Project Final report

Fuad Aljadani

04/01/22

Last Updated: 04/01/22

Revision /Update: V 0.1

Revision History

Date	Initials	Version	Section Updated	Notes
04/01/22	ARJ	0.1	Draft created	

Table of Content

Definitions

The following definitions, acronyms, and abbreviations are found in this document:

Table 1 - Definitions

Term	Definition
Inverted Pendulum	a pendulum that has its center of mass above its pivot point
Algorithm	a process or set of rules to be followed in calculations or other problem-solving operations
Cart motor	a retro-fit circuit board that by-passes all three 6-pin round signal connection ports

Exclusive summary

The inverted Pendulum system is an example system in physics, dynamics, and control field. The center of mass of the Inverted Pendulum is about its pivot point and the Inverted Pendulum can be hung stably in its inverted position by using the control system to monitor the angle of the rod and moving the pivot point horizontally back below the center of gravity when the rod starts to fall over to maintain its balance. So, the purpose of this project is to repair a digital control for an inverted pendulum system which can keep the Inverted pendulum in a balanced position.

In this project, we need to repair a broken Inverted Pendulum for the ENT department in order to use for teaching purposes. In order to meet our goal which is to get the broken Inverted Pendulum fully repaired and operate again, we need to judge whether all the parts in the Inverted Pendulum are operating normally. Then we will decide which components need to be replaced or need to be repaired and what software programming is needed for system control to work. After our assessment on all the components, we will implement a system solution which includes a power supply that will generate power for the microcontroller and the cart motor. Also, this program will have a microcontroller that will meet the requirement for both hardware and software as this program will have a microcontroller that is able to communicate with the power supply and the cart motor by cables in order to give the instruction to the system. Also, the microcontroller should be capable of having a USB port in order to download the program. It will also be able to receive and send data to process encoder data and control the DC motor such that the inverted pendulum is balanced in an upright position. We will outline our test plans and conduct them to identify our project's performance against our project's goals which is to get the broken Inverted Pendulum fully repaired and operate again with a new software system.

The current brain that controls the inverted pendulum is the microcontroller, in our project is the Motorola MMC2001 Mcore Microcontroller. The CMD 2001 is a plug and play system with the supplied PB2001 Controller Module and it is a fully configured system for the Motorola MMC2001 Mcore Microcontroller. The whole system includes a development board with Mbug Monitor/Debugger, 64K x 16Flash EEPROM, 128K x 16 SRAM, DB9 serial Cable, 9v-300ma Wall Plug, hardware manual and UTL2001 software disk. Also, there are programming utilities and support software included in the UTL2001 software disk, and the free software is available including the GNU Assembler, C compiler, C++ compiler. We troubleshooted the system and tried to figure out what are the components that need to be fixed or replaced in order to get the system back to operation.

It was proven that the Inverted Pendulum could work manually by doing an experiment of disconnecting the cable from the logic board with the help of our professor. There are two cables in the power supply and one of them is connected to the card motor, a retro-fit circuit board that by-passes all three 6-pin round signal connection ports which will give a 17V to the cart motor. At the same time, this cable is able to send data from the cart motor to the power supply. Data can be sent through the pins from the logic board to the cart motor. We have been told by our professor that this system is operational in manual mode, and perhaps only needs a new microcontroller, USB communication connections and a program to bring this system back into operation. When we tried to experiment with this system, we observed that when the power was on, the DC motor was not able to respond. Also, we made sure no data was sent through the logic board by disconnecting the cable from the logic board and we were able to control the DC motor by the two push buttons on the power supply. The DC motor was able to turn left and right directions by pushing different push buttons, and one of the LED lights on the power supply will display different colors when different push buttons are pressed in order to show the different directions the DC motor is moving towards. According to all above observations in the system, we proved that this system could work manually and only needs a new microcontroller, USB communication connections and a program to bring this system back into operation. The hardware is working fine, except the weighted cart needs more weights to allow it to travel along the rails more smoothly and the motor gear that contacts the rail needs to be replaced.

The current system is using INT_PORT in its logic board and it is connected to the power supply by a cable. Not only the cable can connect to the power supply but also allows sending data to or receiving data from the cart motor. Also, the INT_PORT is able to access the external interrupt source. This allowed the communication between microcontroller and cart motor. Also, the power wires are coming out from the power supply to the logic board. Specifically, to TB1 which is a Power connector, and it is used as an input power supply to the board.

