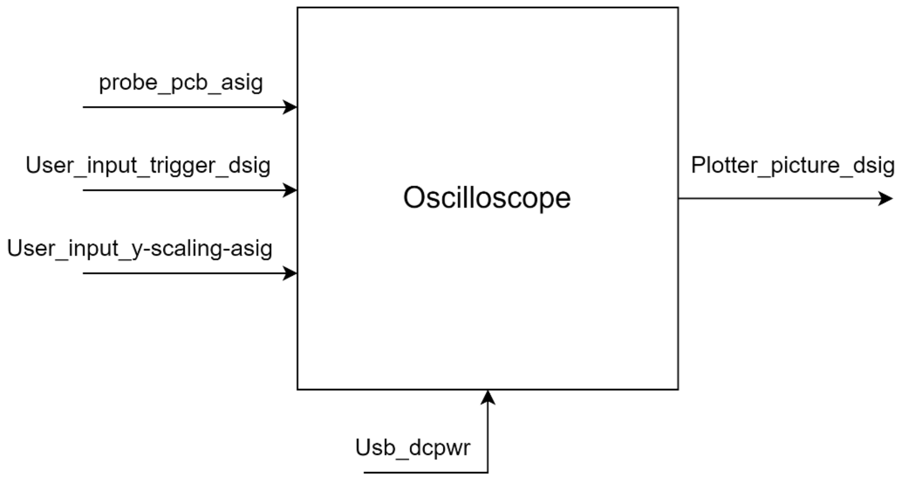
System Verification Documentation

Table of Contents

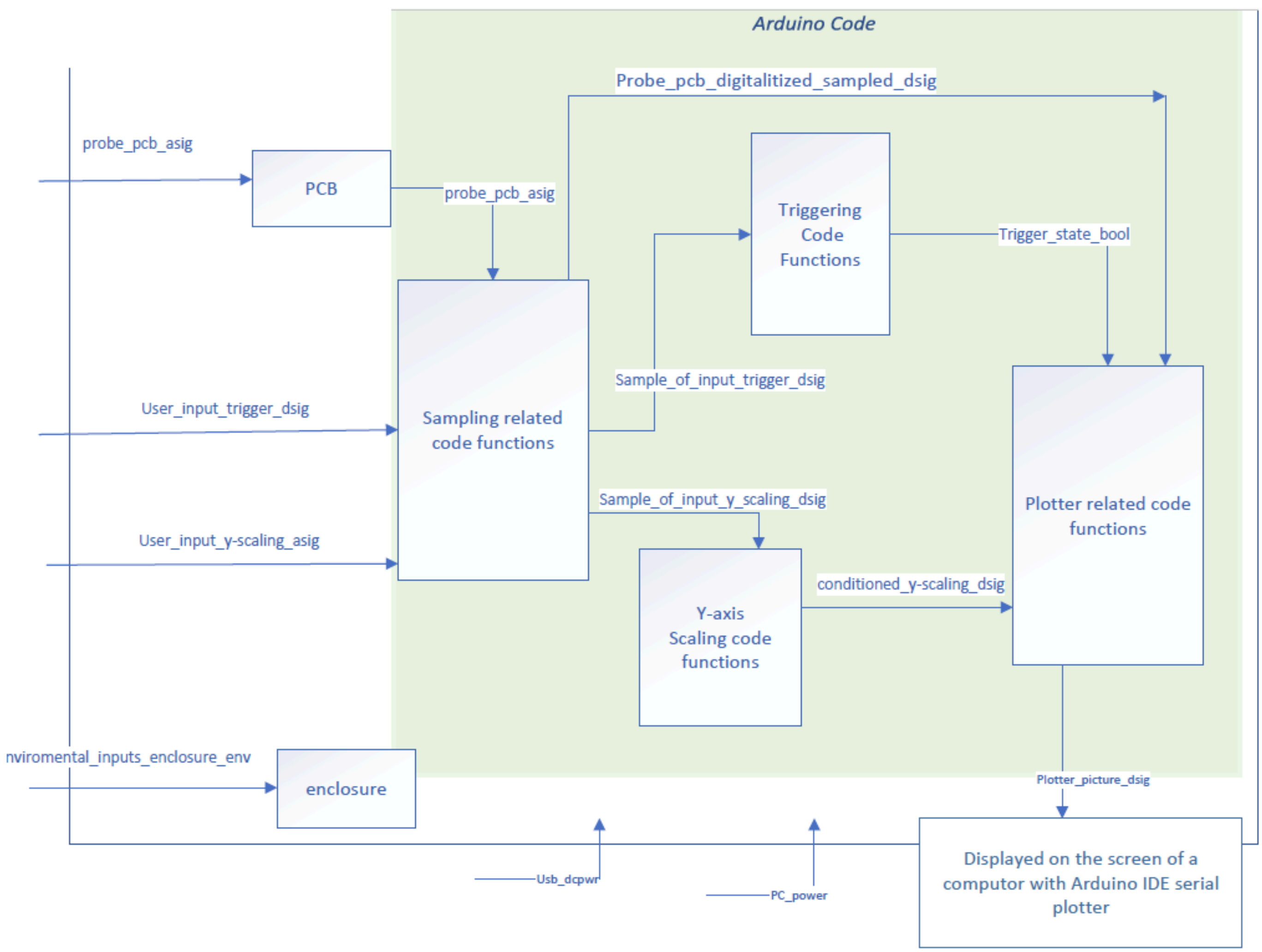
Black Block Diagram



System Function Description:

The system shown above is capable of measuring and displaying two analog signals simultaneously from easily detachable BNC probes. The oscilloscope will be able to sample each incoming signal at a rate of at least 25kHz. The oscilloscope will then generate a voltage waveform graph of the incoming signals in real time onto a connected laptop display via a USB connection. The system contains a user adjustable voltage knob that scales the y-axis between 0 –5V, and a trigger switch which will activate a user configurable trigger point to generate a freeze frame of the voltage waveform.

