

Relatório do Trabalho do Arduino “Descobrir o LED”.

O objetivo do trabalho com o Arduino era realizar um jogo em que através do botão de pressão (porta 11) tentávamos “acertar” no led que se encontrava aceso. Por outras palavras, à medida que os leds iam acendendo sequencialmente, ao clicarmos no botão de pressão (ao qual nomeie de “bset”) pretendia-se descobrir qual o led que se encontrava aceso. Deste modo, se o led não fosse descoberto a sequência prosseguia, caso contrário permanecia aceso. Posteriormente se pretendêssemos reiniciar a sequência, apagando todos os leds já descobertos, utilizava-se o botão de pressão com a funcionalidade de fazer reset (ao qual nomeie de “breset”).

Em primeiro lugar, comecei por atribuir ao valor 12 a designação de *breset* e ao valor 11 o nome de *bset*, para depois sempre que os mesmos sejam mencionados estes sejam substituídos por essas constantes.

Em segundo lugar, decidi criar quatro funções para depois as utilizar posteriormente. Criei a função *reset*, utilizando um ciclo *for* em que os leds dos pinos 2 a 11 apagavam, recomeçando a sequência novamente. Uma função *set* para que quando fosse atingido a vitória os leds acendessem. De seguida, também criei uma função a que atribui o nome de *leds_acesos* na qual se encontra um ciclo que vai percorrer um conjunto de leds, assumindo que todos estão acesos. Ou seja, depois de se verificar se todos os leds estão realmente acesos e algum não estiver colocamos a variável *led_aceso* a zero e assim como no ciclo seguinte a variável zero não é diferente de zero ele encerra o ciclo (*return 0*). Por outras palavras, se juntarmos esta função com outra que irei referir de seguida, os leds que se encontram apagados permanecerão assim (uma vez que o *return* é 0), acendendo e apagando apenas os que se encontram acesos. Por último, para resolver o problema de as luzes não piscarem constantemente durante um certo intervalo de tempo decidi implementar a função *millis*. Deste modo, criei uma função para que os leds acesos, quando se verificasse a vitória, desligassem e apagassem durante 20 segundos através da função *millis*. Ou seja, a partir de um ciclo *while* a função vai verificar que quando o tempo atual menos o tempo inicial for inferior aos 20 segundos (tempo que se pretende que os leds estejam a piscar-*tempo_espera*), vai ligar e desligar os leds através da função *set*, que permite ligar, e da função *reset*, que permite desligar até o ciclo terminar (*return 0*).

No *void setup* defini os botões *bset* e *breset* como entrada e os leds como saída.

Consequentemente no *void loop* atribui os valores à variável *tempo* e *tempo_espera* referidas anteriormente, com 500ms e 20000ms respetivamente. Seguidamente, dentro do ciclo *for* pretendi que se pressionássemos o botão *bset* o led acendia e cada clique era mostrado no monitor série. Mas aqueles que estivessem apagados tinham de continuar a piscar e por isso defini a terceira condição, para que depois não surgisse o problema de os leds que se encontravam ligados voltassem a desligar novamente. Posteriormente, se clicássemos no botão *breset* o contador voltava a recomeçar a contar os cliques do botão *bset*. Mas como o objetivo, no final, era não só colocar os leds acesos a piscar simultaneamente como também permitir que o contador voltasse

a zero, para depois eles recomeçarem a piscar sequencialmente, defini que o contador era igual à vitória. Este facto permitia assim que o contador esquecesse os outros cliques, pois se verificarmos o *return* da *vitória* é zero. Seguidamente, indiquei que os leds deviam de começar a piscar sequencialmente a partir do pino 2, para garantir que o mesmo voltava a acontecer como de início. Por outras palavras, esta função permite também que se a condição *leds_acesos* (função que criei acima) se verificar irá se iniciar a função *vitória* (que também criei acima). Deste modo apenas as luzes que se encontram acesas é que iriam acender e apagar.

Neste trabalho, utilizei regularmente o site de documentação de referência do arduino (<https://www.arduino.cc/reference/en/>), onde por exemplo verifiquei o modo da utilização do ciclo *while* e da função *millis*. Através dos exemplos dados consegui entender como iria organizar o meu código quando utilizasse as mesmas.

Trabalho realizado por:

Inês Martins Marçal.