INSTITUTO POLITÉCNICO DE BRAGANÇA
Escola Superior de Tecnologia e Gestão

Embedded systems

2nd Practical Work
(Arduino and hardware)

Gregor Umbelino (a52981) // Pedro Decolin (a52982)

ESTiG 2022/2023

OBJETIVO:

O objetivo do presente trabalho é recriar um brinquedo no estilo "Bop it". Este brinquedo dá uma série de comandos via alto-falante para o usuário e o objetivo é cumprir o que foi pedido dentro do tempo limite. Por exemplo, o Bop it diz: "aperte o botão amarelo". Se o usuário apertar o botão correto em tempo hábil, pontos são acumulados e a próxima instrução é falada. Se ele erra ou demora muito para agir, ele perde a rodada.

Exemplo de um Bob it

Portanto, iremos adaptar o hardware que temos para recriar este brinquedo. Utilizaremos o Display LCD para o envio das instruções para o usuário. Para as tarefas a serem cumpridas, utilizaremos alguns dos sensores disponíveis: push button, joystick, encoder, sensor de distância. O buzzer será usado para indicar, por meio de alguns sons, início de jogo, passagem de nível, sucessos ou falhas do usuário. O módulo de 8 LEDs será usado para indicar a dificuldade que o usuário se encontra (fácil, médio, difícil). O servo motor será utilizado na dificuldade "difícil", sendo acionado alternadamente, a fim de atrapalhar a concentração do usuário.

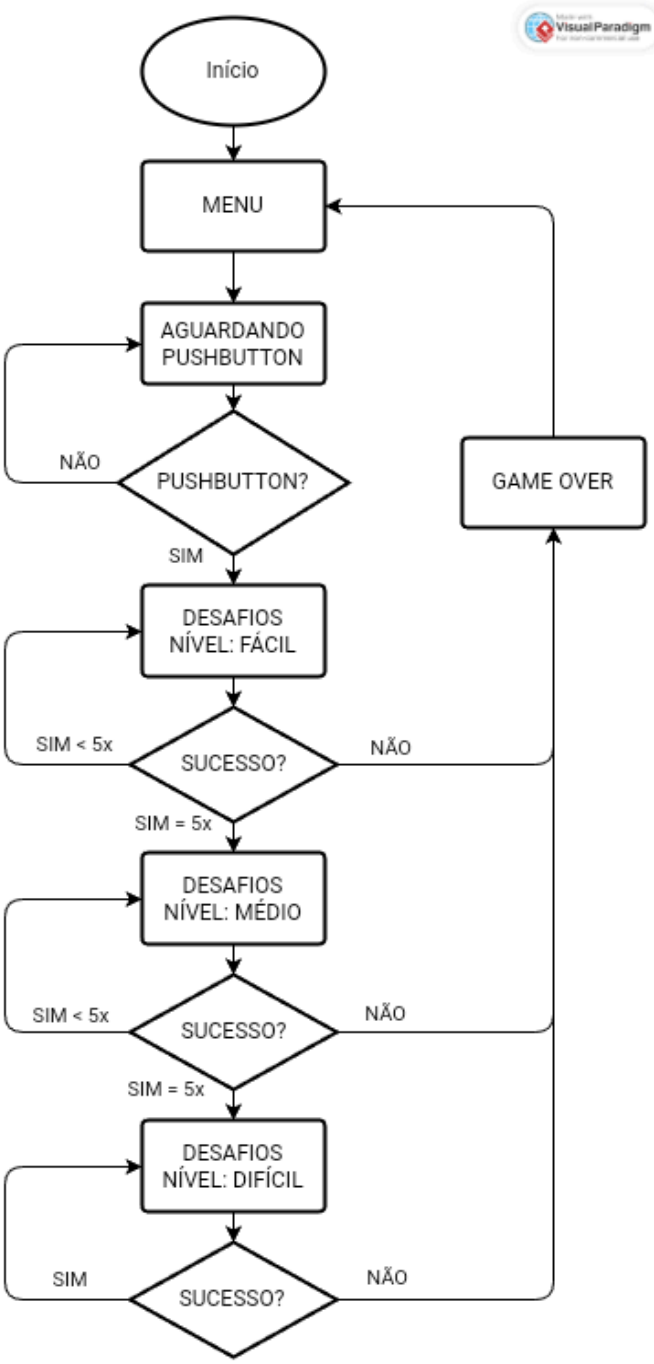
O fluxo do jogo será o seguinte: ao ligar o sistema, o LCD exibe o highscore atual (começa em 5, depois é atualizado conforme o jogador ultrapassa-o) e manda a instrução de apertar o push button para começar. Ao fazer isso, a primeira instrução é mandada ao LCD. Em todo momento, caso o usuário acerte em tempo hábil, ele acumula pontos e recebe uma nova tarefa, caso não cumpra em tempo hábil, ele volta ao início, reiniciando as dificuldades. O jogo é dividido em 3 dificuldades:

- Fácil: serão utilizados o push button (pressionar), encoder (gira horário/anti-horário, pressionar), sensor de distância (aproximar a mão). O usuário terá certo tempo para realizar as tarefas.
- Médio: será incluído o joystick (para cima, para baixo, para direita, para esquerda) na lista de desafios. O usuário terá menos tempo para realizar as tarefas.
- Difícil: O servo motor será utilizado para atrapalhar o usuário. Além disso, o usuário terá, a cada desafio, menos tempo para realizar as tarefas.

O usuário começa no fácil e após concluir 5 tarefas com sucesso, passa ao médio, que após 5 tarefas concluídas com sucesso também passa para o difícil. Ele permanece no difícil até errar. O display de 8 LEDs será usado para indicar a dificuldade, tendo um número de LEDs acesos proporcional ao fácil e ao médio, porém, na dificuldade máxima ficam alternando após cada desafio concluído.

DESENVOLVIMENTO:

Inicialmente foi desenvolvida a lógica do sistema. Construímos um fluxograma que explicita como o fluxo da execução ocorre:



Após isso, recorremos à plataforma Wokwi para desenvolver uma simulação do projeto e nos auxiliar a escrever o código. A simulação usa apenas aqueles componentes utilizados efetivamente no projeto, conectados nas portas identificadas no enunciado do relatório. A simulação é executada com sucesso e mostra como o sistema deve funcionar e pode ser acessada em simulação.

O código final é extenso e decidimos por ressaltar as partes mais importantes. O código na íntegra pode ser acessado em nosso repositório.

```
void loop() {
  mostrarMenu();

  for(i = 0; i < 5 && flag; i++) {
    escolherDesafio();
  }

  if (flag) {
    pcb.write(4, LOW);
    pcb.write(5, LOW);
    pcb.write(6, HIGH);
    pcb.write(7, HIGH);
    transitarNivel();
  }

  for(i = 0; i < 5 && flag; i++) {
    escolherDesafio();
  }

  if (flag) {
    transitarNivel();
    pcb.selectNone();
  }

  while (flag) {
    intervalo <= 500 ? intervalo = intervalo : intervalo -= 100;
    posServo < 90 ? posServo = 180 : posServo = 0;
    pcb.toggle(0);
    pcb.toggle(1);
    pcb.toggle(2);
    pcb.toggle(3);
    pcb.toggle(4);
    pcb.toggle(5);
    pcb.toggle(6);
    pcb.toggle(7);
    myServo.write(posServo);
    escolherDesafio();
  }

  gameOver();
}
```

Função loop principal

```
//// - Escolha
  void escolherDesafio () {
    nivel >= 1 ? v_aleatoria = random(4) : v_aleatoria = random(3);

    switch (v_aleatoria){
      case 0:
        flag = desafioPushbutton();
        break;
      case 1:
        flag = desafioDistancia();
        break;
      case 2:
        flag = desafioEncoder();
        break;
      case 3:
        flag = desafioJoystick();
        break;
      default :
        Serial.println("ERRO");
        flag = false;
        break;
    }
  }
```

Seleção de um desafio

```
//// - Desafios
  bool desafioPushbutton () {
    tempo_inicial = millis();
    tone(buzzer, 1397); // F
    Serial.println("Pressione o botão!");
    Serial.print("Score: ");
    Serial.println(score);
    noTone(buzzer);
    while (!digitalRead(pushbutton)) {
      tempo_final = millis();
      if (tempo_final - tempo_inicial == intervalo) {
        return false;
      }
    }
    score++;
    return true;
  }
```

Exemplo de funcionamento de um desafio

CONCLUSÃO:

Inicialmente, a ideia do grupo era utilizar todos os sensores e atuadores. Além do que implementamos, gostaríamos de adicionar: motor de passos para distrair o jogador, assim como o servo motor; sensor de temperatura, como sendo um desafio adicional. Neste caso, não achamos aplicabilidade viável, visto que não era possível afetar o sensor em tempo hábil apenas utilizando a temperatura das mãos, assim impossibilitando torná-lo um desafio. Entretanto, ficamos muito satisfeitos com o resultado, visto que a simulação com o hardware real foi executada como o esperado. A simulação pode ser acessada aqui.