c) Sempre que o período da onda quadrada for mudado, envie através da porta série (9600 baud, 1start bit, 1 stop bit, 8 bits, sem paridade) o novo valor do período da onda usando a interrupção da porta serie.

BAUDRATE EQU 0FDH

CSEG AT 0H

JMP MAIN

CSEG AT 03H

JMP ISR\_EXT0

CSEG AT 2BH

JMP ISR\_TIMER2

CSEG AT 50H

MAIN:

;INICIALIZAR GPIO

MOV P0, #0FFH

CLR P2.0

;CONFIGURAR TIMER 2

CALL CONFIG\_TIMER2

CALL CONFIG\_INT

CALL CONFIG\_SERIE

SETB ET2

SETB EA

MAINLOOP:

JMP MAINLOOP

CONFIG\_TIMER2:

MOV T2CON, #4

MOV RCAP2H, #0FFH

MOV A, P0

CPL A

INC A

MOV RCAP2L, A

MOV TL2, A

MOV TH2, RCAP2H

ISR\_TIMER2:

CPL P2.0

CLR TF2

RETI

CONFIG\_INT:

SETB P3.2

SETB IT0

SETB EX0

RET

ISR\_EXT0:

MOV A, P0

RR A

CPL A

INC A

MOV RCAP2L, A

MOV SBUF, A

JNB TI, $

CLR TI

RETI

CONFIG\_SERIE:

CLR ES

MOV TMOD, #20H

MOV TL1, #BAUDRATE

MOV TH1, #BAUDRATE

SETB TR1

MOV SCON, #50H

RET

END

a)Gerar uma onda quadrada em P2.0 cujo período é dado á entrada de P0. Usando o timer2 crie um programa que leia o valor de P0.

CSEG AT 0H

JMP MAIN

CSEG AT 2BH

JMP ISR\_TIMER2

CSEG AT 50H

MAIN:

;INICIALIZAR GPIO

MOV P0, #0FFH

CLR P2.0

;CONFIGURAR TIMER 2

MOV T2CON, #4

MOV RCAP2H, #0FFH

MOV A, P0

CPL A

INC A

MOV RCAP2L, A

MOV TL2, A

MOV TH2, RCAP2H

SETB ET2

SETB EA

MAINLOOP: MOV A, P0

RR A

CPL A

INC A

MOV RCAP2L, A

JMP MAINLOOP

ISR\_TIMER2: CPL P2.0

CLR TF2

RETI

END

b) Ler o valor de P0 apenas quando P3.2 for ativado(transição negativa do sinal no pino de entrada da interrupção externa 0).

CSEG AT 0H

JMP MAIN ;resposta ao reset

CSEG AT 03H

JMP ISR\_EXT0

CSEG AT 2BH

JMP ISR\_TIMER2

CSEG AT 50H

MAIN: ;INICIALIZAR GPIO

MOV P0, #0FFH

CLR P2.0

;CONFIGURAR TIMER 2

CALL CONFIG\_TIMER2

CALL CONFIG\_INT

SETB ET2

SETB EA

MAINLOOP: JMP MAINLOOP

CONFIG\_TIMER2:MOV T2CON, #4

MOV RCAP2H, #0FFH

MOV A, P0

CPL A

INC A

MOV RCAP2L, A

MOV TL2, A

MOV TH2, RCAP2H

ISR\_TIMER2: CPL P2.0

CLR TF2

RETI

CONFIG\_INT:SETB P3.2

SETB IT0

SETB EX0

RET

ISR\_EXT0: MOV A, P0

RR A

CPL A

INC A

MOV RCAP2L, A

RETI

END

#include <89c51ic2.inc>

// Resolução GUIA7 - ISR'S, PORTA-SÉRIE.

//Rotina usando o timer2 em auto-reload que gere isr de 1ms

//na rotina isr gerar uma onda quadrada no pino p2.0

//SEMPRE que ocorram 250 isr\_s deve ser apresentado no display uma contagem de 0 até 9

//Aqui está resolvida a alinea 1.

CSEG AT 002BH //Isr timer2

JMP ISR\_T2

CSEG AT 30H

JMP INICIO

TABLE:

DB 0CH,0CFH,54H,46H,87H,26H,24H,4FH,4H,6H

DISPLAY:

MOV DPTR,#TABLE

MOVC A,@A+DPTR

MOV P1,A

FIM: RET

INICIO:

MOV R0,#0

MAIN:

MOV T2CON,#04H //Timer 2 auto-reload

MOV TH2,#HIGH(-1000)

MOV TL2,#LOW(-1000)

MOV RCAP2L,#LOW(-1000)

MOV RCAP2H,#HIGH(-1000)

SETB TR2

SETB EA //HABILITA TODAS AS ISRS

SETB ET2 //HABILITA ISR TIMER2

MOV R3,#250

JNB TF2,$

ISR\_T2:

DJNZ R3,FIM\_ISRT2

CALL LOOP //Rotina para imprimir os 9 numeros.

