

SEL0629

Aplicação de Microprocessadores I

Prática 2 Teclado e Buzzer

Prof. Marcelo Andrade da Costa Vieira

Teclado para tocar as 7 notas musicais no buzzer

- Objetivo:
 - Aprendizado do microcontrolador PIC18F45k22
 - Uso das portas de I/O, temporizadores e interrupção
 - Uso da Linguagem C para microprocessadores
 - Uso do Compilador - Mikro C Pro for PIC
 - Uso do teclado matricial e buzzer
- Prática:
 - Utilizar o teclado matricial para tocar as 8 notas musicais da escala de Dó Maior no buzzer.

Notas musicais no Buzzer

- A cada tecla do teclado matricial (de 1 a 8), gerar na saída da porta RC2 ou RE1 do PIC uma onda quadrada de frequência correspondente as notas musicais seguindo a tabela abaixo:

| Tecla | Nota | Símbolo | Frequência (Hz) |
|-------|------|---------|-----------------|
| 1 | Dó | C | 2073 |
| 2 | Ré | D | 2349 |
| 3 | Mi | E | 2637 |
| 4 | Fá | F | 2794 |
| 5 | Sol | G | 3136 |
| 6 | Lá | A | 3520 |
| 7 | Si | B | 3951 |
| 8 | Dó | C | 4146 |

