Developer Guide

System Overview

For this project I created a two-channel oscilloscope out of a microcontroller. Specifically, I used an Arduino Uno to take in and process signals and I used MATLAB to organize and display the data in a user-friendly way. The GUI allows users to select which channels to display, adjust sensitivity, and start and stop readings. The PCB that I created for this project will allow users to connect to the microcontroller with a BNC connector.

Electrical Specifications

Interface	**Definition**
Arduino Power Supply	Vmin = 6V Vmax = 20V V(nominal) = 7 – 12V Imax = 500mA Inom = 200mA
Operating Temperature	-40 to 85 degrees C
Signal Input	Vmin = 0V Vmax = 5V Imax = 40mA

User Guide

Parts

- USB to mini-b cable
- 2x oscilloscope probes
- Microcontroller Oscilloscope

Assembly

- Connect the Oscilloscope to your computer via the USB cable
- Attach oscilloscope probes to the system. The labels indicate which connector is which channel.

Software Setup

- Download the analog processing code and view it through Arduino IDE.
- Upload the code to the system. Make sure the correct COM port is selected.
- Download the display code.
- Start MATLAB and run the display code.
- You should now have a window with an axes, two buttons, two checkboxes, and a slider.

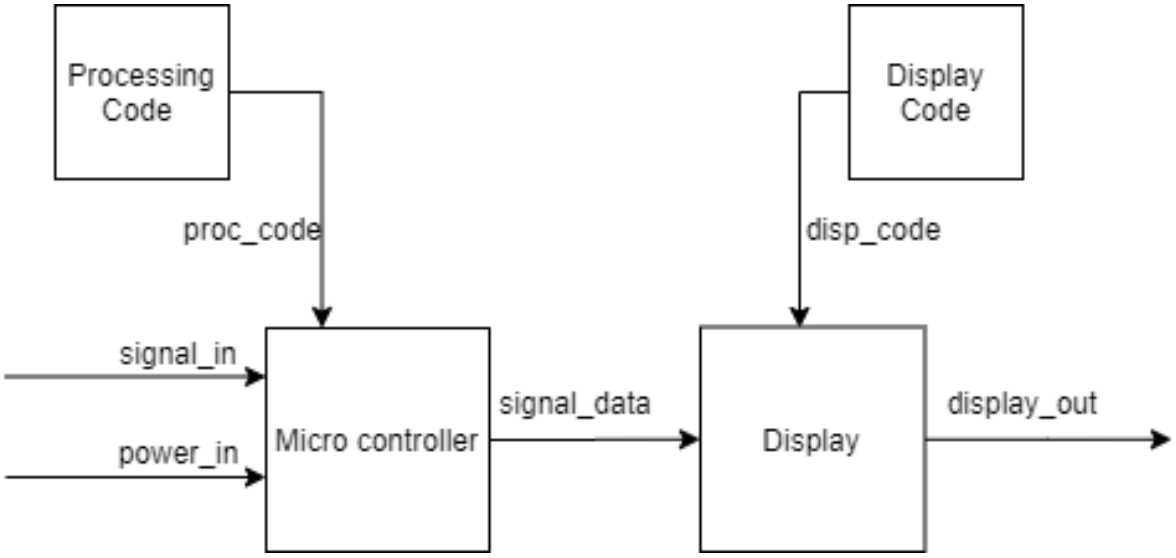
Running the Program

- To start reading voltages press the “Start” button.
- The check boxes on the left indicate which channels are active and will be displayed.

- Plots should start appearing on the graph. Channel 1 plots are red, and channel 2 plots are blue.
- Press the “Freeze” button at any point to freeze the graph. Press the button again to unfreeze and continue plotting.
- The “Horizontal Sensitivity” slider on the right adjusts the horizontal scale. It goes from one to one tenth scale.
- To exit the program, deselect both channels and exit out of the window.

