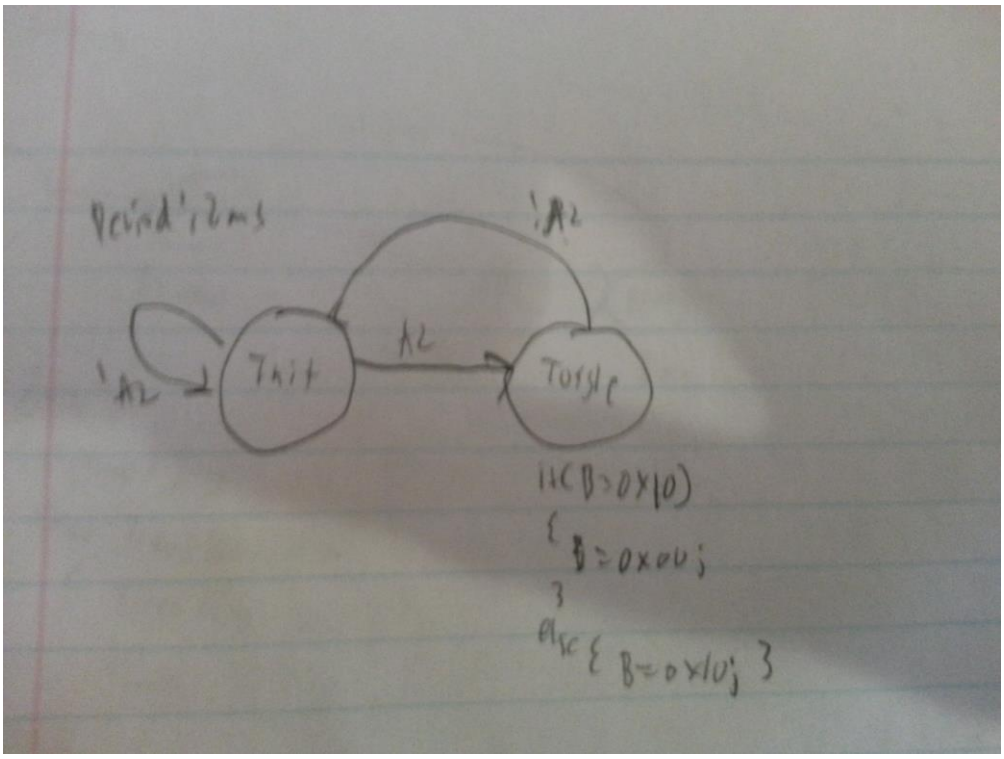
Prelab 6 Report




```
// copy pasted lab6_ex1 code…
#include <avr/io.h>
#include <avr/interrupt.h>
#include "io.c"


volatile unsigned char TimerFlag = 0;
unsigned long _avr_timer_M = 1;
unsigned long _avr_timer_cntcurr = 0;
unsigned char tmp = 0x00;

void TimerOn()
{
	TCCR1B = 0x0B;
	OCR1A = 125;
```

```
	TIMSK1 = 0x02;
	TCNT1 = 0;
	_avr_timer_cntcurr = _avr_timer_M;
	SREG |= 0x80;
}

void TimerOff()
{
	TCCR1B = 0x00;
}

// Bit-access function
unsigned char SetBit(unsigned char x, unsigned char k, unsigned char b) {
	return (b ? x | (0x01 << k) : x & ~(0x01 << k));
}
unsigned char GetBit(unsigned char x, unsigned char k) {
	return ((x & (0x01 << k)) != 0);
}

void TimerISR()
{
	TimerFlag = 1;
}

ISR(TIMER1_COMPA_vect)
{
	_avr_timer_cntcurr--;
	if(_avr_timer_cntcurr == 0)
	{
```

```
                TimerISR();
                _avr_timer_cntcurr = _avr_timer_M;

        }
}

void TimerSet(unsigned long M)
{
        _avr_timer_M = M;
        _avr_timer_cntcurr = _avr_timer_M;
}


enum States {Init, s0} State;

void Tick()
{
        unsigned char tmp2 = ~PINA;
        switch(State)
        {                       // Transitions
                case Init:
                if(GetBit(tmp2,2) == 1)
                {
                        State = s0;
                }
                else
                {
                        State = Init;
                }
```

```
            break;


            case s0:
   if(GetBit(tmp2,2) == 0)
   {
               State = Init;
   }
            else
            {
                        State = s0;
            }
            break;



            default:
            State = Init;
            break;
} // Transitions

switch(State)
{
            // State actions
            case Init:
            break;


            case s0:
            if(tmp == 0x00)
            {
                        tmp = 0x10;
```

```
                }
                else
                {
                        tmp = 0x00;
                }
                PORTB = tmp;
                break;

                default:
                PORTB = 0x00;
                break;
        } // State actions
}


int main(void)
{
        DDRA = 0x00; PORTA = 0xFF;
        DDRB = 0xFF; PORTB = 0x00;

        // intermediate variable used for port updates
        /* Replace with your application code */

        State = Init;
        TimerSet(2);
        TimerOn();
        while (1)
        {
                Tick();
```

```
            while(!TimerFlag);
            TimerFlag = 0;
        }
}
```