Horiba Minimate 1681B SPEX Monochromator User Manual

Liam O'Connor

August 27, 2020

1 Introduction

The monochromator is a 1681B Minimate which was refurbished to run with a standard 4-wire stepper motor. The original control computer was replaced with a Sparkfun Redboard qwiic microcontroller. This board is based on the ATmega328 chip which can be programmed throught the Arduino IDE.

2 Arduino Control

The "brain" of the monochromator is a Sparkfun Redboard with qwiic capabilities. The qwiic system is a jst connection to allow easy i^2c daisy-chaining of multiple devices. We used this to connect a 3x4 numeric keypad and 0.96" (128x64) dot matrix OLED display, both made by Sparkfun electronics.

The coding was done in arduino IDE with the following installed libraries.

- Wire.h
- SPI.h
- SFE_MicroOLED.h
- SparkFun_Qwiic_Keypad_Arduino_Library.h

3 Operation Limitations

The monochromator can be powered on with a rocker switch on the rear panel. Once powered on the OLED screen requires a few seconds to initialize. Once powered on the user should confirm the machine is in the home position (360nm) by pressing the asterisk key once, and then once again after a brief pause. Once the machine resets a wavelength can be entered between 360nm and 1020nm. The pound key will send the grating to the entered wavelength. Once there it can be pressed again to increase the grating position by 10nm.

Any user should view the code that has been installed to help understand the limitations of the machine. A home switch has been installed, but no switch exists to stop the machine from driving beyond the safe range. The acceptable range to enter on the keypad is 360nm to 1020nm. If a number outside this range is entered the code will report an error message, and tell you to try again. If the calibration equation is changed in the code these coded end stops will also need to be changed.

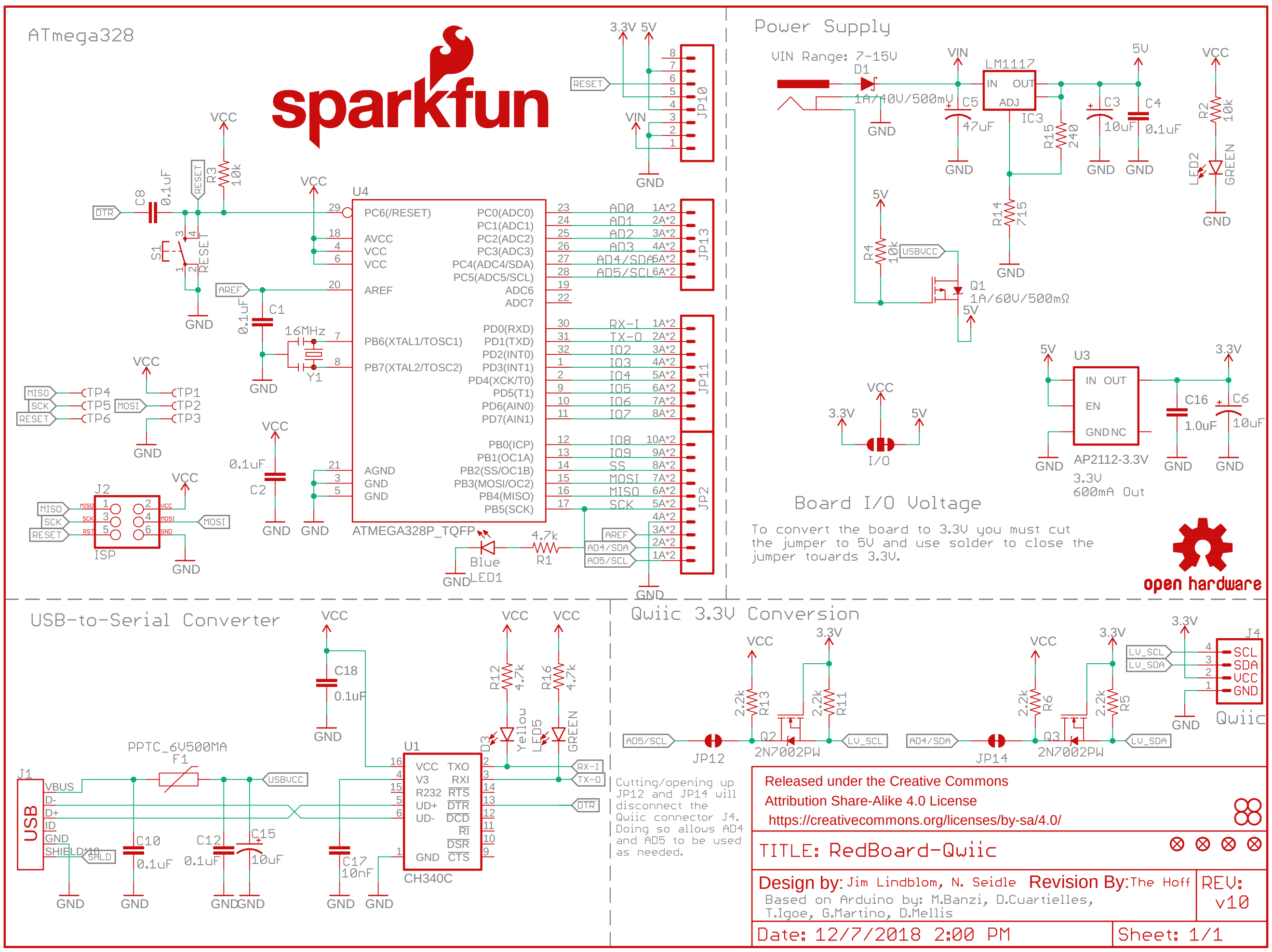
Users should beware that the slit position and distance to sensor can effect the wavelength detected. If a spectroradiometer has a bare fibre optic line placed directly into the exit slit edge effects from singular fibres can occur. adjusting the detector field of view can result in changes up to 10nm. A table (Table 1)of measured values has been included to show the non-linear offset found during calibration.

keyed in (nm)	ASD (nm)	offset (keyed-ASD)
360	360	0
380	380	0
400	399	1
420	419	1
440	438	2
460	458	2
480	478	2
500	497	3
520	516	4
540	536	4
560	556	4
580	576	4
600	596	4
620	616	4
640	636	4
660	656	4
680	676	4
700	696	4
720	716	4
740	736	4
760	756.5	3.5
780	776.5	3.5
800	797	3
820	818	2
840	838.5	1.5
860	859	1
880	880	0
900	901	-1
920	922	-2
940	943	-3
960	964	-4
980	986	-6
1000	1007	-7
1020	1028	-8

Figure 1: Spectral offset of ASD measured values compared to values entered on keypad, at one ASD pistol position.

