Project

Class Project: Voltmeter

Florida Polytechnic University

EEL4746C: Microcomputers

Fall 2018

Student Name: Peter A. Dranishnikov

Student ID: U0000005258

Lab Partner(s): N/A

Section: 01

Experiment Date: December 3rd, 2018

Table of Contents

Introduction

This project involves the use of the analog-to-digital (ADC) converter feature of the Atmega328p as a voltmeter to display the voltage as a color of an LCD RGB backlight.

Discussion

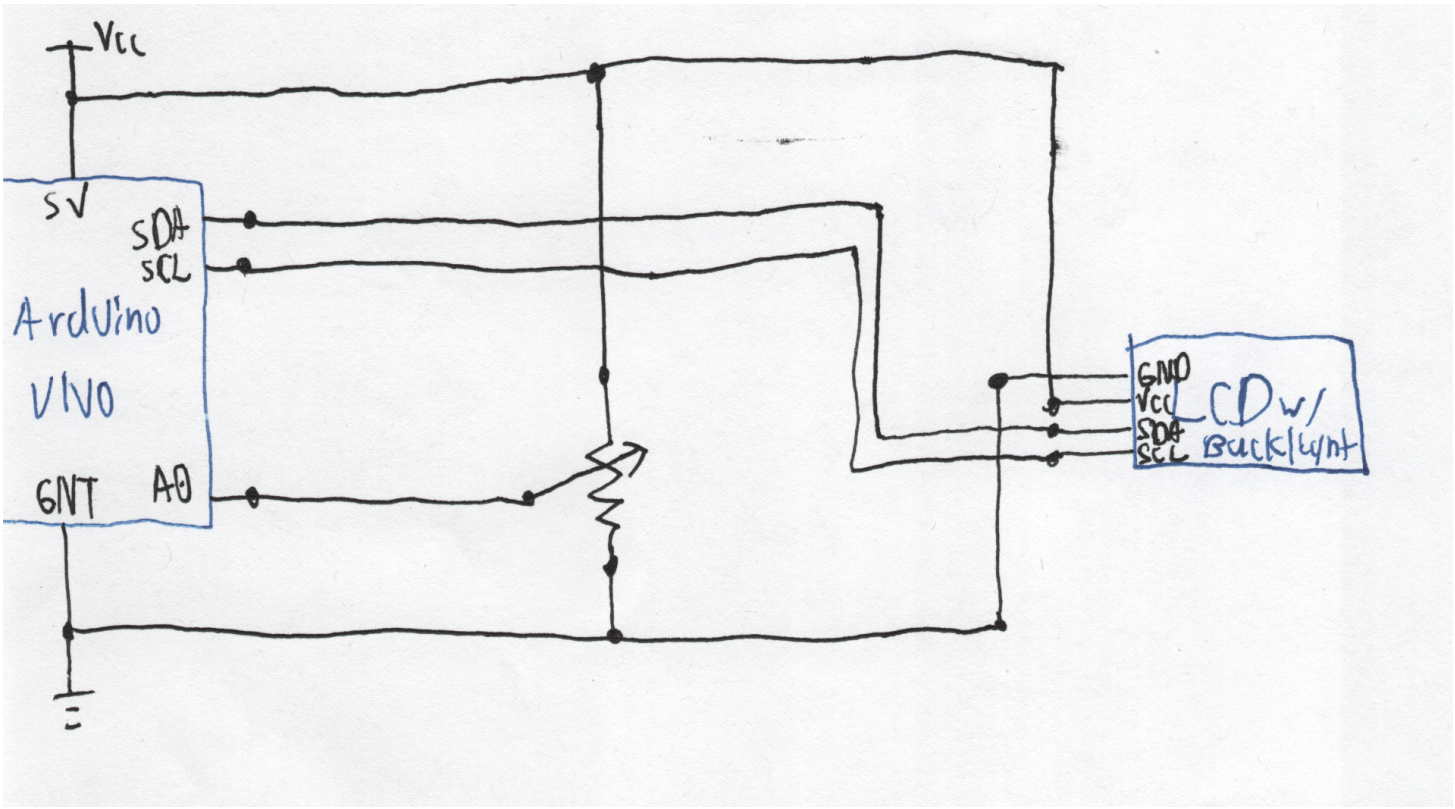
Source code:

```
/*************************************************************
* Filename: project_main.c
* Version: 1.0
* Description: 5 VVoltmeter with LCD backlight output
* Author: Peter Dranishnikov (see additional note)
* Target: Atmel AtMega328p
* Compiler: avr-gcc
* Last modified: Friday, December 7th, 2018
*** NOTE *** Portions of code taken from example files, written by Dr. Al-Nashif
*************************************************************/
#include <stdlib.h>
#include "eel4746c-1.h"

typedef struct twi_struct {
  uint8_t twbr;
  uint8_t twsr;
  uint8_t twar;
  uint8_t twdr;
  uint8_t twcr;
  uint8_t twamr;
} twi_t;

#define twen 2
#define twsto 4
#define twsta 5
#define twint 7

#define timedelay 200

#define RGB_MODE_REGISTER1  0x00
#define RGB_MODE_REGISTER2  0x01
#define BLUE_REGISTER       0x02
#define GREEN_REGISTER      0x03
#define RED_REGISTER        0x04
#define RGB_OUTPUT_REGISTER 0x08


volatile twi_t *twi= (twi_t *) 0x00B8;

void i2c_init();
void i2c_start();
void i2c_write(uint8_t payload);
void i2c_stop();

int main()
{
    uint8_t address=0xC4;// LCD RGB Address
    uint8_t numhi, numlo = 0;
    adc->ADCSRA = 0x87;
    adc->ADMUX = 0x40;

    i2c_init();

    i2c_start();
    i2c_write(address);
    i2c_write(RGB_MODE_REGISTER1);
```

```
    i2c_write(0x00);
    i2c_stop();