Teclado Matricial

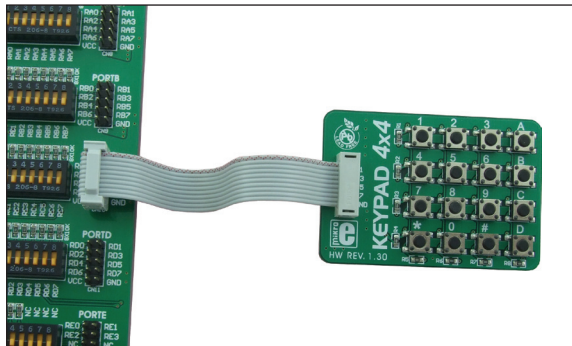


Figure 2: Keypad 4x4 connected to development system

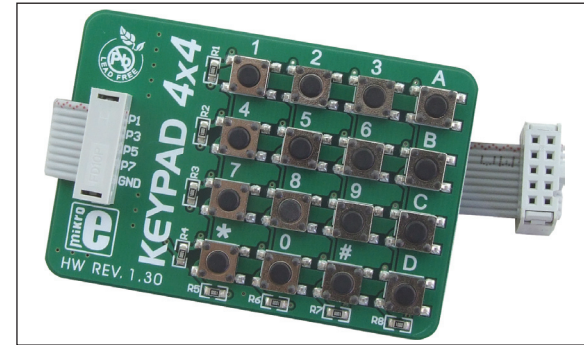


Figure 1: Keypad 4x4

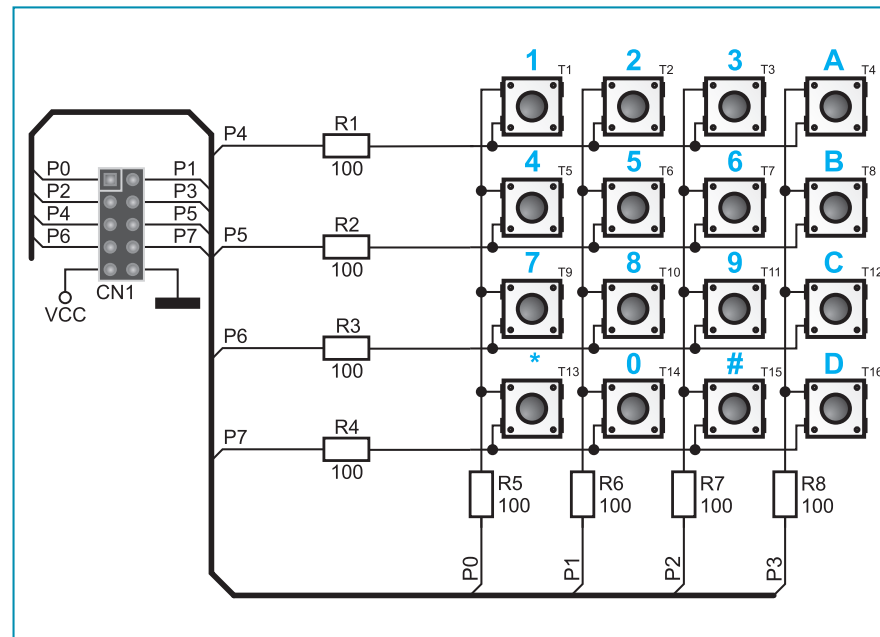


Figure 3: Keypad 4x4 connection schematic

Controle do Teclado

- Utilizar varredura do teclado matricial (Conectar na Porta B)
 - Na porta que for conectado o teclado, definir 4 pinos de I/O como entrada e 4 pinos de I/O como saída;
 - Conectar as linhas do teclado matricial nas entradas e as colunas nas saídas (ou vice-versa);
 - Nos 4 pinos configurados como entrada, definir um estado lógico 1 quando nenhuma tecla é pressionada (usar resistores de pull-up externos ou internos^{*}).
 - Nos 4 pinos configurados como saída, gerar nível lógico 0 na forma de varredura. Esse nível aparecerá em alguma entrada quando o botão correspondente for pressionado.

^{*}Veja registradores INTCON2 e WPUB

REGISTER 9-2: INTCON2: INTERRUPT CONTROL 2 REGISTER

| | | | | | | | |
|---------------------------------|---------|---------|---------|-----|--------|-----|-------|
| R/W-1 | R/W-1 | R/W-1 | R/W-1 | U-0 | R/W-1 | U-0 | R/W-1 |
| $\overline{\text{RBP}}\text{U}$ | INTEDG0 | INTEDG1 | INTEDG2 | — | TMR0IP | — | RBIP |
| bit 7 | | | | | | | bit 0 |

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

- bit 7 **$\overline{\text{RBP}}\text{U}$:** PORTB Pull-up Enable bit
 1 = All PORTB pull-ups are disabled
 0 = PORTB pull-ups are enabled provided that the pin is an input and the corresponding WPUB bit is set.
- bit 6 **INTEDG0:** External Interrupt 0 Edge Select bit
 1 = Interrupt on rising edge
 0 = Interrupt on falling edge
- bit 5 **INTEDG1:** External Interrupt 1 Edge Select bit
 1 = Interrupt on rising edge
 0 = Interrupt on falling edge
- bit 4 **INTEDG2:** External Interrupt 2 Edge Select bit
 1 = Interrupt on rising edge
 0 = Interrupt on falling edge
- bit 3 **Unimplemented:** Read as '0'
- bit 2 **TMR0IP:** TMR0 Overflow Interrupt Priority bit
 1 = High priority
 0 = Low priority
- bit 1 **Unimplemented:** Read as '0'
- bit 0 **RBIP:** RB Port Change Interrupt Priority bit
 1 = High priority
 0 = Low priority

Note: Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the global enable bit. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt. This feature allows for software polling.

REGISTER 10-12: WPUB: WEAK PULL-UP PORTB REGISTER

| | | | | | | | |
|-------|-------|-------|-------|-------|-------|-------|-------|
| R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 |
| WPUB7 | WPUB6 | WPUB5 | WPUB4 | WPUB3 | WPUB2 | WPUB1 | WPUB0 |
| bit 7 | | | | | | | bit 0 |