4 Data Sheets

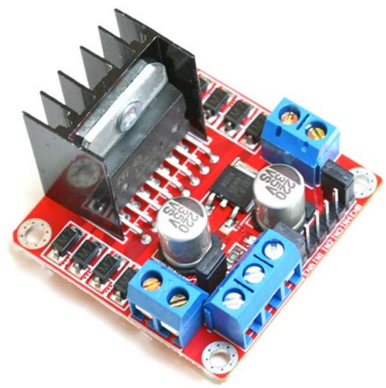
4.1 Sparkfun Redboard


ATmega328
sparkfun
Power Supply
VIN Range: 7-15V
LM1117
AP2112-3.3V
3.3V
600mA Out
Board I/O Voltage
To convert the board to 3.3V you must cut the jumper to 5V and use solder to close the jumper towards 3.3V.
open hardware
USB-to-Serial Converter
CH340C
ATMEGA328P_TQFP
Quiic 3.3V Conversion
Cutting/opening up JP12 and JP14 will disconnect the Quiic connector J4. Doing so allows AD4 and AD5 to be used as needed.
Released under the Creative Commons Attribution Share-Alike 4.0 License
https://creativecommons.org/licenses/by-sa/4.0/
TITLE: RedBoard-Qwiic
Design by: Jim Lindblom, N. Seidle
Revision By: The Hoff
Based on Arduino by: M.Banzi, D.Cuartielles, T.Igoe, G.Martino, D.Mellis
REV: v10
Date: 12/7/2018 2:00 PM
Sheet: 1/1

User Guide

L298N Dual H-Bridge Motor Driver

This dual bidirectional motor driver, is based on the very popular L298 Dual H-Bridge Motor Driver Integrated Circuit. The circuit will allow you to easily and independently control two motors of up to 2A each in both directions.It is ideal for robotic applications and well suited for connection to a microcontroller requiring just a couple of control lines per motor. It can also be interfaced with simple manual switches, TTL logic gates, relays, etc. This board equipped with power LED indicators, on-board +5V regulator and protection diodes.

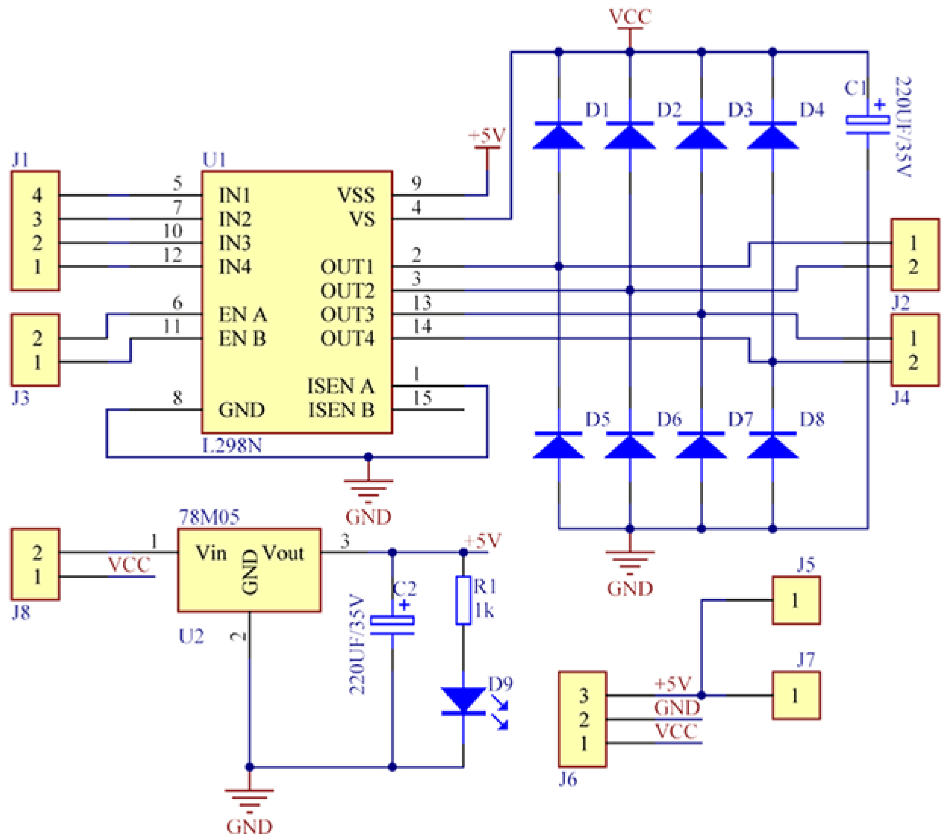


SKU: MDU-1049

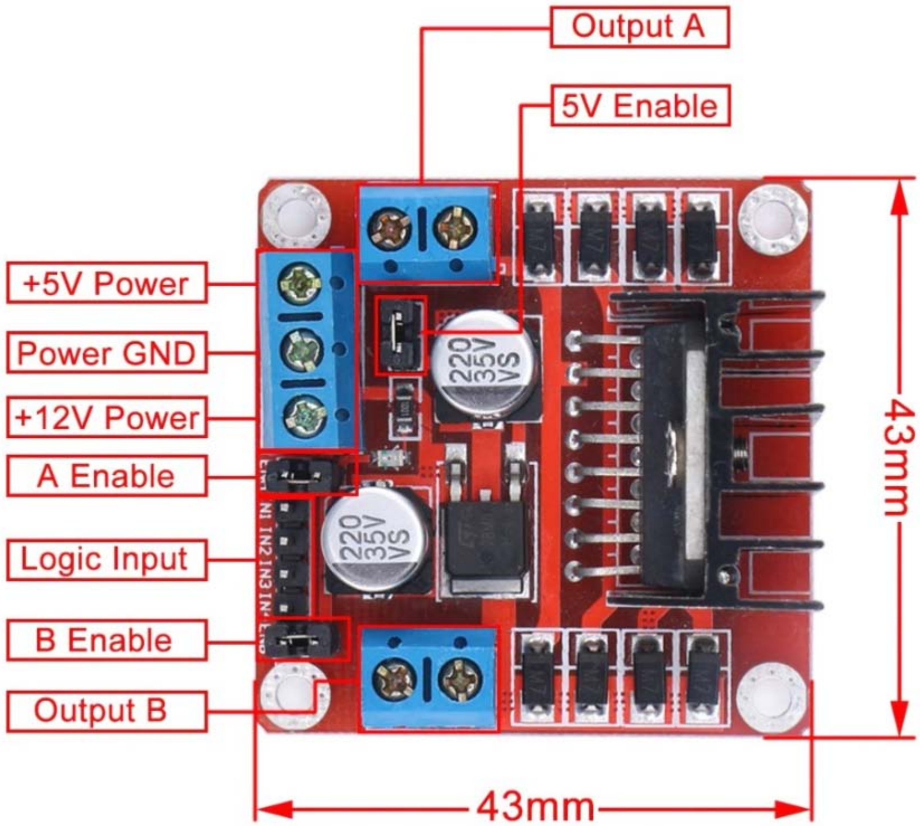
Brief Data:

