Conectando o PIC

Explorando Recursos Avançados

David José de Souza
Nicolás César Lavinia
Mosaico Engenharia Eletrônica

Sumário

CAPÍTULO 1 - INTRODUÇÃO 1

A MOSAICO 1
NOSSOS OBJETIVOS 1
OS PRÉ-REQUISISTOS 2
A SISTEMÁTICA DO TREINAMENTO 2
Conhecendo o microcontrolador base 2
Conhecendo o hardware utilizado 2
Aulas práticas 2
Projeto final 3
Apêndices 3
CD-Rom 3
Padrões textuais 3

CAPÍTULO 2 - O PIC 16F877 5

INTRODUÇÃO 5
PINAGEM 6
NOMENCLATURA DOS PINOS 6
A ESTRUTURAÇÃO INTERNA 9
OS CICLOS DE MÁQUINA 10
A MEMÓRIA DE PROGRAMA 11
Vetor de reset 11
Vetor de interrupção 11
Tamanho da memória e páginas 11
Pilha (Stack) 12
Mapa da memória de programa 13
A MEMÓRIA DE DADOS VOLÁTIL (RAM) 13
Registradores especiais 13
Registradores de uso geral 14
Tamanho da memória e bancos 14
Mapa da memória de dados 16
MEMÓRIAS NÃO VOLÁTEIS (E^2PROM E FLASH) 16
AS INTERRUPÇÕES 17
Interrupção de Timer 0 17
Interrupção externa 17
Interrupção por mudança de estado 17
Interrupção da porta paralela (PSP) 18
Interrupção dos conversores A/D 18
Interrupção de recepção da USART 18
Interrupção de transmissão da USART 18
Interrupção da comunicação serial (SSP e I²C) 18
Interrupção do CCP1 (Capture/Compare/PWM) 18
Interrupção do Timer 2 18
Interrupção do Timer 1 18
Interrupção de fim de escrita na E^2PROM/FLASH 18
Interrupção de colisão de dados (Bus Collision) 19
Interrupção do CCP2 (Capture/Compare/PWM) 19
Operando com as interrupções 19
DEMAIS RECURSOS E PERIFÉRICOS 21
CARACTERÍSTICAS ELÉTRICAS 23

CAPÍTULO 3 - RESUMO DO SET DE INSTRUÇÕES 25

OS TERMOS UTILIZADOS 25
A CONSTRUÇÃO DOS NOMES DAS INSTRUÇÕES 26

TECLAS E LEDS 30
BUZZER 30
MEMÓRIA E^2PROM EXTERNA 30
COMUNICAÇÃO SERIAL RS-232 30
CONVERSÃO ANALÓGICO / DIGITAL (A/D) 30
SISTEMA DE TEMPERATURA 31
CONECTOR DE EXPANSÃO 31
GRAVAÇÃO IN-CIRCUIT 31

CAPÍTULO 5 - AULA 1: AS PRIMEIRAS EXPLORAÇÕES (I/OS E TIMERS) 33

INTRODUÇÃO 33
TEORIA E RECURSOS DO PIC 33
Estudo das Portas 33
Estudo dos Timers 39
LÓGICA DO EXEMPLO 46
ESQUEMA ELÉTRICO 47
FLUXOGRAMA 48
CÓDIGO 52
DICAS E COMENTÁRIOS 58
EXERCÍCIOS PROPOSTOS 58
ANOTAÇÕES 59

CAPÍTULO 6 - AULA 2: VARREDURA DE DISPLAY DE 4 DÍGITOS 61

INTRODUÇÃO 61
TEORIA E RECURSOS DO PIC 61
LÓGICA DO EXEMPLO 63
ESQUEMA ELÉTRICO 65
FLUXOGRAMA 66
CÓDIGO 71
DICAS E COMENTÁRIOS 80
EXERCÍCIOS PROPOSTOS 80
ANOTAÇÕES 81

CAPÍTULO 7 - AULA 3: OPERAÇÃO COM DISPLAY DE CRISTAL LÍQUIDO (LCD) 83

INTRODUÇÃO 83
TEORIA E RECURSOS DO PIC 83
Inicialização do LCD 85
Comandos do LCD 87
LÓGICA DO EXEMPLO 91
ESQUEMA ELÉTRICO 92
FLUXOGRAMA 93
CÓDIGO 98
DICAS E COMENTÁRIOS 106
EXERCÍCIOS PROPOSTOS 106
ANOTAÇÕES 107

