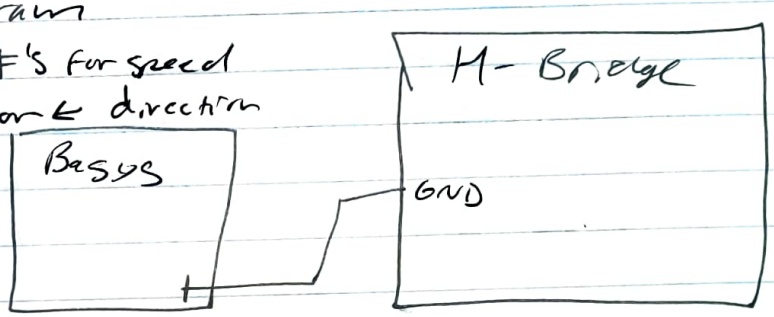
Brett Fischbach
R11793371

Lab 1 Notebook

1-25 - Safety training finished — 1 hr 4-5pm
- Lab table assigned
B - Motor, Basys, Battery, charger received from stockroom
- Names & certs posted on table

1-26 - Met @ 2PM - 7PM
- ✓ Designated A & B Motor sides
LBM - ✓ Need 1A Fuse or Breaker
- ✓ Need H-bridge datasheet
- Set up Block Diagram
- Basys Switches: 3 for #'s for speed
- 1 for → or ← direction
- 1 for enable

★ PWM sends square wave larger on % means larger speed

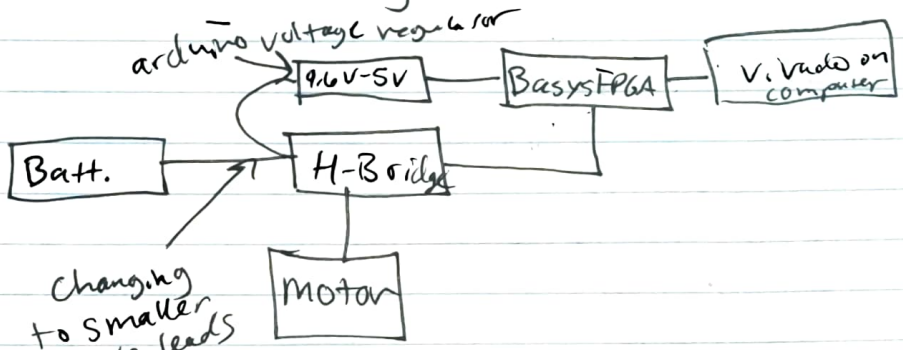


- ✓ Clean workspace
- ✓ missing oscilloscope
- ✓ 10 comparators
- ✓ 10 of commuter resistors
- Battery adaptor 9.6 → 5V → 3.3V
- ✓ Organized Work Area (tubi!)
- ✓ Cleaned work area
- Group Dinner
- Scheduled next meeting for Saturday afternoon

1-27 Pro-Point opportunity Sandia National Laboratories
THK - Pulsed Power, Nuclear Stockpile, National Security, NM, CA
Research into fusion, high energy density, sandia.gov/careers, Reheds posting 685503, SEER

5:30PM → 9:00PM

1-28 - Start PPT - Re-write Budget - Share
- Gantt Chart
- Basys Board Pictures, H-Bridge
- Design sequence and Layout for Block Diagram
- Verilog - turn on switch LEDs
-



- Look at guide for Verilog
- Design platform for boards

5:30PM - 8PM

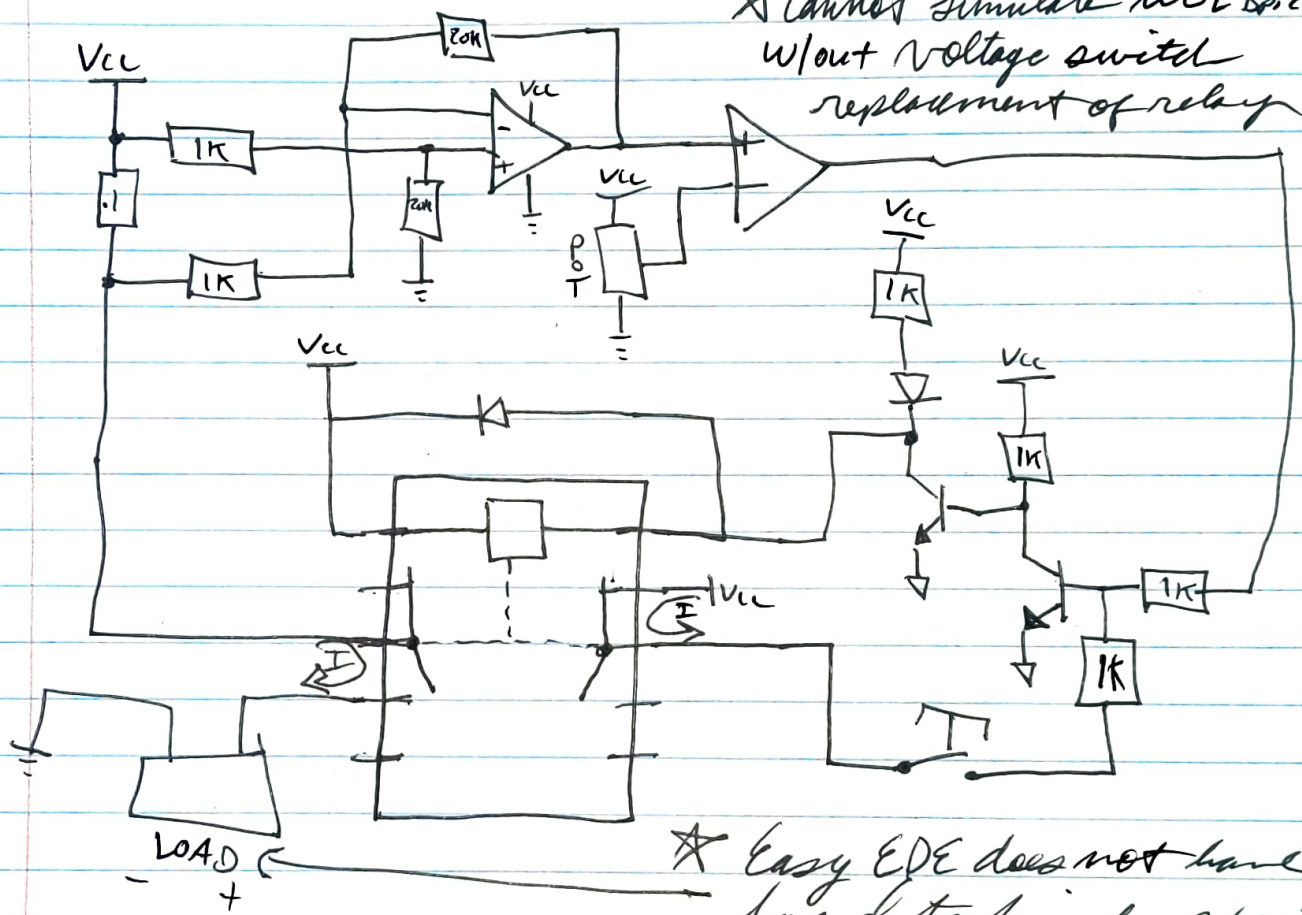
1-30-24 - Check on SGE Stylus & USB to Hdmi Adapter
- Fix text size on footers & headers
- add name assignments
- Figure out Sense A pin, Voltage? Current?
- Use multimeter to determine connections on H-Bridge
- create LTSpice circuit for hardware overflow protection w/ Sens A as Vcc input
- 4pm - 6pm

1-31-24 - Created circuit in Easy EDA, but it cannot simulate, No Load symbol and no potentiometer
- LTSPICE has no switch/relay for a simulation
- Searched for integrated circuits on digikey found INA 3221AIRGVR which is in the proper range of voltages/current for OCP

- Mini Project Goals (1) use BASYS FPGA to make motor spin based on switches SW0-SW7 using L-298H bridge
(2) use shunt resistors and the basys board to detect direction of current flow through the motor, and the amount of current. If current exceeds 1 A stop the motors. Display current mag. and direction on BASYS Display
(3) Implement Hardware based OCP

(2) Proposed OCP from GREAT SCOTT!