- Input Voltage: 3.2V~40Vdc.
- Driver: L298N Dual H Bridge DC Motor Driver
- Power Supply: DC 5 V - 35 V
- Peak current: 2 Amp
- Operating current range: 0 ~ 36mA
- Control signal input voltage range :
- Low: -0.3V ⩽ Vin ⩽ 1.5V.
- High: 2.3V ⩽ Vin ⩽ Vss.
- Enable signal input voltage range :
  - Low: -0.3 ⩽ Vin ⩽ 1.5V (control signal is invalid).
  - High: 2.3V ⩽ Vin ⩽ Vss (control signal active).
- Maximum power consumption: 20W (when the temperature T = 75 ℃).
- Storage temperature: -25 ℃ ~ +130 ℃.
- On-board +5V regulated Output supply (supply to controller board i.e. Arduino).
- Size: 3.4cm x 4.3cm x 2.7cm

Schematic Diagram:



Board Dimension & Pins Function:



1681 Minimate-2
340E and 340S Spectrometers

The 1681 MINIMATE-2, the 340E and the 340S are compact, general-purpose monochromators/spectrometers that can deliver high purity radiation without sacrificing throughput. As shown in the optical diagram (Figure 1), incoming light focused onto the entrance slit is directed by a 45° mirror (M1) to a collimating mirror (M2). From here, light is dispersed by the grating (G) onto mirror M3. Since the grating sorts light according to wavelength depending on the grating angle, only a narrow, specific band of light will be directed out the exit slit by the last mirror (M4). The narrowness of that bandpass is determined by the slit width.

Two slit arrangements are offered for these spectrometers. The 1679F Straight Slits slide into slots at the entrance and exit ports of the spectrometers. The 1679N or the 1679X Slits, on the other hand, are continuously adjustable from 50 to 5000 microns.

MINIMATE-2 Model 1681A has its grating rotated, and wavelength selected, by means of a knob on its front panel.

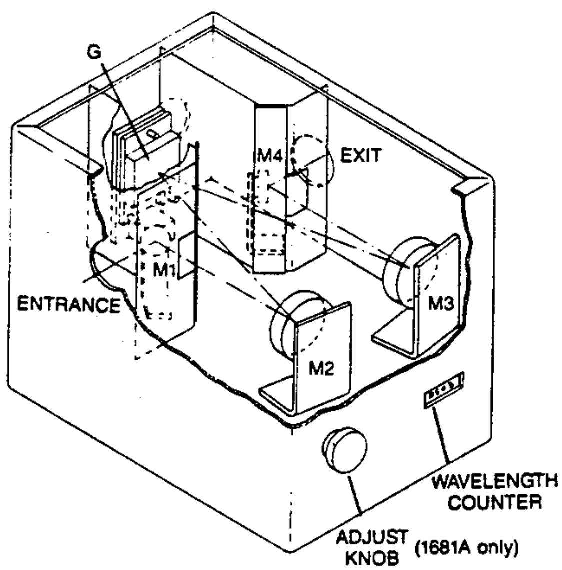


Figure 1
Minimate-2 Model

MINIMATE-2 Model 1681B comes with internal drive circuitry that couples directly to a SPEX MINIDRIVE-2, CD2 COMPUDRIVE scan controller, or the DM3000 spectroscopy computer.

MINIMATE-2 Model 1681C eliminates mirror M4 and has a longer-focal-length mirror in position M3 to create a wide, flat field at the focal plane that is ideal for OMA's. The range of wavelengths available from your instrument will be dictated primarily by the choice of grating.

The 340E Spectrometer has a longer focal length than the 1681 Minimate, the longer focal length increases the wavelength dispersion at the exit plane. The additional room allowed by the longer focal-length mirror makes it possible for the 340E to accept optional swingaway mirrors on the entrance and exit ports.

The 340S Spectrometer has a 220 mm focal-length mirror and a fixed 45° entrance port creating an f/4 input aperature. Dual entrance ports are not available with the 340S.

1 Specifications (with 1200 gr/mm grating)

	1681 A and B	1681 C	340E	340S
Focal Length mm	220	340 exit	340	220 input/ 340 output
Dispersion nm/mm	3.7	2.5	2.5	2.5
Resolution nm	0.25 at 546 nm, 50 μm slit	0.5	0.15	0.15
Aperture f/n	4	4	5.9	4
Mechanical Range nm	30-1000	0-1000	200-1000	200-1000
Accuracy nm	±0.5	±0.5	±0.5	±0.5

2 Optical Components

Your spectrometer has been aligned at the factory and, except for unpacking, the instrument should be ready to operate. Should you find, however, that the wavelength of the mechanical counter (Section 3) does not agree with the actual wavelength passing out the exit slit, proceed to Section 4. (Counter reads true for 1200 gr/mm gratings only).

The top cover of your MINIMATE-2 can be removed after the four screws have been removed.

2.1 Gratings

The grating supplied with your spectrometer can be installed by placing it in the mounting bracket (Figure 2B). (Figure 2A shows the adjusting screws).

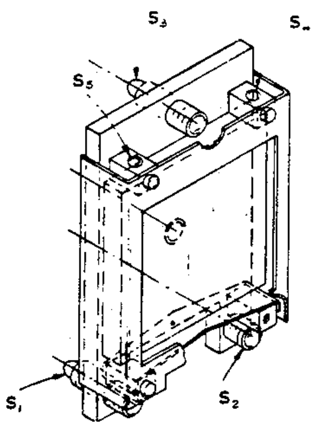


Figure 2A
Grating Mount

An alternative mount is shown in Figure 2B. In this case, a slotted leaf spring is pulled up to remove a grating, pulled back and pushed down to install. The keyhole slot in the leaf spring catches the head of a stud in the rear of the grating holder.

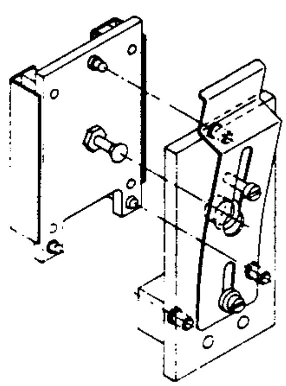


Figure 2B

NEVER TOUCH THE SURFACE OF A GRATING, FOR FINGERPRINTS CAN RARELY BE REMOVED TO ANYONE'S SATISFACTION.

