PID Pendulum
Final Presentation

By: Joe S, Matt H, Austin C, and Kat S

Final Design

Austin

Agenda

- Project Description
  - ConOps Summary
- System Requirements Verification
- Detailed Design
  - Mechanical, Electrical, Software
  - AI&T, Functional Block Diagram
  - Operational Flowchart
- Fabrication and Integration
- Comprehensive Performance Testing
- Schedule
- Project Budget
- Conclusion and Lessons Learned

Kay

Concept of Operations Summary

Stakeholders: Educators and Academics, Industrial Training

System Description: PID demonstrator: illustrating physical and digital effects of changing algorithm constants to the user.

Operational Environment: Indoor, high-traffic environment.

Support Environment:
- COTS (common off the shelf) parts
- Constant Uptime
- Constant Interaction

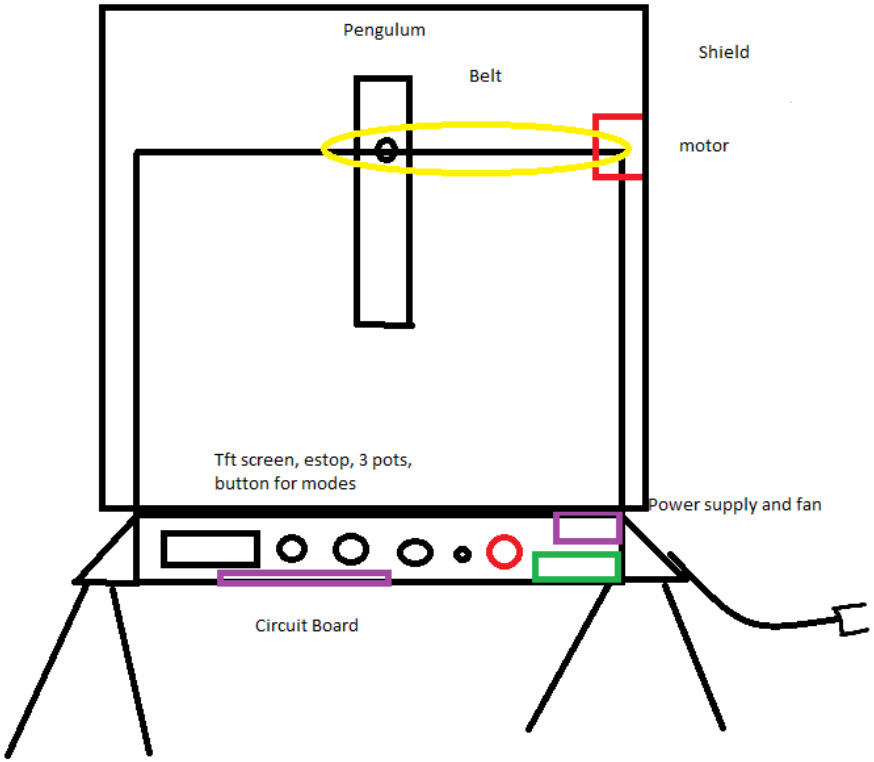
Operating Modes:
- Off Mode
- Idle Mode
- Spin-Up Mode
- Autonomous Mode
- User Interactive Mode

Risks:
- Pinch Points
- Swinging Pendulum
- Failure
- Short Circuit
- Overheating

Impact: Hazardous when misused. Requires physical space, and creates waste.

Kat

The Initial Idea:



Kat

Component Level Requirements

Green = Passed

Blue = Untested

Red = Failed

	Component Testing				
Test Number	Component	Testing Date	Testing Breakdown	Verification Methods	Passing Criteria
1.1	Frame Stability	3/20/23	After welding the frame, apply known forces similar to what the system will experience to test the holding power of the welds	Testing, Inspection	No yielding or welding delamination occurs
1.2	Motors	3/1/23	Motors are rotating with the proper torque output when powered by code.	Inspection, Test	Motor rotates with the required torque.
1.3	Encoder	3/1/23	Ensure that rotation in the motor results in a changing of the received values. Also make sure the motor can be zeroed in code.	Inspection, Demonstration	Encoder values properly change and interface with the code.
1.4	Limit Switches	3/10/23	Once activated a signal is sent to the program	Inspection, Demonstration	The switch provides a high or low status corresponding to its state
1.5	Electrical/Board Components	3/10/23	That all small components work on the board.	Demonstration, Test, Inspection	All components are in working order
1.6	Motor Controller	3/10/23	Ensure that the motor controller is able to send proper steps to the motor to provide precise control without overheating itself.	Test, Demonstration	The motor controller is able to properly and quickly manipulate the motor's position
1.7	USB-C for programming	3/12/23	Ensure the connection with the board is working. Test if the code transfers to the board.	Test, Inspection	Make sure code downloads correctly
1.8	Board is properly grounded as per standards	3/10/23	Ensure grounding in the parts that require it to meet expected engineering standards.	Test, Inspection	All boards components are grounded
1.9	All soldered connections are secure	3/12/23	Ensure that all pieces soldered are secure and connected to the board.	Inspection	There are no loose pins or incorrect connections

Matt

Component Level Requirements (continued)

1.10	Power Supply	3/5/23	All components will be supplied with a stepped down voltage of at least 19V from the 120V standard power outlet.	Test, Inspection	Components are not supplied with more current than they can handle.
1.11	3 Potentiometers	3/11/23	When turned, the values read are changed.	Test, Demonstration	Values changed when turned.
1.12	Estop	3/20/23	The system will be turned on and running and once the Estop is pressed the system will receive no power	Test, Inspection, Demonstration	The button will press freely and will send a value to stop.
1.13	Power	3/20/23	That the system will turn on and off.	Test, Demonstration	Once the button is pressed the system will be turned off and on
1.16	TFT Screen	3/5/23	Values will be provided to the TFT screen and will then be displayed	Test, Inspection	The TFT screen correctly displays the values it is provided
1.17	Leg Fit	3/20/23	That the legs will fit into the frame and be cut to the same lengths. They will not cause stability problems with the frame.	Test, Inspection	A proper slip fit is created without any slop.

Matt

Subsystem Level Requirements

	Subsystem Testing				
Test Number	Component	Testing Date	Testing Breakdown	Verification Methods	Passing Criteria
2.1	Belt Tensioning Assembly	3/24/23	The system shall be able to tighten the belt automatically	Test, Demonstration, Inspection	The shall be no slack in the belt and constant tension provided
2.2	Sprocket and Belt System	3/24/23	Sprocket and Belt shall move together as a system when the sprocket is rotated	Test, Inspection	The sprocket moves the belt properly
2.3	Friction Coefficient and Slip Fit of Pendulum Arm	3/20/23	A proper slip fit is created between the two machined parts at the top of the carriage that allows the axle to freely rotate without too much slop being created	Test, Inspection	A proper slip fit is created between the axle and its holder.
2.4	The Fit of the Legs and System Stability With Them Vs. Without Them	3/20/23	Determine if the legs are needed or not to lower the center of gravity of the system to increase stability.	Test, Inspection	Does the system need the extra stability?
2.5	Electrical Internal Temps don't get too high	3/29/23	The system shall not rise to temperatures that will potentially harm other components when left running for too long	Test, Inspection	The system shall not reach temperature over 150 F

Matt

System Level Requirements

	System Level Testing				
Test Number	Component	Testing Date	Testing Breakdown	Verification Methods	Passing Criteria
3.1	Test the self balancing PID	4/15/23	The system shall use a PID loop to balance the pendulum.	Test, Demonstration	The pendulum will balance itself.
3.2	Ensure that the interface is legible to unknowing users	3/20/23	The system shall display clearly the graph of the PID loop, the P, I, and D constants. And, the system will tell the user what mode the system is currently in..	Test, Demonstration	The TFT screen will correctly display the graph and changing values.
3.3	Ensure that manipulating the interface buttons and knobs creates the proper output in the system during user control	4/16/23	The system shall allow user inputs to change the PID loop using potentiometers and cycle through the system's various modes using the embedded buttons.	Test, Inspection	Test if the user variables change the way the pendulum behaves.
3.4	Test and ensure harmonic oscillation is working	4/12/23	The system shall allow for the pendulum in a rest position to be swung up and begin the PID loop.	Test, Demonstration	The pendulum when started will swing from the rest position to the PID loop position.

Matt

System Requirements Verification

Requirement Type→	Performance							Functional			Design								Interface			Resource		Verification Reference
Requirement Title→	Self Tensioning	Low Power Mode	Autonomous Mode	User Control Mode	Mode Selection	Emergency Protection	TFT PID Constant Display	Require Power	Spin-Up in 20s	System Temp.	Span	Colored Interface	Frame Material	Pendulum Length	Factor of Safety	Fatigue Strength	Element Protection	Vibrations	Mechanical	Electrical	Software	Mass	Weight	This column identifies the specific verification summary related to the activity on each line. Multiple summaries may apply to a single line item.
Level of Assembly																								
System																								Section 3.X Summaries
Frame Stability	IMT					AI									AT	AT	ATI	DIT	DIT				I	Section 3.5
Pendulum						IT								I					DT			I	I	Section 3.6
User Interactability and Legibility		DIT		DIT	DIT		DIT			MT		IT								IT	DIT			Section 3.2
Harmonic Oscilation									DITV											IT	DTV			Section 3.4
Autonomous Operation			DITV							MT										IT	DTV			Section 3.1
Subsystem/Assembly																								Section 2.X Summaries
Frame Assembly	IMT										AIM		IT						DT					Section 2.6
Circuit Board		DTV	DT	DT	DIT	DIMT	I	IM		MT		IT								DIMT				Section 2.5
Sprocket and Belt Assembly	DT																							Section 2.2
Belt Tensioning Assembly	DIT																		DT					Section 2.1
User Interface		DT	DT	DT	DT		DITV					DIT							DT	IT				Section 2.7
Component																								Section 1.X Summaries
Motors		DIT	DIT	DIT				MT	DIT										DIT	IT	DTV			Section 1.2
Encoder		DIT	DIT	DIT			ITV	MT	DIT											IT	DTV			Section 1.3
Limit Switches			DIT			DIT		MT												IT	IT			Section 1.4
Motor Controller		DIT	DIT	DIT				MT												IT	IT			Section 1.6
USB-C Port			T	T			IT	MT												IT	DTV			Section 1.7
Board Grounded		IM						IM																Section 1.8
Power Supply		IM				DIT		DMT		MT														Section 1.9
3 Potentiometers				DIT			ITV	MT												IT	DTV			Section 1.11
Estop						DIT		MT												DITV				Section 1.12
Power Switch								IT												DIT				Section 1.13
Toggle Button		DIT	DIT	DIT	DIT			MT												IT	DT			Section 1.14
Capacitive E-stop						DIT		MT												DITV				Section 1.15
TFT Screen				DITV				MT				IT								IT	DT			Section 1.16
Fan Cooling						MT		IMT		DMT										IT	DTV			Section 1.18

