** durante mais de um tempo configurável) e se não for o único carro no parque, fazer piscar o semáforo verde e vermelho alternadamente a 1 Hz e impedir a entrada de novos carros. O desaparecimento do carro “avariado” faz com que o sistema regresse ao funcionamento normal.



3. Acesso ao Hardware

Entradas: **sens_in**, **sens_out**, **sens_horiz** e **sens_vert**.

Saídas: **sign_red**, **sign_green**, **mot_up** e **mot_down**.

Atenção:

- Se o sistema de controlo funcionar mal, os fins de curso de emergência “Guardas” serão ativados e só se poderá sair dessa situação por intermédio de um “reset” ao sistema de controlo que recentra a cancela. Esta funcionalidade existe sem intervenção do estudante para proteger embates da cancela. O programa feito pelos estudantes não deve tentar ler os fins de curso.
- Não mover a cancela manualmente nunca!**
- O kit laboratorial apresentado utiliza sensores de fim de curso mecânicos para as “Guardas” e sensores VIRTUAIS para a vertical e a horizontal, isto é, **sens_horiz** e **sens_vert** são sensores que não existem na realidade apesar de as variáveis serem devidamente atualizadas
- Por motivos pragmáticos de operacionalidade, o tipo de motor utilizado é um motor do tipo *stepper*, de passo a passo (com redução mecânica) mas uma cancela real utiliza sempre outro tipo de motor, possivelmente controlado por um variador de velocidade (que não faz sentido estudar nesta altura nesta UC). O controlo do motor é feito por intermédio de **mot_up** e **mot_down**.

4. Sequência de passos para o trabalho

1. Comece por analisar o equipamento que lhe é fornecido e as ligações feitas e em falta; os pinos exatos a utilizar serão fornecidos durante a aula (devem-lhe ser fornecido 7 fios Macho-Macho)
2. Implemente as ligações elétricas em falta de acordo com as indicações fornecidas na aula; as ligações em falta são relativas às entradas e saídas (sens_*, sign_*, mot_*, led_*); para facilidade, utilize fios com cores adequadas; refira-se à figura 1 para as funcionalidades pretendidas
3. Descarregue o programa base lab04.ino para uma diretoria sua do drive F:
4. Altere o software de acordo com as ligações fornecidas durante aula, diferentes grupo-a-grupo. Tenha atenção à função setup() e à configuração que cada pino deve ter como saída ou entrada (com *pullup*). Teste todo o sistema, entradas saídas e fins de curso.
5. Implemente a funcionalidade que ativa o **led_sens_horiz** quando **sens_horiz** ficar ativo e o mesmo para **led_sens_vert** e **sens_vert**
6. Implemente as funcionalidades “base” do controlador do parque (sem a supervisão); coloque a capacidade do parque de estacionamento numa constante do seu programa; **teste com 4 lugares**.
7. Acrescente depois a funcionalidade de supervisão; coloque o tempo para considerar carro avariado numa constante do seu programa; **teste com 3 segundos**.

5. Final de aula – submissão e questionário

1. No sistema operativo, copie este ficheiro **lab04.ino** para
LAB04_TXX_GYY_PrimNomeUltNomeAAA_PrimNomeUltNomeBBB.ino
Submeta no moodle até 24 horas depois do final da aula
2. Submeta também a digitalização do sistema de máquina de estados efetivamente implementado
3. Copie para si os ficheiros relevantes
4. Apague todo o conteúdo do drive F:
5. Com todo o cuidado, retire as ligações feitas na aula e deposite os 7 fios na caixa da sua bancada
Entregue / sinalize equipamentos avariados / com mau funcionamento

6. Anexo – trechos de código

```
// PINs =====> Fill-in your own pins
const int pin_----- = -----;
...
void setup()
{
    ...
    // Define Input and Output Pins
    pinMode( pin_-----, INPUT_PULLUP);
    pinMode( pin_-----, OUTPUT);
    ...
}
...
void loop_10ms(void)
{
    ...
    if (sens_in) {
        motor_up = 1;
        motor_down = 0;
    } else if (sens_out) {
        motor_up = 0;
        motor_down = 1;
    } else {
        motor_up = 0;
        motor_down = 0;
    }
    ...
}
```