Normally, a grating that has been aligned in the instrument need not be realigned each time it is installed. However, should realignment be necessary, consult Section 4. See Section 5 for technical considerations.

2.2 Mirrors

The mirrors in your spectrometer require no routine maintenance. Still, care should be exercised to prevent damage to their surfaces which will seriously degrade throughput. In general, your spectrometer should be kept in an atmosphere free of dust, corrosives, and smoke.

Fingerprints, if they are located soon enough, can usually be washed off by squirting the surface of the mirror with research-grade methanol from a squeeze bottle. For best operation, the collimating mirrors should be fitted with masks. Position masks so to block the inner edges of the mirrors (Figure 3).

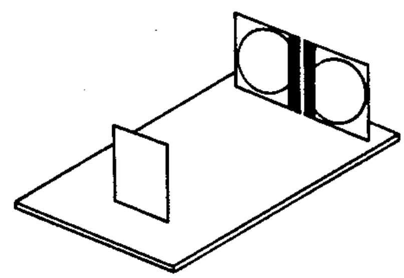


Figure 3
Grating Masks

2.3 Slits

Both entrance and exit slits influence the bandpass of your instrument. Because of this, the most consistent results call for operating the monochromator with both slits at the same width.

2.3.1 1679F Interchangeable Slits

These slits slide into mounts on the entrance and exit ports of your spectrometer. Calibrated in widths of 0.25, 0.5, 1.25, 2.5 and 5.0 mm, they yield bandpasses of approximately 0.9, 1.8, 4.5, 9.0 and 18 nm with the 3.7 nm/mm dispersion of a standard 1200 gr/mm grating in a 1681. In a 340E, 340S, or 1681C, this converts to 0.6, 1.2, 3.6, and 12.5 nm.

2.3.2 1679N and 1679X Continuously Adjustable Slits

Your spectrometer has continuously adjustable slits located at the entrance and exit ports. Note that each is calibrated in units of 100μ. Since the dispersion of the 1681 with standard gratings is 3.7 nm/mm, rotating the slit barrel by one division allows an additional 0.37 nm of light to pass through the instrument. For example, a slit setting of 0.5 mm translates into a 1.85 nm bandpass.

2.4 Optional Grating Turret for 1681/340 Spectrometers

The SPEX dual grating turret allows the researcher to vary spectrometer dispersion or range with a simple twist of a knob. Such performance is handy when the spectrometer is fitted with a multichannel device or when the spectrometer must be scanned efficiently over an extended range.

2.4.1 Operation

The position of the dual grating turret is determined by a knob on the top of the spectrometer. To change grating selection, pull up gently on the knob and turn it 180 degrees until it hits a stop. Then gently lower the turret into place. To change back to the original grating, repeat sequence, rotating in the opposite direction.

3 Mechanical Components

3.1 Wavelength Counter

On your spectrometer's front panel is a four-digit wavelength counter that displays the central wavelength of light that will pass through the spectrometer. This counter is calibrated in nanometers. The displayed wavelength is correct *only* when a 1200 gr/mm grating is installed. For other gratings, multiply the displayed wavelength by grating groove density/1200. You should also be aware that grating monochromators pass more than one order of diffracted light. That is, for any counter reading λ, the monochromator also transmits wavelength λ/n, where n is an integer. In the majority of cases, other aspects such as grating blaze and source output negate any spurious contributions from overlapping orders. Yet the existence of the phenomenon should always enter into the interpretation of results.

3.2 Drive System

3.2.1 Model 1681A, 1681C

The central wavelength of the 1681A and 1681C is determined by rotating the knob on the front plate while observing the counter reading. Always approach the final wavelength from at least 10 nm below.

3.2.2 Model 1681B, 340E/S

Your spectrometer scans its spectral range with a drive system that is linear in wavelength. Gratings can be rotated in steps as small as 0.02 nm (standard grating).

The spectrometer should be connected to a source of the *proper* voltage, then turned on.

The SPEX MINIDRIVE-2, CD2 COMPUDRIVE and DM3000 Spectroscopy Computer systems connect directly to the spectrometer for full control. Or you could use your own device or computer. To help you interface the systems, a driver schematic is included in the appendix.

4 Grating Adjustment

CAUTION: Never touch the diffracting surface of a grating. The surface is easily marred and cannot be cleaned as a mirror can.

The following three adjustments should be made on a grating the first time it is installed in the instrument. Whenever a grating is reinstalled the adjustments should be checked, but no adjustment will generally be required.

The wavelength calibration and vertical adjustment can be done in either order, but the rocking should be done last. For all three procedures the cover of the monochromator must be removed, and a mercury pen lamp installed at the entrance slit.

4.1 Wavelength Calibration

Set both slits to 0.25 mm, and set the monochromator to 546.1 nm. Unless the wavelength is grossly out of calibration, a green line will be seen through the exit slit. If not, adjust the wavelength until the line is centered; if it does so within 0.5 nm of 546.1, no calibration is required. Otherwise, set the monochromator at 546.1 nm and (referring to Figure 2), adjust screw S1 on the mount of the grating to bring the line through the instrument. If more than 1/8 turn is required, also adjust S2 in the opposite direction to balance the total adjustment between them.

Further accuracy may be had by viewing the image through the slit and adjusting for maximum intensity.

Replace the monochromator's cover. If photoelectric detection is available, scan slowly from 554 to 548 nm to ascertain that the peak occurs within 0.5 nm of 546.1.

4.2 Vertical Adjustment

Set the monochromator at 546.1 nm. Observe the image at the exit slit. If it is not vertically centered, center it by adjusting screw S3 (Figure 2) on the grating.

4.3 Rocking Grating

The vertical adjustment must have already been done. Set the monochromator at 0.0 and check whether the exit slit image is still vertically centered. If it is not, the grating must be rocked. While sitting at 0 nm repeat the vertical adjustment as above; then return to 546.1 nm. Adjust screw S4 (Figure 2) to recenter the image; if more than 1/8 turn is required, adjust S5 in the opposite direction to balance the adjustment between them. Set the monochromator to 0 nm and check the centering; if it is off, repeat the vertical adjustment. When the grating is properly aligned, the image will be vertically centered at both 0 and 546.1 nm.

