

# Primeiro experimento - LEDs

## Material utilizado

Placa, soquetes, cabos jacarés, microcontrolador 8051, leds, pontas de prova para multímetro, multímetro, capacitores, resistores, fonte de tensão, fios.

## Introdução

A primeira prática tinha como objetivo montar um circuito com 4 leds que deveriam acender um após o outro, sendo que, ao chegar no último led, o ciclo deveria reiniciar a partir do primeiro led. Para isso, foi utilizado o microcontrolador 8051 além dos outros componentes citados na seção "Material utilizado".

## Procedimento experimental

De acordo com o esquema fornecido para a realização da prática, montamos os componentes adequados na placa. Após programarmos o microcontrolador, o circuito funcionou conforme os objetivos da prática.

## Código do primeiro experimento

```
$mod51

INICIO1:

mov R0, #1b

INICIO2:

    mov P2, R0

    CPL P2.0
    CPL P2.1
    CPL P2.2
    CPL P2.3

    mov R1, #1010000b

DELAY:

    mov R2, #1010000b

DELAY2:

    mov R3, #111111b

DELAY3:
```

```
    nop
    dec R3
    cjne R3, #0b, DELAY3
    dec R2
    cjne R2, #0b, delay2
    dec R1

cjne R1, #0b, DELAY

    mov A, R0
    r1 A
    mov R0, A

cjne R0, #10000b, inicio2

jmp inicio1

END
```

## Conclusão

O experimento foi realizado com sucesso, com os leds acendendo na ordem correta, conforme proposto para esta prática.

## Segundo experimento – botões

### Material utilizado

Placa, soquetes, cabos jacarés, microcontrolador 8051, leds, pontas de prova para multímetro, multímetro, capacitores, resistores, fonte de tensão, fios, botões

### Introdução

A segunda prática teve como objetivo o uso de 4 botões para acender cada um dos 4 botões inseridos na placa no experimento anterior. Cada led deveria acender enquanto o botão adequado estivesse apertado.

### Procedimento experimental

Acrescentamos botões e os componentes indicados no esquema fornecido para as práticas do laboratório ao projeto montado no primeiro experimento.

## Código do segundo experimento

```
$mod51
inicio:

    jnb P2.4, led1
    jnb P2.5, led2
    jnb P2.6, led3
    jnb P2.7, led4
    jmp inicio

led1: clr P2.0
      setb P2.0
      jmp inicio

led2: clr P2.1
      setb P2.1
      jmp inicio

led3: clr P2.2
      setb P2.2
      jmp inicio

led4: clr P2.3
      setb P2.3
      jmp inicio

END
```

## Conclusão

O experimento foi realizado com sucesso e os botões foram usados para acender os leds correspondentes.

## Terceiro experimento – LCD

### Introdução

O objetivo deste experimento era utilizar um display LCD para exibir o nome da dupla responsável pelo projeto.

### Procedimento experimental

Acrescentamos o LCD ao circuito da prática anterior. Além disso, acrescentamos os componentes necessários ao funcionamento do

circuito, tais como o potenciômetro para regular a intensidade do display LCD.