Top-level block diagram (all blocks in the system)



Complete Interfaces and Properties (entire system):

Interface Name	Interface Property
Probe_pcb_asig	Vmax: 5V Vmin: 0V Pins A0 & A1 input
User_input_trigger_dsig	Switch State = HIGH Switch State = LOW PIN 3
Trigger_state_bool	True False
User_input_y-scaling_asig	Vmax: 5V Vmin: 0V
Environmental_inputs_enclosure_env	None
Sample_of_input_trigger_dsig	14 Bit encode Periodically
Sample_of_input_y_scaling_dsig	14 Bit encode Switch State = HIGH Switch State = LOW
Conditioned_y-scaling_dsig	14 Bit encode Float 0-5
Probe_pcb_digitalized_sampled_dsig	14 Bit encode Digital Logic Level High: 1 Digital Logic Level Low: 0
Plotter_picture_dsig	14 Bit encode Float Value Range (0.00 – 5.00V)
USB_Dcpower	Vmax: 5.25V Vmin: 4.4V Imax: 500mA Inominal: 100mA
PC_power	[Depends on the connected computers specifications]

Test plans

Requirement 1 Test Plan

Engineering Requirement: The system must have at least two Channels.

Process:

1. With both oscilloscope probes connected to the PCB, one oscilloscope probe will be connected to a channel on the DC power supply, and the other oscilloscope probe will be connected to a second channel on the DC power supply.
2. There will be two distinct signals on the voltage waveform display through the Serial Plotter.

Requirement 2 Test Plan

Engineering Requirement: The system samples at a rate of at least 1MHz.

1. Open Arduino IDE
2. Select appropriate port, bootloader, and board in IDE
3. Connect USB from laptop to Arduino
4. Compile and upload sketch onto Arduino
    a. Sketch will have the inputs hardcoded in (no prompt or dynamic changing)
5. Connect the oscilloscope probe ground to the ground pin of the Arduino
6. Connect the oscilloscope probe to the PWM output pin specified in the Arduino sketch.
7. Power on oscilloscope
8. The Tektonics 4304 Oscilloscope can automatically detect the frequency of a signal. This feature is accessed under the measurements badge.
    a. Measure
    b. Add measurement
    c. Add frequency measurement
    d. Menu off
    e. Given a table with the measurements (frequency)
9. Observe that a PWM signal is generated with the attributes listed in interface table.

Requirement 3 Test Plan

Engineering Requirement: The probes must disconnect from the system using robust connectors.

Process:

1. Show that the BNC probes can be easily attached and detached from the BNC connectors. Take both probes off and screw them back on.

Requirement 4 Test Plan

Engineering Requirement: There must be a configurable trigger and adjustable time and voltage axis.

process:

1. Initiate the serial plotter so that it is actively displaying voltage waveforms,
2. Show the voltage potentiometer adjusting the y-axis.
3. Show that the trigger switch freezes the display as soon as the switch is flipped to represent the configurable trigger feature.

Requirement 5 Test Plan

Engineering Requirement: The system must respond to button presses in under 20 milliseconds.

process:

1. Show that the potentiometer and trigger switch work instantaneously on the serial plotter.
2. Check and uncheck the value boxes on the top of the serial plotter to show that the system responds to button presses near instantaneously.

Videos

Engineering Requirement: There must be a configurable trigger and adjustable time and voltage axis.

scaling.MP4

triggering.MP4

Engineering Requirement: The system must have at least two Channels.

2channels.MP4

Engineering Requirement: The probes must disconnect from the system using robust connectors.

modularity.MP4

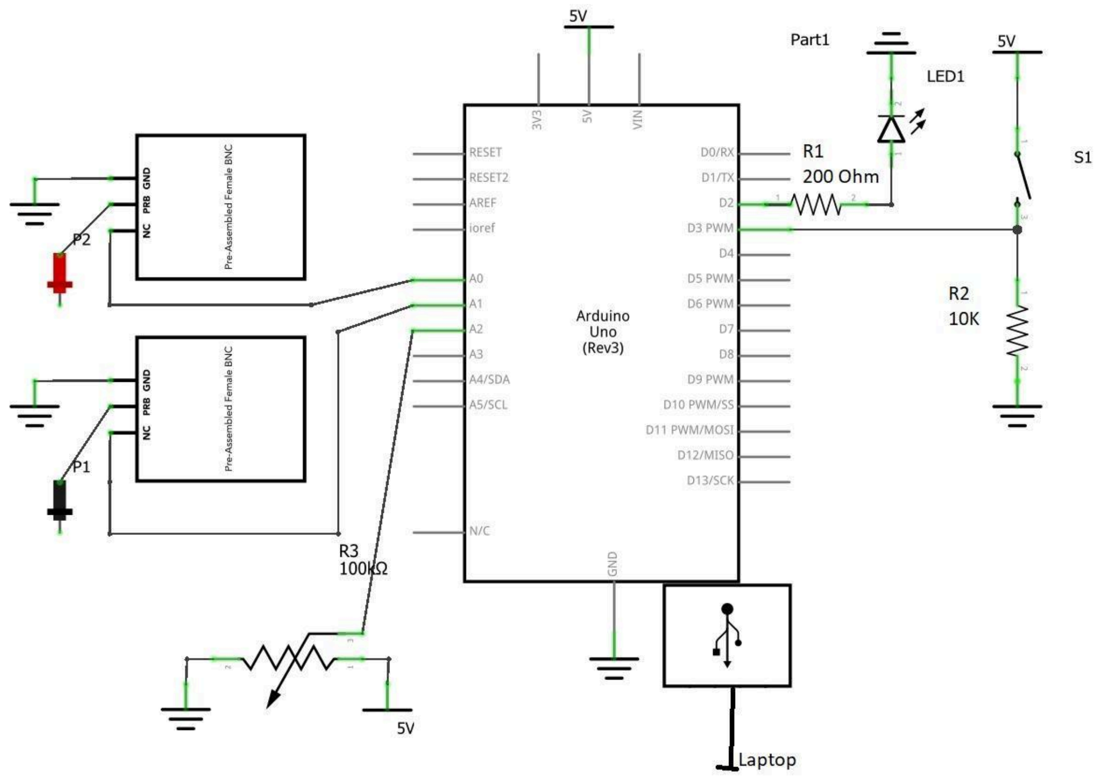
Engineering Requirement: The system must respond to button presses in under 20 milliseconds.

responsetime.MP4

Engineering Requirement: The system samples at a rate of at least 1MHz.