5 Technical Note and Applicable Equations

5.1 Fundemental grating equation applied to Czerny-Turner mount

$m\lambda = d (\sin \alpha + \sin \beta)$ where

m = order
λ = wavelength
d = grating of incidence
α = angle of incidence
β = angle of diffraction

5.2 Theoretical Resolving Power

$$R_T = \frac{\lambda}{\Delta\lambda} = \frac{\nu}{\Delta\nu} = (\sin \alpha + \sin \beta)\frac{W}{\lambda} = mN$$

λ = wavelength
ν = wavenumber
N = total number of grating grooves
W = width of grating ruling
m = order of diffraction

These expressions all yield numbers which are wavelength dependent though if resolution is expressed as $\Delta\nu$, it is independent of the wavelength or frequency observed.

5.3 Factors Influencing Resolution

Source — Since resolution is a linear function of grating width (i.e., optical path difference), resolution deteriorates if the source illuminates less than the full width of the grating. As a consequence, the source or condensing lens should fully illuminate the collimating mirror. This can usually be checked visually by opening the spectrometer or, in the case of energy outside the visible spectrum, by making certain that throughput is reduced when the edges of the collimating mirror are obstructed.

Slit Width — Slits seem to be the most misunderstood parts of the spectrometer. A good habit to cultivate is thinking in terms of spectral bandpass rather than mechanical slit width. Bandpass is a function of the reciprocal linear dispersion which, in turn, depends on the wavelength, the grating constant, the focal length of the instrument and the spectral order (see Section 2.3).

5.4 Factors Influencing Throughput

Source — Maximum throughput is attained whenever the source subtends at least as large a solid angle at the slit as does the collimating mirror in the spectrometer. When the source is too small, or cannot be brought close enough to the entrance slit, the relay optics must meet this criterion.

Slits and Grating — When using photoelectric detection, the careful choice of combinations of slit widths and grating orders can often pay large dividends in throughput. For instance, by going from the first to the second order of the grating the dispersion is doubled (usually at a price of about 10-15% loss in efficiency); this doubled dispersion permits the slits to be set twice as wide (for the same resolution) and the throughput increases quadratically. Changing to a grating with a smaller ruling constant (more grooves/mm) provides the same gains without the loss in grating efficiency resulting from the use of higher orders.

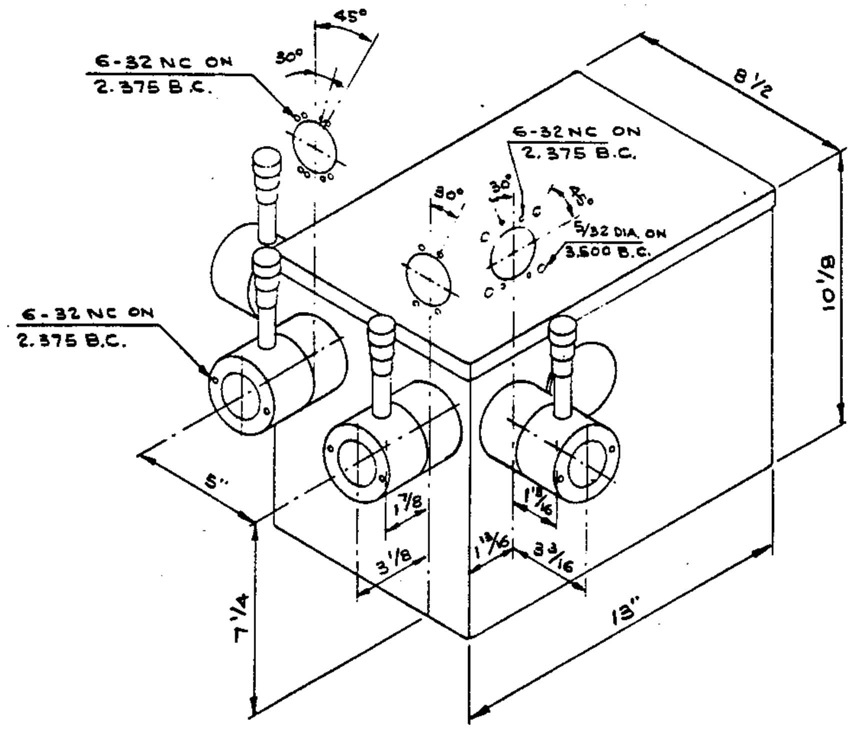
The grating is most efficient at its blaze wavelength. The efficiency falls faster toward shorter wavelengths than toward longer wavelengths. A good rule of thumb for a grating with a blaze at λ, is that the efficiency will fall about 50% of maximum at $2\lambda/3$ and 2λ.

Overlapping spectral orders can become increasingly troublesome at longer wavelengths. At any position of the grating rotation the wavelengths at the exit slit are uniquely determined by the geometry of the optical system. The difficulty arises since the same geometry also brings integral higher frequency harmonics (shorter wavelengths) to the exit slit at the same time. Thus, unless the detector is blind to the unwanted orders, or is made blind by filtering, it will respond to wavelengths other than those of interest. For instance, a photomultiplier with an S-1 photocathode viewing a grating rotated to bring wavelengths at 1 micron to the exit slit will also receive and respond to energy at 500 nm. Other photocathodes are sensitive to ultraviolet when used in the visible. In such cases a glass lens or the glass bulb of the photomultiplier itself may do the filtering since soda-lime glass cuts off at approximately 300 nm.

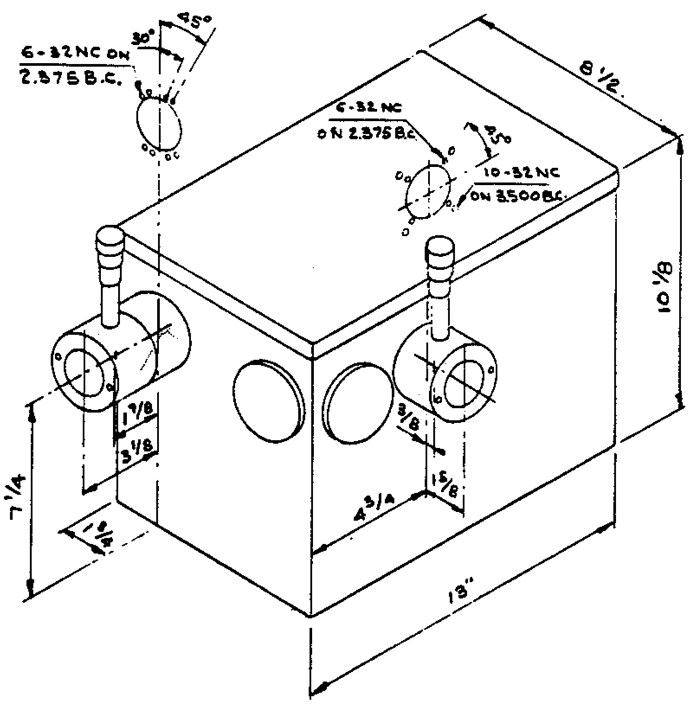
6.0 Pinouts of the Control Electronics

Pin #	CONTROL INPUT (25-pin female)
9	N/C
10	DIRECTION
11	STEP
12	LSWL (Limit Switch Low)
13	LSWH (Limit Switch High)
21	RET
22	RET
23	DIR/RET
24	STEP/RET
25	LSW/RET