Verification Method:
A - Analysis
D - Demonstration
I - Inspection
M - Measurement
T - Test
V - Validation of Records

Notes:
1. Color Scheme: Activities completed highlighted in GREEN; Activities to be completed highlighted in GOLD
2. Summary References: Resolution (3.3.1), Frame Rate (3.3.2), Mechanical I/F (3.3.3), Electrical I/F (3.3.4), Mass (3.3.5), Power (3.3.6).
3. Support electronics used with camera assembly.

Matt

Harmonic Oscillations

Date: 04/16/2023

Activity Title: Harmonic Oscillations

Activity Location: Joe's Shop

Configuration: Pendulum, Carriage, Motor, Rods, Belt, Encoder

Operators: Matt, Joe, Austin

Results: Pendulum can go from down position to apex position

Observations: The system struggles to catch the pendulum in the apex position

Activity Conclusion: Successfully swings from rest position to PID loop position

Matt

Self Balancing

Date: 04/16/2023

Activity Title: Self balancing

Activity Location: Joe's Shop

Configuration: Pendulum, Carriage, Motor, Rods, Belt, Encoder

Operators: Matt, Joe, Austin

Results: Pendulum can balance itself for approximately 20-50 seconds

Observations: The motor does not have enough torque to meet our acceleration requirements

Activity Conclusion: Successfully balances itself

Matt

TFT Screen

Date: 04/16/2023

Activity Title: TFT Screen

Activity Location: Joe's Shop

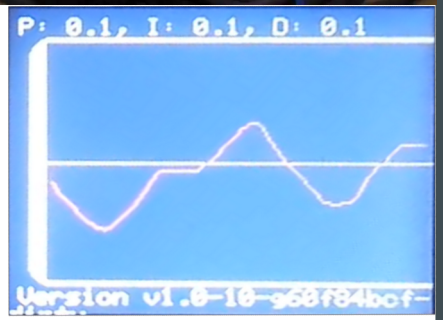
Configuration: TFT Screen, given inpu

Operators: Joe

Results: Displays harmonic oscillations

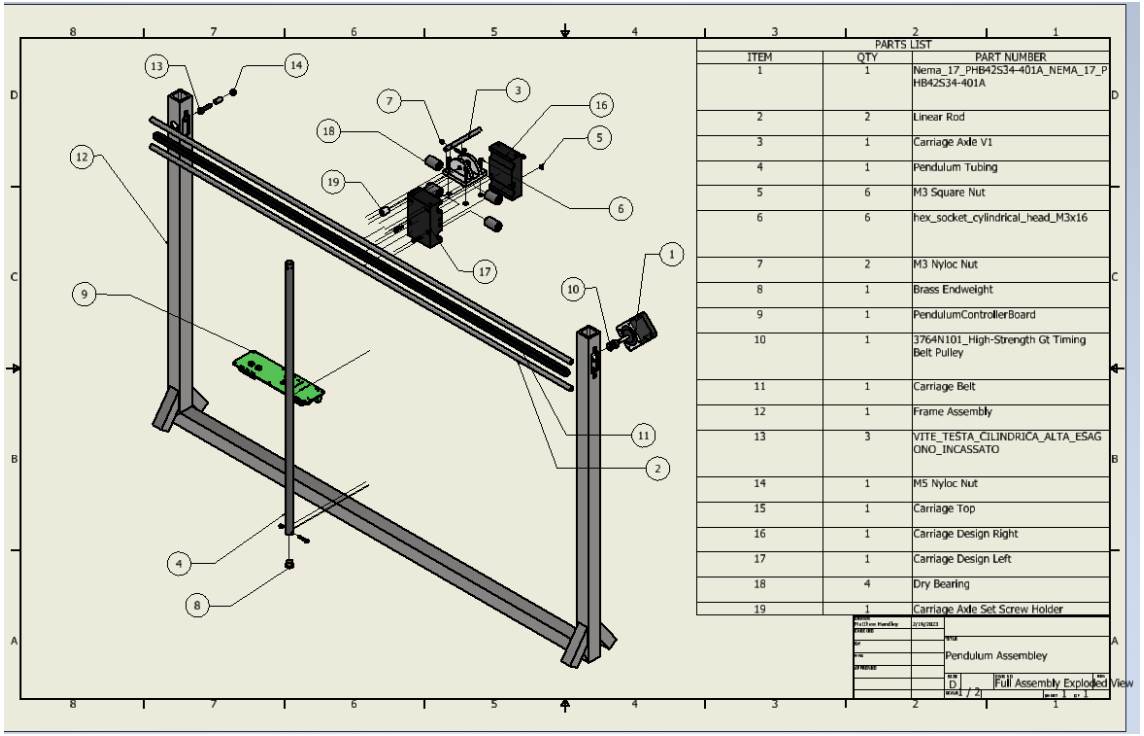
Observations: TFT screen displays values accordingly

Activity Conclusion: Successfully displays graph and changing values



Matt

Detailed Design



PARTS LIST

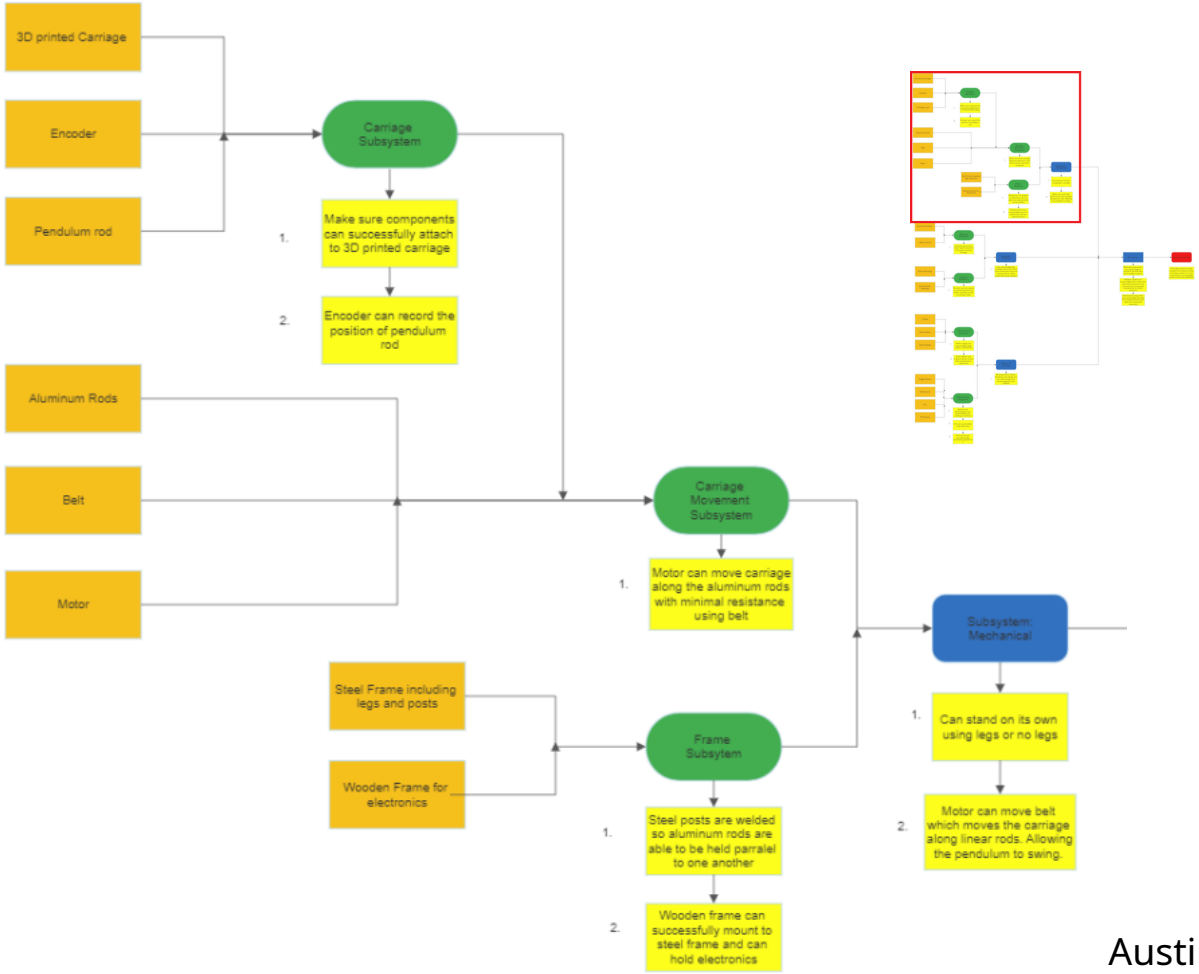
ITEM	QTY	PART NUMBER
1	1	Nema_17_PHB42S34-401A_NEMA_17_PHB42S34-401A
2	2	Linear Rod
3	1	Carriage Axle V1
4	1	Pendulum Tubing
5	6	M3 Square Nut
6	6	hex_socket_cylindrical_head_M3x16
7	2	M3 Nyloc Nut
8	1	Brass Endweight
9	1	PendulumControllerBoard
10	1	3764N101_High-Strength Gt Timing Belt Pulley
11	1	Carriage Belt
12	1	Frame Assembly
13	3	VITE_TESTA_CILINDRICA_ALTA_ESAGONO_INCASSATO
14	1	M5 Nyloc Nut
15	1	Carriage Top
16	1	Carriage Design Right
17	1	Carriage Design Left
18	4	Dry Bearing
19	1	Carriage Axle Set Screw Holder