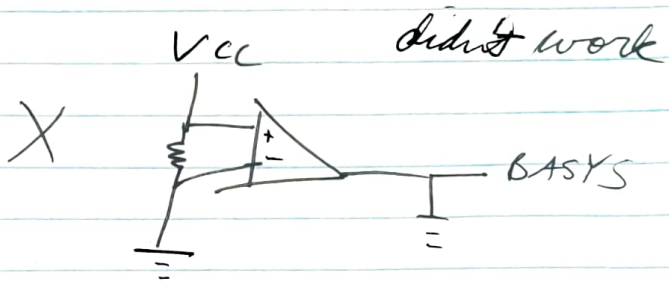
☆ Cannot simulate in LTspice w/out voltage switch replacement of relay

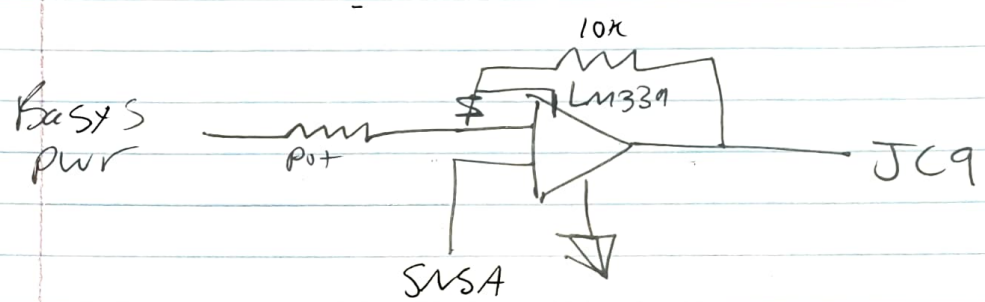


☆ Easy EDE does not have load to finish schematic

6PM — 10:30PM

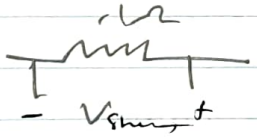
2-1 - Potential IC Chip INA3221AIRGVR
- sends out a warning signal to Basys board when programmed limits are hit
- reports shunt & Bus Voltage, is 4mm by 5mm
- LT spice sim of





- Looked into plug and play ammeters, a bust
  - mess w/ circuit and are expensive

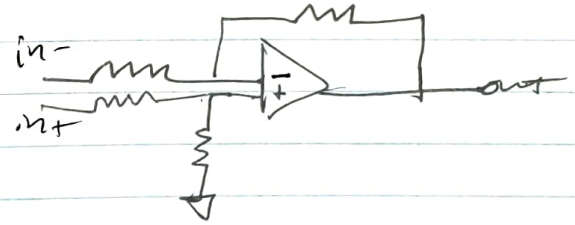
How a shunt resistor works



$V = IR$

$I = V(1\Omega) = 1A$

use w/ diff opamp use zero-drift amplifier



- Plug in motor to see if it works & to figure out SVSA pin
  - Battery needed charging
  - Motor worked
- Sense A had voltage of shunt resistors
- PWM - 100MHz - remember * in always loop (called as register)
- Found Verilog PWM online, modifying to fit w/ 2 ports, + edge rising trigger

[3:0], input wires

Still Planning on GS OC, waiting on components

- PWM Modules for 100kHz, 100Hz see which works
- 7:30pm - 11pm

2-2 - Building GS OC as parts came in today * Fuse backup OCP

-

PWM Pseudocode

Module declaration

wire dec

register dec

100Hz 100

1MHz 1,000,000

10,000

assign PWM to a register

assign p_duty & n_duty

p_d + n_d = 100 000

for each wire (sw-3) variation

p -> (mode = 0'b0000) ? 6000

↓

```
always @ (posedge clk)
loop    begin
            if p_count<P_duty
                p_count=p_count+1;
                PWM_REG = 1;
            else if n_count<n_duty
                //
                PWM_REG=0;
            else
                p_count=0;
                n_count=0;
loopend     end

endmodule
```

- generate ppt slide topics

2-5 3:00PM - Basys, H-bridge, PWR Regulator connected to motor.

- Motor moves forward and backwards, but no variable speed, only max
- we would need to reduce p & n duty by a factor of 10 until variable speed is achieved
- need to replace battery
- Fuse or circuit? Probably fuse
- Which clock are we using? (100 MHz or 1 MHz)
- AI generated Symbol/Logo
- not enough bits in register to match with 12 vs. 21
- Assigned scheduled deliverables
- Did Bulk of PPT
- Logan soldered Relay Ckt
- Mark worked on SOCP
- Ran through presentation

a: 00PM

2/11 - 5:30PM Mark's Code Review & 3-D Printing Design

Modules - Motor control, Main, PWM_Source, Current control, XADC Module, Display, Bintodec

Main

- Top Module
- Basys input switches
- Leds
- 7-segment display ports
- JA PMOD
- System Clock
- JXADC PMOD

Pseudo code (from Marks Completed Code)

Declaration of inputs and outputs

// Display I/O

input switches 0-15

output wires(leds) 0-15 shown are on/off only

output seg 0:6 for 7 segment display

☆ output dp display

☆ output an 3-0 anode

input sys clock 100MHz from BASYS 3

// ADC Ports

assortment of pin ~~out~~ inputs declared for the adc to read the voltage on the pins

// Motor Control

☆ Output wires (switches) 0-7 JA (~~Pmod~~ BASYS -> H-Bridge) wires

Wires have a value of low, high, X (unknown in sims) or Z (high impedance)

// PMOD pins
parameter ENABLE, IN1, IN2 : 1, 9, 10 pins
motor
wire M_interrupt;
wire [19:0] d_data

parameter ADC_channel 16-Bit hexadecimal 16 X ADC Channel 6

// Current_control over_curr
.Enable (sw[15])
.direction (sw[5])
assortment of pin assignments

// Display current control
.data (from Pmod ans)

// Motor Control
.enable(sw[4])
.direction(SW[5])
.interrupt(M_interrupt) from PMOD pins
.mode[sw[0:3]] speed modulation in 4 bit Bin
.clk(SYSCLK)
.Enable JA ENA_PMOD
.In1 ↓ IN1_PMOD
.In2 IN2_PMOD

end

Summary - Declare/instantiate which inputs/outputs go to which module (allows sharing of variable

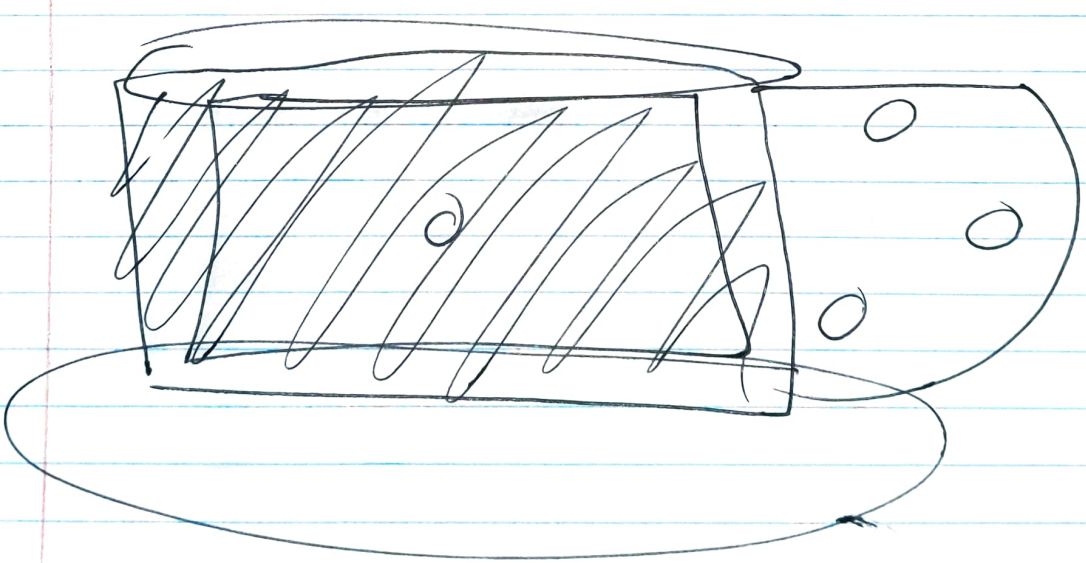
PWM_SRC.V - converts system clock cycles to motor cycles 60-25kHz
- receives Clk & SW[3:0] as inputs, outputs PWM to motor