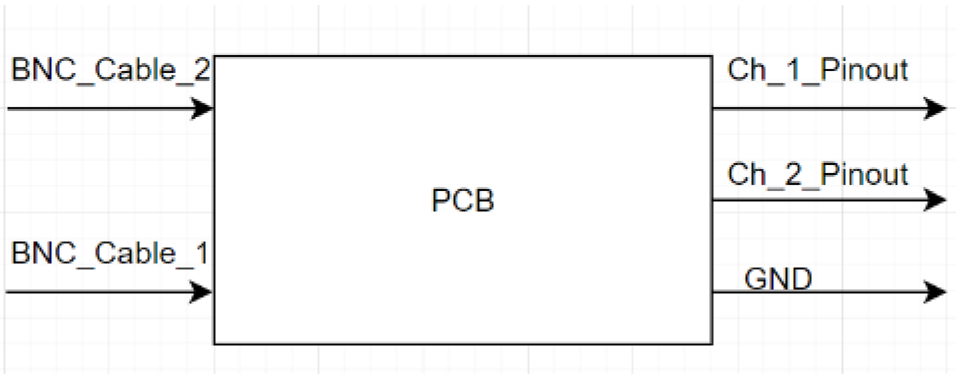
Sampling Rate Frequency from Tektronix Oscilloscope

Project Level Electrical Schematic:



PCB Design

Black Box Diagram:



