Microcontroladores 8051

EXERCÍCIOS RESOLVIDOS

Modos de endereçamento direto

LABEL	INSTR	OPERANDO	COMENTÁRIOS
$include(REG51.inc)			; inclui no programa assembly o arquivo (REG51.inc)
	ORG	0000H	; o programa inicia na linha 0000H da EPROM
	MOV	A,#01010101B	; carrego no ACC por binário o valor 01010101B ou 55h.
	MOV	A,#HIGH (0FH)	; carrego A com a parte alta de uma variável de 16 bits, ou seja, carrego os 8 últimos bits
	MOV	B,#LOW(0FH)	; carrego B com a parte baixa de uma variável de 16 bits, ; ou seja, carrego os 8 primeiros bits
	MOV	B,#HIGH(65535	; carrego B com a parte alta de uma variável de 16 bits, ou ; seja, carrego os 8 últimos bits de 65535)=FFH
	MOV	A,#LOW(65535)	; carrego A com a parte baixa de uma variável de 16 bits, ; ou seja, carrego os 8 últimos bits de 65535)=FFH
	MOV	A,#(255-250)	; carrego em A a diferença da operação 255-250 = 5 = 05H
	MOV	A,#HIGH(255-240)	; carrego em A a parte alta da operação (15=0FH) que é ; 0FH
	MOV	B,#LOW(255-240)	; carrego em B a parte baixa da operação (15=0FH), que é ; 0FH.
	MOV	A,#'B'	; carrego em A o valor do código ASCII da letra B, que é ; 66 em decimal, ou 42H (em hexa)
	MOV	A,#'C'	; carrego em A o valor do código ASCII da letra C, que é ; 67 em decimal ou 43H.
	MOV	A,#0FFH	; carrego no acumulador A(ACC) o valor 11111111B
	MOV	A,#00H	; carrego no acumulador A(ACC) o valor 00000000B
	MOV	B,#0F0H	; carrego no registrador B o valor 11110000B
	MOV	R0,#0F0H	; carrego no registrador R0 o valor 11110000B
	MOV	A,#192	; carrego no ACC, por decimal o valor 1000000B ou 0C0H
	SJMP	$	; o programa fica parado nesta instrução
	END		

Programa com uso do recurso label

LABEL	INSTR	OPERANDO	COMENTÁRIOS
$include(REG51.inc)			; inclui no programa assembly o arquivo (REG51.inc)
	ORG	0000H	; programa começa no 0000H da Eprom (memória de ; programa)
	MOV	A,#0FFH	; escrevo no acumulador (ACC) o valor 0FFH
SALTO:	DEC	A	; decrementa A
	MOV	P2,A	; carrego o valor de A para porta P2
	SJMP	SALTO	; salto para o endereço dado pelo nome (LABEL) “SALTO”
	END		

Programa com instrução que utiliza o registrador DPTR

LABEL	INSTR	OPERANDO	COMENTÁRIOS
$include(REG51.inc)			; inclui no programa assembly o arquivo (REG51.inc)
	ORG	0000H	; inicia em 0000H
INICIO:	MOV	DPTR,#0050H	; carrega o valor 0050h no registrador DPTR
	MOV	A,#00H	; carrega o ACC com 00H
	JMP	@A+DPTR	; o programa salta para o endereço de DPTR+@A = DPTR
	ORG	0050H	; inicia na posição 0050H
	MOV	DPH,#00H	; nesta posição de memória de programa(0050h), escrevo ; a instrução para carregar o valor 0030h para o ; registrador DPTR por meio de DPH e DPL, isto é, por ; partes de 8 bits.
	MOV	DPL,#30H	
	JMP	@A+DPTR	; salta para a posição de endereço 0030h
	ORG	0030H	; programa está agora na posição 0030h.
	LJMP	INICIO	; “long jump” salta o programa para label inicio.
	END		