## Código do terceiro experimento

```
$mod51

ORG 0000H
ADC_DATA EQU P0                ; Define Name to Port Pins
ADC_EOC BIT P3.2
ADC_ALE_SC BIT P3.4
ADC_OE BIT P3.5
EN EQU P3.7
RS EQU P3.6
LCD EQU P1

                MOV ADC_DATA,#0FFH    ;Assign port 1 as I/P port

                ACALL DELAY

MAIN:
                ACall ADC_CON
                MOV P1,A
                SJMP MAIN

;ADC Programming Start

ADC_CON:
                SETB ADC_EOC            ; Make EOC AS Input Port
                CLR ADC_ALE_SC
                CLR ADC_OE

BACK:
                SETB ADC_ALE_SC        ; Give High to Low Pulse to Latch Address
and for Start of Conversion
                ACALL DELAY
                CLR ADC_ALE_SC

HERE12:
                JB ADC_EOC,HERE12      ;Wait till ADC Complete Its Task
HERE11:
                JNB ADC_EOC,HERE11     ;After ADC Put Data To Output Signal Goes
High Again
                SETB ADC_OE            ; Set OE High For converted Data to
Available on Controller,
                ACALL DELAY            ; For Further Operation

                MOV A,ADC_DATA         ;Load Data To Acc.
                CLR ADC_OE             ; Save Converted Digital Data on Memory

                RET

; Delay
; App. 243 µ Sec.
```

```

DELAY:
    MOV R3,#1
HERE1:
    MOV R1,#10
HERE2:
    MOV R2,#10
HERE3:
    DJNZ R2,HERE3
    DJNZ R1,HERE2
    DJNZ R3,HERE1
    RET

;-----

ORG 00H

ACALL INICIALIZAR_LCD

    MOV R1,#1D
    MOV R7,#0D
    MOV DPTR,#NOME1
ESCREVENDO1:
    MOV A, R7
    MOVC A, @A + DPTR ; MOVE O CONTEÚDO DO ENDEREÇO QUE ESTÁ SENDO PASSADO
    PARA O ACUMULADOR
    INC R7
    CJNE A, #0,EX_ESCRITA ;ESCREVE UMA LETRA NO LCD

;PULANDO LINHA
    CLR RS
    MOV LCD, #0C0H
    SETB EN
    CLR EN
    LCALL TEMPO

    MOV R7,#0H
    MOV DPTR,#NOME2
ESCREVENDO2:
    MOV A, R7
    MOVC A, @A + DPTR
    INC R7
    CJNE A,#0,EX_ESCRITA2

LOOP:
    JMP LOOP

EX_ESCRITA:
    CLR EN
    SETB RS
    MOV LCD,A
    SETB EN
    CLR EN
    LCALL TEMPO
    JMP ESCREVENDO1

EX_ESCRITA2:
    CLR EN
    SETB RS
    MOV LCD,A
    SETB EN
    CLR EN
    LCALL TEMPO

```

```

        JMP ESCRIVENDO2

;*****
;FUNÇÃO PARA CONFIGURAR O LCD PARA UM DETERMINADO MODO DE OPERAÇÃO
INICIALIZAR_LCD:
    SETB EN ;HABILITA O LCD
    CLR RS ;AJUSTE PARA PASSAR INSTRUÇÃO
    MOV LCD, #38H ; configura para 8 bits a duas linhas
                    ; e a matriz de caracter para 5x7
    CLR EN ;DESABILITA O LCD

    ;TEMPO PARA O LCD SE AJUSTAR AO COMANDO
    LCALL TEMPO

    SETB EN ;HABILITA O LCD
    CLR RS ;AJUSTE PARA PASSAR INSTRUÇÃO
    MOV LCD, #38H ; configura para 8 bits a duas linhas
                    ; e a matriz de caracter para 5x7
    CLR EN ;DESABILITA O LCD

    ;TEMPO PARA O LCD SE AJUSTAR AO COMANDO
    LCALL TEMPO

    SETB EN
    CLR RS
    MOV LCD, #0FH ; CURSOR PISCANTE
    CLR EN

    LCALL TEMPO

    SETB EN
    CLR RS
    MOV LCD, #06H ;CURSOR COM AUTO INCREMENTO PARA A DIREITA
    CLR EN

    LCALL TEMPO

    SETB EN
    CLR RS
    MOV LCD, #01H ;LIMPA O LCD E COLOCA O CURSOR NA PRIMEIRA POSIÇÃO DO LCD
    CLR EN

    LCALL TEMPO

RET

;*****
; ATRASO DE 2ms
TEMPO:
    MOV R0, #250D
    MOV R1, #4D
CONT: DJNZ R0, CONT
        MOV R0, #248D
        DJNZ R1,CONT
RET
;*****

;NOMES PARA SEREM ESCRITOS
NOME1:
    DB 'FELIPE RIBAS',0
NOME2:

```

```
DB 'IAGO ALEKSANDER',0
END
```

## Quarto experimento – conversor AD

### Introdução

Neste experimento o objetivo era inserir no circuito um conversor AD para exibir dígitos de 0 a 9 conforme a leitura do potenciômetro.