KiCAD Schematic

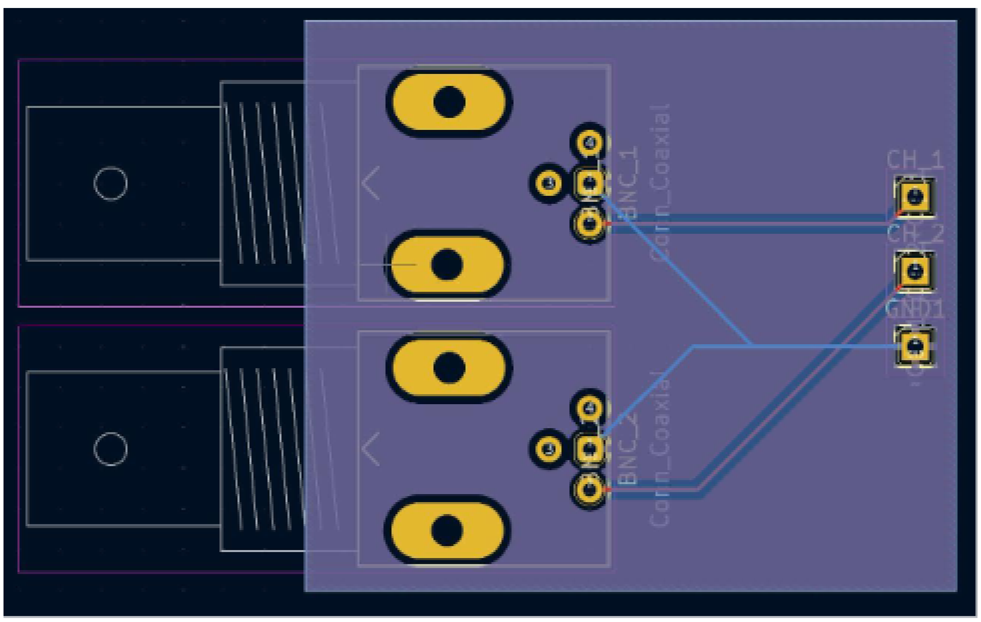


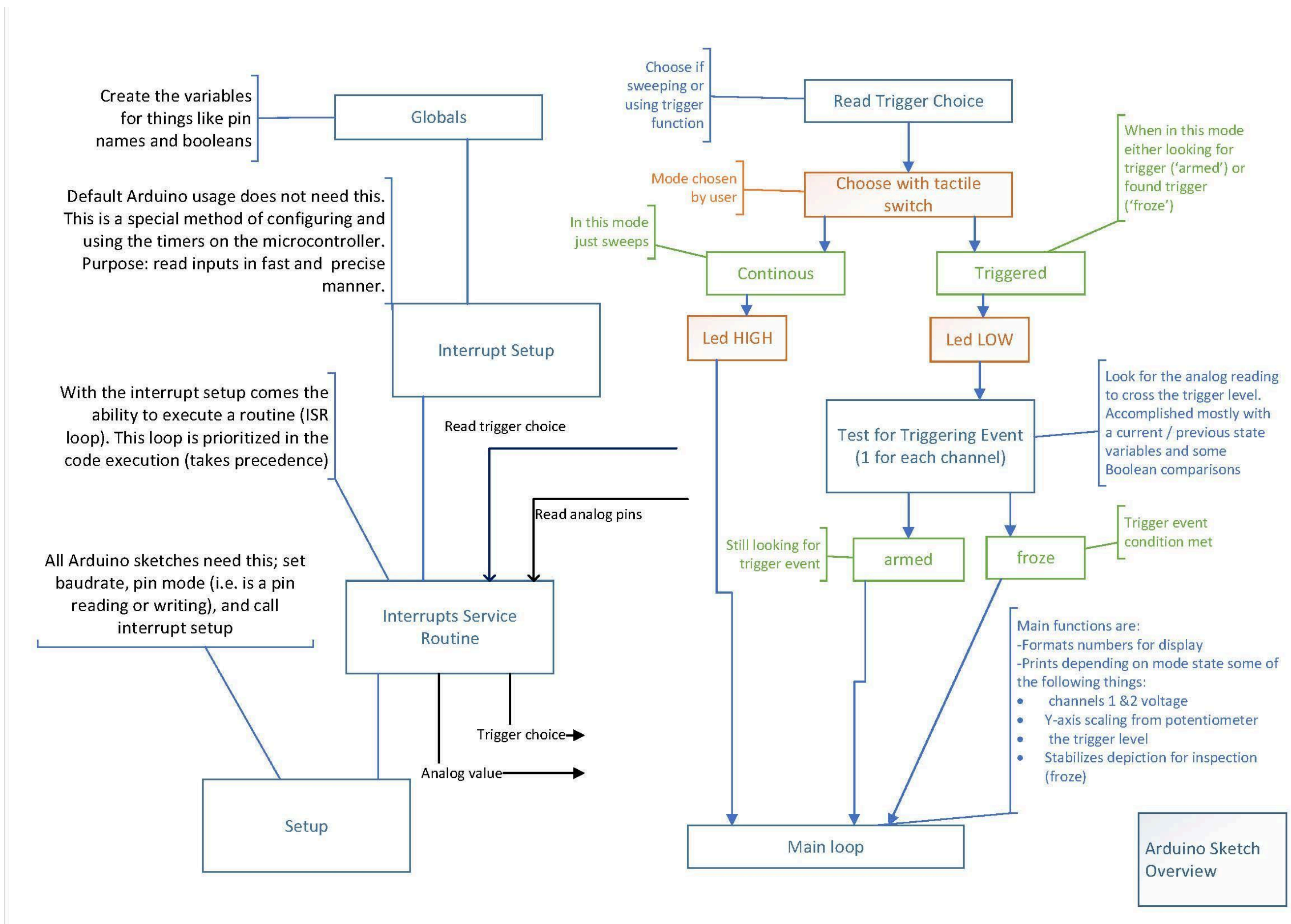
Figure 1

Description:

The PCB for our Oscilloscope design includes two slots for two different BNC Connector right angle jacks (seen left, Figure 1), which will then connect to BNC probes to provide modularity and multiple channels for the Arduino microcontroller to interpret data from. Each BNC Connector slot has a pin that relays the analog signal to a pinout (seen right, Figure 1) on the PCB for a total of two separate pinout analog channels labeled CH_1 and CH_2 respectively. Each BNC Connector is connected to the ground plane via a pair of wires that relay to a GND pinout which wire to the Arduino microcontroller. There are a total of two inputs for each BNC cable, and three output pinouts (2 channel analog signal pinouts, and a ground pinout for each).

Arduino Sketch Overview

Arduino Sketch block diagram



Complete code broken out by language (Double click to open the sub-document)

```
/*
  Names of Group Members: Gabriel Naomi | Asa Wiese
  Sketch: Working on final 1

  Citations:
  ****************************************************************
  File:  Oscilloscope.ino
  Title:  Standalone Arduino 6 channel  Triggered Oscilloscope
  Author: Meeker6751
  Verson: 2018.4.23
  URL:  Arduino forum/

  File: Arduino Interupts
  Author:  amandaghassaei
  URL: https://www.instructables.com/Arduino-Timer-Interrupts/
  ****************************************************************/
bool toggle0 = 0;  //for testing sample freq

#define ul unsigned long
/************************************************************/
#define triggered false  //  choose with switch
#define continuous true  //
bool SweepMode;          //  'continuous' or 'triggered'
/************************************************************/
#define armed true   //  these 2 inside of trigered mode
#define froze false  //  trigrd
bool TriggerState;   //  'armed' or 'froze'
/************************************************************/
#define falling false            //  trigger slope positive
#define rising true              //  trigger slope negative
bool TriggerDirection = rising;  //  'rising' or 'falling'
/************************************************************/

/************************************************************
volatile means can use in ISR like a global
************************************************************/
volatile float pot_state;
//bool toggle0 = 0;  //for testing sample freq

// adjustable variables
int TriggerVolts = 2;  //  trigger vdc; 0 <= TriggerVolts <= 5
int zero_line = 0;

// loop procedure variables
float ChannelFloor;
```

Interrupts and Interrupt Service Routine

The Timers

As mentioned, there are three timer peripherals on the Arduino Uno. To interface with them in this context we need to specify values in an eight-bit register provided for each. This is like a function for each timer that has eight parameters. There are tables in the ATMega32b datasheet that spell out what bits to put in the register for certain behaviors. For CTC we will be adjusting the last 4 bits. These logistics will be illustrated later. This section is meant to give an orientation to the timers. The following is a figure of the register described in the datasheet for an example timer (timer2) as well as a table that highlights the difference between the timers that are most relevant to CTC.

Bit	7	6	5	4	3	2	1	0	
(0xB1)	FOC2A	FOC2B	–	–	WGM22	CS22	CS21	CS20	TCCR2B
Read/Write	W	W	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Figure 2 Example timer register

Figure 3 Max compare match register for each timer.

Prescaler

Timers count up with one increment every clock cycle. The Arduino clock runs at 16 MHz In principle the timers increment that fast. A prescaler breaks that clock speed up so that a timer increments after a chunk of clock periods has elapsed instead of just one. This allows for practical usage of the timers. By default the timers are prescaled by 64. The following equation illustrates how to calculate the timer speed with a prescaler:

Prescaler Equation	
Pre-scaler dictates the speed of your timer	**(timer speed (Hz)) = (Arduino clock speed (16MHz)) / prescaler**

Figure 4 Prescaler Equation [1, p. 3]

Prescaler options
1
8
64
256
1024

Prescaler Specifics

Each timer has a register that controls specific attributes.