Our proposed solution to balance the Inverted Pendulum is explained as follows. The electricity will start to flow to the transformer as soon as the outlet is plugged into the wall. Then the switch will turn on and the power supply will power up both the cart motor and the logic board. The cart motor will receive 17V of power, the encoder of the DC motor will start to give measurement and the DC motor will receive the order so that the DC motor will start to move left and right so as the inverted pendulum will start to swing. The encoder of the DC motor will keep measuring and it will send data to the microcontroller, and the encoder of the swing will send data of swing position to the microcontroller so that the microcontroller will give command to the inverted pendulum until the inverted pendulum is balance.

In our phase 2 project, we are using an Arduino 2560 microcontroller for this project due to the availability of the I/O pins in Arduino 2560 microcontroller. This microcontroller has more than 6 I/O pins so if one pin burns out, we can use another pin instead of replacing the whole microcontroller. The original voltage from the power supply was 17V, but the highest voltage the microcontroller can take is 12V. So, we have designed a shield for the microcontroller which is going to have a voltage regulator and the voltage regulator will reduce the voltage to 9V. We also have built a safety remote control to prevent the user from being hit by the Pendulum swing.

The remote control has three push buttons which are push buttons to control start/stop, push button to move the cart motor left direction, and push button to move the cart motor right direction. The remote control is connected to the project directly on the microcontroller shield through a 20 inches long ribbon cable. Also, the motor control is using four pins, including three digital pins and one ground pin.

Introduction/ Scope/ Marketing Requirements

The scope of this project is to fully restore an Inverted Pendulum that was provided to us. The objective is to repair the Inverted Pendulum to working order. The customer will then implement it for learning purposes. Major parts that we will keep will be the power supply, DC motor, rod, cart mounting platform and base, track rails, power cable. ON/OFF switch and encoders. The parts that will need to be replaced are the microcontroller and the two serial communication ports and cables (DB9 and DB25). We will replace the two serial communication ports with more modern communication ports such as USB. These RS232 ports were used to program the current microcontroller from a PC and possibly a host PC and a user interface to send system start and stop commands.

- The pendulum will have a microcontroller which executes the algorithm to balance the pendulum.

- The power supply will give power to the microcontroller as well as the cart motor that will swing the pendulum. Power supply is the source of power, and it is also the middle piece of this inverted pendulum software system. It has two cables going out not only for powering but also for allowing data communication between the cart motor and the logic board for receiving and sending data. The cable goes to the cart motor allowing data communication and power convey, the power goes to the cart motor from the power supply is 17V. The cable from the power supply going to the logic board only allows data communication. The wires that come out of the power supply to the logic board also give 17V.

- We will use a new microcontroller to replace the older, no longer operational microcontroller. We will mount the new microcontroller on the base platform, in the same general mounting location for the current microcontroller. It includes a processor that can process and respond to instructions sent by the code. The microcontroller does this by interpreting data it receives from its I/O peripherals using its central processor. The microcontroller will need to have input/output pins, analog inputs (hardware serial ports), USB connection, and a reset button.

- Considerations:
    - Option1: Arduino mega 2560 is used to connect to the software on computers in order to help read the input signal and turn it into an output signal.
    - Option 2: Teensey 3
    - Option 3: Launchpad MSP430

smaller mechanical parts such as the cart tracking gears. These might need replacing as their teeth show a bit of wear and this could cause the cart housing to skip/stall/misalign on the track rail.

- Small mechanical parts might need replacing as their teeth show a bit of wear and this could cause the cart housing to skip/stall/misalign on the track rail.
    - cart tracking gears
    - Weight (might need more weight)
    - Document operation user manual
    - The power supply (The power supply is powering up the chip and the DC motor by testing. Also, all the parts in the power supply are receiving power and working fine.