Programa com o uso de interrupcões

LABEL	INSTR	OPERANDO	COMENTÁRIOS
$include(REG51.inc)			; inclui no programa assembly o arquivo (REG51.inc)
SW3	EQU	P3.2	; SW3 igual a P3.2
	ORG	0000H	; inicia em 0000H (reset)
	LJMP	INICIO	; salta para rotina inicio
	ORG	0003H	; INT0 (EX0) – serviço de interrupção de INT0.
	LJMP	INT_0	
	ORG	0100H	; inicia no end 0100H
INICIO:	MOV	A,#00000001B	; move para acumulador o valor 00000001H
	CLR	C	; zera a flag de carry
	MOV	P2,A	; move o conteúdo de A para P2
	SETB	EX0	; habilita a flag de interrupção INT0
	SETB	EA	; habilita todas as interrupções
	LJMP	$	; o programa fica parado nesta instrução
; rotina para atender a interrupção INT_0			
INT_0:	RLC	A	; rodo o conteúdo do acumulador mais o carry à esquerda
	MOV	P2,A	; move o conteúdo do acumulador para a porta P2
TECLA:	JNB	SW3,TECLA	; salto condicionalmente para o Label Tecla se o bit SW3 ; for nível lógico zero
	RETI		; retorno de interrupção
	END		