### Procedimento Experimental

Após inserirmos o microcontrolador responsável pela conversão AD, inserimos o potenciômetro usado para indicar o número que deve ser exibido no LCD. Componentes como resistores e capacitores também foram inseridos na placa conforme esquema fornecido em apresentação para esta prática.

### Código do quarto experimento

```
$mod51

ADC_DATA EQU P0                ; Define Name to Port Pins
ADC_EOC BIT P3.2
ADC_ALE_SC BIT P3.4
ADC_OE BIT P3.5

LCD EQU P1
EN EQU P3.7
RS EQU P3.6

ORG 0000H
    SJMP INICIO

ORG 0Bh
    cpl P3.1                ;complementa o clock (P3.1)
    reti

ORG 0030H
INICIO:
    MOV ADC_DATA,#0FFH      ;Assign port 1 as I/P port
    ACall INICIALIZAR_LCD
    ACALL DELAY

    ;habilitacao das interrupcoes
    mov ie,#10000011b      ;habilitacao da interrupcao INT0 e do timer0
    mov ip,#00000010b      ;prioridade maxima para interrupcao do timer0
```

```

        ;setb it0

        ;timer_0 configurado para gerar interrupcao, clock interno e modo auto-
reload
        mov tmod,#00001010b
        mov tl0,#0FDh      ;contagem de apenas 2us
        mov th0,#0FDh      ;valor de reload para o timer
        mov r0,#030h        ;move para r0 30h
        setb tr0            ;inicia contagem de tempo, alternancia do sinal a cada
1us

```

```

;*****
MAIN:

```

```

        ACALL ADC_CON
        ACALL DATA_LCD
        SJMP MAIN

```

```

;*****
;ADC Programming Start

```

```

ADC_CON:
        SETB ADC_EOC          ; Make EOC AS Input Port
        CLR ADC_ALE_SC
        CLR ADC_OE

BACK:
        SETB ADC_ALE_SC      ; Give High to Low Pulse to Latch Address
and for Start of Conversion
        ACALL DELAY
        CLR ADC_ALE_SC

HERE12:
        JB ADC_EOC,HERE12    ; Wait till ADC Complete Its Task

HERE11:
        JNB ADC_EOC,HERE11   ; After ADC Put Data To Output Signal Goes
High Again
        SETB ADC_OE          ; Set OE High For converted Data to
Available on Controller,
        ACALL DELAY          ; For Further Operation

        MOV A,ADC_DATA       ; Load Data To Acc.
        CLR ADC_OE           ; Save Converted Digital Data on Memory

        RET

```

```

;*****
;inicializar lcd e configura-lo

```

```

INICIALIZAR_LCD:

        SETB EN              ; habilita o lcd
        CLR RS               ; ajuste para passar instrucao
        MOV LCD, #38H        ; configura para 8 bits a duas linhas
                                ; e a matriz de caracter para 5x7

        CLR EN               ;desabilita o lcd

        LCALL TEMPO          ;a short delay

        SETB EN

```



```

        CLR      RS
        MOV LCD, #38H

        CLR EN

        LCALL TEMPO

        SETB EN
        CLR RS
        MOV LCD, #0EH      ; modo cursor piscante
        CLR EN

        LCALL TEMPO

        SETB EN
        CLR RS
        MOV LCD, #06H      ; modo cursor com auto incremento para a direita
        CLR EN

        LCALL TEMPO

        SETB EN
        CLR RS
        MOV LCD, #01H      ; limpa o lcd e coloca o cursor na primeira posicao
        CLR EN

        LCALL TEMPO

RET

;*****

; App. 243 µ Sec.

DELAY:
        MOV R3,#1
HERE1:
        MOV R1,#10
HERE2:
        MOV R2,#10
HERE3:
        DJNZ R2,HERE3
        DJNZ R1,HERE2
        DJNZ R3,HERE1

        RET

;*****
; atraso de 2ms

TEMPO:
        MOV R0, #250D
        MOV R1, #4D
CONT:   DJNZ R0, CONT
        MOV R0, #248D
        DJNZ R1,CONT

        RET

;*****
; limpar tela