Declares 21 bit wire p_duty positive portion of cycle
" " " register count stores count variable
reg PWM_REG;
assign PWM = PWM_REG;
(P_duty based on SW3 2 1 0)

initialize count to zero

loop to generate n_duty based on p_duty, setting PWM on while count is greater than p_duty, Ø otherwise

Motor control - inputs from top (enable, Clk, reset, interrupt)



- Current display issues, displaying instantaneous current at different points on the signal, rather than a consistent average

- higher PWM cycles (>90%) reads correct current

- any lower is either inconsistent or Ø

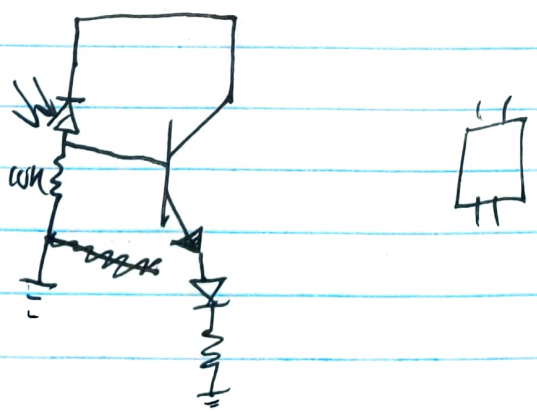
- another solution: display only the highest value for .1s period
  ☆ worked, loaded to

2-12 Presentation Assembly 8:30PM - 12:30PM
- Gifted vids
- Wrote slides
- Logan attempted to fix H-bridge, Failed
- Practiced presentation
- Cleaned up prints

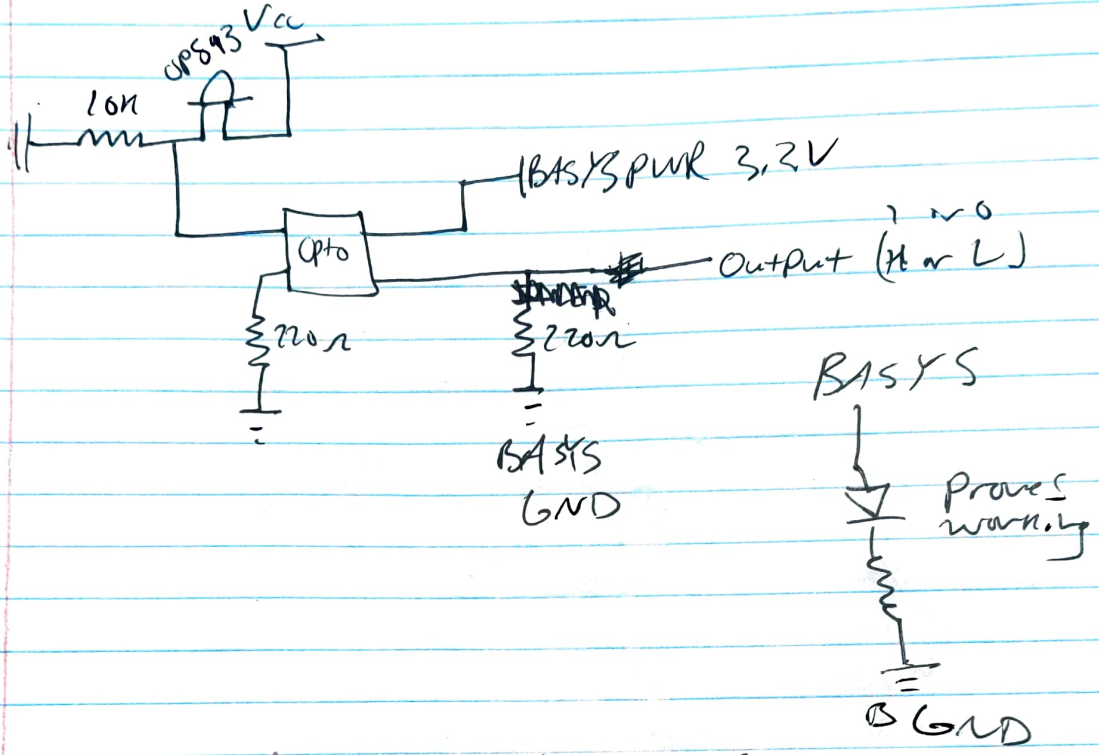
2-25 - Worked on State Machine Diagram for track navigation for 2hrs

2-26 - Looked up IR sensors & finished PPT

2-2 - Using OP593 after Clark's harsh criticism of Thermopile sensors
- OP593 is a phototransistor



Used Optocoupler to convert analog to digital



3/3 7:~~PM~~ - Realized we needed an IR Emitter to test Morse recognition
- Looking into Farnsworth technique
-

- 20wpm, $u = \frac{1.2}{C}$, u is one unit of morse in seconds

~~Hertz~~ C is the speed of transmission in wpm

$u = \frac{1.2}{20} = .065$, $Hz = 1/.06 = 16.7Hz$

t 5 units = .3s

1 = •---, 2 = ••---, 3 = •••--

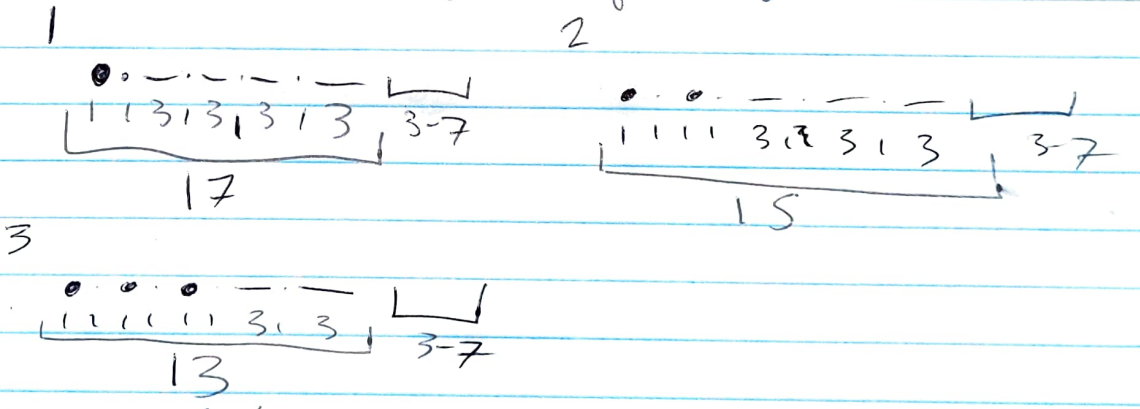
3331 Journal continued

- Found code for Morse Buzzer, need to convert to IR LED
- LED output on Arduino completed, now need to configure Morse → ~~binary~~ decimal conversion in Vivado
- .06s → 16.7 Hz | 100 MHz
  Frequency of IR inputs (words, 6233 Hz units) | f of BASYS Clock

$$\frac{100M}{16.7} \rightarrow 5,988,023.1 \text{ clock cycles per word}$$

- Will IR Emitter in project have delays after each ~~#~~?
- Assume Yes
  - Program must find whole rest, then read the value between rest

