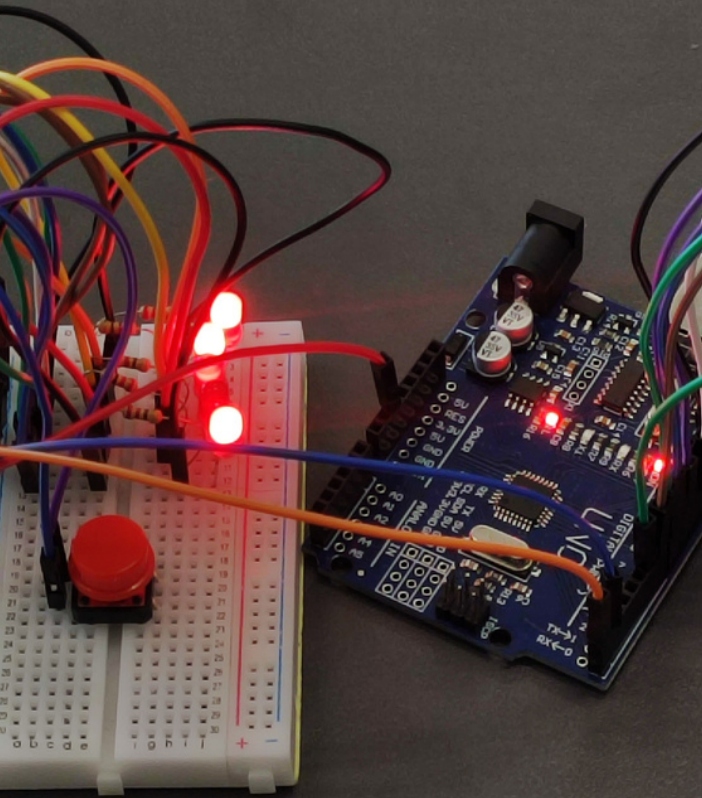


# RELATÓRIO

TRABALHO PRÁTICO 1



REALIZADO POR:

Pedro Seco  
nº2021273328

Rodrigo Oliveira  
nº2023231040

1 2 9 0



UNIVERSIDADE D  
COIMBRA

LICENCIATURA EM  
ENGENHARIA  
INFORMÁTICA

“GUESS THE  
NUMBER”

TECNOLOGIA DA  
INFORMÁTICA  
PL4

NOVEMBRO DE 2023

## Introdução

No âmbito da disciplina de Tecnologia da Informática, foi-nos desafiado a elaborar o primeiro trabalho prático. Neste trabalho, o objetivo consistiu na elaboração de um jogo utilizando dois arduinos: o arduino mestre que controla o jogo e que gera um número aleatório, e o arduino secundário que irá ser usado por um outro jogador para tentar adivinhar o número. Além disso, de forma a que os arduinos possam comunicar entre si pela porta série, utilizamos as funções *Serial.write()* e *Serial.read()*.

Assim, este relatório serve para explicar de uma maneira mais clara e pormenorizada o código elaborado e referir os maiores obstáculos ultrapassados na elaboração do mesmo.

## Desenvolvimento

Para o funcionamento deste trabalho, foram utilizados, como já mencionado anteriormente, dois arduinos. No circuito do arduino mestre utilizamos a *breadboard* para dispor os 5 LEDs, um botão de início de jogo, resistências e fios de ligação. No arduino secundário utilizamos a *breadboard* para dispor 5 LEDs, dois botões (um de incremento e outro de tentativa), resistências e fios de ligação.

No que toca à elaboração do código, iremos começar por explicar o arduino mestre.

Começámos por declarar as variáveis que vão ser usadas. 'buttonPin' define o pino onde o botão está conectado, 'buttonState' guarda o estado atual do botão, 'NumeroRand' guarda o número aleatório que irá ser gerado. Para lidar com o Debounce, criámos a variável 'lastDebounceTime' para guardar o tempo da última mudança de estado do botão, 'debounceDelay' que é um valor predefinido (50 milissegundos) para evitar leituras falsas no botão. Por fim, criamos a variável 'lastButtonState' que guarda o último estado do botão.

Passando às funções criadas. De forma a não repetir código, criamos a função "acender\_apagarLEDs(int valor)" que controla o estado dos LEDs e que recebe como parâmetro um valor (HIGH para acender os LEDs ou LOW para apagar). Além disso, foi criada a função "gerirLEDs(int valor, int led)" que recebe dois parâmetros, "valor" e "led" e que acende os LEDs de acordo com o número aleatório que será gerado. Nesta função, foi criado um ciclo "for" que vai iterar 5 vezes de forma a poder controlar os 5 LEDs. Dentro do ciclo, para definirmos o estado de cada LED utilizamos o DigitalWrite(). Ora, no DigitalWrite, "led + i" calcula o número do pino do LED atual no loop e "(valor >> i) & 1" faz uma operação de shift à direita para deslocar bits do "valor" para a direita "i" vezes. É através de uma máscara "& 1" que o último bit é isolado. Assim, com esta função é possível controlar os LEDs de acordo com o número aleatório.