CAPÍTULO 8 - AULA 4: CONVERSOR ANALÓGICO-DIGITAL INTERNO 109

INTRODUÇÃO 109
TEORIA 109
RECURSOS DO PIC 111
LÓGICA DO EXEMPLO 117
ESQUEMA ELÉTRICO 118
FLUXOGRAMA 119
CÓDIGO 122
DICAS E COMENTÁRIOS 130
EXERCÍCIOS PROPOSTOS 130

FLUXOGRAMA 138
CÓDIGO 140
DICAS E COMENTÁRIOS 146
EXERCÍCIOS PROPOSTOS 146
ANOTAÇÕES 147

CAPÍTULO 10 - AULA 6: OS MÓDULOS CCP (CAPTURE / COMPARE / PWM) 149

INTRODUÇÃO 149

Lógica do exemplo

Nosso exemplo para demonstração do LCD é bem reduzido. Simplesmente utilizaremos este display para informar ao usuário qual botão foi pressionado. Para isso elaboramos uma rotina chamada ESCREVE que envia a informação passada pelo W para o display. Para evitarmos problemas de temporização, esta rotina já garante um tempo de espera de 1ms. Para os comandos que exigem um tempo maior, um *delay* adicional será dado após a chamada desta rotina.

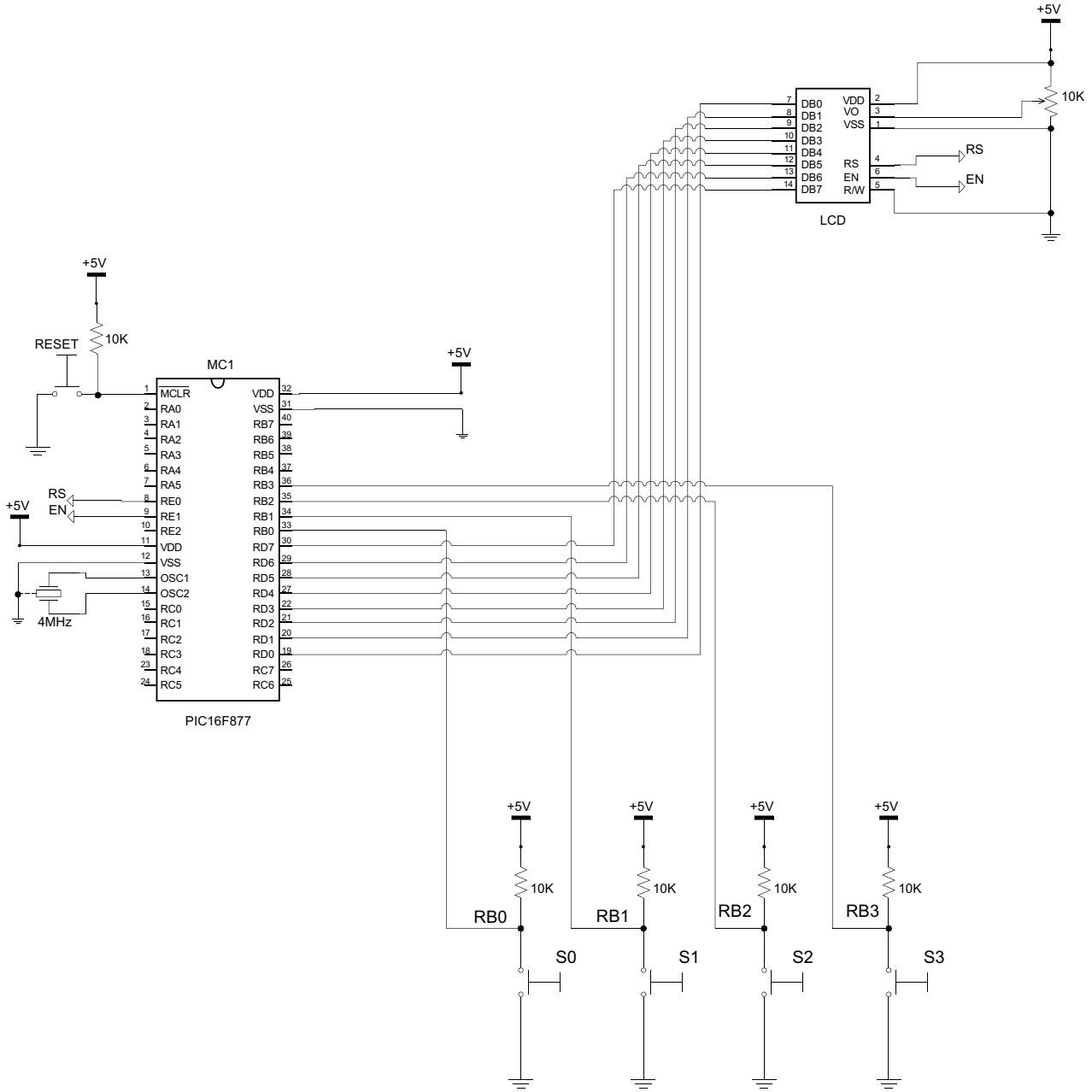
A rotina ESCREVE pode ser usada tanto para enviar comandos quanto dados para o display. Acontece que como esta rotina não verifica ou altera o valor de RS, esta saída deve ser configurada antes da rotina ser chamada.