Pendulum Assembley

Full Assembly Exploded View

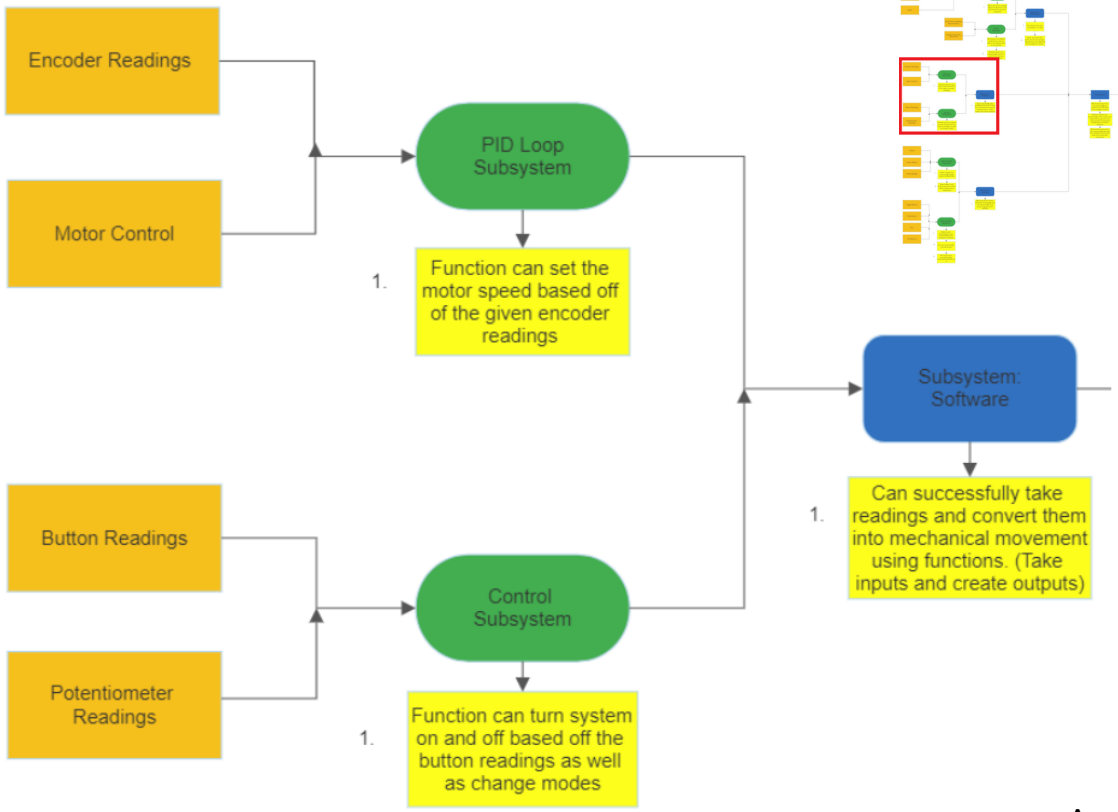
Matt

Integration & Test Flow



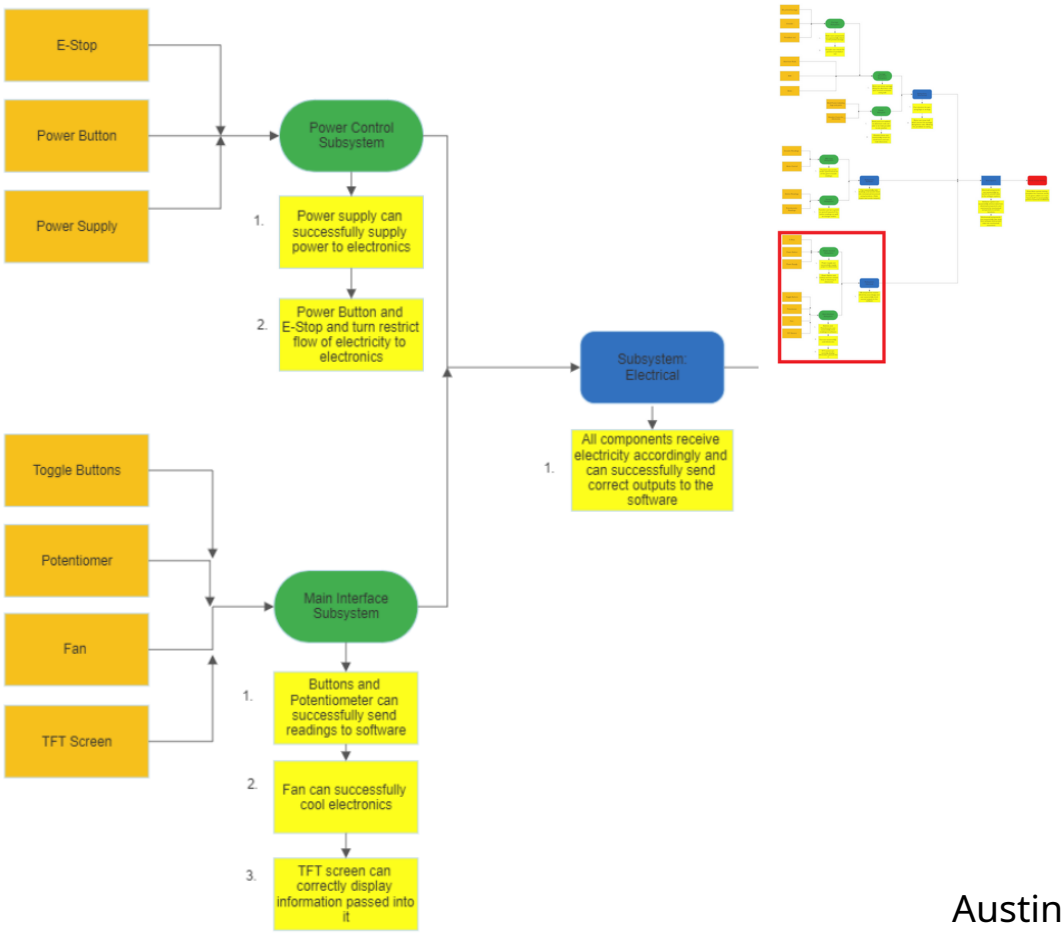
Austin

AI&T Flow (cont.)



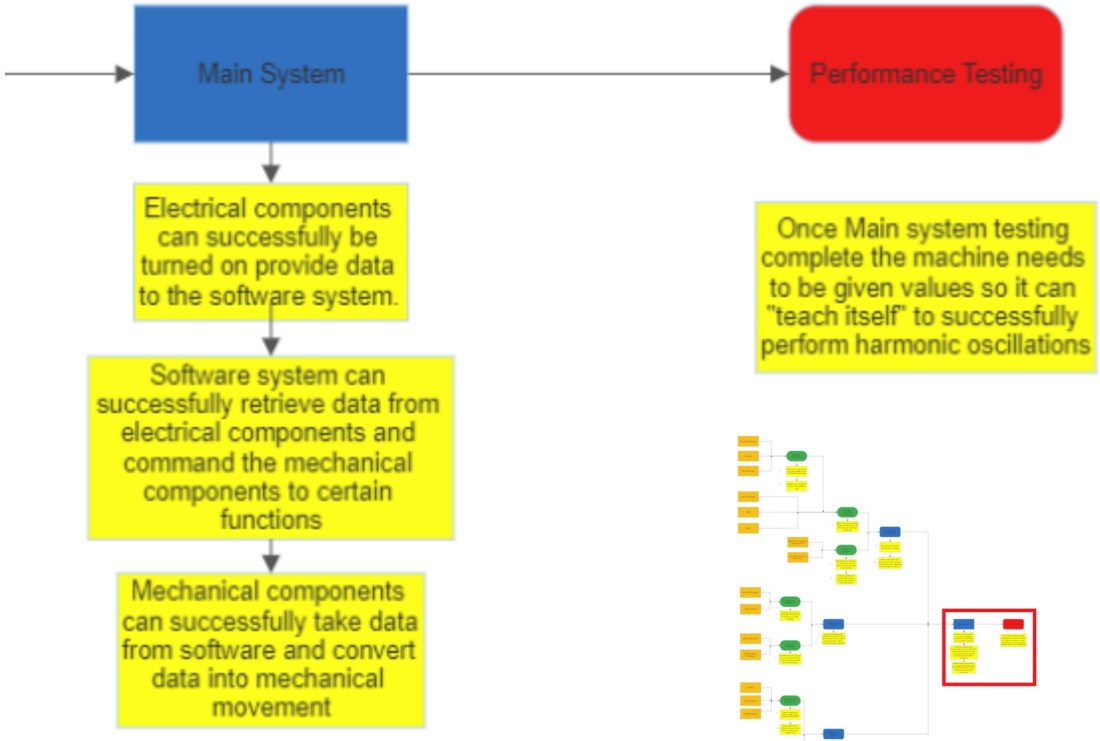
Austin

AI&T Flow (cont.)