    i2c_start();
    i2c_write(address);
    i2c_write(RGB_OUTPUT_REGISTER);
    i2c_write(0xFF);
    i2c_stop();

    i2c_start();
    i2c_write(address);
    i2c_write(RGB_MODE_REGISTER2);
    i2c_write(0x20);
    i2c_stop();

    i2c_start();
    i2c_write(address);
    i2c_write(RED_REGISTER);
    i2c_write(0x00);
    i2c_stop();

    while(1)
    {
        adc->ADCSRA = adc->ADCSRA | 0x40;

        while ((adc->ADCSRA&0x10) == 0 ); // Wait for value to be ready

        numlo = adc->ADCL & 0xE0;
        numhi = (adc->ADCH & 0x03) << 6;

        i2c_start();
        i2c_write(address);
        i2c_write(GREEN_REGISTER);
        i2c_write(numlo);
        i2c_stop();

        i2c_start();
        i2c_write(address);
        i2c_write(BLUE_REGISTER);
        i2c_write(numhi);
        i2c_stop();
    }
    return 0;
}

void i2c_stop()
{
  twi->twcr=((1<<twen)|(1<<twint)|(1<<twsto));
}

void i2c_write(uint8_t payload)
{
  uint16_t i;
  twi->twdr=payload;
  twi->twcr=(1<<twen)|(1<<twint);
  while (((twi->twcr)&(1<<twint))==0x00);
  for (i=0;i<timedelay;i++);
}


void i2c_start()
{
  twi->twcr=0xA4;// or ((1<<twen)|(1<<twint)|(1<<twsta))
  while (((twi->twcr)&(1<<twint))==0x00);
}
```

```
void i2c_init()
{
  uint16_t i;
  /* Enable Internal Pullups */
  twi->twsr=0x00;
  volatile uint8_t *DDRC=(uint8_t *) 0x0027;
  *DDRC=0x00;
  volatile uint8_t *PORTC=(uint8_t *) 0x0028;
  *PORTC=0xFF;

  /* initialize TWI */
  twi->twbr=0x48;
  twi->twsr=0x00;
  twi->twcr=0x04;
}
```

Circuit:



Experimental procedure

Compile, link, and flash to Arduino.

Results/Measurements/Observations

The ADC input was split between the green and blue channels, so the output has multiple gradients and intensities.

Result Discussion

Works as intended.

Conclusion

It is possible to use a microcontroller as a basic voltmeter.