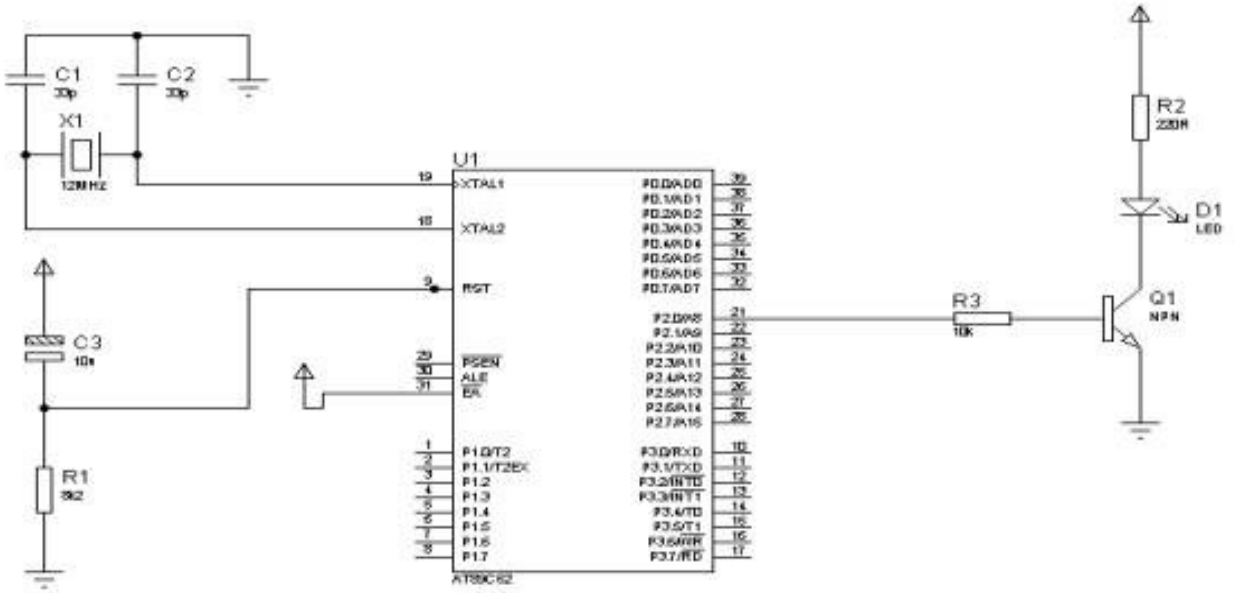
Experiência 01: pisca-pisca com led

Objetivo

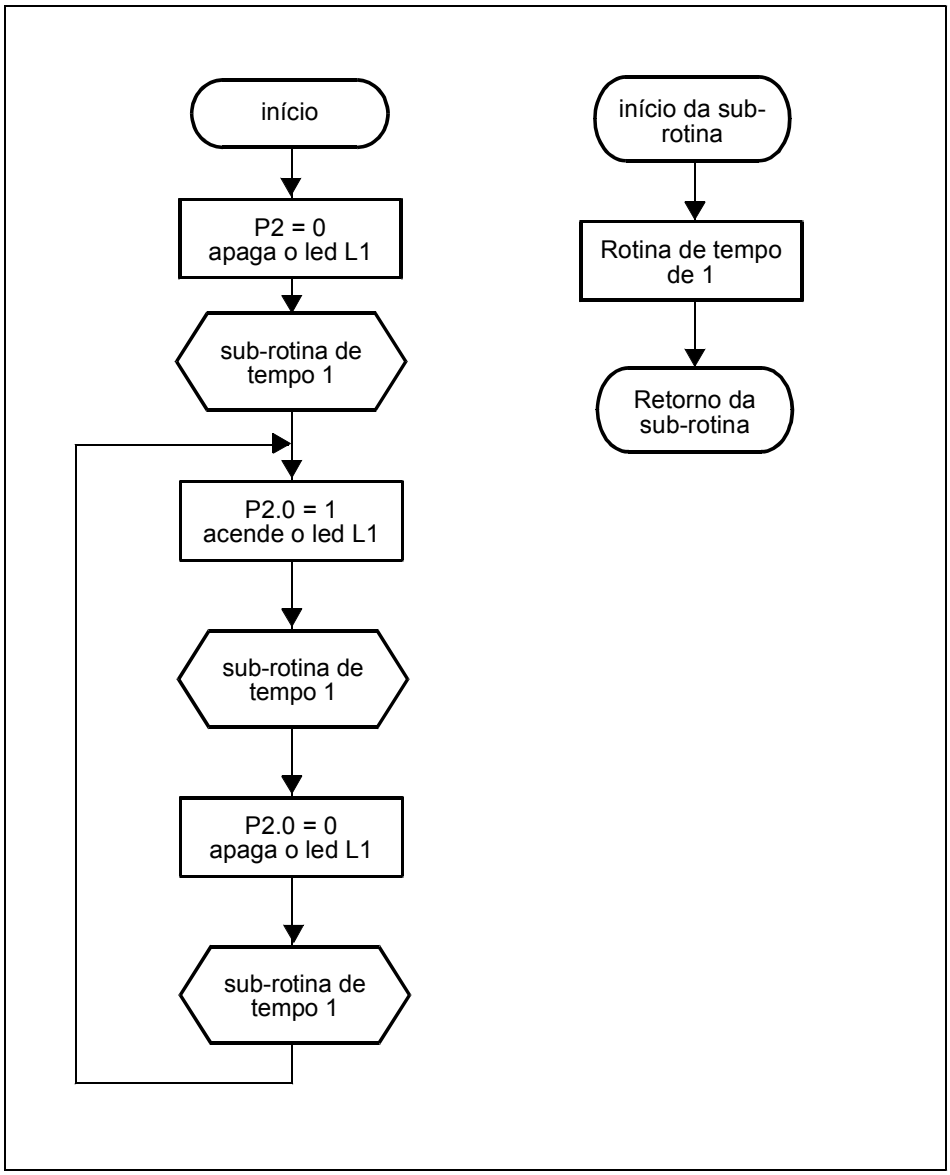
Montar um circuito que consiste em acender e apagar um led continuamente, com o tempo de 1 segundo entre acender e apagar.

É importante analisar o esquema elétrico para fazer um programa que seja compatível com o hardware desejado.

Esquema elétrico



Fluxograma


início
P2 = 0
apaga o led L1
sub-rotina de
tempo 1
P2.0 = 1
acende o led L1
sub-rotina de
tempo 1
P2.0 = 0
apaga o led L1
sub-rotina de
tempo 1
início da sub-
rotina
Rotina de tempo
de 1
Retorno da
sub-rotina

Programa

LABEL	INSTR	OPERANDO	COMENTÁRIOS
$include(REG51.inc)			; inclui no programa assembly o arquivo (REG51.inc)
TEMPO	EQU	19702	; tempo de 50ms para cristal de 11 MHz
	ORG	0000H	; o programa começa na linha 0000h da EPROM ; (memória de programa).
	MOV	P2,#00H	; escrevo na porta P2 o valor 00000000B ou 00H, ou ; seja, zero a porta P2
PRINCIPAL:	LCALL	TEMPO_1S	; chamo sub-rotina de tempo de 1 segundo
	SETB	P2.0	; liga o bit P2.0 da porta P2
	LCALL	TEMPO_1S	; chamo sub-rotina de tempo de 1 segundo
	CLR	P2.0	; desliga o bit P2.0 da porta P2
	LCALL	TEMPO_1S	; chamo sub-rotina de tempo de 1 segundo
	LJMP	PRINCIPAL	; salta para o label principal
; rotina de tempo de 1 segundo, que utiliza o timer 0 do Microcontrolador 8051			
TEMPO_1S:	MOV	R0,#20H	; carrega Reg. R0 com numero 20 em hexa
	CLR	TR0	; desliga timer 0
CONT:	CLR	TF0	; reseta flag TF0
	MOV	TL0,#LOW(TEMPO)	; carrega parte baixa do timer 0
	MOV	TH0,#HIGH(TEMPO)	; carrega parte alta do timer 0
	SETB	TR0	; liga timer 0
	JNB	TF0,$	; espera o estouro do timer 0.
	CLR	TR0	; desliga timer 0
	DJNZ	R0,CONT	; decrementa e retorna se Reg. não, zero
	RET		; retorno da sub-rotina
	END		; fim do programa

