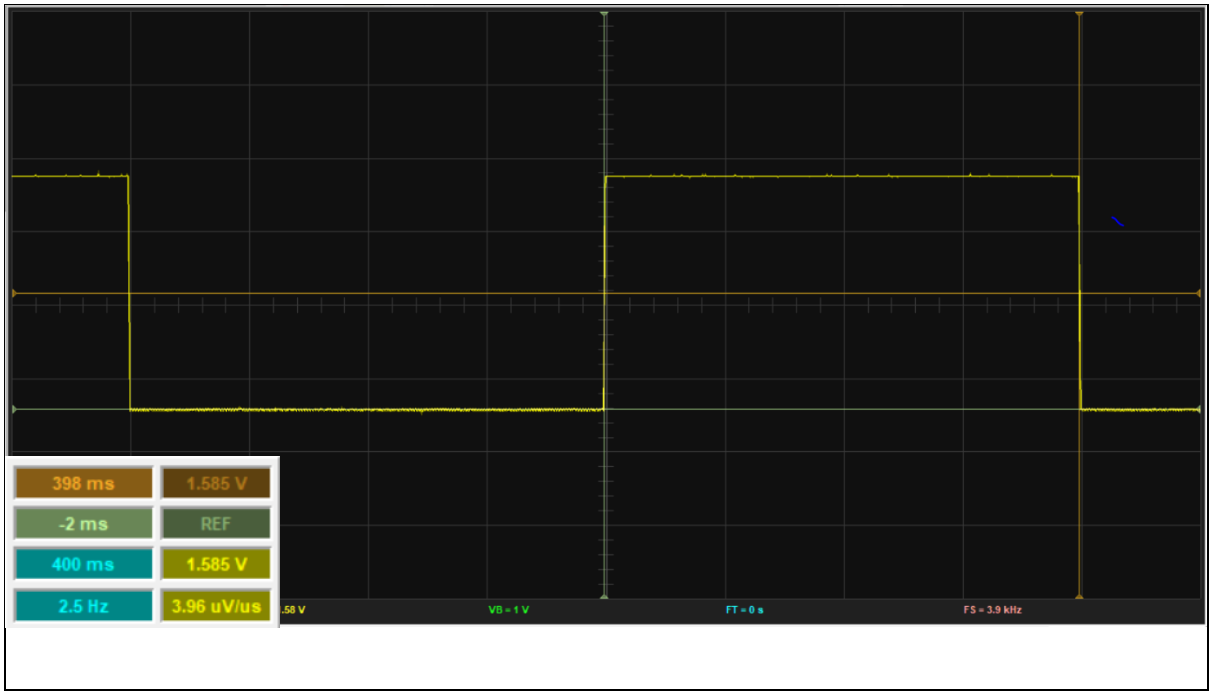
C10 – Interrupt

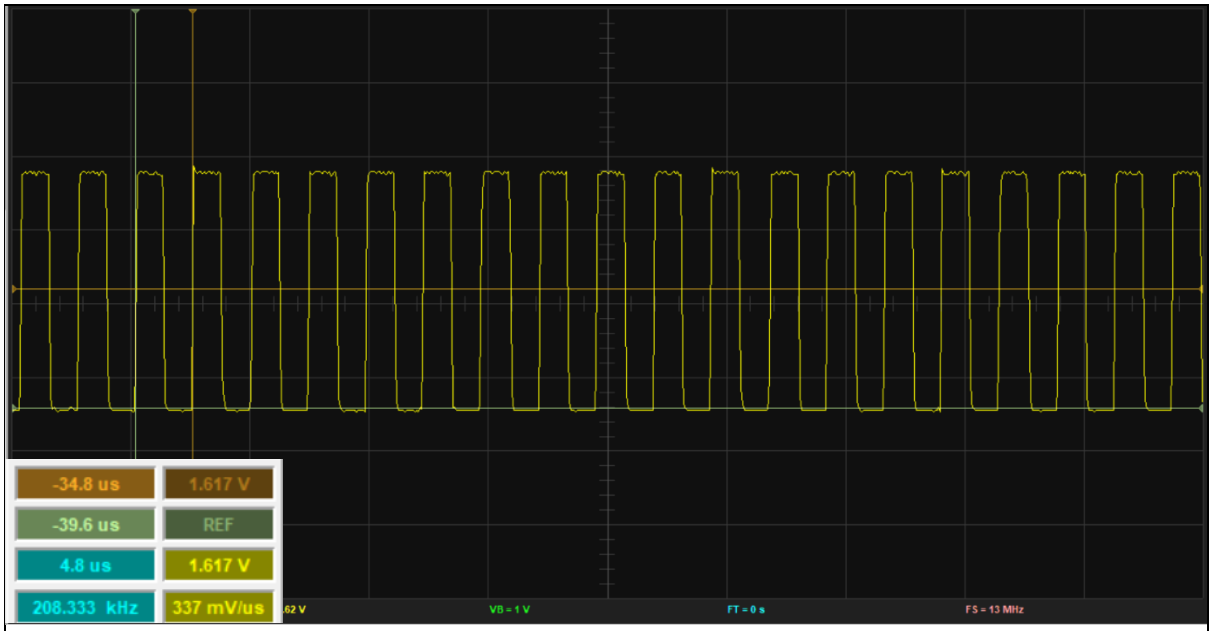
3.1 Timer Interrupts

Implementing a blink program where the LED toggles every 400ms. Using timer one I will set the pre-scaler to 265 since this gives us a maximum time period of 1447.2ms. Using compare mode I will set the compare at 18,750 since this will give an exact period of 400ms.