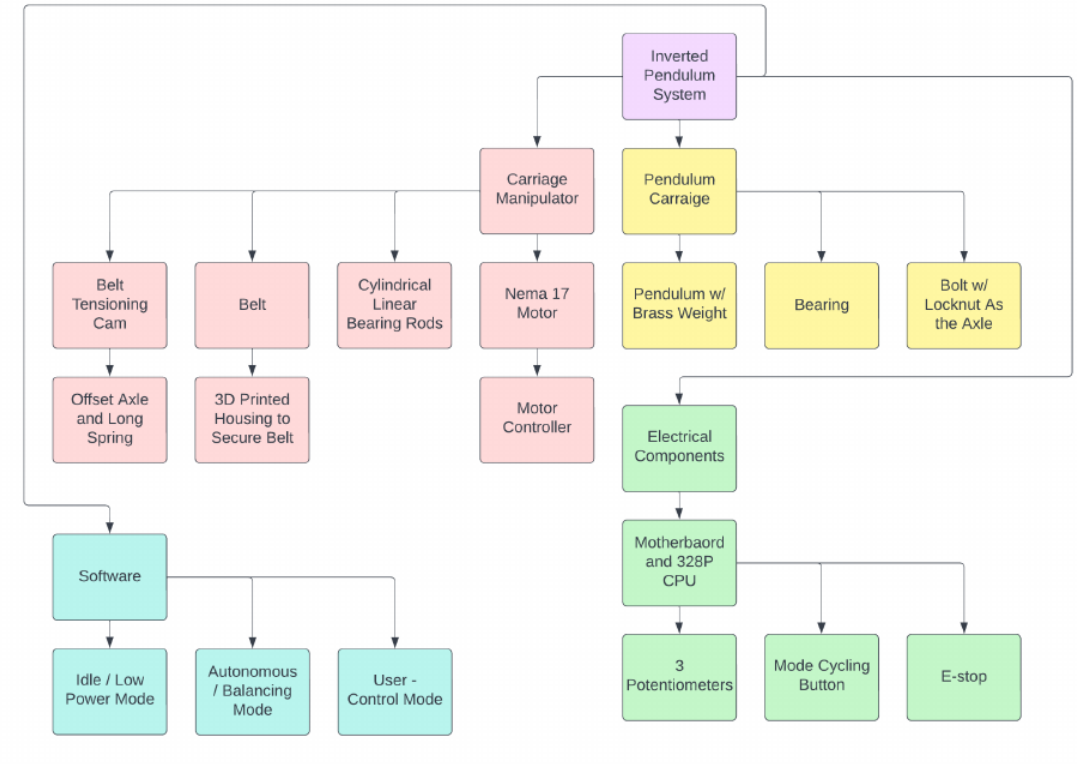
Austin

AI&T Flow (cont.)



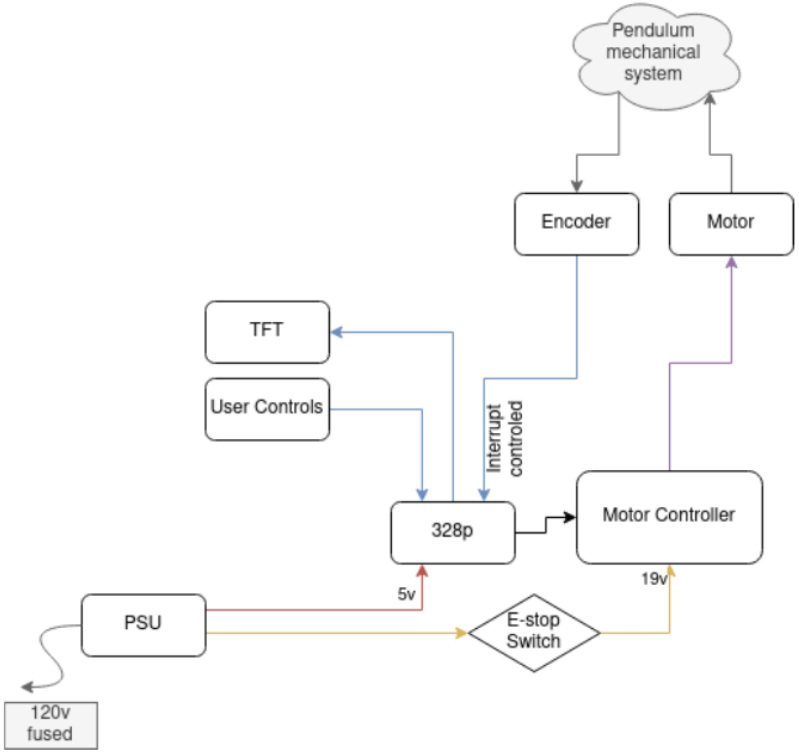
Austin

System Functional Block Diagram



Austin

Operational Flow Chart



Austin

Subsystem Definition

Mechanical

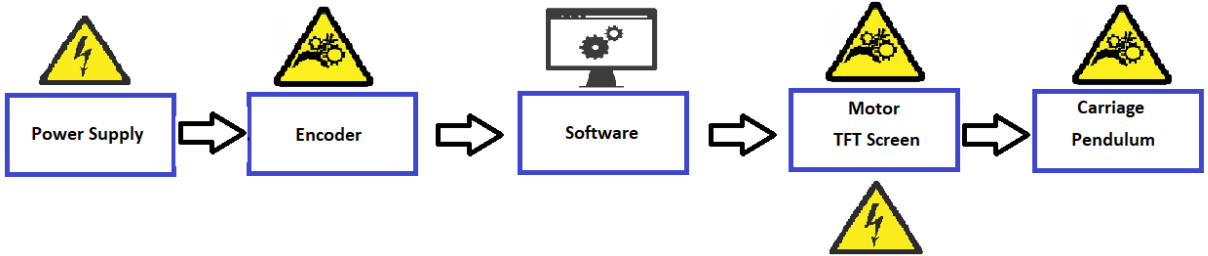
- Carriage
- Aluminium Rods
- Steel Frame
- Wooden Frame
- Motor (output)

Software

- Encoder reading
- Motor control
- Potentiometers readings

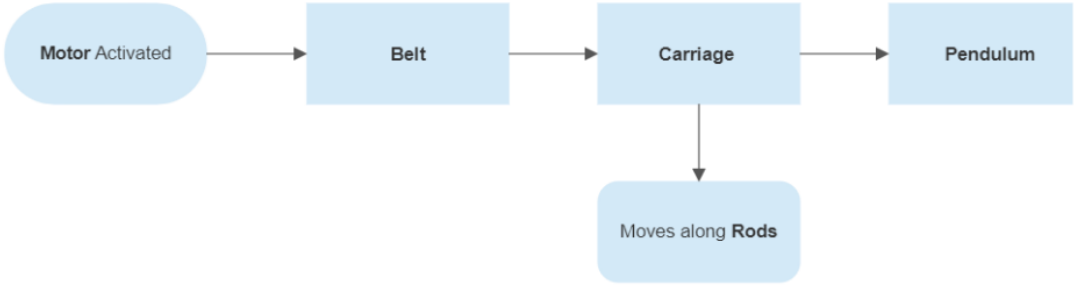
Electrical

- Power switch
- Emergency stop
- Power supply
- TFT screen



Austin

Subsystem 1 Mechanical Design

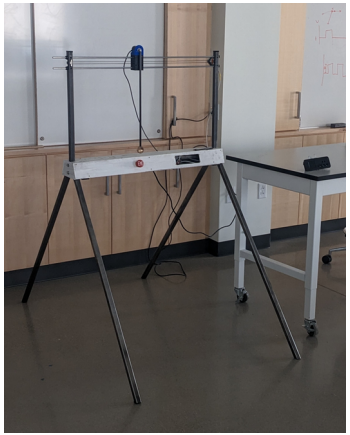


Joe

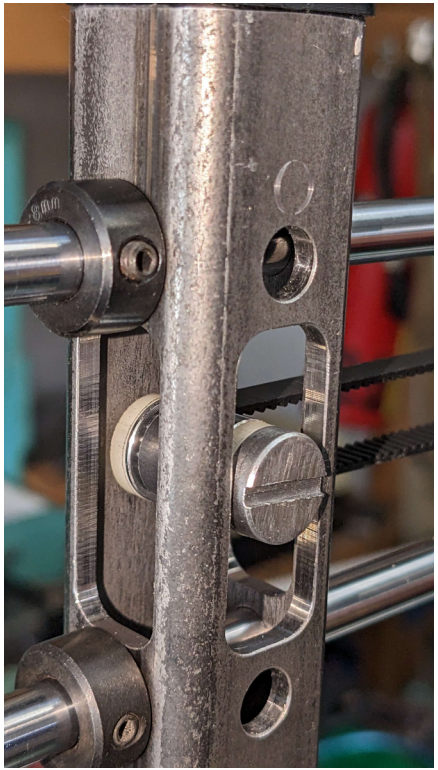
Subsystem 1 Mechanical Design

Carriage:

Full System Assembly:

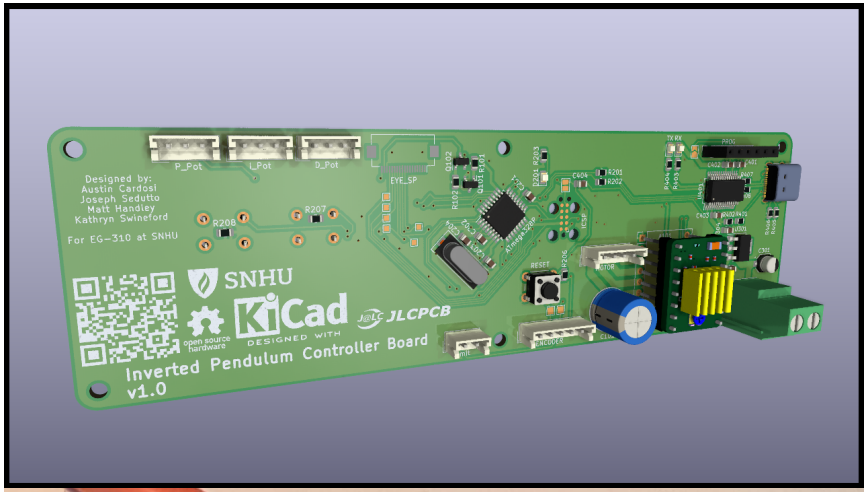


Self Tensioning Cam



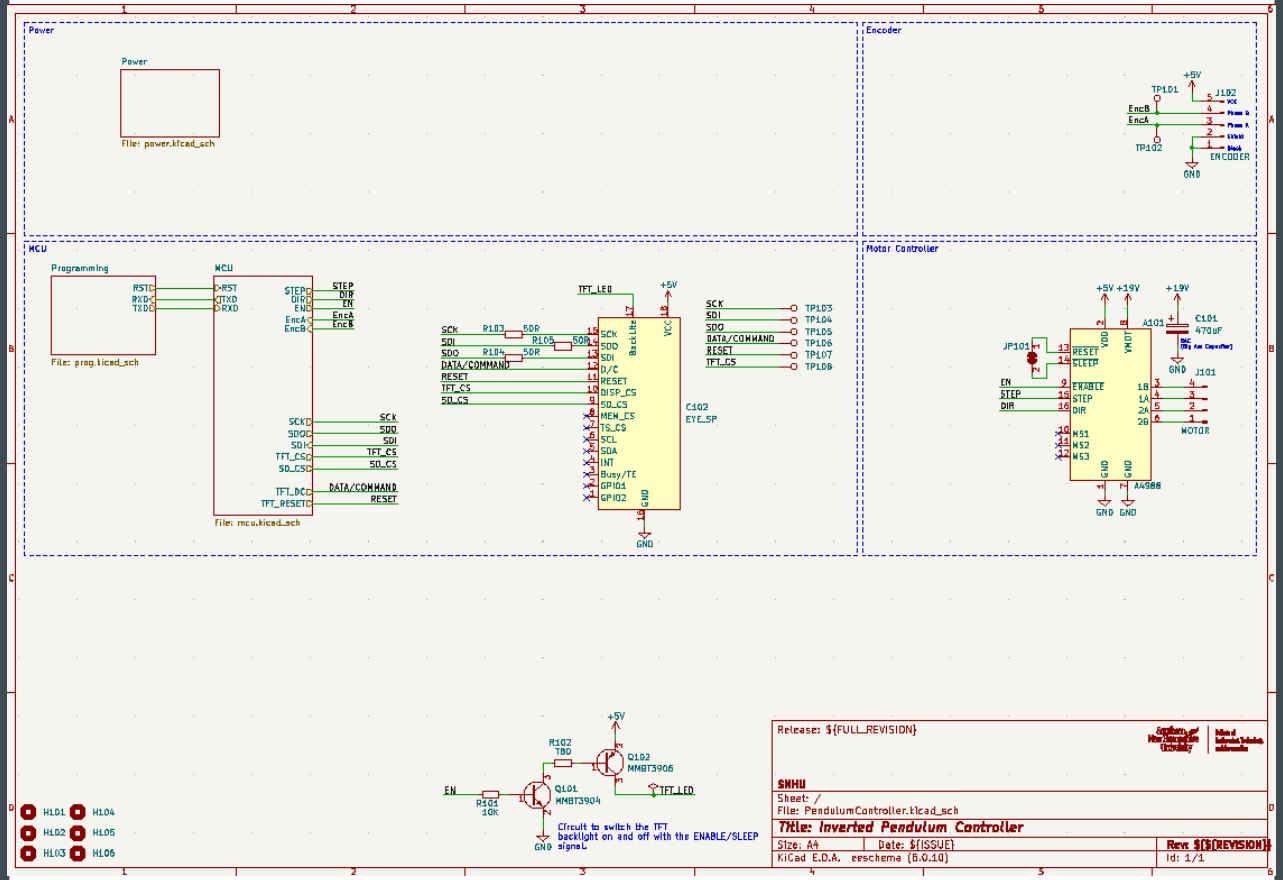
Joe

Subsystem 2 Electrical Design



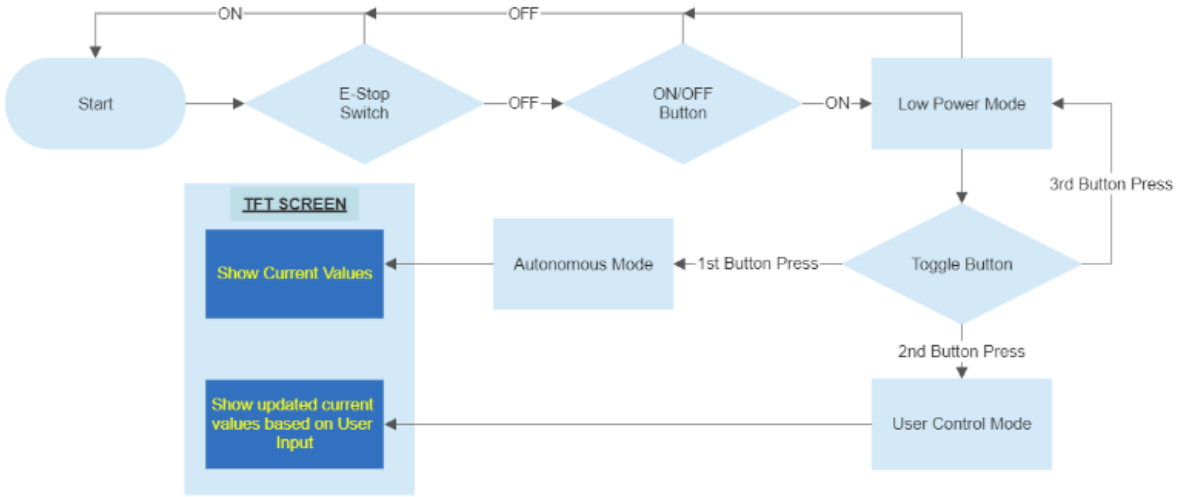
Joe

Subsystem 2 Electrical Design



Joe

Subsystem 3 Software Design



Joe

Subsystem 3 Software Design

PID Balancing Loop:

```
void updateMotor()
{
    if (abs(err) < 800)
    {
        int delay = map(abs(setpoint), 0, 100, 20000, MAX_SPEED);
        if (setpoint > 0)
        {
            digitalWrite(DIR, HIGH);
            pos++;
        }
        else
        {
            digitalWrite(DIR, LOW);
            pos--;
        }

        digitalWrite(STEP, HIGH);
        delayMicroseconds(350);
        digitalWrite(STEP, LOW);
        delayMicroseconds(delay);
    }
}
```

```
err = encoder.read() - (COUNTS_PER_ROTATION / 2);

// Primitive I
if (abs(err) < 20)
{
    err = err * 1.2;
}

setpoint = constrain((err * 0.3) * 10, -101, 101);
setpoint = -setpoint;

Serial.println(err);

updateMotor();
```

Joe

Subsystem 3 Software Design

Encoder Calibration:

```
bool calibrate(Encoder &enc)
{
    Serial.println(F("Begining Calibration."));
    for (int i = 0; i < 500; i++)
    {
        Serial.println(enc.read());
        delay(20);
    }

    enc.write(COUNTS_PER_ROTATION / 2);
    return true;
}
```

Joe

Subsystem 3 Software Design

Spin-Up Code

```
bool spinUpDone = false;
void spinUp()
{
    if (abs(encoder.read()) < 30 && millis() - lastMax > 400)
    {
        freq = freq * -1;
        Serial.println("Hit apex!");
        lastMax = millis();
        encMax = 0;
    }
    setpoint = freq;

    if (abs(encoder.read()) > 2000)
    {
        spinUpDone = true;
        Serial.println("Spinup done.");
    }
}
```

Joe

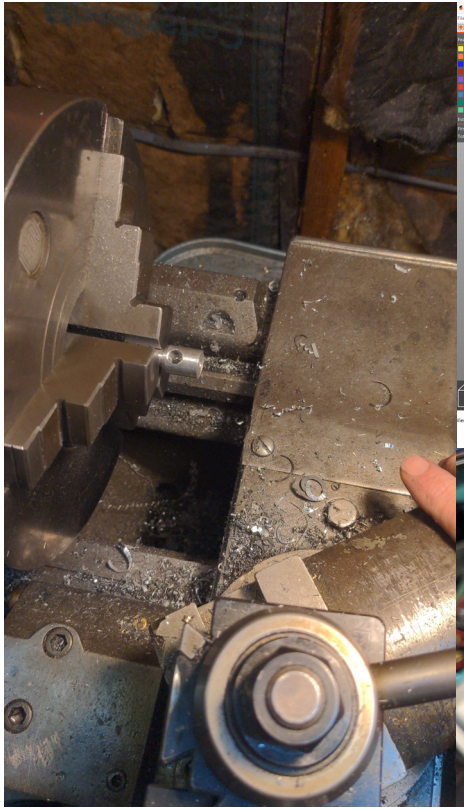
Fabrication & Integration

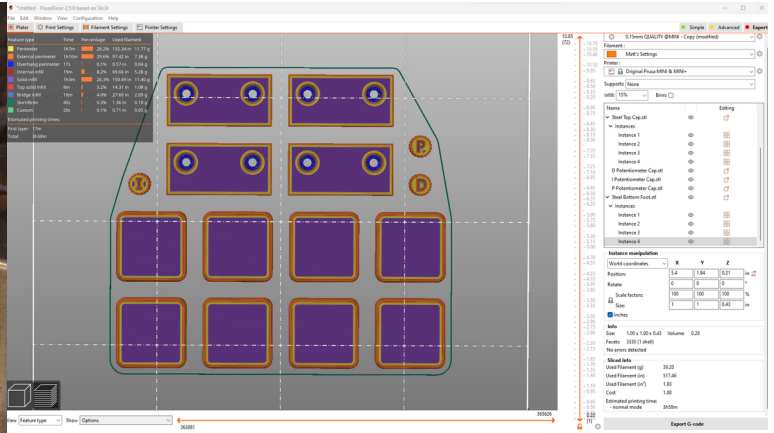
Matt

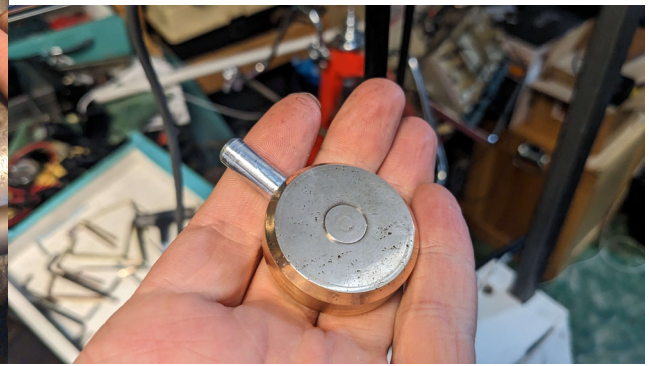
Fabrication and Integration Continued:

Matt

Fabrication and Integration Continued:







Matt

Comprehensive Performance Testing

- **1.0 Frame Stability**
  - The frame can hold >25lbs without deformation
- **1.9 All soldered connections are secure**
  - When shaken all pin connections successfully stay in place
- **1.13 Power**
  - The pendulum stops moving when the power button is pressed
- **1.16 TFT Screen**
  - The screen displays digital PID calculations and constants with a 30 hz refresh rate
- **1.17 Leg Fit**
  - Legs were placed into the frame without slip
- **2.1 Belt Tensioning Assembly**
  - The belt is completely parallel meaning there is no slack
- **2.5 Electrical internal Temps don't get too high**
  - Internal temperature stays well under 150F
- **3.1 Test the self balancing PID**
  - The pendulum balances
- **3.4 Test and ensure harmonic oscillations are working**
  - The pendulum starts in the rest position and can swing itself to the PID loop position

Austin

Schedule Milestones

		Name	Duration	Start	Finish
1		Start	0 days	1/3/22 8:00 AM	1/3/22 8:00 AM
2		**Proj Dev Plan (PDP)**	39 days	**1/3/22 8:00 AM**	**2/24/22 5:00 PM**
3		Research	2 days	1/3/22 8:00 AM	1/4/22 5:00 PM
4		CDR report due	0 days	2/24/22 11:00 PM	2/24/22 5:00 PM
5		ConOps	5 days	1/5/22 8:00 AM	1/11/22 5:00 PM
6		Design concepts	2 days	1/3/22 8:00 AM	1/4/22 5:00 PM
7		Report writing	6 days	1/5/22 8:00 AM	1/12/22 5:00 PM
8		PDP Draft/SRR	0 days	1/21/22 8:00 AM	1/21/22 8:00 AM
9		PDP report	3 days	1/21/22 8:00 AM	1/25/22 5:00 PM
10		PDP Due	0 days	1/25/22 5:00 PM	1/25/22 5:00 PM
11		PDP Presentation	0 days	2/24/22 5:00 PM	2/24/22 5:00 PM
12		**Design**	15 days	**1/3/22 8:00 AM**	**1/21/22 5:00 PM**
13		**Preliminary design**	14 days	**1/3/22 8:00 AM**	**1/20/22 5:00 PM**
14		Mechanical	7 days	1/3/22 8:00 AM	1/11/22 5:00 PM
15		Electrical	5 days	1/3/22 8:00 AM	1/7/22 5:00 PM
16		Programming	14 days	1/3/22 8:00 AM	1/20/22 5:00 PM
17		Func Proto Build/Test	3 days	1/10/22 8:00 AM	1/12/22 5:00 PM
18		**Detailed design**	8 days	**1/12/22 8:00 AM**	**1/21/22 5:00 PM**
19		Mechanical	5 days	1/12/22 8:00 AM	1/18/22 5:00 PM
20		Electrical	5 days	1/13/22 8:00 AM	1/19/22 5:00 PM
21		Programming	7 days	1/13/22 8:00 AM	1/21/22 5:00 PM
22		System Requirements	5 days	1/3/22 8:00 AM	1/7/22 5:00 PM
23		Critical Design Review	0 days	1/21/22 5:00 PM	1/21/22 5:00 PM
24		Spring Break	5 days	2/25/22 8:00 AM	3/3/22 5:00 PM
25		**Fabrication and Integrat**	14 days	**3/4/22 8:00 AM**	**3/23/22 5:00 PM**
26		Build	7 days	3/4/22 8:00 AM	3/14/22 5:00 PM
27		Assembly	7 days	3/15/22 8:00 AM	3/23/22 5:00 PM
28		Functioning Prototype Milest	0 days	3/23/22 5:00 PM	3/23/22 5:00 PM
29		Test Readiness Review	0 days	3/23/22 5:00 PM	3/23/22 5:00 PM
30		**Testing**	8 days	**3/23/22 5:00 PM**	**4/4/22 5:00 PM**

Kat

Schedule Milestones (cont.)

30		**Testing**	8 days	**3/23/22 5:00 PM**	**4/4/22 5:00 PM**
31		Testing plan	0 days	3/23/22 5:00 PM	3/23/22 5:00 PM
32		Subsystem testing	4 days	3/24/22 8:00 AM	3/29/22 5:00 PM
33		Integrated testing	4 days	3/30/22 8:00 AM	4/4/22 5:00 PM
34		**Final phase**	10 days	**4/5/22 8:00 AM**	**4/18/22 5:00 PM**
35		Parts order	1 day	4/5/22 8:00 AM	4/5/22 5:00 PM
36		Final Report Draft	10 days	4/5/22 8:00 AM	4/18/22 5:00 PM
37		**Final Report Due**	5 days	**4/19/22 8:00 AM**	**4/25/22 5:00 PM**
38		Final iteration	5 days	4/19/22 8:00 AM	4/25/22 5:00 PM
39		**Final presentations**	5 days	**4/26/22 8:00 AM**	**5/2/22 5:00 PM**
40		Final demonstration	5 days	4/26/22 8:00 AM	5/2/22 5:00 PM
41		Finish	0 days	4/18/22 5:00 PM	4/18/22 5:00 PM

Kat

Project Budget

Item	Qty	Reference(s)	Value	Price
1	1	A101	A4988	$0.00
2	1	C101	470uF	$0.00
3	1	C102	EYE_SP	$0.00
4	2	C201, C202	100nf	$0.00
5	2	C203, C204	22pf	$0.00
6	1	C301	10uf	$0.00
7	6	C302, C303, C401, C402, C403, C404	100nF	$0.00
8	1	C304	1uF	$0.00
9	1	D201	STAT	$0.00
10	1	D301	PWR	$0.00
11	1	D401	RX	$0.00
12	1	D402	TX	$0.00
13	6	H101, H102, H103, H104, H105, H106	PiMountingHole	$0.00
14	1	J101	MOTOR	$30.00
15	1	J102	ENCODER	$0.00
16	1	J201	ICSP	$0.00
17	1	J202	Limit	$0.00
18	1	J302	19V Input	$0.00
19	1	J401	USBC	$0.10
20	1	J402	PROG	$0.00
21	1	JP101	SolderJumper_2_Bridged	$0.00
22	1	JP201	RSTBridge	$0.00
23	1	JP401	SolderJumper_2_Open	$0.00
24	1	Q101	MMBT3904	$0.00
25	1	Q102	MMBT3906	$0.00
26	4	R101, R207, R208, R402	10K	$0.00
27	1	R102	TBD	$0.00
28	7	R201, R202, R203, R206, R301, R403, R404	1K	$0.00
29	1	R401	4K7	$0.00
30	2	R405, R406	5K1	$0.00
31	2	R407, R408	22R	$0.00
32	1	RV201	P_Pot	$0.00
33	1	RV202	I_Pot	$0.00
34	1	RV203	D_Pot	$0.00
35	1	SW201	RESET	$0.00
36	1	SW202	START_SW	$0.00
37	1	SW203	STOP/MODE_SW	$0.00
38	2	TP101, TP102	TestPoint	$0.00
39	6	TP103, TP104, TP105, TP106, TP107, TP108	TestPoint	$0.00
40	1	U201	ATmega328P	$0.70
41	1	U301	NCV1117ST50T3G	$0.50
42	1	U401	FT232RL	$0.10
43	1	Y201	16mHz	$0.01
44	1	Carbon Fiber Tubing	4ft	$19.94
45	4	Linear Ball Bearing	8mm	$7.99
46	4	Dry Bearings	8mm	$6.99
47	1	Timing Pulley		$16.23
48		Steel		$50.00
49		Aluminum Stock		$0.00
50		Brass Stock		$0.00
51		3D Print		$20.00
52		Aluminum Rods		$20.00
		Loctite		$0.00
		Sealant		$0.00
		E-Stop		$12.00
		Belt		$5.00
		Wood		$0.00
			Total	$189.56

Kat

Lessons Learned

- Stepper motors aren’t great for this application: they don’t like rapidly changing variable speeds and low speed operation.

- Always have a backup plan to each component

- GitHub is a godsend. Having all your files stored in one place and backed up on the cloud makes for a highly iterative and efficient workflow.

- Communication is key when working with a team

- Should have ordered parts earlier

Austin

Thank you all for your help!

To Professor Husson and Professor Guo:

-Thank you for guiding us through this project and giving us your insight as we met hurdles.

To Professor Carlstrom and Joe Donovan:

-Thank you for funding the project and helping us order parts. Thank you for your insight on how and whether our system will mesh properly.

To Michael Sedutto:

-Thank you for welding our frame and allowing us to use your CNC machine. Without you, we wouldn't have a mechanical system.

Matt

Questions?