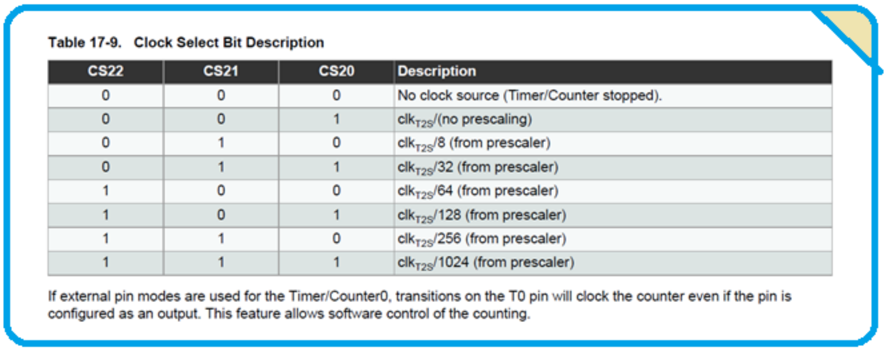


Table 17-9. Clock Select Bit Description

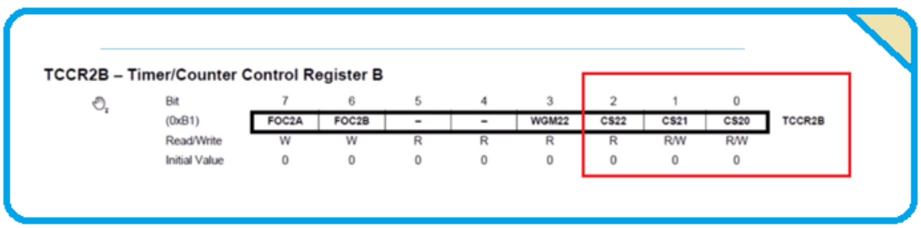
CS22	CS21	CS20	Description
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	clk_{T2S}/(no prescaling)
0	1	0	clk_{T2S}/8 (from prescaler)
0	1	1	clk_{T2S}/32 (from prescaler)
1	0	0	clk_{T2S}/64 (from prescaler)
1	0	1	clk_{T2S}/128 (from prescaler)
1	1	0	clk_{T2S}/256 (from prescaler)
1	1	1	clk_{T2S}/1024 (from prescaler)

If external pin modes are used for the Timer/Counter0, transitions on the T0 pin will clock the counter even if the pin is configured as an output. This feature allows software control of the counting.

Figure 5 Prescaler clock select bit.

[2, p. 126]

Prescaler: Relevant bits in bit register



TCCR2B – Timer/Counter Control Register B

Bit	7	6	5	4	3	2	1	0	
(0xB1)	FOC2A	FOC2B	–	–	WGM22	CS22	CS21	CS20	TCCR2B
Read/Write	W	W	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Figure 6 Relevant bits in bit register

Prescaler: adjusting -> code example

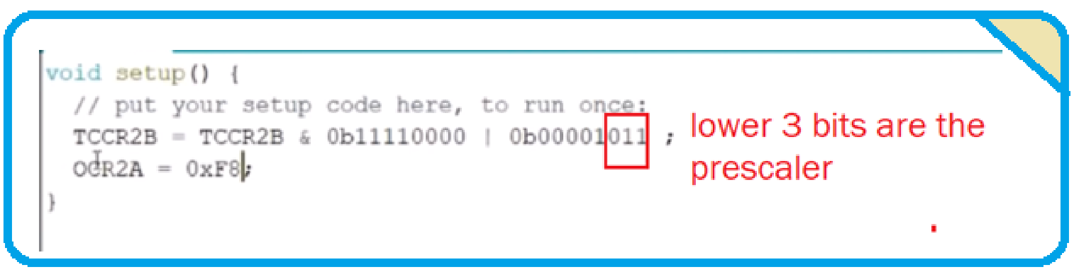


Figure 7 Prescaler: adjusting -> code example

[3]

Compare Match Value

	What the timer counts up to (max compare match register)
Timer 0	255
Timer 1	65535.
Timer 2	255

The other key attribute of the CTC method is the compare match value. This is important because it is how you achieve a particular interrupt frequency. Recalling each timer increments up to a certain value. One can specify that value. The mode where you specify what it counts up to is called the OCRA.

Compare Match Value: Equation

	Interrupt frequency equation
Interrupt frequency	**interrupt frequency (Hz) = (Arduino clock speed 16,000,000Hz) / (prescaler * (compare match register + 1))** the +1 is in there because the compare match register is zero indexed

Figure 8 Interrupt frequency equation

Compare Match Value: register guidance from datasheet

Table 17-8. Waveform Generation Mode Bit Description

Mode	WGM2	WGM1	WGM0	Timer/Counter Mode of Operation	TOP	Update of OCRx at	TOV Flag Set on(1)(2)
0	0	0	0	Normal	0xFF	Immediate	MAX
1	0	0	1	PWM, Phase Correct	0xFF	TOP	BOTTOM
2	0	1	0	CTC	OCRA	Immediate	MAX
3	0	1	1	Fast PWM	0xFF	BOTTOM	MAX
4	1	0	0	Reserved	–	–	–
5	1	0	1	PWM, Phase Correct	OCRA	TOP	BOTTOM
6	1	1	0	Reserved	–	–	–
7	1	1	1	Fast PWM	OCRA	BOTTOM	TOP

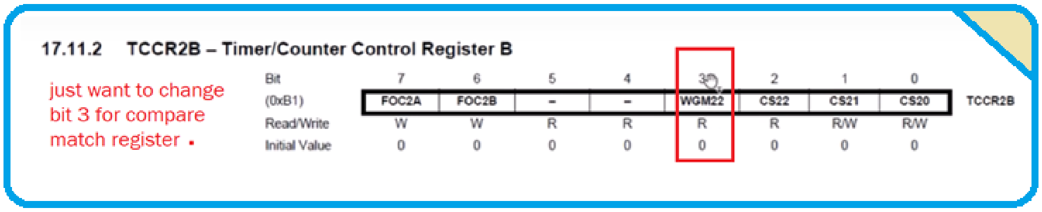
counts up to whatever OCRA is >

Notes: 1. MAX= 0xFF
2. BOTTOM= 0x00

Figure 9 Compare match data sheet guidance

[2, p. 130]

Compare Match Value: Register example



[2]