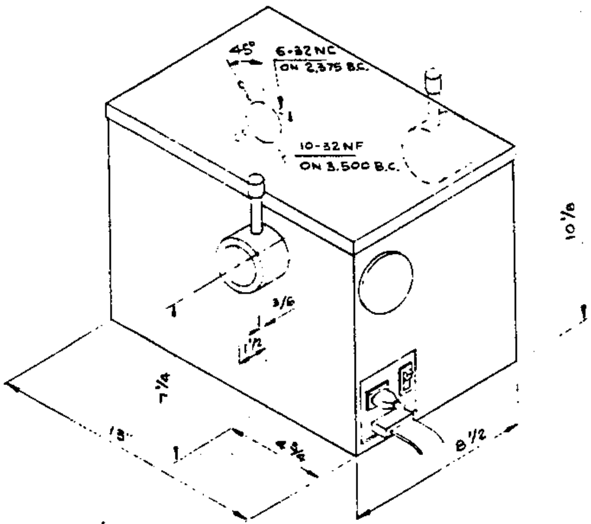
Maximum Speed is 500 steps/s.



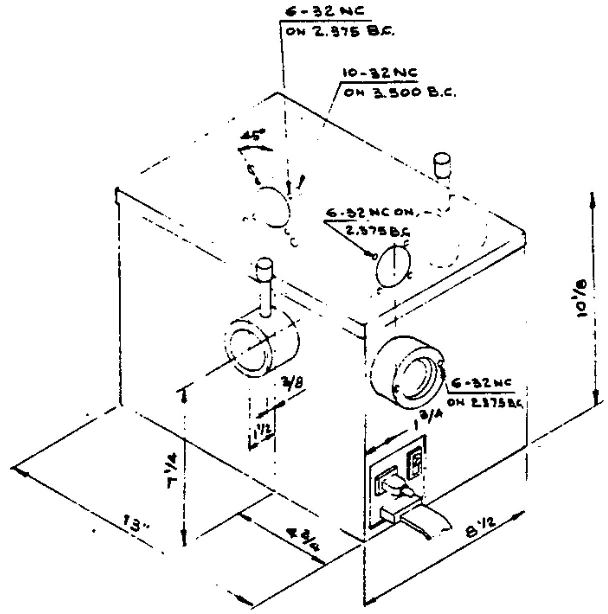
340E Spectrometer



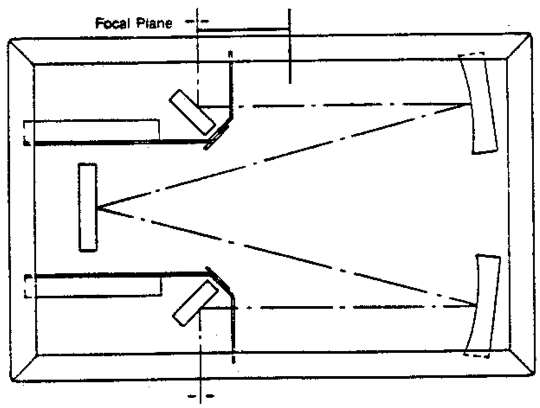
340S Spectrometer



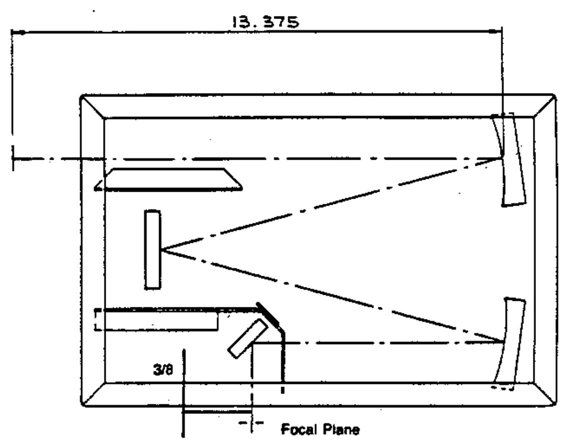
1681 A & B MINIMATE



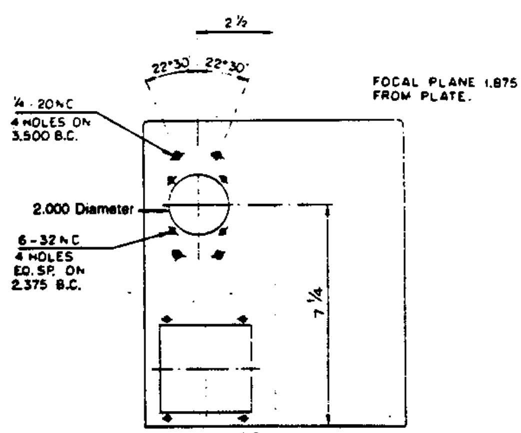
1681 C MINIMATE



1681 A & 1681 B MINIMATE 2



1681 C MINIMATE 2



1681 C Exit Port Schematic

5 Code

```
/*
Program to control stepper driven monochromator
Program will send grating to initial position defined in code, step a set interval of wavelength each time a key is hit
ex. program starts, send stepper to 100nm, hit # to go to 105nm, hit it again to go to 110nm, etc. hit * to home
home is defined by 14911 steps off limit switch, which corresponds to 360nm
max physical range is set to 1030nm, 1031nm from ASD
written by Liam O'Connor January 2020
 */

#include <Stepper.h>
#include <Wire.h>  // Include Wire if you're using I2C
#include <stdlib.h>   //library which includes ascii to in converter
#include <SFE_MicroOLED.h>  // Include the SFE_MicroOLED library
#include "SparkFun_Qwiic_Keypad_Arduino_Library.h" //Click here to get the library: http://librarymanager/All#SparkFun_keypad

KEYPAD keypad1; //Create instance of this object

/////////////////////////
// MicroOLED Definition //
//The library assumes a reset pin is necessary. The Qwiic OLED has RST hard-wired, so pick an arbitrarty IO pin that is not being used
#define PIN_RESET 9
//The DC_JUMPER is the I2C Address Select jumper. Set to 1 if the jumper is open (Default), or set to 0 if it's closed.
#define DC_JUMPER 1

///////////////////////////////////////////////////////////////////
// MicroOLED Object Declaration //
MicroOLED oled(PIN_RESET, DC_JUMPER);    // I2C declaration
char button;                          //define what the button press equals
const double homeWavelength = 360.0;       //wavelength of home position
double wavelength = homeWavelength;    //global variable for the wavelength the user enters
double currentWavelength = 360.0;      //wavelength machine was set to before entering new numbers
const double stepsPerWave = 18.76;          //conversion factor  of motor steps to wavelength change
const double interval = 10.0;               //number of nm stepped each interval 2,5 ,or 10 nm
int steps;                        //number of steps motor will move, equal to wavelength by stepsPerWave conversion factor
const int stepsPerRevolution = 200;  // change this to fit the number of steps per revolution for your motor
// initialize the stepper library on pins 2 through 5:
Stepper myStepper(stepsPerRevolution, 2, 3, 4, 5);

void setup() {             ///////////////////////////////////////////////////////////////////////////////////////////////////
  // initialize the serial port:
  Serial.begin(9600);             //connect to serial monitor
  myStepper.setSpeed(100);               // set the speed at X rpm:
   delay(100);
  pinMode(7, INPUT_PULLUP);               // initalize home limit switch, Normally open switch returns 1/HIGH until pressed
  Wire.begin();
  oled.begin();    // Initialize the OLED
  oled.clear(ALL); // Clear the display's internal memory
  oled.display();  // Display what's in the buffer (splashscreen)
  delay(1000);     // Delay 1000 ms
  oled.clear(PAGE); // Clear the buffer.

  randomSeed(analogRead(A0) + analogRead(A1));

}



void loop() {      //main loop   ////////////////////////////////////////////////////////////////////////////////////////////////
  Serial.println("Monochromator control");

  if (keypad1.begin() == false)   // Note, using begin() like this will use default I2C address, 0x4B.
              // You can pass begin() a different address like so: keypad1.begin(Wire, 0x4A).
  {
    Serial.println("Keypad does not appear to be connected. Please check wiring. Freezing...");
    while (1);
```

```
  }
  //Serial.print("Initialized. Firmware Version: ");
  //Serial.println(keypad1.getVersion());
  Serial.println("Enter a wavelength in nanometers between 360-1020 and press # to go.");
  texttoScreen();          // the  function to display the wavelength on the OLED
  do{
  getbutton();            //function to get keypad inputs, repeats until entry is within range
  if (wavelength >= 1031){                      //check that wavelength is within monochromator range
    printTitle("Out of range, try again", 0);
  }
  else if (wavelength < homeWavelength){
      printTitle("Out of range, try again", 0);
  }
  }
  while(wavelength >= 1031 || wavelength < homeWavelength);
  if (wavelength < 1031 && wavelength > homeWavelength){      //will only go to waelength if in bounds
  gotoWavelength();
  }

  printWave();
  do{                              //loop to move stepper in 5 or 10 nm steps each time the # key is hit until * is hit sending it home
    getbutton();                     //use get button function to check for home key (*) or continue (#) key
     Serial.println(button);
     delay(500);
    if (button == '#'){
      Serial.println("inside if loop, button =  ");
       Serial.println(button);
     delay(500);
    moveInterval();
    }
  }while(button != '*');
  Serial.println("end of program");
  delay(500);
}/////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

void printWave(){                        //////////////////function to print wavelength
  oled.setFontType(0);  // Set font to type 0,1, or 2 for size
  oled.clear(PAGE);     // Clear the page
  oled.setCursor(0, 0); // Set cursor to top-left            //print to oled
  oled.print("wavelength = ");
  oled.print(wavelength);
  oled.print("  * to home, # to move ");
  oled.print(interval);
  oled.display();      // Refresh the display
  delay(500);

}

void moveInterval(){        ////////////////  function to move the stepper by same number of steps each time the continue key (#) is hit
   steps = interval * stepsPerWave;
    myStepper.step(steps);
    wavelength = wavelength +interval;      //increase the wavelength counter
    printWave();                         //display wavelength to oled
}