Backup Slides

Reference/Citations

Rotary Inverted Pendulum. Quanser. Retrieved January 22, 2023,

from https://www.quanser.com/wp-content/uploads/2017/03/ROTPEN-graphics.jpg

Lego, Raspberry and Python Project - Reaction Wheel Inverted

Pendulum. (2022, April 16). YouTube. Retrieved January 22, 2023, from

https://youtu.be/WObG2LoSEwQ

List and Summary of Analyses

- Frame stability - Requirement met
- Motor torque and responsiveness - Requirement met
- Encoder accuracy - Requirement met
- Limit switch proper behavior - Requirement met
- Electrical components properly connected and grounded - Requirement met
- Motor controller provides enough current - Requirement met
- USB-C port for programming - Requirement met
- Power supply converts 120V to 12V - Requirement met
- 3 potentiometers proper behavior - Requirement met
- Estop stops operation in less than 10 ms - Requirement met
- Power switch turns the system on and off - Requirement met
- TFT screen has color display and refreshes - Requirement met
- Left slip fit is tight with no wobble - Requirement met
- Belt tensioning assembly eliminated belt slip - Requirement met
- Sprocket and belt system are square and parallel - Requirement met
- Minimal friction between encoder and pendulum - Requirement met
- Internal temps stay under 90 C - Requirement met
- Pendulum can self balance for more than thirty seconds - Requirement met
- TFT screen creates a legible user interface - Requirement met
- Pendulum can oscillate itself into a starting position - Requirement met
- Pendulum has a user interactive mode where the user can manipulate the P, I, and D constants - Requirement Failed
- Users can toggle between different modes - Requirement Failed

Full Code

Calibrate.h:

```
// Libs
#include <Arduino.h>
#include <Encoder.h>

// Some constants
#define COUNTS_PER_ROTATION 4096

/**
 * @brief Auto calibration routine.
 *
 * @param enc An encoder to calibrate (sinusoid alpha decay).
 * @return true if successfully calibrated,
 * @return false on calibration error.
 */
bool calibrate(Encoder &enc)
{
    Serial.println(F("Begining Calibration."));
    for (int i = 0; i < 500; i++)
    {
        Serial.println(enc.read());
        delay(20);
    }

    enc.write(COUNTS_PER_ROTATION / 2);
    return true;
}
```

Full Code

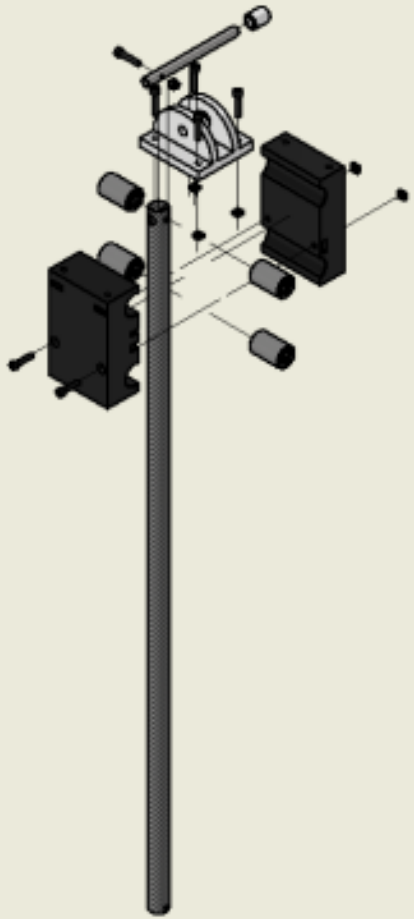
controls.cpp:

```
/**
 * @file controls.cpp
 * @author Joe
 *
 * @brief Inputs and Controls
 */

#include <Arduino.h>
#include <boardPins.h>

void beginControls()
{
    ;
}

int getP()
{
    return analogRead(PPIN);
}

int getI()
{
    return analogRead(IPIN);
}

int getD()
{
    return analogRead(DPIN);
}
```

Full Code

main.cpp

```cpp
/**
 * @file main.cpp
 * @author Joe
 * @brief Source code for EG-310 Pendulum demonstrator
 */

// AVR/System
#include <Arduino.h>
#include <Encoder.h>

// Our libs
#include "display.h"
#include "calibrate.h"

// Pindefs and misc
#include <boardPins.h>

// Instantiate Objects
Display tft = Display(TFT_CS, TFT_DC, TFT_RST);
Encoder encoder(ENC_A, ENC_B);

void setup(void)
{
    // Configure serial
    Serial.begin(115200);
    Serial.println(REV);

    // Begin our display
    tft.begin();
    tft.loading(0); // Update the loading bar

    // Inputs and Outputs
    pinMode(STAT_LED, OUTPUT);
    pinMode(ENABLE, OUTPUT);
    pinMode(DIR, OUTPUT);
    pinMode(STEP, OUTPUT);
    pinMode(PPIN, INPUT);
    pinMode(IPIN, INPUT);
    pinMode(DPIN, INPUT);

    digitalWrite(ENABLE, HIGH);
    digitalWrite(STAT_LED, HIGH);
    digitalWrite(DIR, HIGH);

    // Done loading
    tft.loading(100);

    // calibrate(encoder);
    Serial.println(F("Done. Encoder is now at "));
    Serial.println(encoder.read());
}

const int MAX_SPEED = 900;
int setpoint = -100;
int pos = 0;
int err = 0;
int divisor = 0;

void updateMotor()
{
    if (abs(err) < 800)
    {
        int delay = map(abs(setpoint), 0, 100, 20000, MAX_SPEED);
        if (setpoint > 0)
        {
            digitalWrite(DIR, HIGH);
            pos++;
        }
        else
        {
            digitalWrite(DIR, LOW);
            pos--;
        }

        digitalWrite(STEP, HIGH);
        delayMicroseconds(350);
        digitalWrite(STEP, LOW);
        delayMicroseconds(delay);
    }
}

bool myDir = true;

void loop()
{
    // Plot table
    // tft.plot(encoder.read() / 100);

    // Draw screen
    // Serial.println("Drawing New Issues");
    // tft.displayIssues();

    // Blink
    // digitalWrite(STAT_LED, HIGH);
    // delay(100);
    // digitalWrite(STAT_LED, LOW);
    // delay(100);

    // -  (pos * 0.01)
    err = encoder.read() - (COUNTS_PER_ROTATION / 2);

    // Primitive I
    if (abs(err) < 20)
    {
        err = err * 1.2;
    }

    setpoint = constrain((err * 0.3) * 10, -101, 101);
    setpoint = -setpoint;

    Serial.println(err);

    updateMotor();
}
```