Seguindo o roteiro descrito na parte teórica desta aula, implementamos também uma rotina de preparação do LCD, chamada INICIALIZACAO_DISPLAY. Esta rotina configura o sistema para comunicação com 8 vias, 2 linhas, sem cursor visível e com movimento automático do cursor para a direita. Além disso, ela já limpa a tela e posiciona o cursor na primeira linha, primeiro caractere da esquerda.

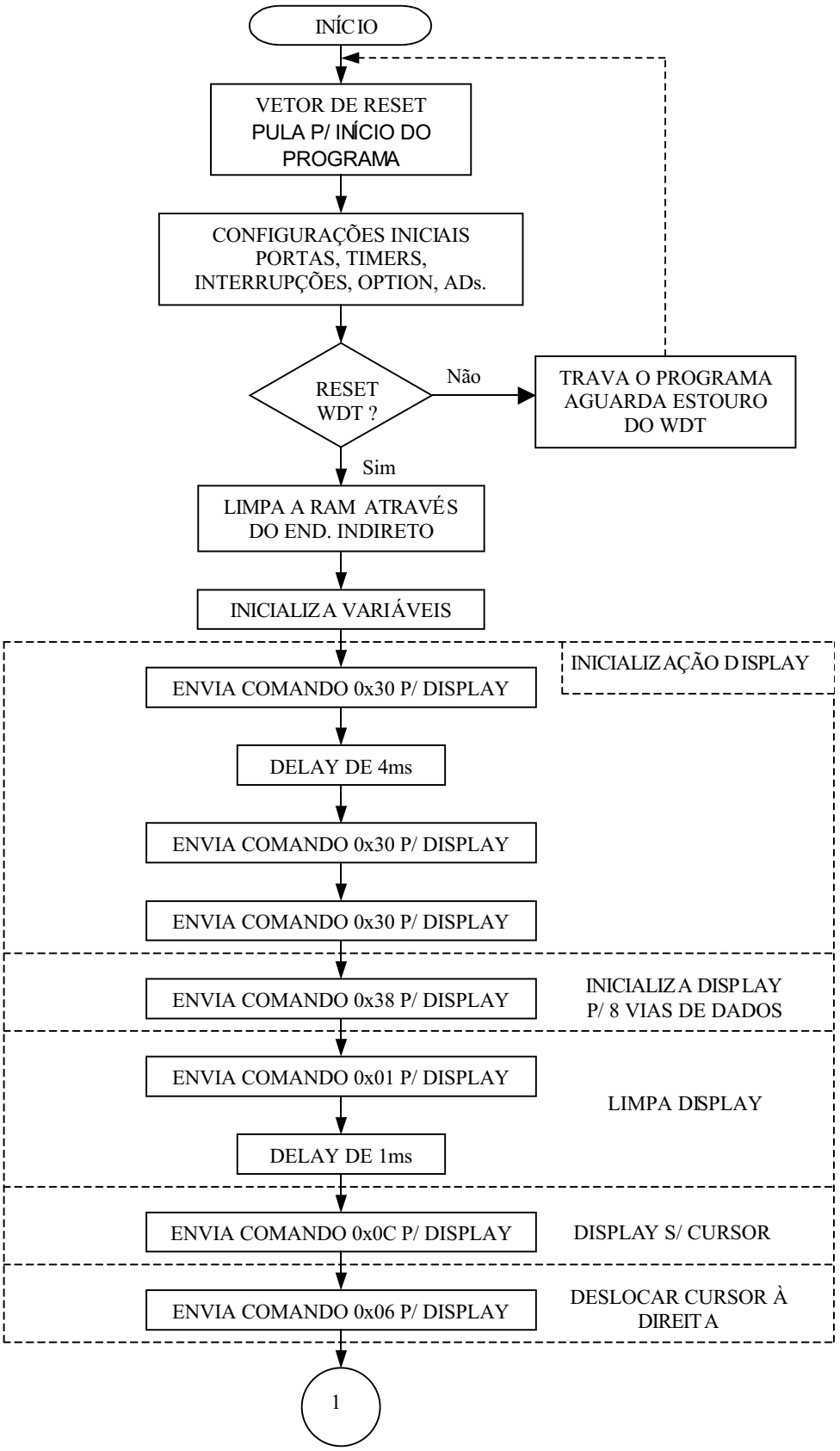
Para cada botão pressionado, posicionamos o cursor em um local diferente da tela e escrevemos o número do botão em questão. Após a liberação, uma tela padrão é utilizada.