Experiência 02 – seqüencial com led

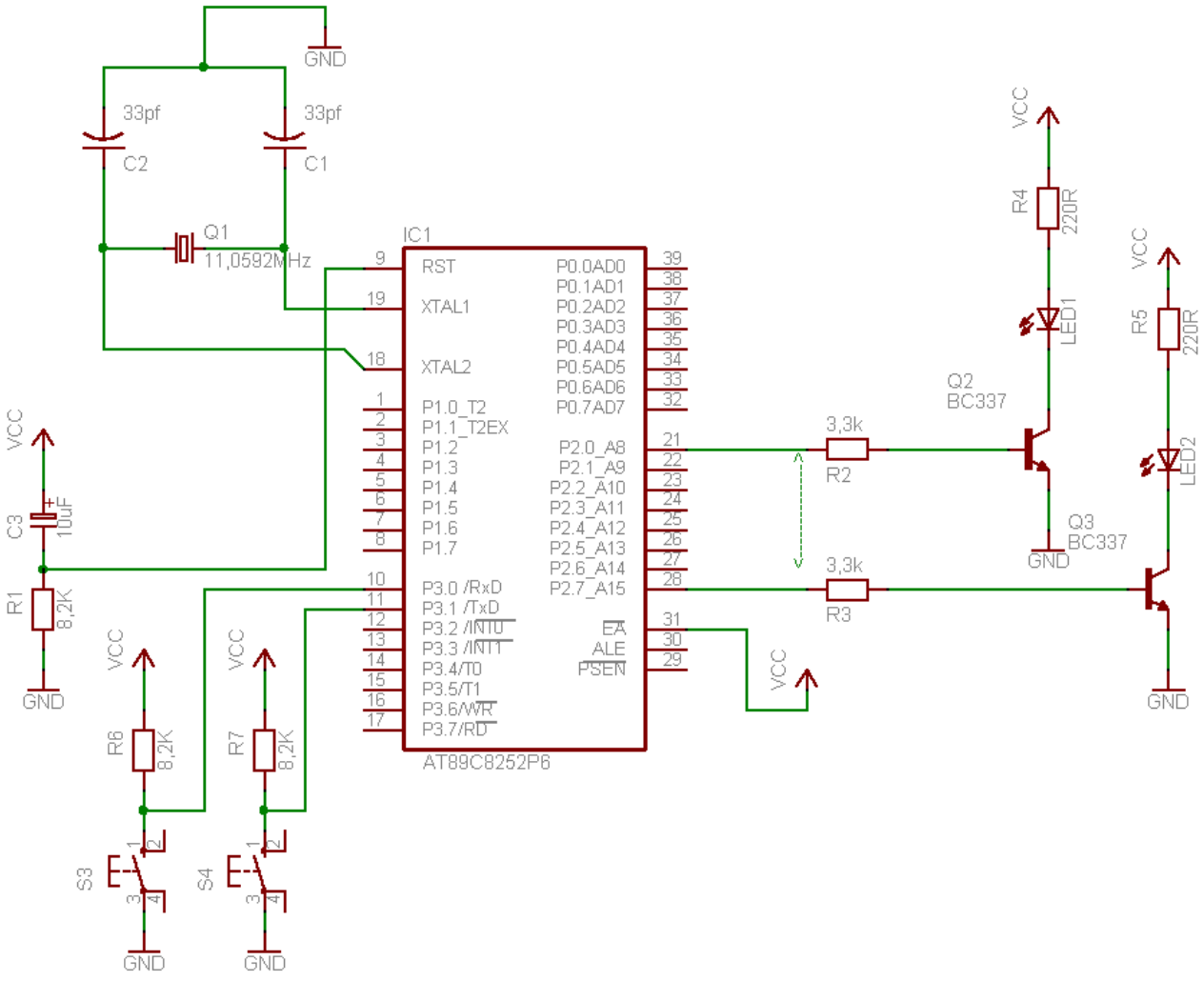
Objetivo

Montar um circuito que consiste em acender sequencialmente 8 leds, com o tempo de 1 segundo entre acender e apagar cada led. Há, também, dois botões com a função de determinar o sentido de rotação dos leds como segue abaixo.