Compare Match Value: code example

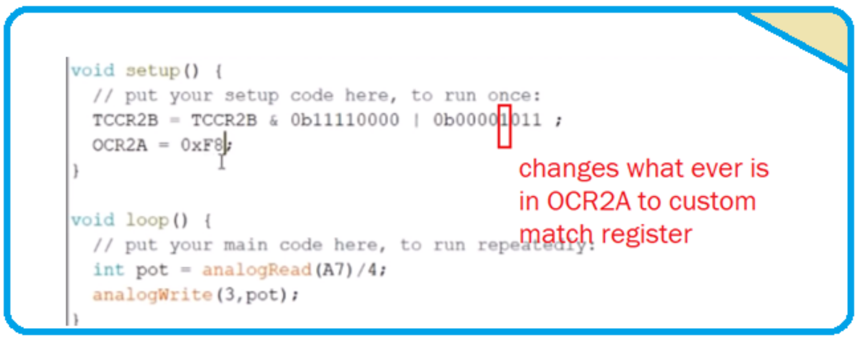
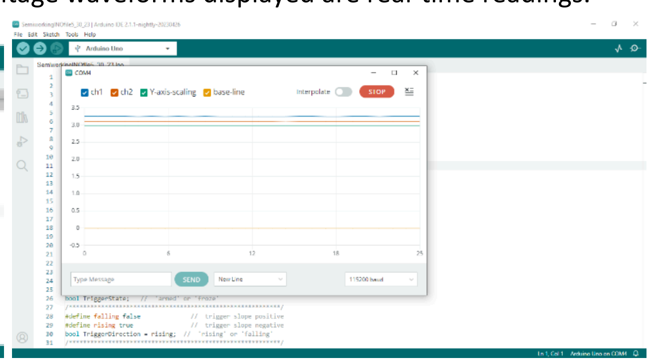


Figure 10 Compare Match Register Code Example

[3]

Depicting Signals with the Serial Plotter

Using the built in Arduino Integrated Development Environment (IDE), an INO file can be downloaded onto the laptop that is intended to display the output voltage waveform. Within the IDE, there is a serial plotter button on the top right corner. When clicked, the laptop screen will begin to display the voltage waveform readings from the BNC probes. The voltage waveforms displayed are real-time readings.

Triggering Tree

The user chooses with a switch whether the system is in sweeping mode or triggered. In sweeping mode it just depicts the signal continuously. Int triggered mode it descends down that branch and there are two sub branches, armed and froze. In the armed state it is searching for the triggering even. In the froze mode it has been triggered. Please refer to the flow chart at the beginning of this section for embellishment on the topic.

Y-axis scaling

The Y-axis scaling was interesting because we came up with a solution from just playing with the Arduino during class. We were playing with the Arduino so that group member could get familiar with the plotting features and the structure of an Arduino sketch. We breaded circuits with components like LED and potentiometers and uploaded some sketches that we found on the Arduino forum. Later when wrestling with different ideas for how to allow the user to scale the plot we recalled something we noticed while messing around. In particular it was that the plotter would be adjusted on the y-axis to accommodate the lowest and highest numerical values being plotted. We reasoned that we could add a potentiometer and then the ceiling of the display window would mostly follow the potentiometer. We also added a zero signal line to constitute the floor.

This user adjustable potentiometer is included to scale the y-axis of the voltage waveform display. When this potentiometer is adjusted, the y-axis maximum value scales from the initial 0-5V range to a customizable value less than 5V. This feature is especially useful if the user intends to analyze voltages that are smaller than 1V, as the graph will "zoom" into finer, smaller voltage values for a closer picture.