No "void setup()" inicializamos a comunicação com o "Serial.begin", configuramos os pinos dos LEDs como saída através de um ciclo for, de modo a reduzir linhas de código, configuramos o pino do botão como entrada pull-up. Com a ajuda da função que criamos, "acender\_apagarLEDs()", fizemos com que os LEDs ao início estivessem todos ligados, passando à função o valor "HIGH". Por fim, utilizamos o "randomSeed(analogRead(0))" para podermos gerar o número aleatório.

Por fim, no "void loop()", utilizamos o Debounce e quando o botão implementado na *breadboard* fosse pressionado, iria ser gerado um número aleatório e guardado na variável "NumeroRand", e através da nossa função que criamos anteriormente, "gerirLEDs()", os LEDs iriam acender de acordo com esse número aleatório que foi gerado. Além disso, quando pressionado irá enviar o carácter "S" ao arduino secundário. Para terminar, depois do arduino secundário tentar adivinhar o número, através do "Serial.available > 0" o arduino mestre vai verificar se há alguma coisa para verificar na porta. Se houver, vai ser guardado na variável "adivinha" e vai comparar o valor lido nesta variável com o número aleatório que foi gerado ("NumeroRand"). Se forem iguais, vai enviar através do

`Serial.write()` um “Y” para o arduino secundário e os LEDs do mestre vão ser ligados. Se não forem, vai apenas mandar um “N” para o arduino secundário.

Agora, vamos passar à explicação do código do arduino secundário. Começamos por declarar as variáveis `botao_tentativa` e `botao_incremento` que representam os pinos dos botões, a variável `led` para definir os estados dos 5 LEDs com a ajuda de algumas funções, `contagem` que representa o número enviado para o arduino mestre, `estado_iniciar` que indica em que fase o jogo se encontra, `buttonState_1` e `buttonState_2` que guardam o estado atual dos botões e, como temos 2 botões para o Debounce iremos precisar das variáveis `lastIncrementoState` e `lastTentativaState` que guardam o estado anterior e das variáveis `lastDebounceTime` e `debounceDelay` pelo mesmo motivo que já referimos acima no mestre.

No `void setup()` fazemos um procedimento igual ao `void setup()` do arduino mestre, com exceção de não utilizarmos o `randomSeed`, pois, neste arduino, não queremos gerar um número aleatório e sim adivinhar esse número aleatório.

No que toca a funções, à semelhança do arduino mestre criamos uma função chamada `acenderLEDs()` que acende todos os LEDs conectados aos pinos de 8 a 12, com um ciclo `for` para iterar pelos pinos de 8 a 12, com o valor `HIGH`. O mesmo foi feito para a função `apagarLEDs()` que desliga os pinos de 8 a 12, com o valor `LOW`. A função `piscarLEDs()` é uma junção das duas funções que acabamos de referir, que estão dentro de um ciclo `While` que vai repetir 3 vezes a instrução de ligar e desligar os LEDs. A função `gerirLEDs()` tem o mesmo objetivo e é exatamente igual à que usamos no arduino mestre que tem como função controlar o estado dos LEDs com base na variável `contagem`, que irá ser incrementada (explicação no `void loop()`). Por fim, a função `leitura()` foi criada de modo a reduzir bastante linhas de código, o seu principal objetivo é ler o estado da porta (`Serial.available()`) e irá mudar o estado com base no caractere que lê: se for um ‘S’, define o `estado_iniciar` como ‘S’, põe a variável `contagem` para 0 e pisca os LEDs ao chamar a função `piscarLEDs()`. Se ler um ‘Y’, acende os LEDs (ou seja, acertou no número) e define a variável `estado_iniciar` como vazio. Por fim, se ler um ‘N’, apaga os LEDs ao chamar a função `apagarLEDs()` e põe a variável `contagem` para zero.

Finalmente, no `void loop()` voltamos a utilizar o Debounce tal como no arduino mestre. Quando o botão do arduino mestre é pressionado, o caractere ‘S’ enviado vai ser recebido pelo arduino secundário através da função `leitura()`. Isto faz com que seja chamada a função `piscar()`, dando assim, início ao jogo. Uma vez que o jogo começa, sempre que o botão de incremento é pressionado, a variável `contagem` vai ser incrementada, sendo usada a função `gerirLEDs()` para alterar o estado dos LEDs conforme o número de vezes que o botão é premido. Quando o jogador chegar ao número pretendido pode pressionar o botão de tentativa, que irá enviar a `contagem` para o arduino mestre através do uso do `Serial.write()`. Por fim, após o arduino mestre verificar se o número enviado pelo arduino secundário corresponde ao número aleatório, é novamente chamada a função `leitura()`. Se esta função receber o caractere ‘Y’, então o número enviado para o arduino mestre estava certo. Logo, os LEDs irão acender-se por ação da função `acenderLEDs`, ficando assim até o jogo ser reiniciado pelo arduino mestre. No entanto, se o caractere recebido for o ‘N’, o número estava errado. Logo, a `contagem` voltará a 0, e os LEDs apagam-se, fazendo com que o arduino secundário possa voltar a tentar acertar o número.

Para concluir, este trabalho foi essencial para a melhor compreensão de toda a matéria que temos vindo a dar nas aulas desde o início do ano. Ao longo do trabalho, fomos tendo alguns obstáculos, mas com o esforço e com alguns conselhos do professor, foi possível ultrapassá-los. Sentimos que estamos mais confortáveis a trabalhar com o arduino e este trabalho contribuiu significativamente para a nossa aprendizagem.