**Square wave**

#include <p18f4520.h>

#pragma config OSC = HS // High-speed oscillator

#pragma config WDT = OFF // Watchdog Timer disabled

#pragma config LVP = OFF // Low-voltage Programming disabled

#pragma config PBADEN = OFF

unsigned char g\_ui8TimerOnflag=0,g\_ui8Num=0;

unsigned char g\_ui8tmrcnt=0;

void Timer0Init()

{

T0CONbits.T0PS0 = 1; // Prescaler of 2

T0CONbits.T0PS1 = 1; // Prescaler of 2

T0CONbits.T0PS2 = 1; // Prescaler of 2

T0CONbits.PSA = 0; // Timer Prescaler assigned

T0CONbits.T0SE = 0; // Increment on low to high transition

T0CONbits.T0CS = 0; // Internal clock

T0CONbits.T08BIT = 0; // 16 BIT MODE

T0CONbits.TMR0ON = 1; // Turns timer on

TMR0H =0xB3;

TMR0L = 0xB4;

// Interrupt Setup

RCONbits.IPEN = 1; // Interrupt priority enabled

INTCONbits.GIE = 1; // Enables Global Interrupts

INTCONbits.PEIE = 1; // Enables Peripheral Interrupts

INTCONbits.TMR0IE = 1; // Enables overflow Interrupt

INTCONbits.TMR0IF = 0; // Sets overflow bit to zero

INTCON2bits.TMR0IP = 1; // Timer 0 Interrupt is high priority

}

void TMRISR (void) ;

#pragma code InterruptVectorHigh=0x08

void InterruptVectorHigh (void)

{

\_asm

goto TMRISR

\_endasm

}

#pragma code

#pragma interrupt TMRISR

void TMRISR (void) // Timer ISR

{

if(INTCONbits.TMR0IF == 1)

{

INTCONbits.TMR0IF=0; //Clear Interrupt Flag to prevent reinitialisation of ISR

TMR0H =0xB3;

TMR0L = 0xB4;

if(g\_ui8TimerOnflag)

{

PORTD = 0x00;

g\_ui8TimerOnflag=0;

}

else

{

PORTD = 0xff;

g\_ui8TimerOnflag=1;

}

}

}

void main (void)

{

Timer0Init();

g\_ui8TimerOnflag=0;

TRISD = 0x00; //Port configured as Output

PORTD = 0x00; //PORTD = 0x00

while(1)

{}

}