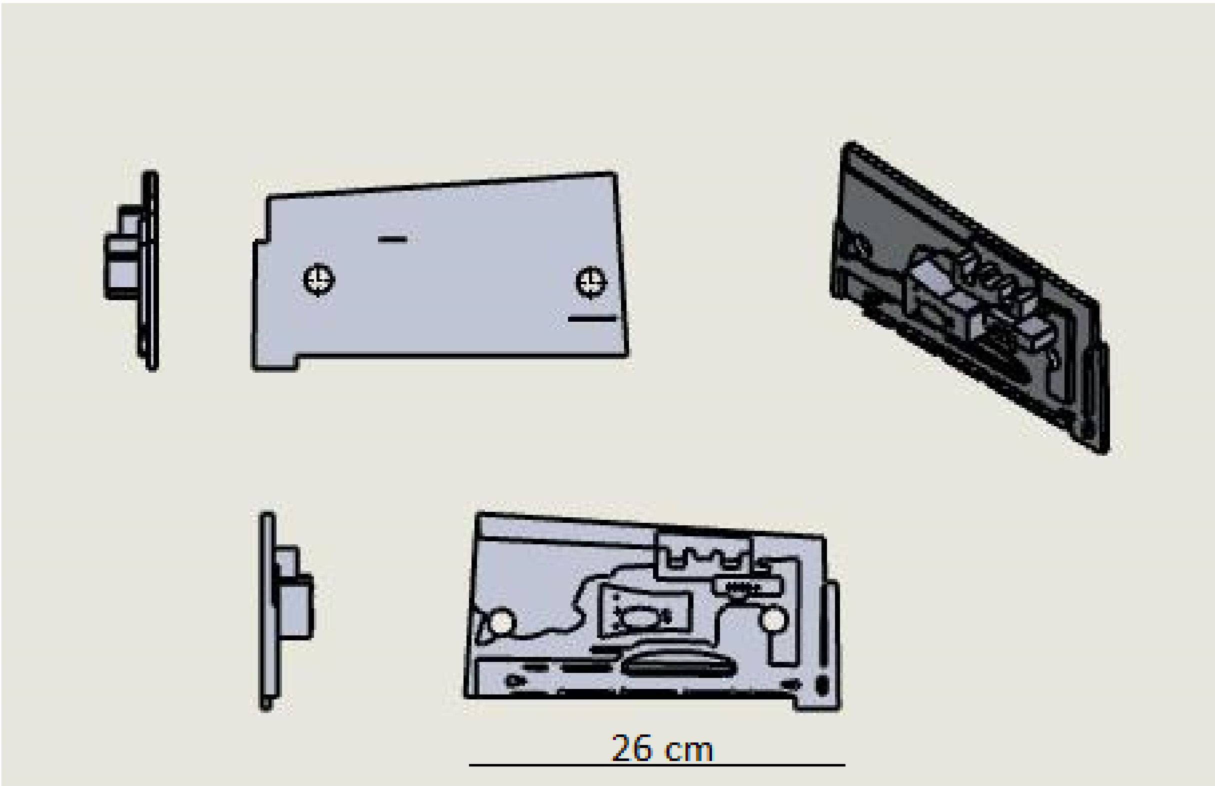
Enclosure

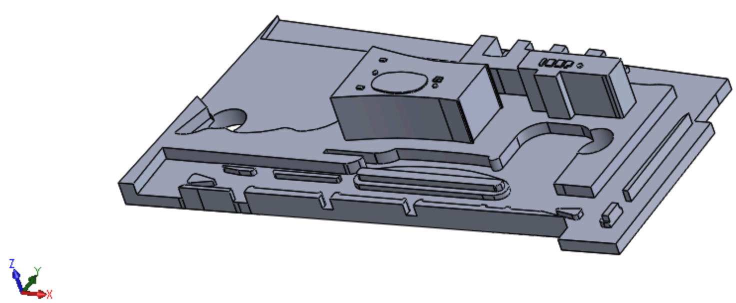
Description:

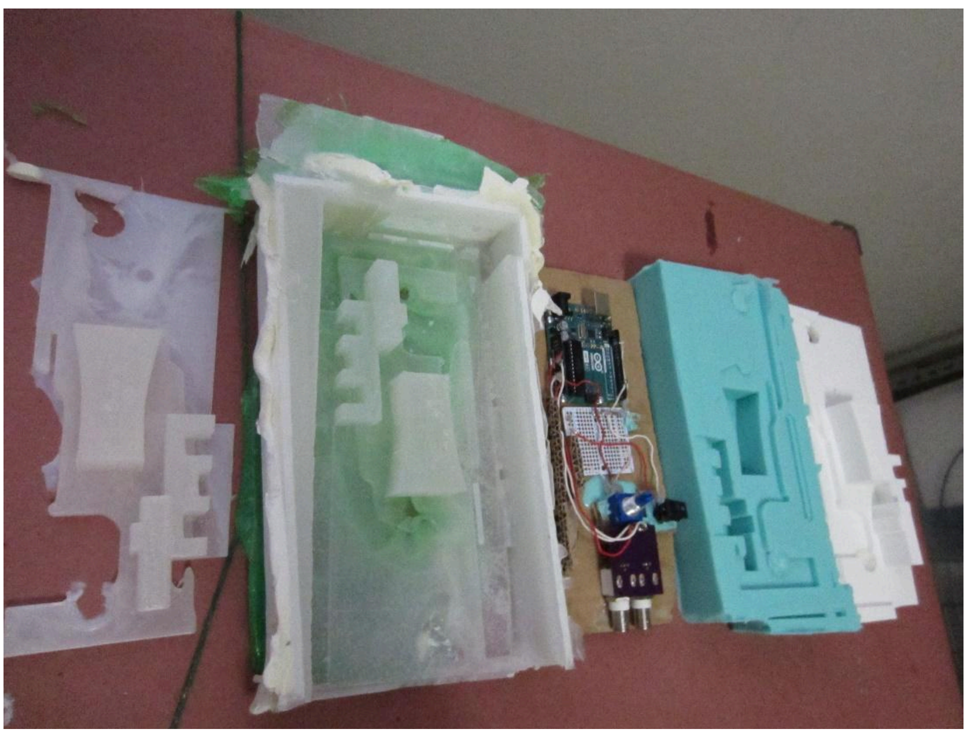
Our initial plan for the oscilloscope enclosure involved a cast mold that concaved to the specific shape of the Arduino, proto-board, switches, and PCB. The internal components need to be surrounded tightly in a plastic wrap before the liquid gel can be poured to create a protective mold enclosure. After multiple attempts of trial and error, the cost of the materials and the time required to create a fitting mold proved too expensive for our final enclosure design.

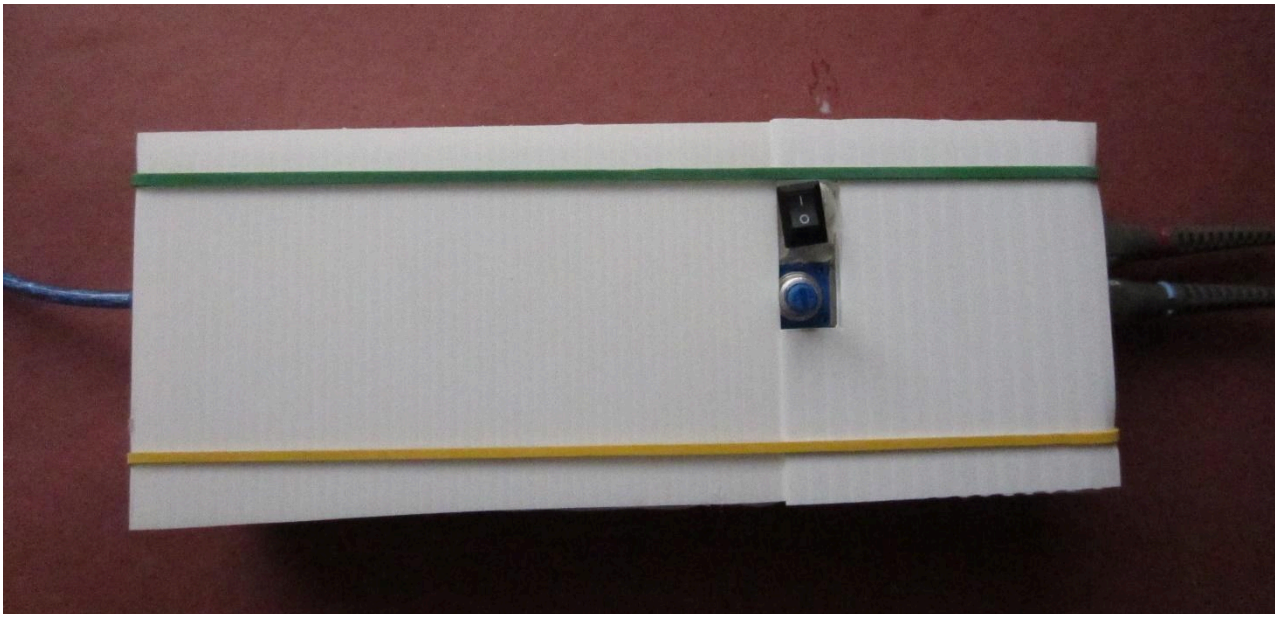
The final exterior is made up of a corrugated plastic that has been hot glued together into a rectangular enclosure to protect the internal components (Arduino microcontroller, proto-board, and PCB). Cutouts in the Styrofoam were made to make the voltage potentiometer, trigger switch, and removable BNC probes available to the user.

