

**Relatório do Projeto de Tecnologia da Informática**

**Introdução**

No âmbito da disciplina de Tecnologia da Informática foi nos atribuído um projeto prático, que consiste no desenvolvimento de um jogo com auxílio ao Arduino e aos conhecimentos adquiridos no decorrer das aulas. Para o jogo, foi necessário o uso de dois Arduinos distintos e, associado a eles, um código desenvolvido por cada um do grupo. O objetivo era desenvolver um jogo em que o primeiro Arduino gerasse um número de forma aleatória e, após avisar o segundo Arduino, este teria de adivinhar esse número, com o auxílio dos LEDs que representam o número em binário.

Para haver clareza, iremos separar o relatório em duas partes, cada uma destinada para um Arduino diferente. O Arduino Mestre irá desempenhar a função de gerador de números aleatórios e o Arduino Secundário irá desempenhar a função de adivinhador desses números gerados.

**Desenvolvimento – Arduino Mestre**

Circuito

Para este trabalho, o Arduino Mestre necessitou de um botão conectado ao pino 7, que indica quando será gerado o número aleatório. Serão também necessários 5 LEDs para a leitura desse número em binário, dos pinos 8 ao 12, tal como solicitado. Cada um desses LEDs utiliza uma resistência de 220Ω, sendo dispostos pela *breadboard*. Para realizar a conexão entre os LEDs e o microcontrolador, foram utilizados ao todo 18 cabos. Para a comunicação entre os dois Arduinos, foram utilizados 3 cabos. A comunicação entre os Arduinos foi efetuada pelos pinos 0 e 1, tal como foi lecionado nas aulas.

Código

No começo do programa, foram atribuídos valores a variáveis que iriam melhorar o desempenho e clareza do código. São utilizadas, essencialmente, para o *debounce* do botão e para eventuais *resets* necessários. Também foi criada uma função à qual chamei *mostra\_leds\_binario* que, tal como o nome indica, serve para mostrar o número gerado nos LEDs dispostos pela *breadboard*, em binário. Essa função solicita três variáveis, sendo o número que foi gerado aleatoriamente, o primeiro e o último LED onde a representação binária começa. Com o auxílio da função integrada no software do Arduino, *bitread()*, fizemos a leitura de cada um dos bits que o número aleatório possuía e, em caso de ser ‘1’, o LED ficava aceso. Caso contrário, o LED permanecia desligava.

Posteriormente, iniciei o *setup()* onde foi inicializado o *Serial* do software, foram indicadas as saídas para os 5 LEDs (de 8 a 12), com o auxílio de um ciclo *for* e a entrada no pino 7, reservada para o botão que gera o número aleatório.

Agora, dentro da função *loop()*, realizei o *debounce* do botão que evita leituras incorretas ou instáveis, provocadas pelo contato imperfeito que pode traduzir-se em múltiplas ativações do mesmo. Para realizar o *debounce* recorri à função integrada no software, a *millis()*, tal como foi lecionado em sala de aula. Após isso, solicitei ao programa que, caso o botão fosse pressionado, ele gerasse um número aleatório entre 0 e 31, através da função *random(0, 32)*. Foram introduzidos os valores 0 e 32 pois esta exclui este último valor, ou seja, faz uma seleção somente de 0 a 31 tal como era solicitado. Além disso, envia para o Serial o caracter *‘S’* para informar o Arduino Secundário que o número foi gerado e ele pode proceder à tentativa de adivinhá-lo.

Após isso, cabe ao Arduino Secundário adivinhar o número gerado e, enquanto isso, o Mestre fica a aguardar que receba algo, através da função *Serial.avaiable()*, também integrada no próprio sistema do IDE. Com ela, podemos verificar se existe algo à espera para ser lido no Serial e, se houver, armazenamos o que foi recebido na variável *incomingByte*. A partir daí, basta-nos verificar se o Arduino Secundário foi próspero na adivinhação do número aleatório. Em caso positivo, o Mestre envia um caracter *‘Y’* e “reseta” o número para 31, para acender todos os LEDs. Em caso negativo, o Mestre envia um caracter ‘N’ para o informar do seu insucesso e fica a aguardar uma nova tentativa por parte do Secundário, até o mesmo acertar ou, eventualmente, ser gerado um novo número. Se esta segunda opção acontecer, o ciclo repete-se todo, avisando o Arduino Secundário que foi gerado um número novo e que ele deverá voltar a tentar adivinhar.

**Desenvolvimento – Arduino Secundário**

Circuito

Para a montagem do circuito do Arduino Secundário foram utilizados o Arduino, 5 LEDs que ilustram o número de "tentativa" em binário, dois botões, um para aumentar de um em um o número "tentativa" e um para enviar este número ao Arduino Mestre, 20 cabos ao todo, 5 resistências de 220 Ohm (uma para cada LED) e uma *breadboard.*

Código

No início do programa, antes dos métodos *setup* ou *loop,* são definidas um número de variáveis relacionadas com o *debounce,* com o estado do jogo (começado ou não) e com o piscar dos LEDs (que acontece no início de cada jogo). Tal como no Arduino Mestre, é definida uma função *show\_binary\_leds* cujo propósito é, dado um número, o pino inicial e o final, liga os LEDs correspondentes aos bits 1 do número em binário, como já foi referido.

Durante o método *setup*, é chamado o método *Serial.begin* para comunicação em série, são definidos os números dos pins para conexão aos LEDs (9 a 13) com o modo OUTPUT, os cinco LEDs são ligados e são definidos os pins de conexão aos dois botões (6 e 7) com o modo INPUT\_PULLUP.

No método *loop* é recebida a informação por comunicação em série, sendo que 'S' começa o jogo, 'Y' acaba o jogo, e 'N' repõe o valor do número "tentativa" para 0, para uma nova tentativa ser feita. São feitos também os três piscar de LEDs que acontecem no início do jogo. Se o jogo estiver em curso, é feito o *debouncing* de cada botão para que quando carregado ou o número tentativa possa ser aumentado, ou o número atual possa ser enviado ao Arduino Mestre.

**Conclusão**

De uma forma geral, este projeto permitiu a ambos os integrantes do projeto a compreender e assimilar melhor determinadas funcionalidades do Arduino e o seu potencial enquanto microcontrolador. Apesar de algumas adversidades durante a produção do código, por exemplo, na conjugação do Debounce com as condições que nós queríamos implementar, conseguimos ultrapassá-las e aprender com alguns erros que cometemos ao longo do projeto.

Ivo Leonardo Almeida Freire, nº 2022257546

Gabriel Alexandre Serens de Almeida, nº 2023226622