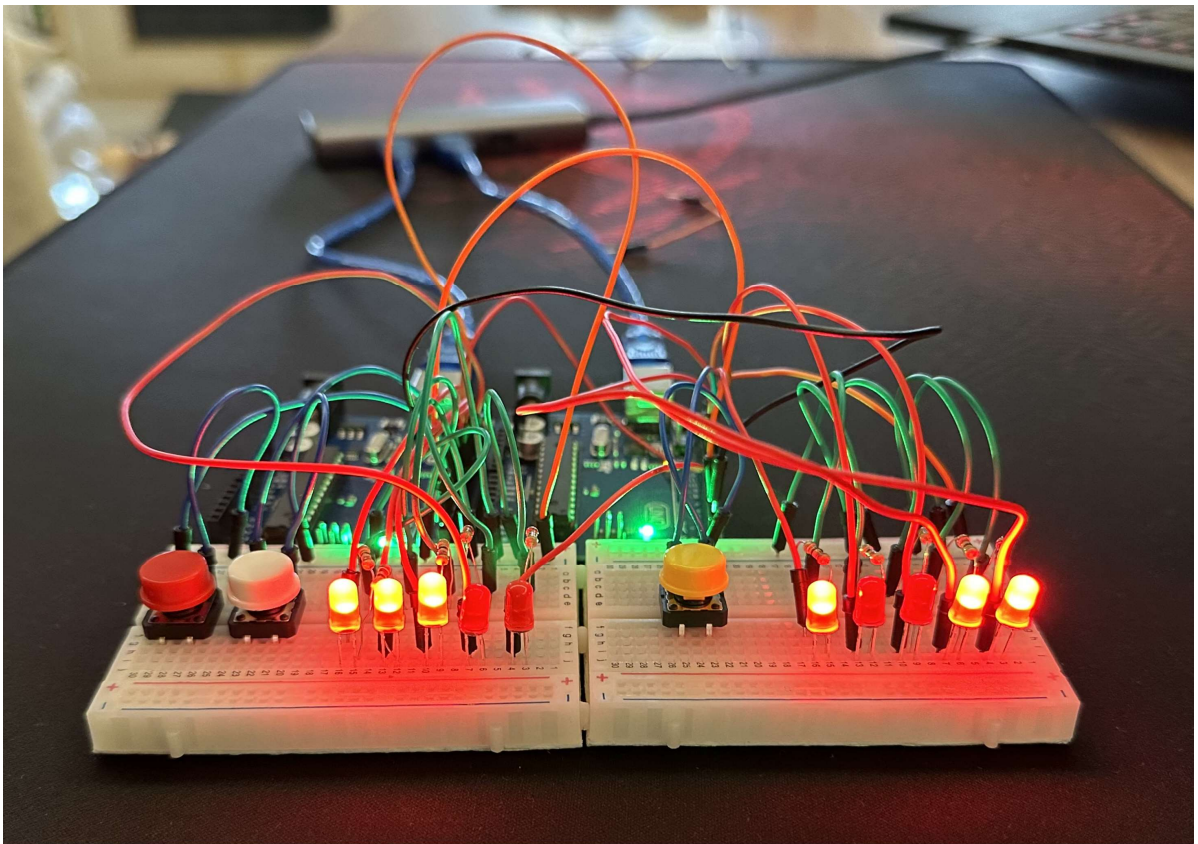


Trabalho prático 1

Tecnologia da Informática 2023/2024

Guess the number



Licenciatura em Engenharia Informática | 10/11/2023

Trabalho realizado por:

João Pedro Mota Cruz- 2022258375

Tiago Mustra Santos- 2023213359

Introdução:

Para a realização do nosso trabalho, apesar dos dois circuitos só diferirem no facto do arduino Secundário ter dois botões e o Mestre apenas ter um, foi necessário fazer dois códigos, um para controlar cada um.

Explicitação dos códigos:

Inicialização e Setup():

No arduino Mestre, é inicializada a constante "BUTTON_MESTRE", que representa o botão mestre, que está ligado ao pino digital 7. São também declaradas as variáveis "num_aleatorio" que representa o número aleatório (que terá o seu valor alterado mais tarde), "BUTTON_MESTRE_lastState", "valueByte", "buttonState", "lastButtonState", "lastDebounceTime" e "debounceDelay".

Utilizámos a linha de código "randomSeed(analogRead(0))" para que a partir do valor lógico lido no pino 0, seja gerado uma sequência de números aleatórios diferentes, para que quando se execute o código várias vezes, se obtenham sequências de números aleatórios diferentes.

