Program 1---16 bit timer in mode1

**Assume that XTAL = 11.0592 MHz, write a C program** continuously gets a single bit of data from PI. 7 and sends it to P1.0, Simultaneously **generate a square wave of 2 kHz frequency on pin P1.5. Crystal frequency is 11.0592MHz.**

**(a) T = 1 / f = 1 / 2 kHz = 500 us the period of square wave.**

**(b) 1 / 2 of it for the high and low portion of the pulse is 250 us.**

**(c) 250 us / 1.085 us = 230 and 65536 – 230 = 65306 which in hex FF1AH.**

**(d) TL = 1A and TH = FF, all in hex.**

#include <reg51.h>

sbit SW =P1^7;

sbit IND =P1^0;

sbit WAVE =P1^5;

void timer0(void) interrupt 1

{

WAVE=~WAVE; //toggle pin

}

void main()

{

SW=1; //make switch input

TMOD=0x01;

TL0=0xFF;

TH0=0x1A; //for delay

IE=0x82; //enable interrupt for timer 0

TR0=1;

while (1)

{

IND=SW; //send switch to LED

}

}

Program 2—8 bit autoreload--

Write a C program that continuously gets a single bit of data from PI. 7 and sends it to P1.0, while simultaneously creating a square wave of 200us (as period) on pin P1.5. Use timer 0 in mode 2 to create the square wave. Assume that XTAL = 11.0592 MHz.

#include <reg51.h>

sbit SW =P1^7;

sbit IND =P1^0;

sbit WAVE =P1^5;

void timer0(void) interrupt 1

{

WAVE=~WAVE; //toggle pin

}

void main()

{

SW=1; //make switch input

TMOD=0x02;

TH0=0xA4; //TH0=-92

IE=0x82; //enable interrupt for timer 0

TR0=1;

while (1)

{

IND=SW; //send switch to LED

}

}

Program 3

A switch is connected to pin P3.2. When switch is pressed the corresponding line goes low Display 0A at port0. Write a C program to blink alternate LEDS connected to Port 1 Simultaneously.

#include <reg51.h>

sbit SW =P3^2;

unsigned int i;

void extint0() interrupt 0

{

P0=0x0A;

}

//void delay\_ms(unsigned int j);

void main()

//unsigned int i;

{

SW=1;

IE=0x81;

while(1)

{

P1=0x00;

for(i=250;i>0;i--)

{}

P1=0xAA;

for(i=250;i>0;i--)

{}

}

}

Program 4(timer 0 interrupt and external interrupt1)

Write a C program using interrupts to do the following:

(a) Generate a 10 KHz frequency on P2.1 using T0 8-bit auto-reload

 (b)  Assume that the clock pulse is connected to external interrupt1 EX1. Count the pulses and display it on P0. Assume that XTAL = 11.0592 MHz. Set the baud rate at 9600.

#include <reg51.h>

sbit WAVE =P2^1;

unsigned char cnt;

void timer0() interrupt 1

{

WAVE=~WAVE; //toggle pin

}

//void timer1() interrupt 3

void extint1() interrupt 2

{

cnt++; //increment counter

P0=cnt; //display value on pins

}

void main()

{

cnt=0; //set counter to 0

TMOD=0x02;

TH0=0x46; //10 KHz

IE=0x86; //enable interrupts

TR0=1; //start timer 0

while (1); //wait until interrupted

}

**Changed priority example for above program**

#include <reg51.h>

// Define global variables

sbit WAVE =P2^1;

unsigned char cnt;

// Timer 0 ISR for generating a 10 KHz frequency on P2.1

void timer0\_ISR() interrupt 1

{

WAVE=~WAVE;

}

// External Interrupt 0 ISR for counting pulses and displaying on P0

void extint0\_ISR() interrupt 0

{

// Increment the counter

cnt++;

// Display the value on P0

P0 = cnt;

}

void main()

{

// Initialize the counter

cnt = 0;

// Configure Timer 0 in mode 2 (8-bit auto-reload) for generating a 10 KHz frequency

TMOD |= 0x02;

TH0 = 0x46; // 10 KHz frequency at 11.0592 MHz crystal frequency

// Enable Timer 0 interrupt (high priority)

ET0 = 1;

PT0 = 1; // Set high priority for Timer 0 interrupt (or IP=0x02)

TR0 = 1;

// Configure External Interrupt 0 for counting pulses

IT0 = 1; // Edge-triggered

EX0 = 1; // Enable External Interrupt 0

// Enable interrupts globally

EA = 1;

// Main loop

while (1) {

}

}

Let’s program the external interrupt such that, when falling edge is detected on the INT0 pin then the microcontroller will toggle the P1.0 pin.

**Programming steps**

1. Enable global interrupt i.e. EA = 1
2. Enable external interrupt i.e. EX0 = 1
3. Enable interrupt trigger mode i.e. whether interrupt is edge triggered or level triggered, here we will use falling edge trigger interrupt, so make IT0 = 1.

/\*

\* 8051\_External\_Interrupt

\*/

#include <reg51.h> /\* Include x51 header file \*/

sbit LED = P1^0; /\* set LED on port1 \*/

void Ext\_int\_Init()

{

EA = 1; /\* Enable global interrupt \*/

EX0 = 1; /\* Enable Ext. interrupt0 \*/

IT0 = 1; /\* Select Ext. interrupt0 on falling edge \*/

}

void External0\_ISR() interrupt 0

{

LED = ~LED; /\* Toggle pin on falling edge on INT0 pin \*/

}

void main()

{

Ext\_int\_Init(); /\* Call Ext. interrupt initialize \*/

while(1);

}