

02/12/20

```
#define V_MAX 3.3

void init_adc(void)
{
    ADCSRA |= _BV(ADPS2) | _BV(ADPS1); // F_ADC = F_CPU/64
    ADCSRA |= _BV(ADEN);                // Enable ADC
}

uint16_t read_adc(void)
{
    ADCSRA |= _BV(ADSC);                // Start ADC conversion
    while (ADCSRA & _BV(ADSC));         //wait until result is valid
    return ADC;
}

int main(void)
{
    uint16_t result;
    double voltage;

    init_debug_uart0();
    init_adc();

    for (;;)
    {
        result = read_adc();             //get position of an ADC
        voltage = V_MAX / 1023 * result; //calculate the voltage
        printf("%4d : %6.5lfV\n", result, voltage);
        _delay_ms(1000);
    }
}
```

Initially I could not quite get the full voltage range for my potentiometre I found that it went away when I connected the wires closer to the potentiometer so I expect this is because it is due to spurious capacitance or inductance.

$$\begin{aligned} \text{noise} &= \log_2 \frac{3.3}{V_{\max} - V_{\min}} \\ \text{noise} &= \log_2 \frac{3.3}{3.3 - 0} \end{aligned}$$

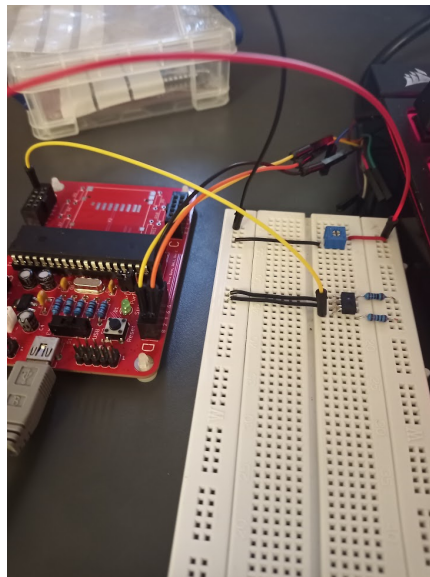
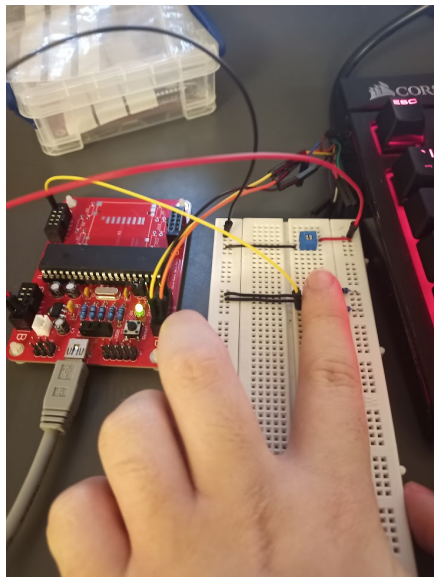
$$\text{noise} = \log_2 1024$$
$$\text{noise} = 10$$

Determining $V > V_a$ I am confident that there is nothing in proximity to the sensor, and when $V < V_p$ I am confident that there is an object in proximity to the sensor.

$V_a = 3.22V$ $n=1000$
 $V_p = 1.61V$ $n=500$

I then edited the code so that it would turn the LED on or off depending on the voltage measured on whether the voltage measured at the photosensor.

```
void channel_adc(uint16_t n)
{
    if(n < 500)
    {
        PORTB |= _BV(PB7);
        printf("\nLED On\n");
    }
    if(n > 1000)
    {
        PORTB &= ~_BV(PB7);
        printf("\nLED Off\n");
    }
}
```



```
LED On
447 : 1.44194V
589 : 1.90000V
824 : 2.65806V
927 : 2.99032V
974 : 3.14194V
995 : 3.20968V

LED Off
1015 : 3.27419V

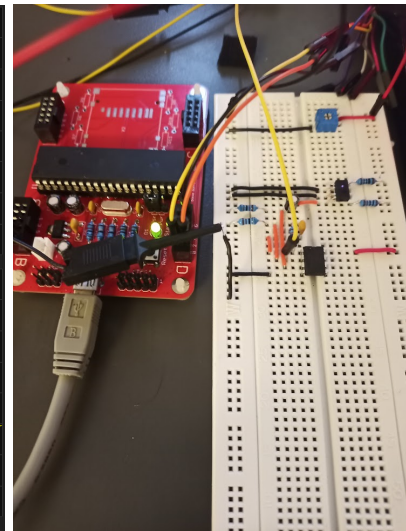
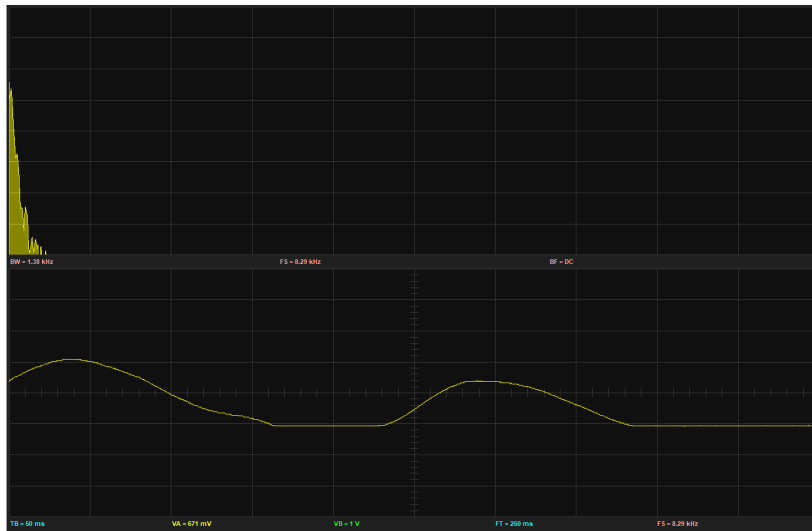
LED Off
1015 : 3.27419V

LED Off
1016 : 3.27742V

LED Off
1016 : 3.27742V

LED Off
1016 : 3.27742V
```

Using an active band pass filter I was able to get the bitscope to register my pulse using very small fluctuations in the proximity sensor.



I then began modifying the code to calculate my bpm.

```
for (;;)
{
    result = read_adc(); //get position of an ADC
    if (result > 250)
    {
        if (pulse == 0)
        {
            t_end = t_start;
            t_start = count;
            if (edge)
            {
                bpm = 60 * 100.0 / (t_end - t_start)
            }
            edge = 1;

            pulse = 1;
        }
        for(;;)
        {
            _delay_ms(10);
            result = read_adc(); //get position of an ADC
            count += 10;
        }
    }
    printf("bpm = %6.5lf\n", bpm);
}
```