- Put silos on sheet for 1 hour (sent VL code)
- How is a word divided? each of our characters are shorter

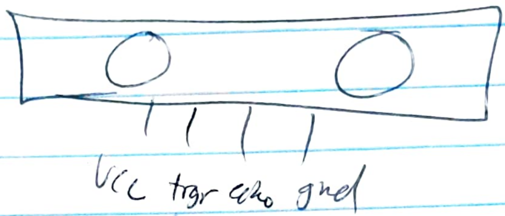


- Got test Arduino working w/ IR LED & photodiode
- Worked on track nav w/ Mark
- Logan worked on breakout board

3-5-23 - ~~the~~ Ultrasonic Sensor

Distance = time · Speed of Sound / 2

344 m/s



5V - DC

f = 40Hz

sends pulse

outputs time in us

100MHz toggle

~~40~~

send Enable pin through trigger input (10us TX)

every 10us send trgr

$\frac{uS}{148}$ = # of inches

inputs
echo pin clk
module_enable

outputs
~~reg[size] distance~~
wire detected, trigger

```
reg timer_enable;

reg[] distance; //variable measured in MS

localparam targetdistance; //constant measured in inches
```

assign LED DETECTION

```
assign detected = distance < (targetdistance * ratio)
```

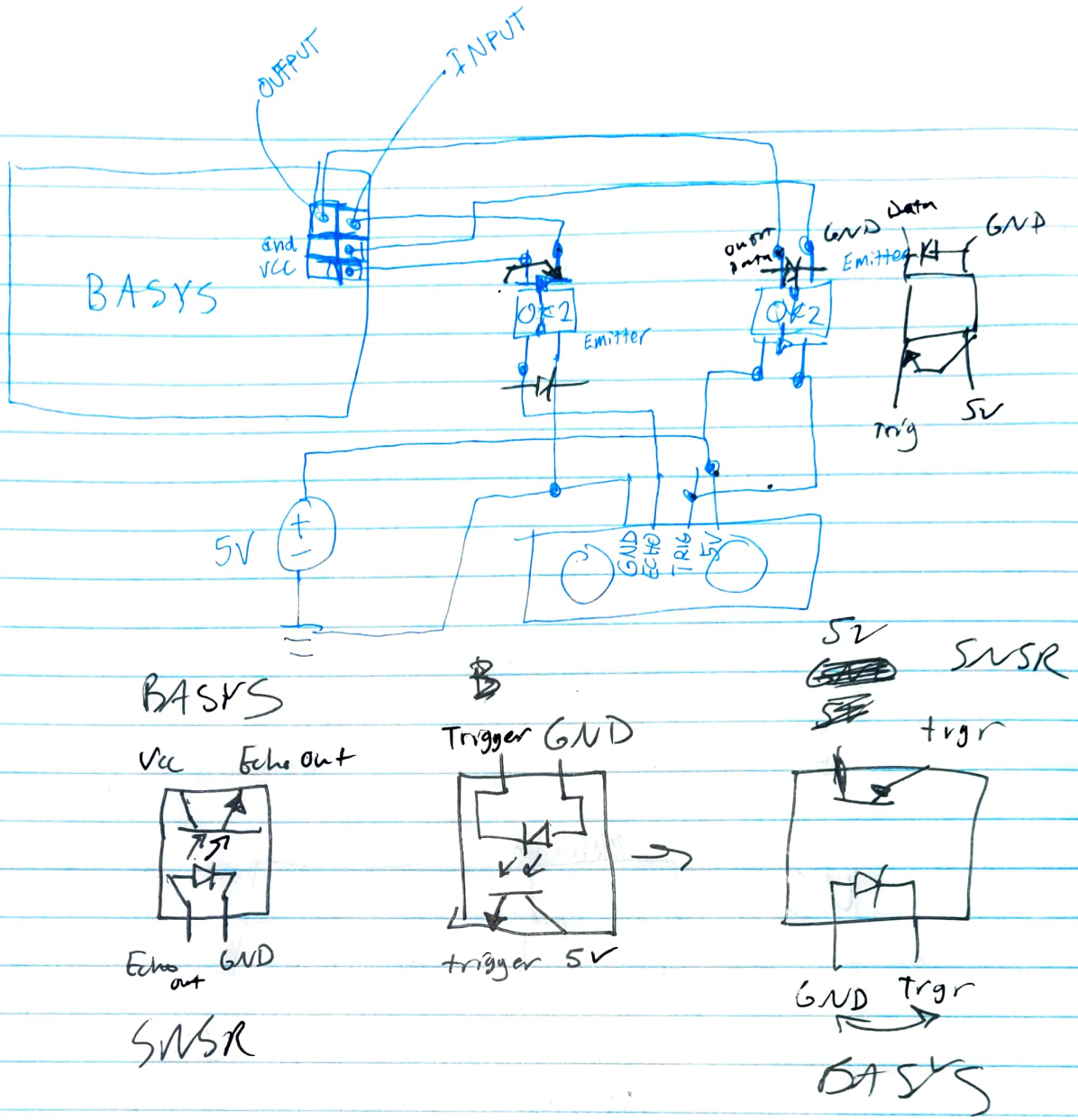
(timer, 148)

```
reg[] timer (units of MS) (max = 23,300)
reg[] counter = 0;
```

(at beginning, timer = max-value so that we dont get false positive)

```
always @(posedge echo) begin

    ~~reg~~ timer = 0
    timer_enable = 1

end

always @(negedge echo) begin

    timer_enable = 0
    counter = 0
end
always @(posedge clk) begin

    counter++;
    if(counter > 100) begin
        timer ++;
        counter = 0;
    end
end
```

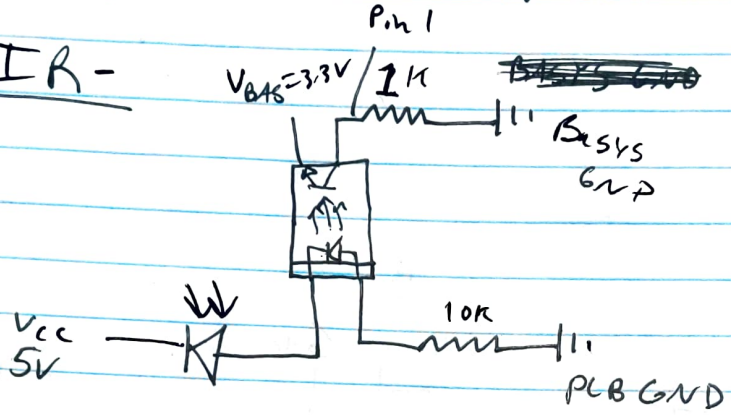
$$\frac{100{,}000{,}000 \text{ cycles}}{\text{sec}} \cdot \frac{1 \text{ sec}}{1{,}000{,}000} = 100 \frac{\text{cycles}}{\text{MS}}$$


OUTPUT
INPUT
BASYS
GND
VCC
Emitter
GND
Data
GND
Emitter
5V
trig
5V
GND
ECHO
TRIG
5V
BASYS
Vcc
Echo out
Echo out
GND
SNSR
Trigger
GND
trigger
5V
5V
SNSR
trgr
GND
Trgr
BASYS

