

# Tegra 3

Danilo Souza, Hugo Santos, Welton Araújo

Emails: {dhcsouza, hugoleonardoeng07, weltonmaxx007}@gmail.com

Matrículas: 10080000801, 10080000701, 10080000501

**Abstract—**

**Index Terms—**

Tabela I  
VARIÁVEIS DE CONTROLE

| Variáveis    | Endereco | Descrição   |
|--------------|----------|---|
| COUNTER      | 0x20     | Contador do número de botões apertados antes quem perdeu a partida                                    |
| LAST_COUNTER | 0x21     | Backup da configuração de CONT  |
| BOT          | 0x22     | Campo de 8 bits responsável por memorizar se o botão foi apertado ou não durante o momento de decisão |
| LAST_BOT     | 0x23     | Backup da configuração de BOT   |
| COUNT1       | 0x24     | Contador 1 para o delay de duração do choque  |
| COUNT2       | 0x25     | Contador 2 para o delay de duração do choque  |
| COUNT3       | 0x26     | Contador 3 para o delay de duração do choque  |

## I. DESCRIÇÃO DA IMPLEMENTAÇÃO

O algoritmo de funcionamento do programa funciona com o auxílio das variáveis da Tabela I onde também são descritas suas funções de controle. A Tabela II contém os pinos do PIC utilizados como saídas e entradas.

Inicialmente, no *loop* da rotina *START*, o programa tem somente dois botões como entrada, *BOTAO\_START* para iniciar e *BOTAO\_N\_PS* para definir o número de jogadores da partida. Na seleção do número de jogadores, os bits *BOT,2* e *BOT,3* são setados ou limpados enquanto que a variável *COUNTER* inicia como 0, 1 ou 2 para 4, 3 ou 2 jogadores, respectivamente. Isso se deve porque, logo que o contador atinge um valor igual a 3, o perdedor deve levar um choque.

Ao pressionar o botão de selecionar o número de jogadores, a rotina *SET\_N\_PS* é chamada onde os bits *ENABLED\_LED\_P3* e *ENABLED\_LED\_P4* são lidos, caso tenham valor lógico alto, significa que os jogadores 3 e 4, respectivamente, estão habilitados e a rotina *DISABLE\_P3P4* é chamada para desabilitá-los. Caso somente 2 estiverem habilitados, o jogador 3 é habilitado pela rotina *SET\_P3* onde bit *ENABLED\_LED\_P3* é setado para alto. Em último caso, se houverem 3 habilitados, a rotina *SET\_P4* é chamada para setar como 1 o bit *ENABLED\_LED\_P4*.

No começo de uma partida, o programa passa a operar no delay da rotina *DELAY\_LED*. Sua função é manter *START\_LED* setado até o momento de limpá-lo e os botões do jogo serem apertados, porém também checa constantemente, na sua “subrotina” *DELAY2\_LED*, as entradas dos botões do jogo para punir com um choque, através das rotinas *SHOCK\_P1*, *SHOCK\_P2*, *SHOCK\_P3* e *SHOCK\_P4*, o jogador que apertar o botão antes da hora certa e, em seguida, esperar o início de uma nova partida.

Quando o *START\_LED* apagar, o programa estará rodando no loop da rotina *MAIN*, definido como momento de decisão, onde os botões do jogo são novamente checados. Caso algum seja apertado, alguma das rotinas *BOTAO\_P1\_PRESS*, *BOTAO\_P2\_PRESS*, *BOTAO\_P3\_PRESS* ou *BOTAO\_P4\_PRESS* são chamadas para, além de incrementar 1 no valor de *COUNTER*, setar o bit *BOT,0*, *BOT,1*, *BOT,2* ou *BOT,3*, respectivamente para cada jogador.

Durante o momento de decisão, os botões *BOTAO\_P1*, *BOTAO\_P2*, *BOTAO\_P3* e *BOTAO\_P4* passam a ser funcionais. No entanto, realmente funcionarão somente o botão daqueles

jogadores que foram habilitados. Por exemplo, caso sejam somente dois oponentes, os botões *BOTAO\_P3* e *BOTAO\_P4* ainda funcionam como entrada, porém os seus estados de pressionado ou não já foram setados para o estado pressionado durante a seleção do número de jogadores, isto é, neste caso, *BOT,2* e *BOT,3* estão setados como 1 fazendo-os indiferentes neste momento. Ademais, a variável *COUNTER* já foi incrementada em 1 para cada jogador desabilitado ou vice-versa para cada jogador é reabilitado. Os valores de *BOT* e *COUNTER* são armazenados em *LAST\_COUNTER* e *LAST\_BOT*.

