

## Trabalho prático 2

### Tecnologia da Informática 2023/2024

### *Ardugotchi*

O objetivo deste trabalho é o de implementar um *Tamagotchi*. Um *Tamagotchi* é uma mascote virtual (circa 1996) que necessita de cuidados e atenção, podendo morrer se não for devidamente cuidada. O *Tamagotchi* a implementar deverá ter três botões (comer, dormir e brincar) e três LEDs (1 LED para cada necessidade: fome, sono e brincar). Um LDR será usado para analisar a luz ambiente, ajudando a determinar se é hora de pôr a mascote a dormir.

**TRABALHO A REALIZAR EM GRUPOS DE 2 ALUNOS/AS, RESPEITANDO A LISTA  
PUBLICADA PELOS DOCENTES DAS TURMAS PL.  
REGRAS PARA CASOS EXCECIONAIS DOCUMENTADAS ABAIXO.**

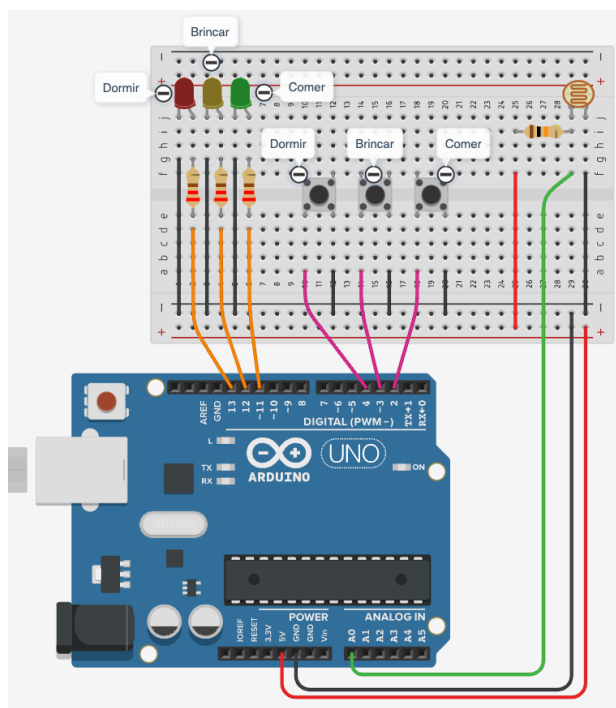
### Lógica de operação

Normalmente, os Tamagotchis costumam utilizar intervalos de base horária, mas no nosso caso isso não é viável – por esse motivo iremos trabalhar com uma base de minutos. O nosso Tamagotchi irá comportar-se de acordo com um conjunto de regras, a saber:

- Quando o Arduino é ligado, o Tamagotchi deve considerar-se acabado de “nascer”, começando-se a contar o tempo.
- A cada 4 minutos (mais ou menos 60s, sendo esta margem aleatória), a mascote precisa de comer, o que deverá sinalizar acendendo o LED verde. Para satisfazer esta necessidade, o utilizador deverá carregar no botão correspondente.
- A cada 3 minutos (mais ou menos 60s, sendo esta margem aleatória), o Tamagotchi precisa de atenção, o que deverá sinalizar acendendo o LED amarelo. Para satisfazer esta necessidade, o utilizador deverá carregar no botão correspondente.
- Ao fim de 10 minutos acordado (mais ou menos 60s, sendo esta margem aleatória), ou se a média das últimas medições da luz ambiente for acima de 2/3 da gama do ADC, (o que acontecer primeiro) o Tamagotchi deverá pedir para o pôr a dormir, acendendo o LED vermelho. Uma vez premido o botão de dormir, o Tamagotchi irá dormir durante 5 minutos, não devendo gerar quaisquer eventos durante esse período e recomeçando a rotina normal após acordar (em ambos os casos - dormir e acordar – deverá imprimir uma mensagem informativa sobre o acontecimento).
- O estado de saúde da mascote será determinado por um contador de penalizações. As penalizações serão adicionadas conforme o tempo que o utilizador demorar a satisfazer as exigências do Tamagotchi, sendo estas de 5 pontos por cada minuto completo – contudo, se o utilizador reagir durante os primeiros 15s após a notificação, deverá haver uma bonificação (subtração) de 5 pontos neste contador (se este for igual a zero continuará a zero).
- A cada minuto decorrido o valor do contador de penalizações deverá ser impresso.