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

bit 7-0

WPUB<7:0>: Weak Pull-up Register bits

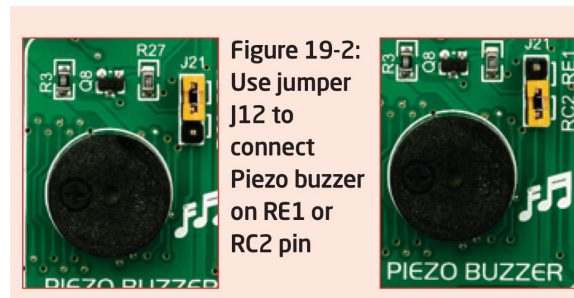
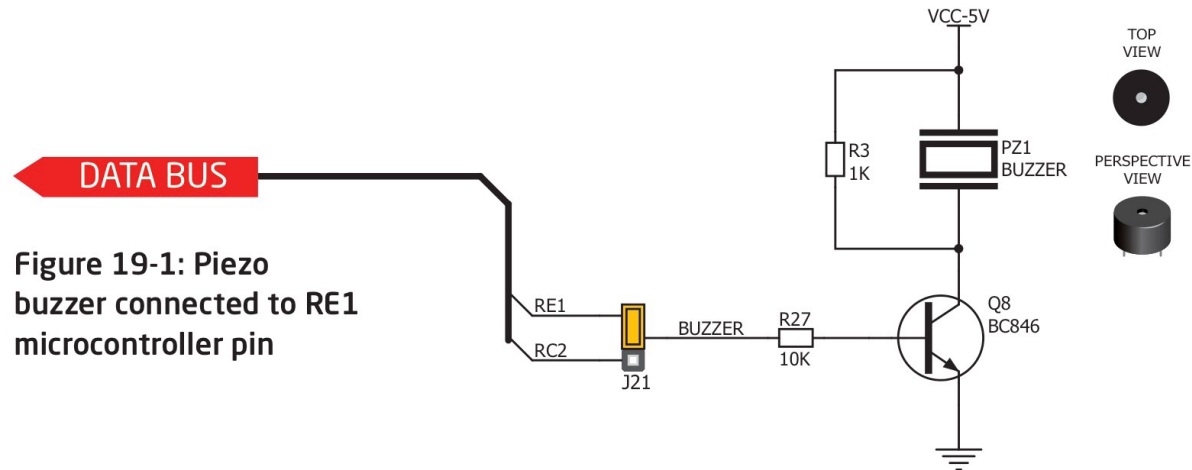
1 = Pull-up enabled on PORT pin

0 = Pull-up disabled on PORT pin

Notas musicais no Buzzer

- Dependendo da tecla pressionada, uma onda quadrada com a frequência correspondente deve ser gerada na entrada do Buzzer;
- Utilizar o temporizador TMR0 (com interrupção) para geração das ondas quadradas no Buzzer;
- A frequência do clock é de 8 MHz.

Buzzer



Prática 2

- Grupo de no máximo 2 pessoas.
- A prática deve ser feita no Micro C Pro for PIC
 - Programa em C
 - Compilar e Simular (debugger)
- Gravar o circuito no Kit EasyPIC v.7
 - Habilitar as chaves necessárias (Display, Botões)
 - Conectar o Teclado 4x4
 - Carregar o programa no PIC
 - Simular: fazer o buzzer tocar a nota de acordo com a tecla apertada
 - Validar a prática: Usar um frequencímetro ou osciloscópio para mostrar a frequência da onda gerada no buzzer e verificar o erro em relação à frequência esperada.

Relatório Prática 2

- Colocar introdução, objetivos, resultados, conclusões, etc.
– *consultar as normas para confecção de relatórios no site da disciplina.*
- Colocar o esquemático do hardware utilizado (Proteus, Orcad ou similar).
- Colocar trechos do programa desenvolvido (comentado).
- Colocar os resultados da validação: utilizar o osciloscópio para verificar os valores de frequência. Colocar tabela com os valores e erros.

Envio do Relatório

- Além do relatório em PDF, enviar também os arquivos do projeto gerados pelo compilador (código em C);
- Enviar os arquivos (*.zip) pelo site do e-Disciplinas até a data definida pelo professor (30/05/2023 – 23:59h).
- Mostrar o circuito funcionando para o professor na aula imediatamente posterior ao envio do relatório.