No arduino secundário, foram inicializadas as constantes "TENTATIVA_BUTTON" e "INCREMENTO_BUTTON", que representam os botões tentativa (vermelho) e incremento (branco) e estão ligados aos pinos digitais 6 e 7, respetivamente. Neste foi também inicializada a variável booleana "jogo", a que é dada o valor inicial falso e as variáveis "valueByte", "num_secundario", "buttonState", "lastButtonState", "lastDebounceTime" e "debounceDelay".

Em ambos foram inicializados os leds conectados aos pinos 8 a 12 e de seguida ligados.

Mais tarde, codificámos o "debounce" dos botões mestre, tentativa e incremento para acabar com os problemas de ruído e instabilidade que podem ocorrer quando este é pressionado e/ou solto, e assim melhorar a fiabilidade da operação de clicar no botão.

loop() e outras funções:

Arduino Mestre:

Quando o botão mestre é pressionado, é gerado o número aleatório entre 0 e 32 (exclusive), utilizando a função `random()`, que vai ser representado em código binário nos leds, através do ciclo `for()`, que lhe sucede e utilizando os “shift”s. Para além disso, o arduino Mestre envia um ‘S’ para o Secundário, de modo a comunicar-lhe que o jogo será iniciado.

Arduino Secundário:

Se a variável `jogo` for falsa (na primeira vez que o ciclo `loop()` é corrido, tem esse valor), se houver dados disponíveis para leitura na porta serial (ou seja `Serial.available() > 0`), e se este arduino receber ‘S’ do outro, então a variável `jogo` passa a ser verdadeira e as luzes piscam, através da função criada `piscar()`, mostrando que o jogo será iniciado. A função `piscar(num)` faz com que os leds pisquem (sincronizados), variando entre ligados e desligados, o número de vezes que quisermos, “num”, estando em cada estado durante 300 milissegundos.

Através da condição `if{}` seguinte, sempre que o botão incremento é pressionado (estado LOW), o número secundário aumenta o seu valor em 1, e através do ciclo `for`, é representado nos leds em binário. De seguida, quando pressionado o botão tentativa, o número secundário criado é enviado para o arduino Mestre.

De volta ao arduino Mestre:

O valor recebido pelo arduino Secundário é atribuído à variável “valueByte” e de seguida a “num_secundario”. Caso o “num_secundario” seja igual a “num_aleatorio”, então o número foi adivinhado, e por isso é enviado um ‘Y’ para o arduino Secundário e os leds são acesos. Caso contrário, é enviado um ‘N’ para arduino Secundário.

Voltando ao arduino Secundário:

Se receber ‘N’ então o número Secundário volta a 0, as luzes são todas apagadas e pode se tentar adivinhar o número de novo, as vezes que quisermos. Se receber ‘Y’, então as luzes são todas acesas e a variável “jogo” volta a ter o valor falso, de modo a que não volte a entrar na condição “`if (jogo) {}`” pois o jogador já ganhou.

Se o botão (mestre) do arduino Mestre voltar a ser permitido, o jogo é reiniciado, piscando 3 vezes.

Se durante qualquer momento do jogo, o botão mestre do arduino Mestre for premido, o jogo reinicia, sendo o número aleatório mudado e as leds do arduino Secundário piscam 3 vezes.

Observações/ lacunas pertinentes:

Durante o desenvolvimento do projeto, diversas dificuldades foram enfrentadas, exigindo soluções para garantir o funcionamento adequado do sistema. Esta parte destaca algumas dessas dificuldades, descreve as estratégias adotadas para superá-las e oferece observações importantes que contribuíram para a evolução do projeto.

-Debouncing (no arduíno secundário):

A implementação do debouncing para os botões do arduíno Secundário representou um verdadeiro desafio no projeto. A dificuldade não foi propriamente o desenvolvimento do debouncer mas sim a sua implementação, pois apesar do funcionamento deste aquando usado no botão mestre no arduino Mestre, quando reutilizado no arduino Secundário, o resultado não fora o mesmo.

A verdade é que na teoria o código estava certo, mas por alguma razão desconhecida este debouncer nao respondia. Depois de muita pesquisa e alterações desnecessárias decidimos prosseguir na resolução de outros problemas e depois de algum tempo o debouncer já estava funcional.

-Representação dos números em binário nos leds.

Alguns desafios foram encontrados na lógica condicional, principalmente no que diz respeito ao controlo dos leds. Foram observados problemas como leds não desligados e reinicialização inadequada. Isto aconteceu maioritariamente devido á má implementação do “shift”, no entanto, após várias experiências, conseguimos superar este desafio!

Licenciatura em Engenharia Informática | 10/11/2023

Trabalho realizado por:

João Pedro Mota Cruz- 2022258375

Tiago Mustra Santos- 2023213359