3-18-23 - Return from Spring Break

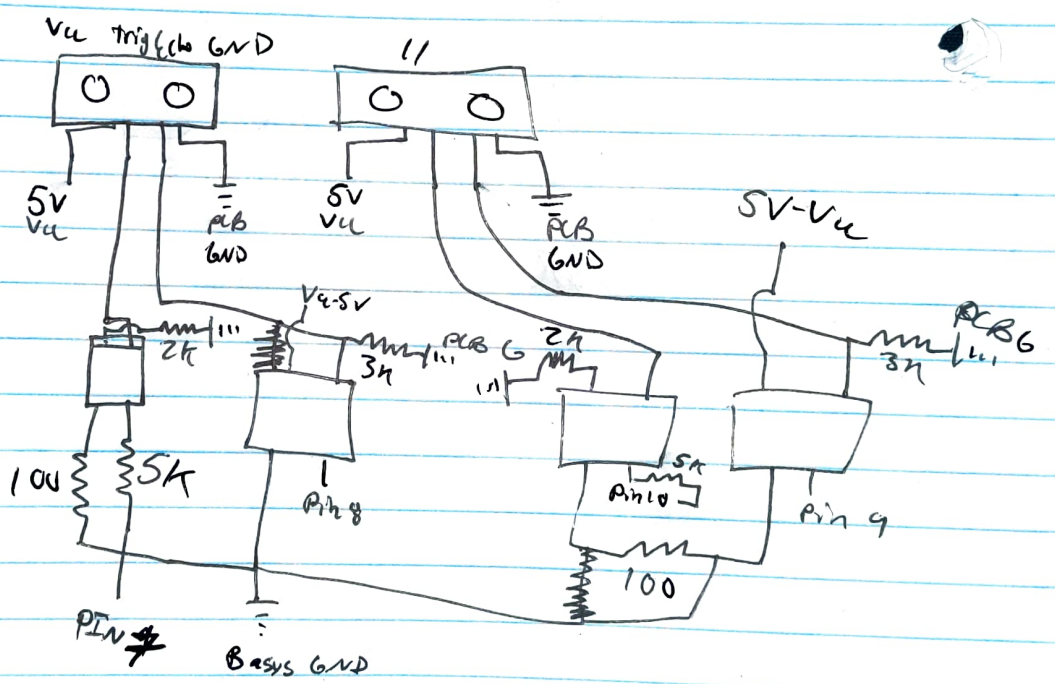
- Need to catch up on US & IR code changes
- Update PPt to white background, black text, blue accents add Nautilus Lego to top left corner, leave black sidebar
- Draw circuit diagram of US & IR arrays on paper and in LT Spice

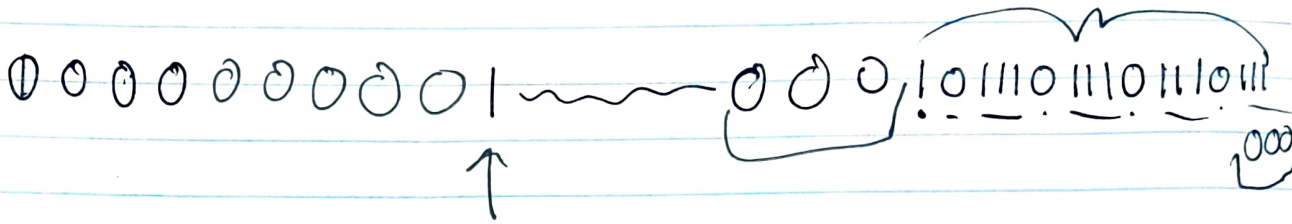
IR -



- Only needed to measure the width
- displays hex value

US -

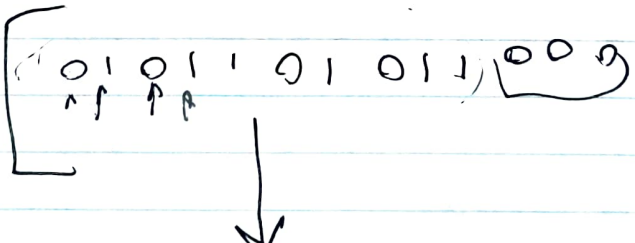


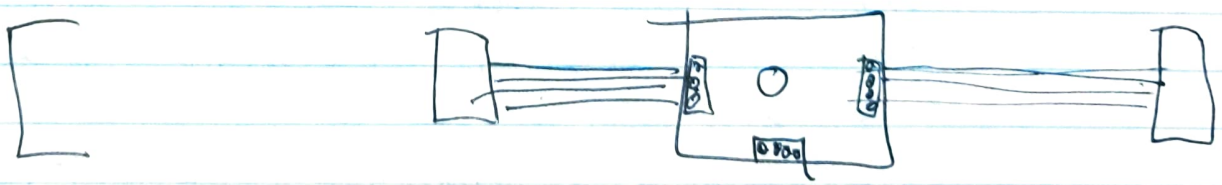


reg[14:0] value_ONE =

```
initial begin
  clk=0;
  IR_in = 1;
  #60
  IR_in= 0;  → 000
  #180
  for (integer i=0; i<16; i++) begin
     IR_in = Value_ONE[0];
     VALUE_ONE >> 1;
  END
```

```
always begin
   #1;
   clk= ~clk
end
```



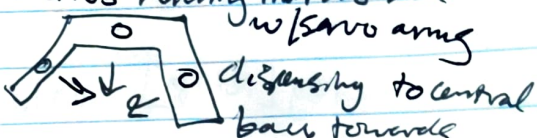


Morse on TB works, Rover breakout board made,
IPS PCB Tested & Implemented
Re-designed parts
Bin-detection working, IR sensor on PCB Not working ~~5V~~ → 3.3V

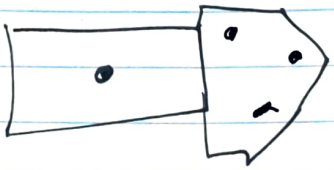
Marble Dispensing Redesigned

3.25

- I Built a new emmitter.
- BASYS Receives input, just need to calibrate timers
- Marble Dispenser design by Logan: 3 silos holding marbles back w/servo arms dispensing to central bay towards a ~~ramp~~ into bin