CPL P2.0 //Comuta o pino

MOV R3,#250 //Assegurar que não se perde o valor de r3

CLR TF2 //Limpa flag do timer para preparar para nova isr.

LOOP:

MOV A,@R0

CALL DISPLAY

INC R0

CJNE R0,#9,LOOP

RET

FIM\_ISRT2:

RETI

END

#include <89c51ic2.inc>

CSEG AT 0023H //Vetor de interrupção a porta série

JMP ISR\_SP

CSEG AT 002BH

JMP ISR\_T2

//Timer a gerar onda quadrada e etc ex1

TABLE2:

DB "P130P"

TABLE:

DB 0CH,0CFH,54H,46H,87H,26H,24H,4FH,4H,6H

DISPLAY:

MOV DPTR,#TABLE

MOVC A,@A+DPTR

MOV P1,A

FIM: RET

INICIO:

MOV R0,#0

MAIN:

MOV T2CON,#04H //Timer 2 auto-reload

;MOV TH2,#HIGH(-1000)

;MOV TL2,#LOW(-1000)

;MOV RCAP2L,#LOW(-1000)

;MOV RCAP2H,#HIGH(-1000)

MOV TH1,#0FDH //Gerador interno de baud-rate para 9600bps

MOV SCON,#50H //Modo - 1 com REN = 1 , receçao ativa

SETB TR1 //Arranca para gerador de baud

SETB TR2

SETB EA //HABILITA TODAS AS ISRS

SETB ET2 //HABILITA ISR TIMER2

MOV R3,#250

JNB TF2,$

RET

ISR\_T2:

DJNZ R3,FIM\_ISRT2

CALL LOOP //Rotina para imprimir os 9 numeros.

CPL P2.0 //Comuta o pino

MOV R3,#250 //Assegurar que não se perde o valor de r3

CLR TF2 //Limpa flag do timer para preparar para nova isr.

LOOP:

MOV A,@R0

CALL DISPLAY

INC R0

CJNE R0,#9,LOOP

RET

FIM\_ISRT2:

RETI

//Receber a string...

ISR\_SP:

MOV A,SBUF //Receber a string no acumulador vinda da porta série.

CLR RI //Limpa a flag de receção, esvazia o buffer para receber próximo.

CJNE A,#TABLE2,NEXT ; Se for igual a P130P então o periodo é de 130ms

CALL ROT130MS

JMP MAIN

NEXT:

CALL ROT2MS

JMP MAIN

ROT130MS:

MOV TH2,#HIGH(-130000)

MOV TL2,#LOW(-130000)

MOV RCAP2L,#LOW(-130000)

MOV RCAP2H,#HIGH(-130000)

RET

ROT2MS:

MOV TH2,#HIGH(-2000)

MOV TL2,#LOW(-2000)

MOV RCAP2L,#LOW(-2000)

MOV RCAP2H,#HIGH(-2000)

RET

END

#include<89c51ic2.inc>

CSEG AT 0H

LJMP MAIN

CSEG AT 03H

LJMP CONFIGT2

CSEG AT 0BH

LJMP LUZ

CSEG AT 13H

LJMP TESTEVEL

BSEG AT 9H

EXCESSO: DBIT 1

CSEG AT 30H

MAIN:

MOV TMOD,#01H

MOV TH0,#HIGH(-10000)

MOV TL0,#LOW(-10000)

MOV R6,#150

MOV R5,#150

MOV R4,#10

MOV TCON,#5

SETB EA

SETB ET0

SETB TR0

SETB EX0

SETB EX1

LOOP:

JMP $

LUZ:

JB EXCESSO,VERMELHO

JMP AMARELO

AMARELO:

DJNZ R6,OUT

CPL P1.1

MOV R6,#150

OUT:

CLR TF0

MOV TH0,#HIGH(-10000)

MOV TL0,#LOW(-10000)

RETI

VERMELHO:

SETB P1.1

CLR P1.0

DJNZ R5,OUT

MOV R5,#150

DJNZ R4,OUT

MOV R4,#10

CLR EXCESSO

SETB P1.0

CLR P1.1

JMP OUT

CONFIGT2:

MOV T2CON,#04H

MOV TH2,#HIGH(-10000)

MOV TL2,#LOW(-10000)

MOV RCAP2H,#HIGH(-10000)

MOV RCAP2L,#LOW(-10000)

SETB TR2

CLR IE0

RETI

TESTEVEL:

CLR TR2

JB TF2,NOTWASTED

SETB EXCESSO

NOTWASTED:

CLR TR2

CLR TF2

CLR IE1

RETI

END