Artifacts

Block Diagram and Interface Definitions



Interface Name	Interface Type	Specifics
signal_in	Analog Voltage	Vmin = 0V Vmax = 5V Imax = 40mA Sample frequency = 10 kHz
power_in	DC Power	Vmin = 6V Vmax = 20V V(nominal) = 7 – 12V Imax = 500mA Inom = 200mA
proc_code	Code	C language Arduino IDE
signal_data	Serial Data	Voltage values from 0.00V – 5.00V for each channel Baud Rate: 19200
disp_code	Code	Matlab code
display_out	Visualization	Displayed waveform on PC monitor

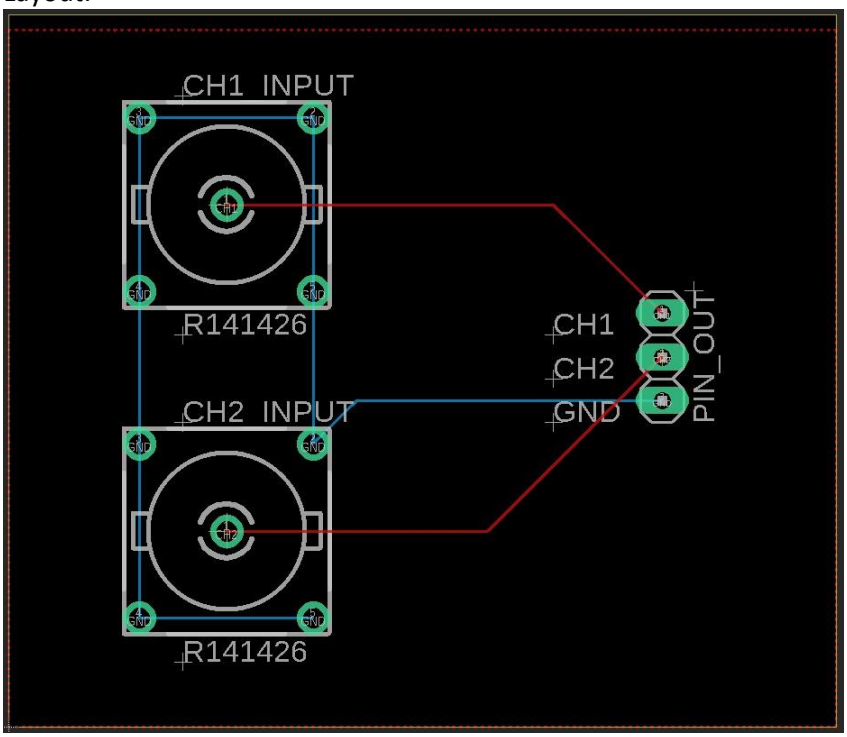
The block diagram shows that the two main parts of the system are the signal processing and the display. The microcontroller takes the signal input, signal_in, and sends the signal data to the computer which will display the data in a user-friendly manner. The signal input is limited to between 0 and 5 volts with a maximum current of 40mA. The sample frequency is 10kHz meaning the oscilloscope can accurately display waveforms with an upper frequency limit of 5kHz. The data is sent in chunks of 300 readings meaning that the oscilloscope has a lower frequency limit of 33Hz.

PCB Information

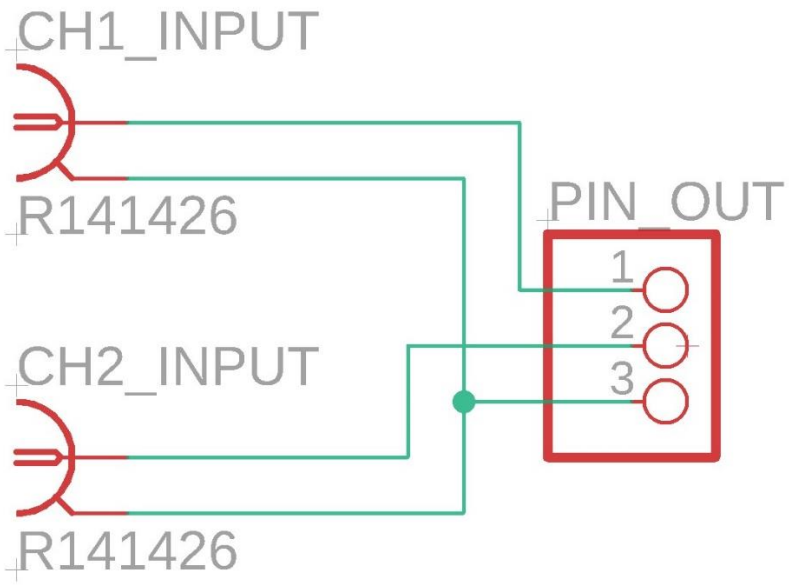
The PCB that I created is a simple BNC to pin header that will allow users to connect probes to the microcontroller via BNC connectors. There is a BNC connector for each channel.

Dimensions: 48mm x 41mm

Layout:



Schematic:



Picture:

Part Information

Component	Description	Quantity
Arduino Uno	Programmable Microcontroller	1
BNC Connector	Female BNC Connector for PCB	2