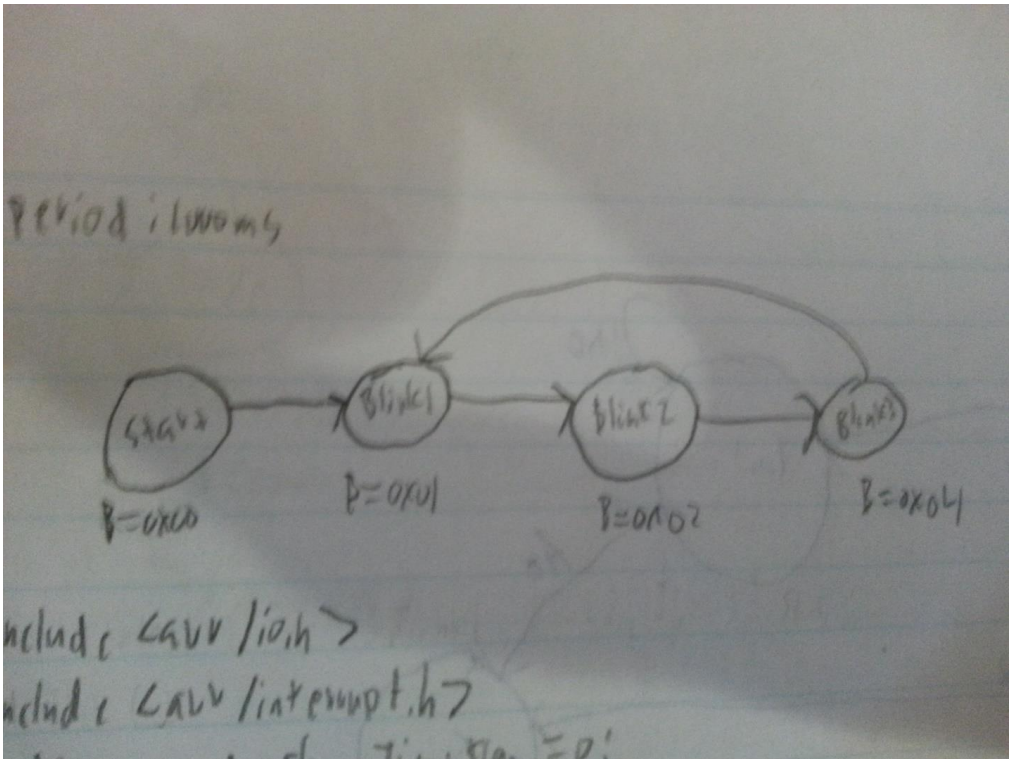
Prelab 5 Report




// copy pasted lab5_ex1 code…..

```
#include <avr/io.h>
#include <avr/interrupt.h>


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, s1, s2} State;

void Tick()
{
        switch(State)
        {                       // Transitions
                case Init:
                State = s0;
                break;

                case s0:
                State = s1;
                break;

                case s1:
                State = s2;
```

```
        break;

        case s2:
        State = s0;
        break;

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

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

        case s0:
        tmp = 0x01;
        PORTB = tmp;
        break;

        case s1:
        tmp = 0x02;
        PORTB = tmp;
        break;

        case s2:
        tmp = 0x04;
```

```
                PORTB = tmp;
                break;

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


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

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

        State = Init;
        TimerSet(1000);
        TimerOn();
        while (1)
        {
                Tick();
                while(!TimerFlag);
                TimerFlag = 0;
        }
}
```