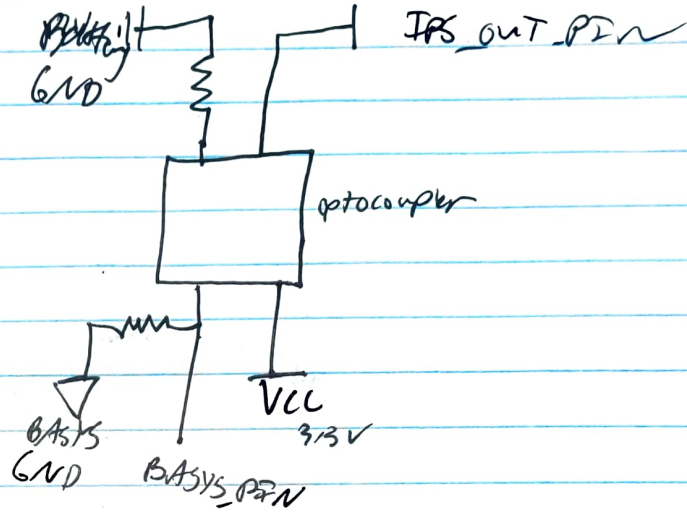
- Switching to 4 Rover IPS sensors



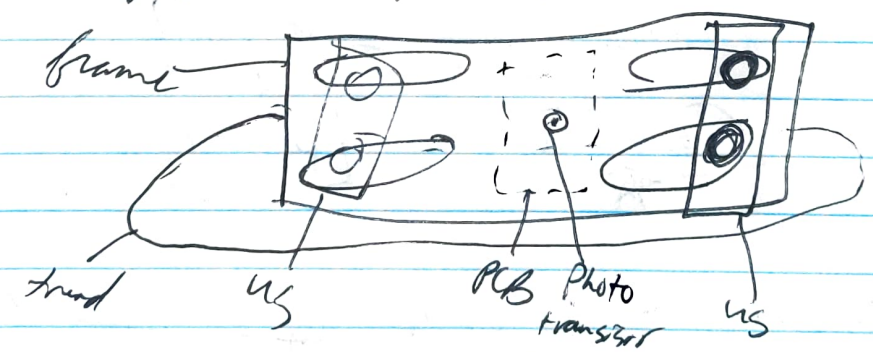
Tuesday

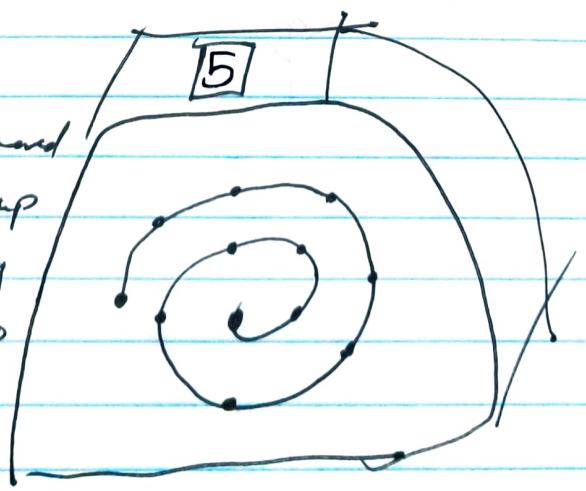
- Ensures track navigation will go more smoothly.
- When center rear sensor ~~flags~~ goes off track, the rover will turn in the most recent on dir of side front sensors
- Mark made unity sim for rover movement
-
- IR Module written - seeks out character/wordspace ~~between~~ before and after morse code dits/dashes.
  - once two 7spaces are found the 7spaces are removed and the data between is counted based on number of highs and length of said highs
  - dots and dits are counted separately as backup for debugging
  - IR Testbench returns Don't care as outputs for all #'s, debugging required
- Ultrasonic sensor now measures length of echo received
  - this value is variable, unlike our previous method of signal to echo rise.
  - LED now shows ~~ON~~ off if object is in target distance, calibration as simple as changing a constant in code

- PCB will house IR & US sensor circuits
  - Secondary IPS PCB also
- Moved to 12V battery



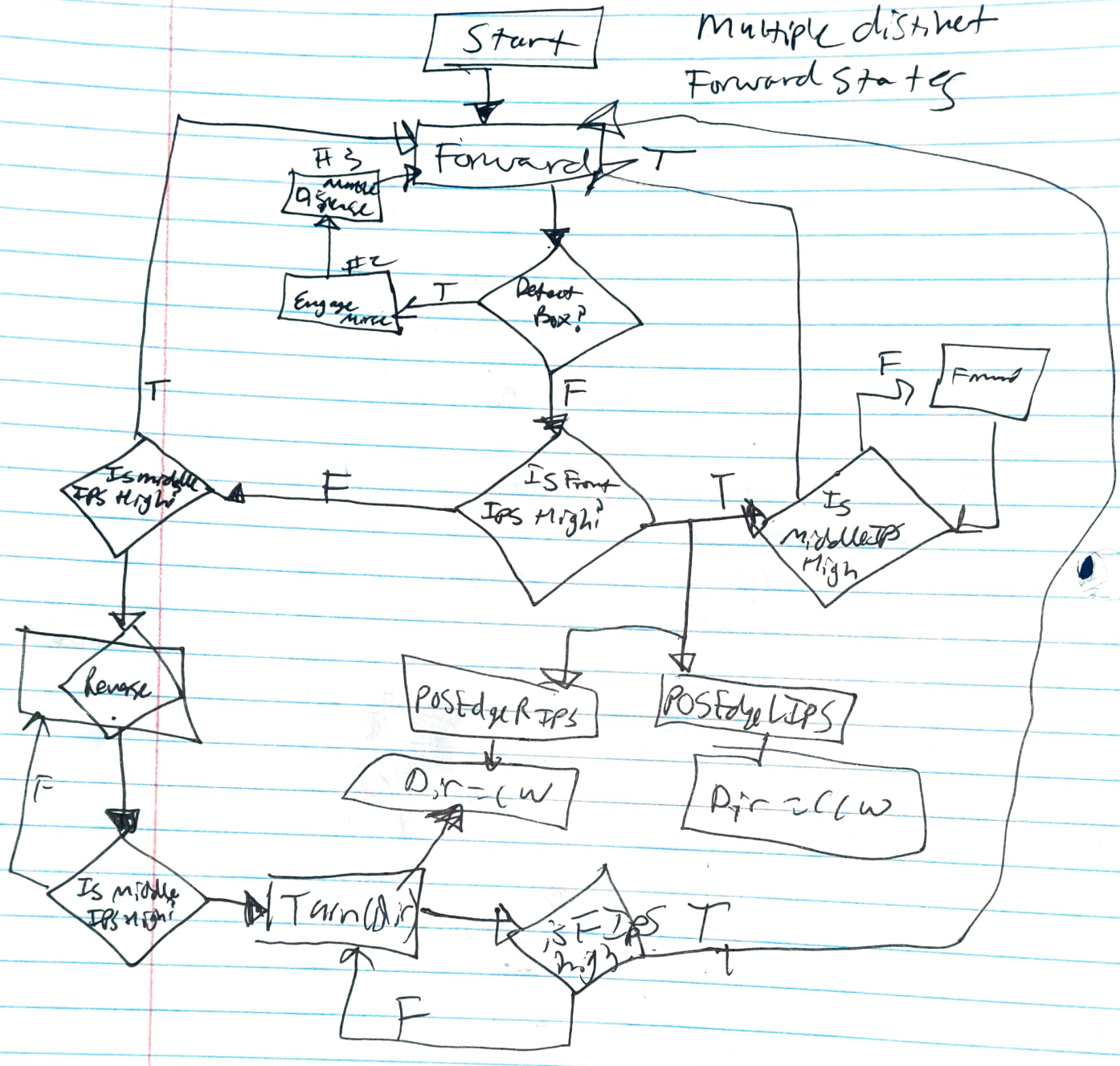
- Mark now controller of GitHub Repository
  - made public for class
  - includes all designs and code for project
- sensor rig designed by Mark
  - hangs off trim side of rover, connects between threads