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


ISR(TIMER1_COMPA_vect) {
    PINB |= _BV(PB7);
}

int main() {

    DDRB |=  _BV(PB7);      //led as output

    //setup timer one
    TCCR1A = 0;

    TCCR1B |= _BV(WGM12);       //set to ctc mode
    TCCR1B |= _BV(CS12);        //set clock prescale to 256

    OCR1A = 18750;          //value as calculated (400ms)

    TIMSK1 |= _BV(OCIE1A);      //output compare A match interrupt
    sei();                  //enable interrupts

    while(1);
    return 0;
}
```


398 ms
1.585 V
-2 ms
REF
400 ms
1.585 V
2.5 Hz
3.96 uV/us
.58 V
VB = 1 V
FT = 0 s
FS = 3.9 kHz

3.2 ADC Interrupts



Indicated frequency = 208,333Hz
208.333 x 2 = 416,666 ADC Samples per Second (x2 since pin in toggled).

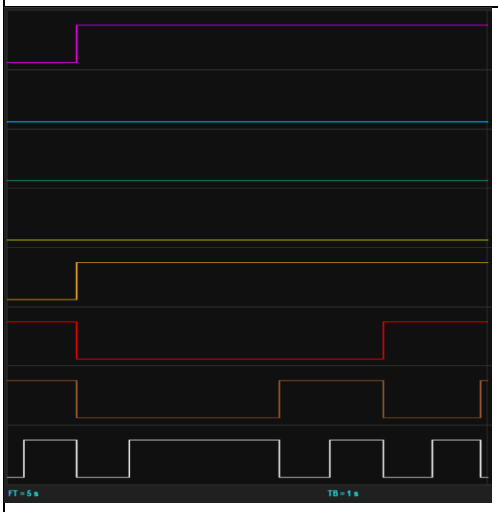
ADC Source Clock pre-scaler = 2
ADC Source Clock Frequency = 6MHz
6,000,000 / 416,666 = 14.4 clock cycles ≈ 13 clock cycles (data sheet)

```
ISR(ADC_vect) {
    PINB |= _BV(PB7);
}
ADCSRA |= _BV(ADIE);              //interrupt enable
sei();
```

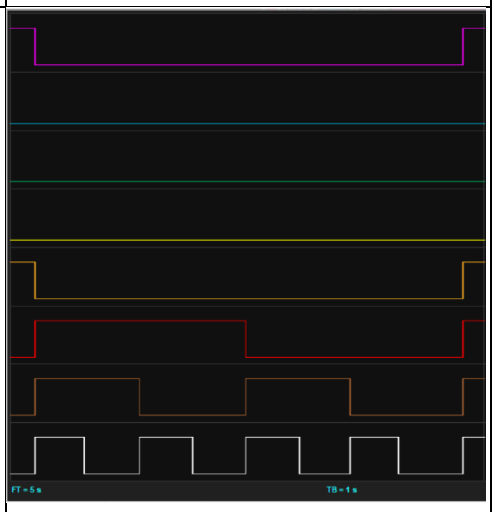
3.3 External Interrupts

```c
int main() {
    i = 0;

    DDRC = 0xff;        //set port c as output
    PORTC = i;

    DDRB |= _BV(PB7);       //led pin as output

    DDRD &= ~_BV(PD2);      //INT0 as input
    DDRD &= ~_BV(PD3);      //INT1 as input

    PORTD |= _BV(PD2);      //INT0, INT1 pullup
    PORTD |= _BV(PD3);


    EICRA |= _BV(ISC01);        //INT0 to falling edge
    EICRA |= _BV(ISC11);        //INT1 to falling edge

    EIMSK |= _BV(INT0);         //enable interrupts
    EIMSK |= _BV(INT1);
    sei();


    while(1);
    return 0;
}
```

```c
volatile uint8_t i;


//if PD2 falls, decrement
ISR(INT0_vect) {
    PORTB |= _BV(PB7);

    if(i != 0) {
        i--;
    }
    PORTC = i;
}

//if PD3 falls, increment
ISR(INT1_vect) {
    PORTB &= ~_BV(PB7);

    if(i != 255) {
        i++;
    }
    PORTC = i;
}
```



Incrementing



Decrementing