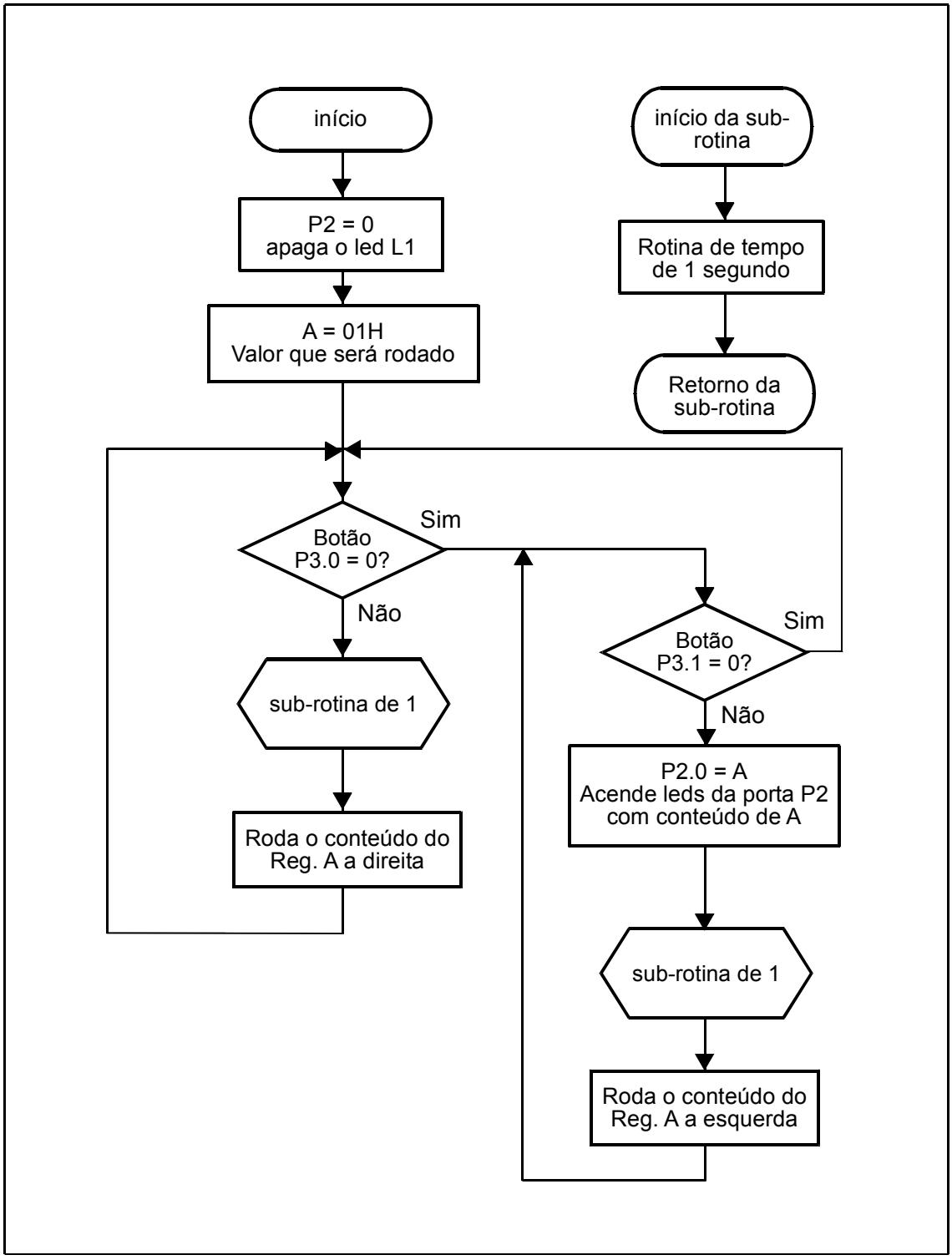
- Botão 1: quando acionado roda a seqüência de led para a esquerda.
- Botão 2: quando acionado roda a seqüência de led para a direita.