Este exemplo não utiliza nenhuma interrupção.

Esquema Elétrico



Fluxograma


INÍCIO
VETOR DE RESET
PULA P/ INÍCIO DO
PROGRAMA
CONFIGURAÇÕES INICIAIS
PORTAS, TIMERS,
INTERRUPÇÕES, OPTION, ADs.
RESET
WDT ?
Não
TRAVA O PROGRAMA
AGUARDA ESTOURO
DO WDT
Sim
LIMPA A RAM ATRAVÉS
DO END. INDIRETO
INICIALIZA VARIÁVEIS
INICIALIZAÇÃO DISPLAY
ENVIA COMANDO 0x30 P/ DISPLAY
DELAY DE 4ms
ENVIA COMANDO 0x30 P/ DISPLAY
ENVIA COMANDO 0x30 P/ DISPLAY
ENVIA COMANDO 0x38 P/ DISPLAY
INICIALIZA DISPLAY
P/ 8 VIAS DE DADOS
ENVIA COMANDO 0x01 P/ DISPLAY
LIMPA DISPLAY
DELAY DE 1ms
ENVIA COMANDO 0x0C P/ DISPLAY
DISPLAY S/ CURSOR
ENVIA COMANDO 0x06 P/ DISPLAY
DESLOCAR CURSOR À
DIREITA
1