```

DATA\_LCD:

```
SETB EN
CLR RS
MOV LCD, #01H      ; limpa o lcd e coloca o cursor na primeira posicao
CLR EN
```

;\*\*\*\*\*

; converter dados lidos em binario para decimal e transferir para o lcd (faixa de valores de 0 a 9)

BINARY\_TO\_BCD:

```
SETB RS      ;Telling the LCD that the data which is being send is to
be displayed
```

;tabela de enderecos para os numeros

```
MOV 030h,30      ;0
MOV 031h,31      ;1
MOV 032h,32      ;2
MOV 033h,33      ;3
MOV 034h,34      ;4
MOV 035h,35      ;5
MOV 036h,36      ;6
MOV 037h,37      ;7
MOV 038h,38      ;8
MOV 039h,39      ;9
```

```
MOV R6,#030h ;move para R6 30h
MOV R5, #00h ;contador inicia com o numero 0
```

VERIFICA:

```
SUBB A, #26 ;subtrair acumulador de 26 com o limite inferior
JC FINISH
INC R5
SJMP VERIFICA
```

FINISH:

```
;soma número desejado + endereço base da tabela de número
mov A, R6
add A, R5
```

```
MOV P1, A
SETB EN
ACALL TEMPO
CLR EN
```

```
MOV R7, 6
```

ESPERA:

```
ACALL TEMPO
DJNZ R7, ESPERA
```

;\*\*\*\*\*

; converter dados lidos em binario para decimal e transferir para o lcd (de 0 a 255 - nao usado)

```
;      MOV B, #64h      ; 100
```

```

;      DIV AB          ;/ 100
;      MOV R5, A      ; store cents in R5
;      MOV A, B        ; get remainder
;      MOV B, #0AH     ; 10
;      DIV AB          ; / 10
;      MOV R7, A      ; save tens in R7
;      MOV A, B        ; get units
;      MOV R6,A        ; save units in R6

;      MOV A, R5
;      ADD A, #30h

;      MOV P1, A
;      SETB EN         ;EN is High
;      ACALL TEMPO      ;a short delay
;      CLR EN          ;EN is low to make a high-low pulse

;      MOV A, R7
;      ADD A, #30h

;      MOV P1, A
;      SETB EN
;      ACALL TEMPO
;      CLR EN

;      MOV A, R6
;      ADD A, #30h

RET          ; return to main

END

```

## Conclusão

O conversor DA funcionou conforme especificado nos objetivos do experimento. Após alterações realizadas no código, a faixa de valores obtida e exibida no display foi configurada para exibir números de 0 a 9 (pois, no projeto original, a escala era maior, variando até 256).

## Projeto final – Genius

### Introdução

O nosso projeto do laboratório de Microcontroladores e Aplicações foi baseado no jogo Genius, com quatro leds e quatro botões. Os botões são usados para o usuário inserir sequências, sendo que essas sequências têm a validade checada. Os leds são

usados para exibir certa sequência, que o usuário deve memorizar a fim de inserir corretamente no momento adequado. O LCD é usado para informar se uma sequência inserida pelo usuário é válida ou não, se o usuário venceu ou perdeu e o número de sequências inseridas.

### **Procedimento experimental**

O microcontrolador 8051 foi programado de forma a enviar as mensagens adequadas para o LCD, sendo que o LCD é a forma de o usuário saber o status dele no jogo (se acertou a sequência, quantas sequências seguidas foram inseridas corretamente, se o botão "start" deve ser pressionado para iniciar o jogo ou se o usuário errou e perdeu o jogo). Quatro LEDS foram inseridos no circuito para mostrar as sequências que deveriam ser inseridas pelo usuário, sendo que tal inserção era feita por quatro botões que também foram adequadamente posicionados na placa. O circuito deste projeto final ficou bem semelhante, fisicamente, ao que foi montado até a terceira prática, com a diferença do código utilizado no LCD e, também, com a diferença de que os botões e LEDS foram utilizados em conjunto com o LCD para que o jogo pudesse ser elaborado.

### **Código projeto final**