Mechanical drawings









Bill Of Materials

Label	Part Type	Properties
CONN1	USB cable	: USB-B
LED1	LEDs	package led10mm; variant 10mm
P1	Multimeter straight black probe	variant straight black probe
P2	Multimeter straight red probe	variant straight red probe
Part1	Arduino Uno (Rev3)	type Arduino UNO (Rev3)
Pre-Assembled Female BNC1	Pre-Assembled Female BNC	variant V1
Pre-Assembled Female BNC2	Pre-Assembled Female BNC	variant V1
R1	220Ω Resistor	pin spacing 400 mil; package THT; power 0.25; bands 5; tolerance ±5%; resistance 220Ω
R2	10k Ω Resistor	package 0402 [SMD]; power 0.25W; tolerance ±5%; resistance 10k
R3	Rotary Potentiometer (Large)	package THT; knob status 50%; type Rotary Shaft Potentiometer; power 0.5W; maximum resistance 100kΩ; track Linear; size Rotary - 16mm
S1	SWITCH-MOMENTARY-2	package tactile-pth-12mm; variant 12mm

Receipts

Michaels
Made by you™
MICHAELS STORE #8743 (541)738-0109
MICHAELS STORE #8743
1550 NW 9TH ST - STE 106
CORVALLIS, OR 97330
8-9181-3585-8759-4308-5108-5114-1873-3112
4091947 SALE 6007 8743 001 5/28/23 11:47
GORILLA MINI HOT 52427302304 2 @ 8.99 17.98 N
CS PLASTALINA 1# 886946243585 2 @ 3.99 7.98 N
TOTAL 25.96
ACCOUNT NUMBER ************4133
Debit 25.96
APPROVAL: 636470 CHIP ONLINE
PIN Verified
Application Label: US DEBIT
AID: A0000000980840
TVR: 8080048000
TSI: 6800
DEVICE ID:0001
TRACE REF:000069510
This receipt expires at 60 days on 07/26/23
Click. Buy. Create. Shop michaels.com today!
Get Savings & Inspiration! Text* SIGNUP to 273283
To Sign Up for Email & Text Messages.
*Msg & Data Rates May Apply
You will receive 1 autodialed message
with a link to join Michaels alerts.
THANK YOU FOR SHOPPING AT MICHAELS
Effective 11/27/2022 Clearance sales are considered
final, sold AS IS and are not
returnable for a refund or exchange.
Dear Valued Customer:
Michaels return and coupon policies are available
at michaels.com and in store at registers.

PCB Order from Osh Park

OSH Park Sales <sales@oshpark.com>
to me

[This email originated from outside of OSU. Use caution with links and attachments.]

OSH Park

3 x SimplePCB(5_9_23).kicad_pcb shipped

Hi wiesea@oregonstate.edu!

Your projects have been shipped to you via USPS Free Shipping.

You can track your order with the code 9400136106089849417192 or track it on the web .

Order Details

Ordered on
May 13th 2023
Paid on
May 13th 2023
Payment Method
Discover ending in [redacted]
Shipped on
May 19th 2023 via USPS Free Shipping

USPS - Free Shipping $0.00	
Order items	$11.15
TOTAL (paid)	$11.15
Shipping Address	Asa Wiese [redacted] United States of America

Female BNC Connectors

Your Digi-Key order is on its way!

Your Package Has Shipped!

Order Summary

Invoice Number:	97504396
Subtotal:	$ 6.18
Shipping:	$ 4.99
Total:	$ 11.17

Shipping Address

ASA WIESE
[redacted]
USA

Shipment Summary

Sales Order Number: 81417528

Shipping Information

USPS First-Class Mail:
92001901064603555809992 Track Package

Note: Your tracking information may not be available from the carrier until the next business day.

REVIEW INVOICE

Need to see your status of other orders? Signup for myDIGIKEY now!

Bibliography

[1] amandaghassaei, "Arduino Timer Interrupts," [Online].

[2] Atmel, *ATmega328P data sheet.*

[3] J. Ilett, "Arduino PWM Frequency Tweaked to 1kHz," [Online].

[4] A. Wiese, Thor of the Universe: The Lightning G-O-D, 2025.

[5] rgco, "Multichannel Arduino Oscilloscope," [Online]. [Accessed April 2023].

[6] G. Recktenwald, *Basic Pulse Width Modulation.*

[7] C. Nadovich, Synthetic Instruments : Concepts and Applications, 2005.: Elsevier/Newnes .

[8] R. Meier, "Roger Meier's Freeware," [Online]. Available: http://freeware.the-meiers.org/.

[9] Meeker6751, "Arduino Oscilloscope (6-Channel)," Arduino Project Hub, [Online]. Available: https://projecthub.arduino.cc/Meeker6751.

[10] Tektronixs, XYZs of Oscilliscope: Primer.

[11] "https://www.electronicwings.com/arduino/pwm-in-arduino," [Online].