É importante analisar o esquema elétrico para fazer um programa que seja compatível com o hardware onde ele vai ser utilizado.

Esquema elétrico



Fluxograma


início
P2 = 0
apaga o led L1
A = 01H
Valor que será rodado
Botão
P3.0 = 0?
Sim
Não
sub-rotina de 1
Roda o conteúdo do
Reg. A a direita
Botão
P3.1 = 0?
Sim
Não
P2.0 = A
Acende leds da porta P2
com conteúdo de A
sub-rotina de 1
Roda o conteúdo do
Reg. A a esquerda
início da sub-
rotina
Rotina de tempo
de 1 segundo
Retorno da
sub-rotina

Programa

LABEL	INSTR	OPERANDO	COMENTÁRIOS
$include(REG51.inc)			; inclui no programa assembly o arquivo ; (REG51.inc)
TEMPO	EQU	19702	; tempo de 50ms para cristal de 11 MHz
	ORG	0000H	;o programa começa na linha 0000H da EPROM ; (memória de programa).
	MOV	P2,#00H	; zera a porta P2 ou seja apaga todos os leds
	MOV	A,#01H	; carrega o acumulador com valor 01H
DIREITA:	JNB	P3.0,ESQUERDA	; testa a porta P3.7 se for nível lógico 1, salta para o ; label ESQUERDA .
	MOV	P2,A	; move para a porta P2 o conteúdo do acumulador
	LCALL	TEMPO_1S	; chama sub-rotina tempo de um segundo
	RR	A	; roda para direita o conteúdo do acumulador
	LJMP	DIREITA	; salta para o label direita
ESQUERDA:	JNB	P3.1,DIREITA	; testa a porta P3.6 se for nível lógico 1, salta para o ; label DIREITA .
	MOV	P2,A	; move para a porta P2 o conteúdo do acumulador
	LCALL	TEMPO_1S	; chama sub-rotina tempo de um segundo
	RL	A	; roda para esquerda o conteúdo do acumulador
	JMP	ESQUERDA	; salta para o label esquerda
; rotina de tempo de 1 segundo, que utiliza o timer 0 do Microcontrolador 8051			
TEMPO_1S:	MOV	R0,#20H	; carrega Reg. R0 com numero 20 em hexa
	CLR	TR0	; desliga timer 0
CONT:	CLR	TF0	; reseta flag TF0
	MOV	TL0,#LOW(TEMPO)	; carrega o TL0 (com a parte baixa de TEMPO = ; 4CF6H) = F6H
	MOV	TH0,#HIGH(TEMPO)	; carrega o TH0 (com a parte alta de TEMPO = ; 4CF6H) = 4CH
	SETB	TR0	; liga timer 0
	JNB	TF0,$	; espera o estouro do timer 0.
	CLR	TR0	; desliga timer 0
	DJNZ	R0,CONT	; decrementa e retorna se Reg. não, zero
	RET		; retorno da sub-rotina
	END		; fim do programa

Programa com uso da instrução CPL

Compile o programa no RIDE51 e simule no modo debug e com janelas **Port 0 e Port 2** abertas e analise cada instrução do programa no modo passo a passo.

LABEL	INSTR	OPERANDO	COMENTÁRIOS
$include(REG51.inc)			; inclui no programa assembly o arquivo ; (REG51.inc)
	ORG	0000H	; o programa começa na linha 0000H da EPROM ; (memória de programa).
BOTAO:	JB	P0.0,BOTAO	; testa o botão e se for nível lógico zero, vai para ; próxima linha
	CPL	P2.0	; complementa a porta P2.0, ou seja inverte a ; condição anterior. Se P2.0 estiver com nível lógico ; 1 vai para nível lógico 0
	JNB	P0.0,BOTAO	; testa o botão e se for nível lógico um, vai para ; próxima linha
	LJMP	BOTAO	; salto incondicional para o label botao
	END		; fim do programa

Mapa da eprom (memoria de programa interna) do programa no formato intel hex

:0B0000002080FDB2A03080FD02000057

:00000001F