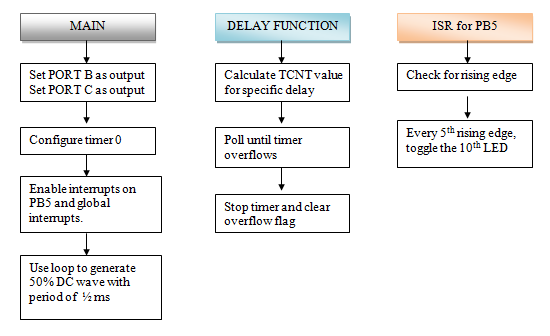
CPE 301 – SPRING 2015

DESIGN ASSIGNMENT 2

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **SUBMISSION ITEM** | **COMPLETED (Y/N)** | **MARKS**  **(/MAX)** |
| 0. | Flowchart of code | Y |  |
| 1. | AVR C Code that compiles and works | Y |  |
| 2. | Verify period and duty cycle (DC) of waveform in simulation | Y |  |
| 3. | Schematics and LED count sequence | Y |  |
| 4. | Snapshot of board with connected components | Y |  |
| 5. | Link to YouTube Video | Y |  |

**0 – Flowchart of code:**



**1 - AVR C code:**

#include <avr/io.h>

#include <avr/interrupt.h>

#define F\_CPU 16000000UL //set clock frequency to 16MHz for compiler

void delayms(int ms); //delay function prototype

int main(void)

{

DDRB |= (1<<PB5); //set portb.5 as output

DDRB |= (1<<PB4); //toggles every 5 rising edges

DDRC |= 0x0F; //make lower 4 pins of port C outputs

DDRD |= 0x0F; //make lower 4 pins of port D outputs

DDRD &= ~(1<<PD4); //set portd.4 as input

TCCR0A = 0; //timer 0 in normal mode

TCCR0B = 7; //external clock on t0 (portb.4)

TCNT0 = 0; //initialize counter0 to 0

PCICR |= (1<<PCIE0); //needed for using PCINT5 (PB5)

PCIFR |= (1<<PCIF0); //needed for using PCINT5 (PB5)

PCMSK0 |= (1<<PCINT5); //using PCINT5 because interrupt will be caused by PB5

sei(); //enable interrupts

while(1) //generate waveform of 50% DC with 0.5 sec period

{

PORTB &= ~(1<<PB5); //portb.5 off

delayms(250); //delay for 250 ms

PORTB |= (1<<PB5); //port.5 on

PORTD = TCNT0>>4; //move upper nibble of counter into portd

PORTC = TCNT0 & 0x0F; //move lower nibble of counter into portc

delayms(250); //delay for 250 ms

}

}

void delayms(int ms)

{

double tcnt = 65536 - (((double)ms)/1000)\*((double)F\_CPU/64); //set TCNT = 2^16 - delay/clock period/64

TCNT1H = ((int)tcnt)>>8; //move upper byte of TCNT

TCNT1L = ((int)tcnt) & 0x00FF; //move lower byte of TCNT

TCCR1A = 0; //timer 1 in normal mode

TCCR1B = 3; //prescaler = 64

while(!(TIFR1 & (1<<TOV1))) //loop until timer overflow happens

{

}

TCCR1B = 0; //stop timer 1

TIFR1 = 1; //clear overflow flag

return;

}

ISR (PCINT0\_vect) //ISR for PCINT5 (PORTB.5)

{

if((PORTB & (1<<PB5)) == 0) //if pin changed from high to low, do nothing

{

}

else //pin changed from low to high

{

static int counter = 0; //used to detect every 5th rising edge

counter++;

if(counter%5 == 0) //every 5 rising edges, pb.4 toggles

PORTB ^= (1<<PB4); //toggle pb.4

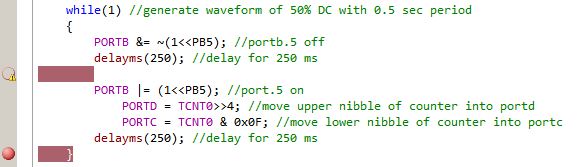
}

return;

}

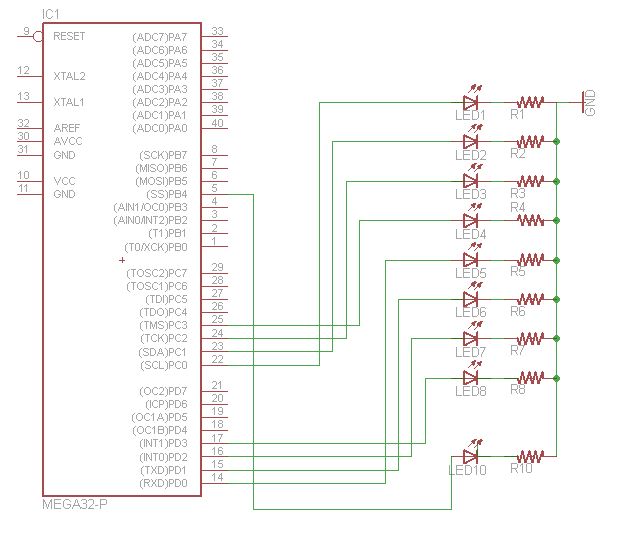
**2 - Verify period and duty cycle of waveform in simulation:**

I set a breakpoint after each delay to verify the delay is 250 ms and that the wave is 50% DC.

C:\Users\Emmanuel\Desktop\first half.JPG 🡨 250 ms after first delay

C:\Users\Emmanuel\Desktop\second half.JPG 🡨 250 ms after second delay

**3 - Schematics:**



LED count sequence: 00000000 🡨 0

00000001

00000010

00000011

...

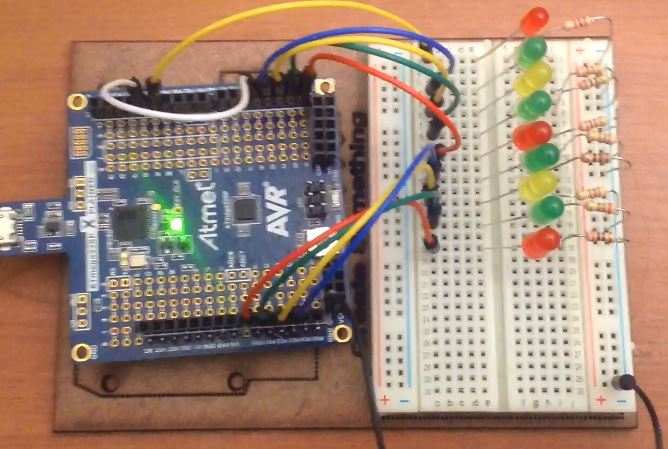
11111111 🡨 255

00000000

- LED #10 toggles every 5 rising edges

- LED #9 not used

**4 - Snapshot of board with connected components:**



**5 - Link to video**: <https://www.youtube.com/watch?v=tIDPQiCQ6v0>

**Student Academic Misconduct Policy**

<http://studentconduct.unlv.edu/misconduct/policy.html>

“This assignment submission is my own, original work”.

Emmanuel Sanchez