Ainda dentro do loop da *MAIN*, existe uma rotina chamada *CHECK\_COUNTER*. Nesta rotina, a variável *COUNTER* é checada à procura de um valor igual a 3 para fazer a chamada da rotina *FIND\_LOSER*.

A rotina *FIND\_LOSER* procura pelo primeiro 0 entre os bits de *BOT,0*, *BOT,1*, *BOT,2* e *BOT,3*, nesta ordem, para setar algum dos bits de *SHOCK\_LED\_P1*, *SHOCK\_LED\_P2*, *SHOCK\_LED\_P3* e *SHOCK\_LED\_P4* como 1 simbolizando o choque, respectivamente para o jogadores 1, 2, 3 e 4. Em sequência, chama a rotina *DELAY\_SHOCK* que limita a duração do choque, limpa os bits *SHOCK\_LED\_P1*, *SHOCK\_LED\_P2*, *SHOCK\_LED\_P3* e *SHOCK\_LED\_P4*, carrega os valores de *LAST\_COUNTER* e *LAST\_BOT* em *COUNTER* e *BOT*, respectivamente, para recomençar o jogo com a última configuração de jogadores.

## II. DESCRIÇÃO DO PROJETO

O projeto baseia-se em um jogo, onde o objetivo é não levar choque, o brinquedo possui 4 controles com um botão cada e uma luz no meio, usada para indicar a hora em os jogadores devem apertar os botões, o último que apertar leva um choque. Ao começo do jogo o LED fica piscando, a rodada termina quando o LED mudar de cor, a partir desse momento o último jogador que apertar o botão leva um choque, que é descarregado no próprio controle, feito de metal. Caso algum

Tabela II  
BITS DE CONTROLE

| Bits           | Pino    | Descrição  |
|----------------|---------|--|
| BOTAO_P1       | PORTA,0 | Botão para mudar o estado do jogador 1 durante o momento de decisão da partida |
| BOTAO_P2       | PORTA,1 | Botão para mudar o estado do jogador 2 durante o momento de decisão da partida |
| BOTAO_P3       | PORTA,2 | Botão para mudar o estado do jogador 3 durante o momento de decisão da partida |
| BOTAO_P4       | PORTA,3 | Botão para mudar o estado do jogador 4 durante o momento de decisão da partida |
| BOTAO_START    | PORTA,4 | Botão para iniciar a partida   |
| BOTAO_N_PS     | PORTA,5 | Botão para escolher o número de jogadores da partida                           |
| SHOCK_LED_P1   | PORTB,0 | LED que simboliza o sinal de choque no jogador 1                               |
| SHOCK_LED_P2   | PORTB,1 | LED que simboliza o sinal de choque no jogador 2                               |
| SHOCK_LED_P3   | PORTB,2 | LED que simboliza o sinal de choque no jogador 3                               |
| SHOCK_LED_P4   | PORTB,3 | LED que simboliza o sinal de choque no jogador 4                               |
| ENABLED_LED_P3 | PORTB,4 | LED indicador de participação do jogador 3 na partida                          |
| ENABLED_LED_P4 | PORTB,5 | LED indicador de participação do jogador 4 na partida                          |
| START_LED      | PORTB,6 | LED que sinaliza o início do momento de decisão                                |

Fig. 1. Ilustração do jogo

jogador aperte o botão antes do tempo, este também levará choque como punição.

Para o projeto foram usados 4 botões, cada botão com 2 LED's, um para indicar que aquele jogador está participando do jogo e outro para indicar que aquele jogador levou um choque, um botão para representar o *start*, um botão para escolher o número de jogadores, sendo no mínimo 2 e no máximo 4 e um led para indicar o começo e término de uma rodada. Quando o jogo começa, a luz do LED fica constante e quando o LED apaga, a rodada termina um LED amarelo do último botão pressionado acende indicando que aquele jogador levou um choque. Os LED's são somente para representar o circuito do choque que poderia ser acionado usando o mesmo sinal de saída utilizado para acender o LED.