Código

```
; * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
; *              CONECTANDO O PIC - EXPLORANDO RECURSOS AVANÇADOS          *
; *                                EXEMPLO 3                               *
; *                                                                        *
; *                CENTRO DE TREINAMENTO - MOSAICO ENGENHARIA              *
; *                                                                        *
; *  TEL: (0XX11) 4992-8775              SITE: www.mosaico-eng.com.br      *
; *       (0XX11) 4992-8003            E-MAIL: mosaico@mosaico-eng.com.br  *
; * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
; *   VERSÃO : 1.0                                                         *
; *     DATA : 10/05/2002                                                  *
; * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

; * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
; *                            DESCRIÇÃO GERAL                             *
; * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
;  ESTE EXEMPLO FOI ELABORADO PARA EXPLICAR O FUNCIONAMENTO DO MÓDULO DE LCD.
;  FOI CRIADA UMA ROTINA PARA ESCREVER COMANDOS OU CACACTRES NO LCD. EXISTE
;  TAMBÉM UMA ROTINA DE INICIALIZAÇÃO NECESSÁRIA PARA A CORRETA CONFIGURAÇÃO
;  DO LCD. OS BOTÕES CONTINUAM SENDO MONITORADOS. UMA MENSAGEM É ESCRITA
;  NO LCD PARA CADA UM DOS BOTÕES, QUANDO O MESMO É PRESSIONADO.
;
; * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
; *                       CONFIGURAÇÕES PARA GRAVAÇÃO                      *
; * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

 __CONFIG _CP_OFF & _CPD_OFF & _DEBUG_OFF & _LVP_OFF & _WRT_ENABLE_OFF & _BODEN_OFF &
_PWRTE_ON & _WDT_ON & _XT_OSC

; * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
; *                          DEFINIÇÃO DAS VARIÁVEIS                       *
; * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
;  ESTE BLOCO DE VARIÁVEIS ESTÁ LOCALIZADO LOGO NO INÍCIO DO BANCO 0

   CBLOCK  0X20                  ; POSIÇÃO INICIAL DA RAM

           TEMPO1
           TEMPO0                ; CONTADORES P/ DELAY

           FILTRO_BOTOES         ; FILTRO PARA RUIDOS

           FLAG                  ; FLAG DE USO GERAL

   ENDC

; * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
; *                DEFINIÇÃO DAS VARIÁVEIS INTERNAS DO PIC                 *
; * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
;  O ARQUIVO DE DEFINIÇÕES DO PIC UTILIZADO DEVE SER REFERENCIADO PARA QUE
;  OS NOMES DEFINIDOS PELA MICROCHIP POSSAM SER UTILIZADOS, SEM A NECESSIDADE
;  DE REDIGITAÇÃO.

   #INCLUDE <P16F877.INC>        ; MICROCONTROLADOR UTILIZADO

; * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
; *                      DEFINIÇÃO DOS BANCOS DE RAM                       *
; * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
;  OS PSEUDOS-COMANDOS "BANK0" E "BANK1", AQUI DEFINIDOS, AJUDAM A COMUTAR
;  ENTRE OS BANCOS DE MEMÓRIA.

#DEFINE   BANK1   BSF     STATUS,RP0      ; SELECIONA BANK1 DA MEMORIA RAM
#DEFINE   BANK0   BCF     STATUS,RP0      ; SELECIONA BANK0 DA MEMORIA RAM
```

```
  MOVLW   0XC8                    ; COMANDO PARA POSICIONAR O CURSOR
  CALL    ESCREVE                 ; LINHA 1 / COLUNA 8
  BSF     RS                      ; SELECIONA O DISPLAY P/ DADOS

                                  ; COMANDOS PARA ESCREVER AS
                                  ; LETRAS DE "TECLA 3"
  MOVLW   'T'
  CALL    ESCREVE
  MOVLW   'E'
  CALL    ESCREVE
  MOVLW   'C'
  CALL    ESCREVE
  MOVLW   'L'
  CALL    ESCREVE
  MOVLW   'A'
  CALL    ESCREVE
  MOVLW   ' '
  CALL    ESCREVE
  MOVLW   '3'
  CALL    ESCREVE

  BSF     TELA_PRINCIPAL          ; SETA FLAG P/ MOSTRAR TELA PRINCIPAL

  GOTO    VARRE                   ; VOLTA P/ VARREDURA DOS BOTÕES

; * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
; *                              FIM DO PROGRAMA                          *
; * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

  END                             ; FIM DO PROGRAMA
```

Dicas e Comentários

Apesar da estrutura do sistema ficar muito simples com a implementação da rotina ESCREVE, nunca se esqueça de confirmar o estado da saída RS antes de utilizá-la. Conforme empregado neste exemplo, recomendamos deixar RS=1 como valor padrão, alterando-o para 0 sempre que desejar escrever algum comando. Em seguida, retornar RS=1 (padrão) para enviar dados.

Para enviar um caractere ao LCD, não se esqueça que deve ser especificado o seu código ASCII. Para facilitar a sua vida, o MpLab efetua esta conversão quando você digita um caractere entre aspas simples (Ex: ‘A’). Neste caso, existe diferença entre maiúsculas e minúsculas. Para caracteres não printáveis, seu código pode ser diretamente definido (Ex: 0x35). O MpLab, assim como a maioria dos programas para Windows, também aceita a especificação de um caractere que não aparece no teclado através da combinação da tecla ALT seguido do código ASCII, imprimindo o resultado na tela. O problema é que nem sempre o código ASCII é equivalente ao mesmo símbolo para o LCD e para a fonte do MpLab.

Apesar da placa McLab2 possuir ligação com o módulo de LCD através de 8 vias de dados é possível utilizá-la para testar e implementar a comunicação com 4 vias. Basta modificar a rotina de inicialização e a de escrita de um byte.

Exercícios Propostos

Agora que o exemplo já foi estudado e esclarecido, aproveite para gerar novos problemas e soluções, seguindo os exercícios propostos:

1. Altere a comunicação para 4 vias;
2. Mantenha a tela principal disponível somente quando o sistema é ligado. Após alguns segundos, mostre uma tela com o nome das quatro teclas e indique a tecla pressionada através de um caractere de seta (?) ou outro qualquer.