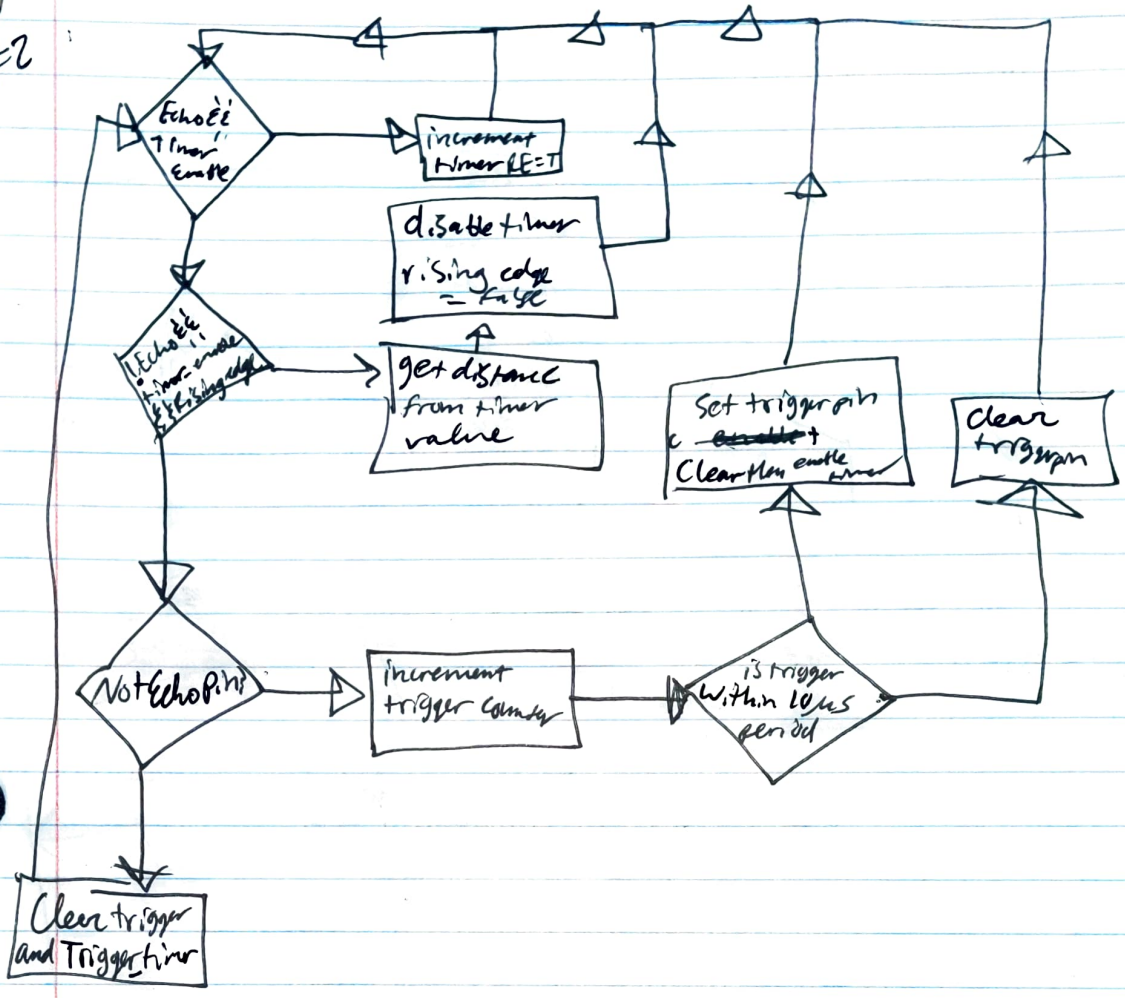


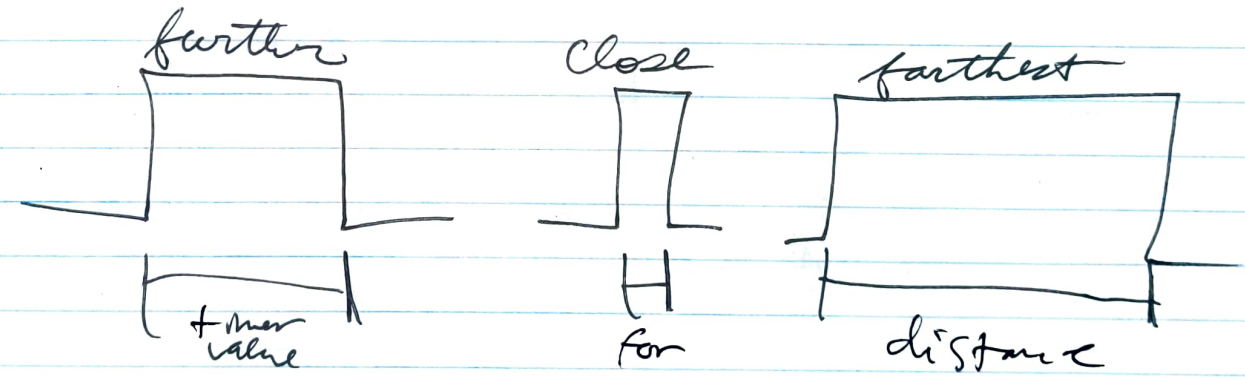
- 7-seg - 5 forward
- 12 LEDs Light up
  Nautilus config
  through Arduino