O Tamagotchi irá morrer se o total de penalizações ultrapassar os 25 pontos, parando a execução do programa - deverá ser impressa uma mensagem a informar quando tal suceder. Para a medição da média dos valores da intensidade luminosa, deverão guardar uma amostra num vetor, a cada minuto, fazendo a média de 6 amostras.

## Regras de ligação (conforme mostra a figura)



### Conexões do circuito

#### 3 LED:

- Pino 11: Comer
- Pino 12: Brincar
- Pino 13: Dormir

#### 3 Botões de pressão:

- Pino 2: Comer
- Pino 3: Brincar
- Pino 4: Dormir

LDR ligado no pino A0, em circuito de divisor de tensão com resistência de 10KOhm ligada aos +5V.

## Notas e recomendações

Tenham atenção à possibilidade de *overflow* de qualquer contador de tempo – cuidado com a manipulação de constantes numéricas (ver slides da aula teórica da semana de 21/11)! Não utilizar `delays()`, sob qualquer pretexto.

## Crítérios de avaliação

O uso de *debouncing* é obrigatório. O recurso a funções será valorizado, assim como a legibilidade e qualidade geral do código que for submetido. Submissões elaboradas sem recurso a estruturas de controlo apropriadas (como é o exemplo dos ciclos) serão penalizadas. O recurso a instruções ou conteúdos não lecionados na cadeira dará lugar a penalização.

A fraude académica (contemplando situações como o plágio ou falsa autoria, entre outras) será tratada de acordo com o Regulamento Disciplinar dos Estudantes da UC.

**Este trabalho será sujeito a defesa sem consulta sem acesso ao código fonte, na última aula PL do semestre, pelo que não deverão faltar.**

## Regras de submissão

O trabalho deverá ser realizado em grupos de 2 alunos, de acordo com as listas criadas nas aulas PL. **Nos casos excecionais dos alunos sem grupo atribuído**, dispensa-se a implementação da funcionalidade do LDR.

As submissões **serão feitas individualmente**, no Infoestudante, com a identificação dos nomes e números de estudante do grupo a que pertencem (ou indicação do/a próprio/a quando se tratar de aluno/a sem grupo).

<b>Data limite para submissão dos trabalhos: 10 de dezembro, pelas 23:59.</b>
-------------------------------------------------------------------------------

**A submissão deverá ser constituída por um ficheiro comprimido ZIP, contendo:**

- O código elaborado para o projeto, na sua totalidade (ficheiros em formato *.ino*).
- Um pequeno relatório (3 páginas, máximo) documentando como foram implementadas as funcionalidades e quaisquer observações/lacunas pertinentes.
- Um pequeno vídeo do circuito montado, em operação durante uma ronda completa de jogo (tenham atenção aos limites de upload). **Para que o vídeo não produza um ficheiro de dimensões exageradas, sugerem-se as seguintes alternativas:**
  - **acelerar o vídeo;**
  - **reduzir a resolução e qualidade (dentro de limites adequados à visualização);**
  - **ajustar as variáveis de temporização no programa usando uma mesma escala para efeitos da gravação (por exemplo, dividindo todos os intervalos de referência por 3 - no caso do intervalo para dormir passaríamos de 10 para 3.3 minutos +- 60s (acordando ao fim de 5/3 minutos, ou seja 1.6 minutos), de 3 para 1 minuto (+- 60s) no intervalo de brincar e de 4 para 1.3 no intervalo para comer (+- 60s).**

**Solicita-se ainda que indiquem no relatório quais as opções que adotaram.**

**Alunos remotos sem kit Arduino deverão efetuar uma submissão normal (com exceção do vídeo),** usando o Tinkercad para implementar o projeto. Nestes casos aplica-se o seguinte:

- Deverão usar a função “Enviar para->Convidar pessoas” para obter um link sem data de expiração para o projeto, de modo que os docentes possam conferir o que foi feito (NÃO COLOCAR O PROJETO COM VISIBILIDADE PÚBLICA!!!!).
- O link do Tinkercad acima referido deverá ser incluído no relatório a submeter.
- Adicionalmente, o código desenvolvido deverá ser submetido utilizando para o efeito um ficheiro de texto no formato *.txt* (em substituição do ficheiro *.ino*).