void goHome(){               //function to step one at a time until home switch is triggered, then roll  back to 350 nm//////////////////////////
  Serial.println("Hit the * key to go home and reset the machine.");
   oled.setFontType(0);  // Set font to type 0,1, or 2 for size
  oled.clear(PAGE);     // Clear the page
  oled.setCursor(0, 0); // Set cursor to top-left            //print to oled
  oled.print("wavelength =");
  oled.print(wavelength);
  oled.print(" hit * to go home");
  oled.display();      // Refresh the display
  delay(1000);
  char button = keypad1.getButton();
```

```
  do{ keypad1.updateFIFO();        //do loop to wait until user hits **** key to send it home
      button = keypad1.getButton();
      Serial.print("button = ");
      Serial.println(button);       //testing printing variables
      delay(200);
      }while(button != '*');
  do{                            // step reverse one step at a time until the home microswitch is triggered
    myStepper.step(-1);
}while(digitalRead(7) != 0);
delay(1000);
myStepper.step(14911);   //steps the grating back to spectrally calibrated 360nm
wavelength = homeWavelength;

Serial.print("I'm Home!");
 oled.setFontType(0);  // Set font to type 0,1, or 2 for size
  oled.clear(PAGE);     // Clear the page
  oled.setCursor(0, 0); // Set cursor to top-left             //print to oled
  oled.print(" I'm home now!");
  oled.display();       // Refresh the display
  delay(2000);
  currentWavelength = homeWavelength;
}



void gotoWavelength(){       //function to go number of steps until wavelength is reached/////////////////////////////////////////////////
  Serial.print(" wavelength in go to wavelength loop = ");
  Serial.println(wavelength);
   //steps = (wavelength - homeWavelength)*stepsPerWave;
   steps = (wavelength - currentWavelength)*stepsPerWave;
   myStepper.step(steps);
   delay(100);
   currentWavelength = wavelength;
 }

void getbutton(){        //function to get button inputs on keypad
////////////////////////////////////////////////////////////////////////////////
  Serial.println("get button function");

  keypad1.updateFIFO();  // necessary for keypad to pull button from stack to readable register
  int digits [5]= {1,1,1,1,1};         //5 coloumn array to store digits from fifo
  Serial.print(digits[0]);                //not needed jut for debugging
  Serial.print(digits[1]);
  Serial.print(digits[2]);
  Serial.print(digits[3]);
  Serial.println(digits[4]);
  int i=0;                  //counter variable to control array index

   Serial.println("in else loop");
  do{ keypad1.updateFIFO();  // don't need as it clears fifo each loop, left while debugging
      button = keypad1.getButton();
      Serial.print("button = ");
      Serial.println(button);       //testing printing variables
      //Serial.println("in do while loop");
      delay(200);
      if (button == '*'){                              // going home function here that back steps untill switch is hit then roll off and reset
        goHome();
        Serial.print(" Going Home");
        break;                              //exit do loop
      }
      else if(button != 0)       // '1'||'2'||'3'||'4'||'5'||'6'||'7'||'8'||'9'|| '#'|| '*')
       {digits[i] = (button - '0');    //fill array with button presses allowing wavelength of 9999 to be entered
       i++;}                  //increment array position
    }while(button != '#');    // fill array until user hits pound key to go to enetered wavelength

 // if (digits[0] == '#'){         //series of if statments that take digits from array and create wavelength between 0-9999nm
```

```
 // }
  if (digits[1] == -13){         // check for -13 as it is ascii for #, which we are using as our enter/go key
    wavelength = digits[0];
  }
  else if (digits[2] == -13){
    wavelength = digits[0]*10 + digits[1];
  }
  else if (digits[3] == -13){
    wavelength = digits[0]*100 + digits[1]*10 + digits[2];
  }
  else if (digits[4] == -13){
    wavelength = digits[0]*1000 + digits[1]*100 + digits[2]*10 +digits[3];
  }
Serial.print("wavelength = ");         //print the wavelength value as the program sees it (if incorrect the  above array and if statements are
buggy)
Serial.println(wavelength);
Serial.print(digits[0]);
Serial.print(digits[1]);
Serial.print(digits[2]);
Serial.print(digits[3]);
Serial.print(digits[4]);
delay(500);


// We prompt for G - but never check for it       ///block of code to wait for serial input to assist with debuging
//Serial.println("Press G and Enter to continue");
//  while(Serial.available() < 1)
  //{
  //}
// now we clear the serial buffer.
//while(Serial.available() > 0)
  //{
//byte dummyread = Serial.read();
  //}
                                    /////////////////testing code ends here
}

// Center and print a small title////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// This function is quick and dirty. Only works for titles one
// line long.
void printTitle(String title, int font){
  int middleX = oled.getLCDWidth() / 2;
  int middleY = oled.getLCDHeight() / 2;

  oled.clear(PAGE);
  oled.setFontType(font);
  // Try to set the cursor in the middle of the screen
  //oled.setCursor(middleX - (oled.getFontWidth() * (title.length()/2)),
         //      middleY - (oled.getFontHeight() / 2));
  oled.setCursor(0, 0);     //set cursor  top left
  // Print the title:
  oled.print(title);
  oled.display();
  delay(3000);
  oled.clear(PAGE);
}


void texttoScreen()     //function to print to screen///////////////////////////////////////////////////////////////////////////////////////////
{
printTitle("Enter wavelength, hit # to go", 0);

  // Demonstrate font 1. 8x16. Let's use the print function
  // to display every character defined in this font.
  oled.setFontType(0);  // Set font to type 0,1, or 2 for size
  oled.clear(PAGE);     // Clear the page
```

```
oled.setCursor(0, 0); // Set cursor to top-left
// Print can be used to print a string to the screen:
oled.print("wavelength =");
oled.display();      // Refresh the display
delay(2000);         // Delay a second and repeat
oled.setFontType(1);  // Set font to type 0,1, or 2 for size
oled.print(wavelength);
oled.display();
delay(500);
}
```