#1 Track navigation / IPS state diagram

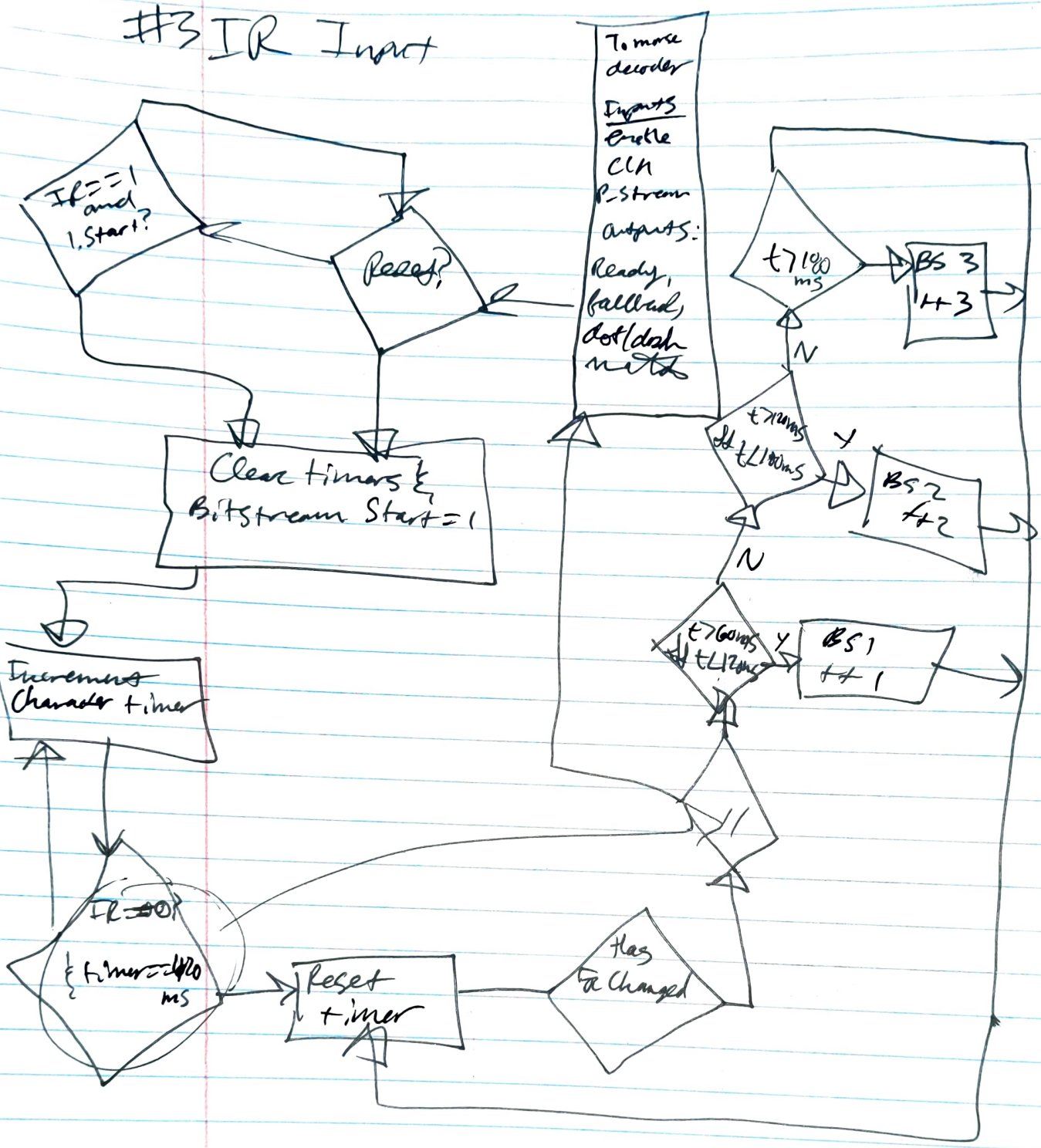


#2

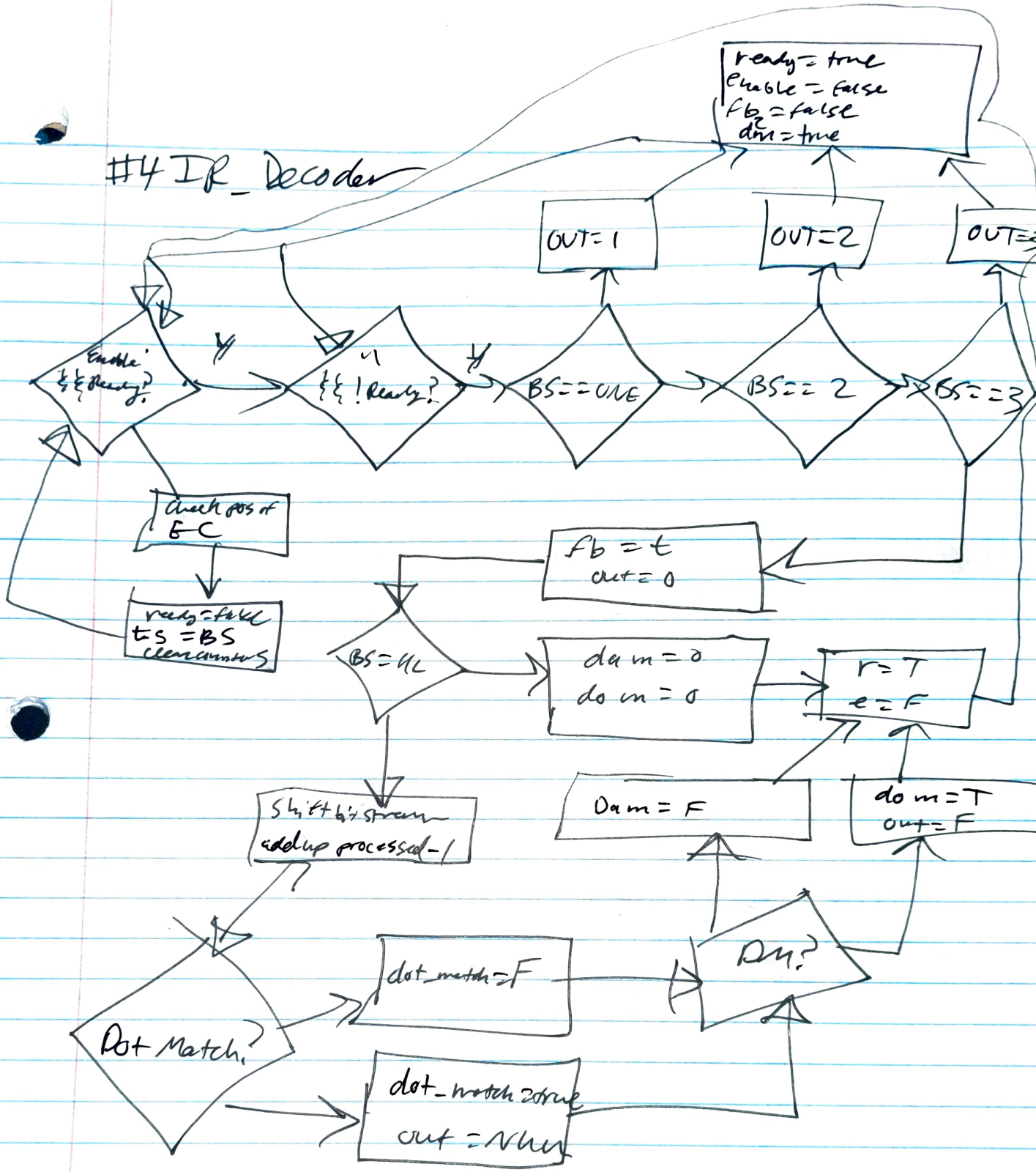

Echo && Timer enable
increment timer RE=T
disable timer
rising edge = false
Echo && timer enable && Rising edge
get distance from timer value
Set trigger pin
Clear then enable timer
Clear trigger pin
Not Echo Pin
increment trigger counter
is trigger within 10us period
Clear trigger and Trigger_timer


further
timer value
close
H
farthest
distance

#3 IR Input

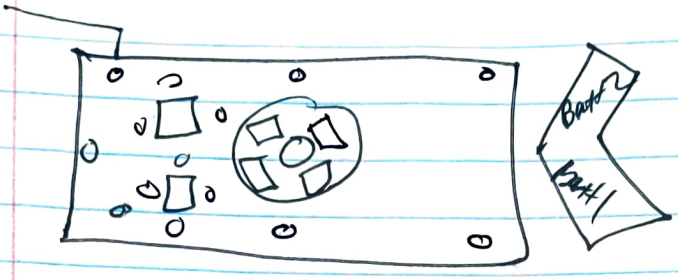

To morse decoder
Inputs
Enable
CLK
P_Stream
Outputs:
Ready,
fallback,
dot/dash
IR==1 and 1.start?
Reset?
Clear timers & Bitstream Start=1
Increment Character timer
IR==0?
& timer==1920 ms
Reset timer
Has IR Changed
t>1920 ms
BS 3
t+3
N
t>960ms && t<1920ms
Y
BS 2
t+2
N
t>760ms && t<1120ms
Y
BS 1
t+1

#4 IR_Decoder

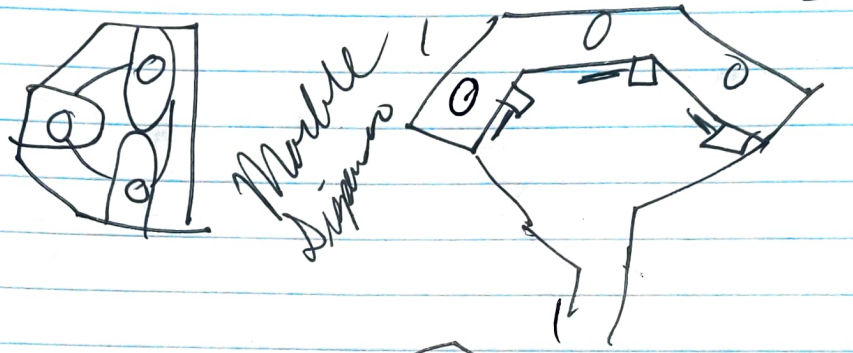

ready = true
Enable = false
fb = false
dm = true
OUT = 1
OUT = 2
OUT =
Enable && Ready?
!Ready?
BS == ONE
BS == 2
BS == 3
Check pos of E-C
ready = false
ts = BS
clear counters
fb = t
out = 0
BS = 0
dam = 0
dom = 0
r = T
e = F
Dam = F
dom = T
out = F
Shift bit stream
add up processed -1
Dot Match?
dot_match = F
Dm?
dot_match = true
out = value

Current CAD Designs

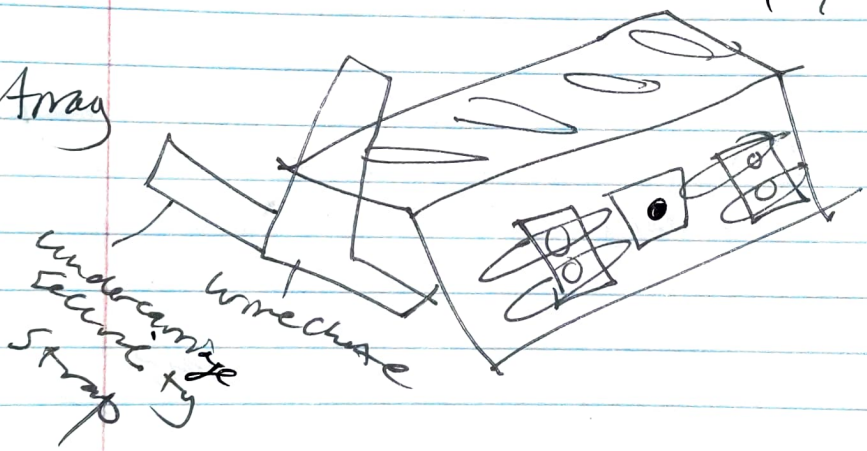
Base Plate



IPS Carrier Assembly



Sensor Array PDU



LOGAN converts to Eagle files

Circuits For PPCB