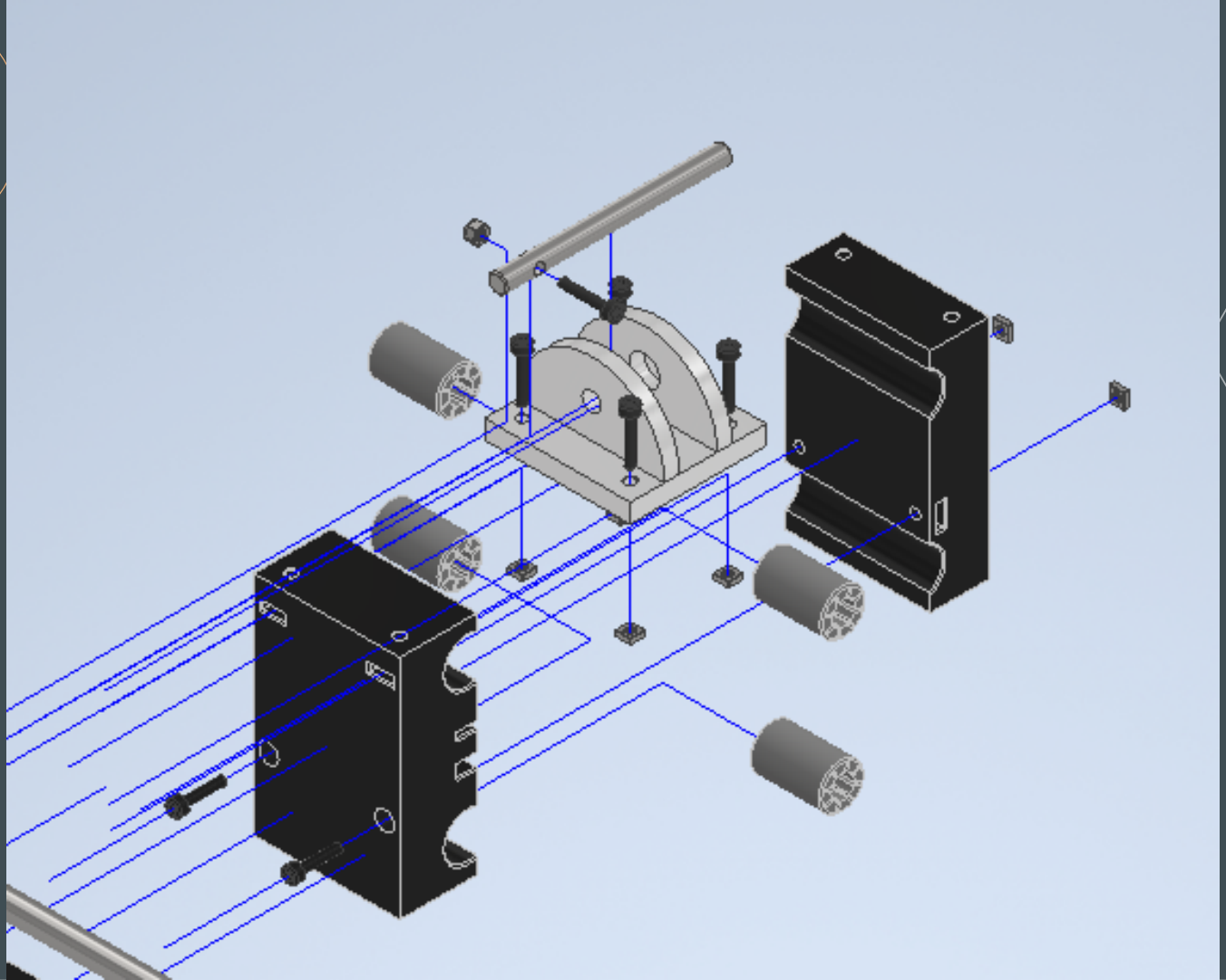
PARTS LIST

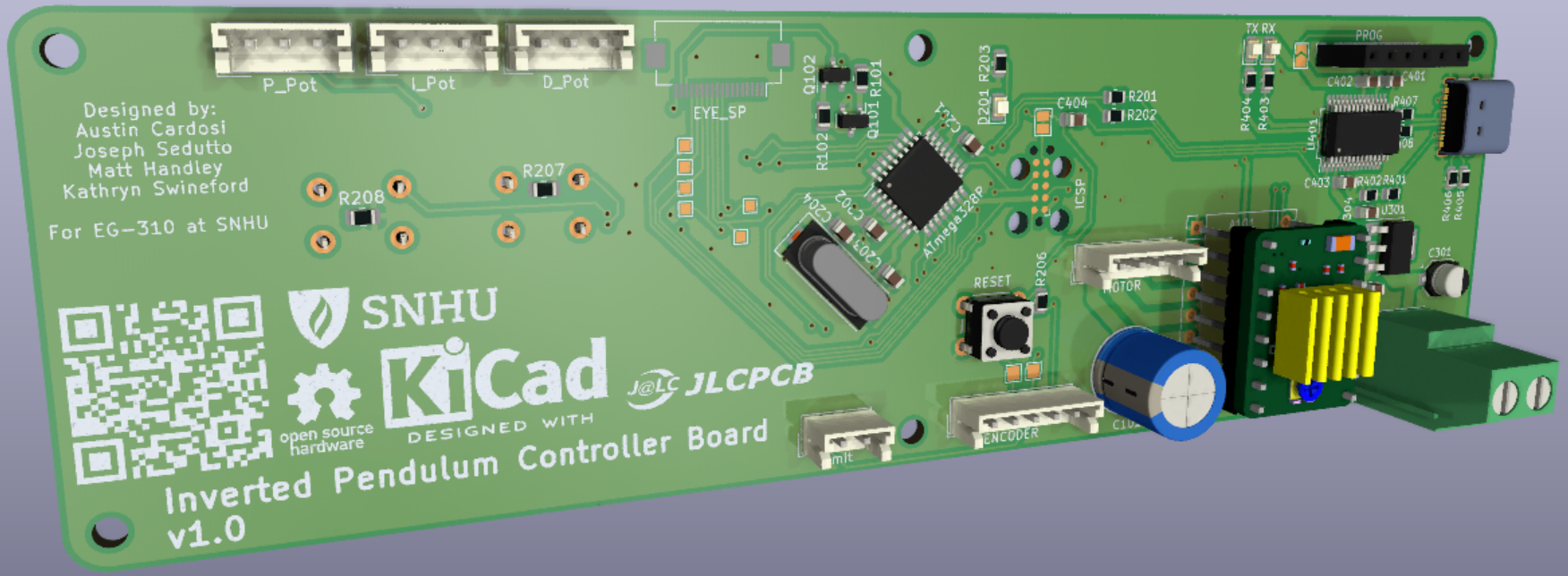
ITEM	QTY	PART NUMBER
1	4	Dry Bearing
2	1	Carriage Design Left
3	1	Carriage Top
4	1	Carriage Axle V1
5	1	Pendulum Tubing
6	1	Carriage Axle Set Screw Holder
7	5	hex_socket_cylindrical_head_M3x16
8	1	M3 Nyloc Nut
9	6	M3 Square Nut
10	1	Carriage Design Right
11	2	VITE_TESTA_CILINDRICA_ALTA_ESAGONO_INCASSATO

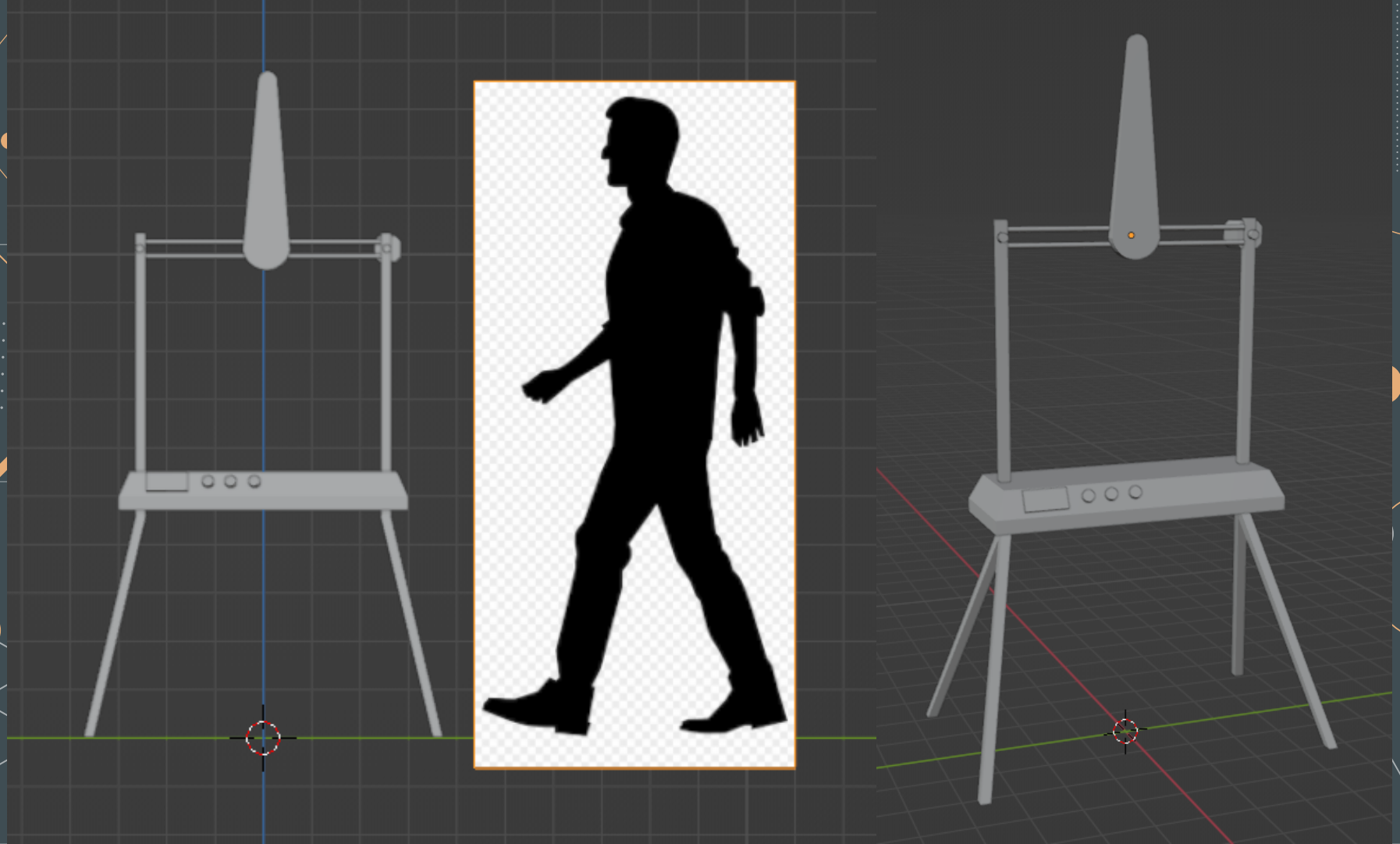
Carriage Version 1 Exploded View

D

7/8




P_Pot
I_Pot
D_Pot
Designed by:
Austin Cardosi
Joseph Sedutto
Matt Handley
Kathryn Swineford
For EG-310 at SNHU
R208
R207
EYE_SP
Q102
Q101
R101
R102
D201
R203
C404
R201
R202
TX RX
PROG
C402
C401
R404
R403
R407
C403
R402
R401
U301
C301
R406
R405
ICSP
ATmega328P
RESET
R206
MOTOR
ENCODER
SNHU
KiCad
DESIGNED WITH
open source hardware
JLCPCB
Inverted Pendulum Controller Board
v1.0



Human Chegg (Matt) pinned a message to this channel. See all pinned messages. Today at 4:19 PM

4:19 PM BOT GitHub

KenwoodFox

[EG-310-InvertedPendulum:feat/initalSoftware] 1 new commit

3418f24 Bump - KenwoodFox

4:21 PM BOT AKMJ Hook

firmware.hex
37.31 KB

BOT AKMJ Hook

Board-Manual.pdf
2.15 MB

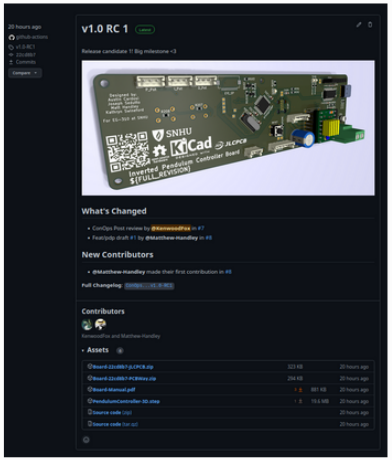
Assembly

The Pendulum 328 is a prototype built from prototype parts with the primary goal of being an educational tool.

Should parts ever need to be re-ordered, or this process used to drive future projects, this section of the documentation details those processes!

Ordering the PCB

The Pendulum 328's CI/CD will automatically attach new builds of the gerber files to each release, when ordering new boards, always use the release tagged Latest. (Not necessarily the release in the photo.)



This guide will use JLCPCB for ordering the boards, but we also build gerbers for PCBWay as well