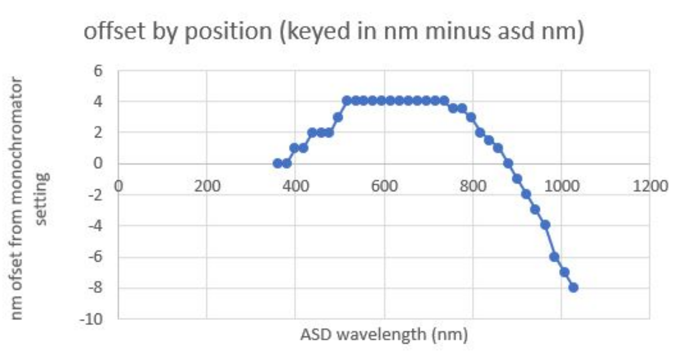


Figure 2: Offset values showing the non-linear response of the monochromator.

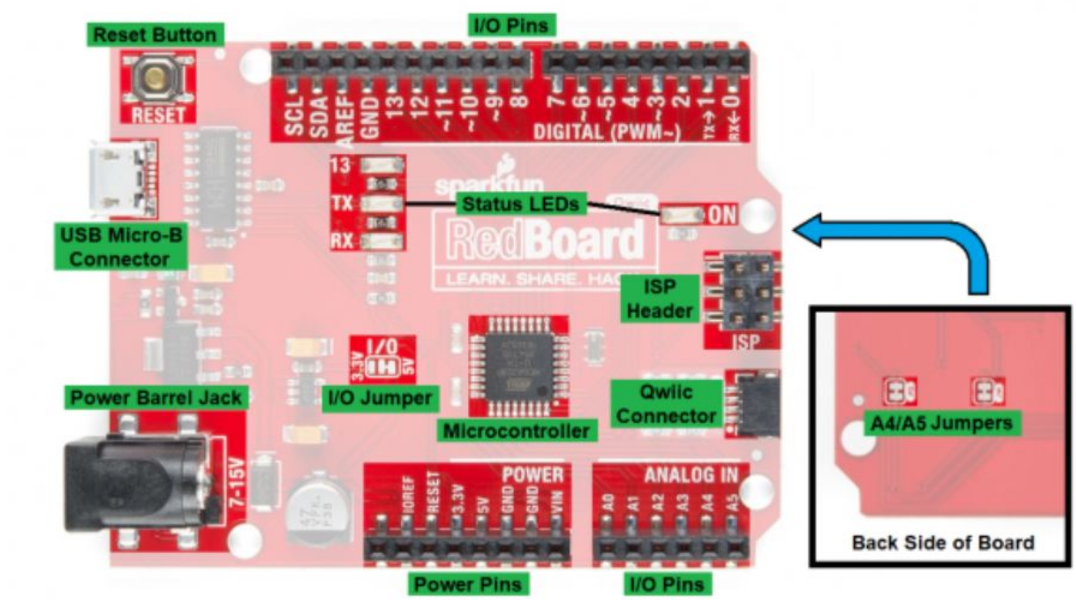


Figure 3: Sparkfun Redboard Qwiic based on ATmega328 chip.