- Phase 2:
    - Customized shield for microcontroller
    - Customized remote control
    - 14 pins connector
    - Two 4-pins connectors
    - 20 inches ribbon cable
    - Voltage regulator
    - Heat sink
    - Three push buttons

Customer / Marketing Requirements

1. The swinging pendulum will stay balanced for at least 30 sec and find its position if external force applies.
2. Will be a user manual provided which includes equipment information and functions
3. Should explained how the software is connecting with the hardware
4. The code should be explained
5. Should be cost effective around $250.00
6. The software should operate continuously

Specification requirements

Engineering / Functional Specifications

Table 3 - Engineering Specifications

#	Source	Specification	Justification	Priority
1.01	**1, 3, 6**	**Power supply**	Will be able to convert electric current from a source to the correct voltage, current, and frequency to power the load. Generating (DC 17V)	**High**
	1, 3, 6	**switch**	120V. Normally open.	**High**
	1, 3, 6	**DC motor**	**motor accepts 17V DC input voltage.**	**High**

	1, 3	**Capacitor**	4700 uF 25 V	**High**
	1	**Rod**	**The rod will not be easily damaged and can handle a certain weight while performing.**	**Medium**
	1, 3	**Encoders**	**The encoder will not fail to detect the motor's rotation**	**High**
	1, 3, 5, 6	**Microcontroller**	Mbug Monitor/Debugger, 64K x 16Flash EEPROM, 128K x 16 SRAM, DB9 serial Cable, 9v-300ma Wall Plug	**High**
	1	**Base and Rails**	**The pendulum will not run out of the rails**	**High**

	2	**User Manual**	**The user manual should include specific information about equipment information and functions**	**Medium**
	3	**Plug**	**Allow electrical equipment to connect to the electrical grid (120V/ 60Hz)**	**High**

Related Projects

- **Inverted pendulum**: An inverted pendulum is a pendulum that has its center of mass above its pivot point. By using the control system to monitor the angle of the rod and moving the pivot point horizontally back below the center of gravity when the rod starts to fall over to maintain its balance, it can be hung stably in this upside-down position.
https://ctms.engin.umich.edu/CTMS/index.php?example=InvertedPendulum&section=SystemModeling
https://www.quanser.com/products/linear-flexible-inverted-pendulum/

- **Linear Flexible Joint**: The Linear Flexible Joint is a passive linear cart that connects to the Linear Servo Base Unit through a linear spring.
https://www.quanser.com/products/linear-flexible-joint/
- **Linear Flexible Joint with Inverted Pendulum**: The Linear Flexible Joint with Inverted Pendulum combines two fundamental control challenges to give students an opportunity to a more advanced modeling and control challenge.

https://www.quanser.com/products/linear-flexible-joint-inverted-pendulum/#overview

- **Linear Servo Base Unit with Inverted Pendulum**: The Linear Motion Control Lab is one of the most popular, flexible, and modular solutions for teaching controls.
https://www.quanser.com/products/linear-servo-base-unit-inverted-pendulum/

User Roles and Responsibilities

In this project, group members are all expecting to have different assigned works to work on to contribute to the project. For our inverted pendulum project, we have a few major tasks that group members need to work on which are power supply, logic board, and cart motor. Within those three big tasks, group members all need to do experiments and research in order to accomplish the whole project. Here are our different roles in contributing to the whole project.

Name	Fuad	Gryffyn	Anthony
Team project roles	Circuit analyzer System documenter Researcher	Circuit analyzer Control(PID)	Main Programmer
Roles	**Timekeeper:** Keeps the group aware of time constraints and deadlines and makes sure meetings start on time.	**Harmonizer:** Strives to create a harmonious and positive team atmosphere and reach consensus (while allowing a full expression of ideas.)	**Devil's Advocate:** Raises counter-arguments and (constructive) objections, introduces alternative explanations and solutions.
Roles	**Checker:** Checks to make sure all group members understand the concepts and the group's conclusions.	**Reporter:** Serves as group spokesperson, summarizing the group's activities and/or conclusions.	**Wildcard:** Assumes the role of any missing member and fills in wherever needed.
Roles	**Recorder:** Takes notes summarizing team discussions and decisions and keeps all necessary records.	**Facilitator:** Moderates team discussion, keeps the group on task, and distributes work.	**Prioritizer:** Make sure the group focuses on most important issues and does not get caught up in details.

Responsibilities - power supply	Identify different components in power supply Testing the functions of components in power supply Research information about power supply identification Drawing schematic for power supply	Drawing schematic for power supply	**N/A**
Responsibilities - cart motor	Identify the connection between power supply and card motor Identify DC motor type Identify encoders types Identify the function of a custom made circuit board.	Underasting how data transferring between the power supply and the cart motor	**N/A**

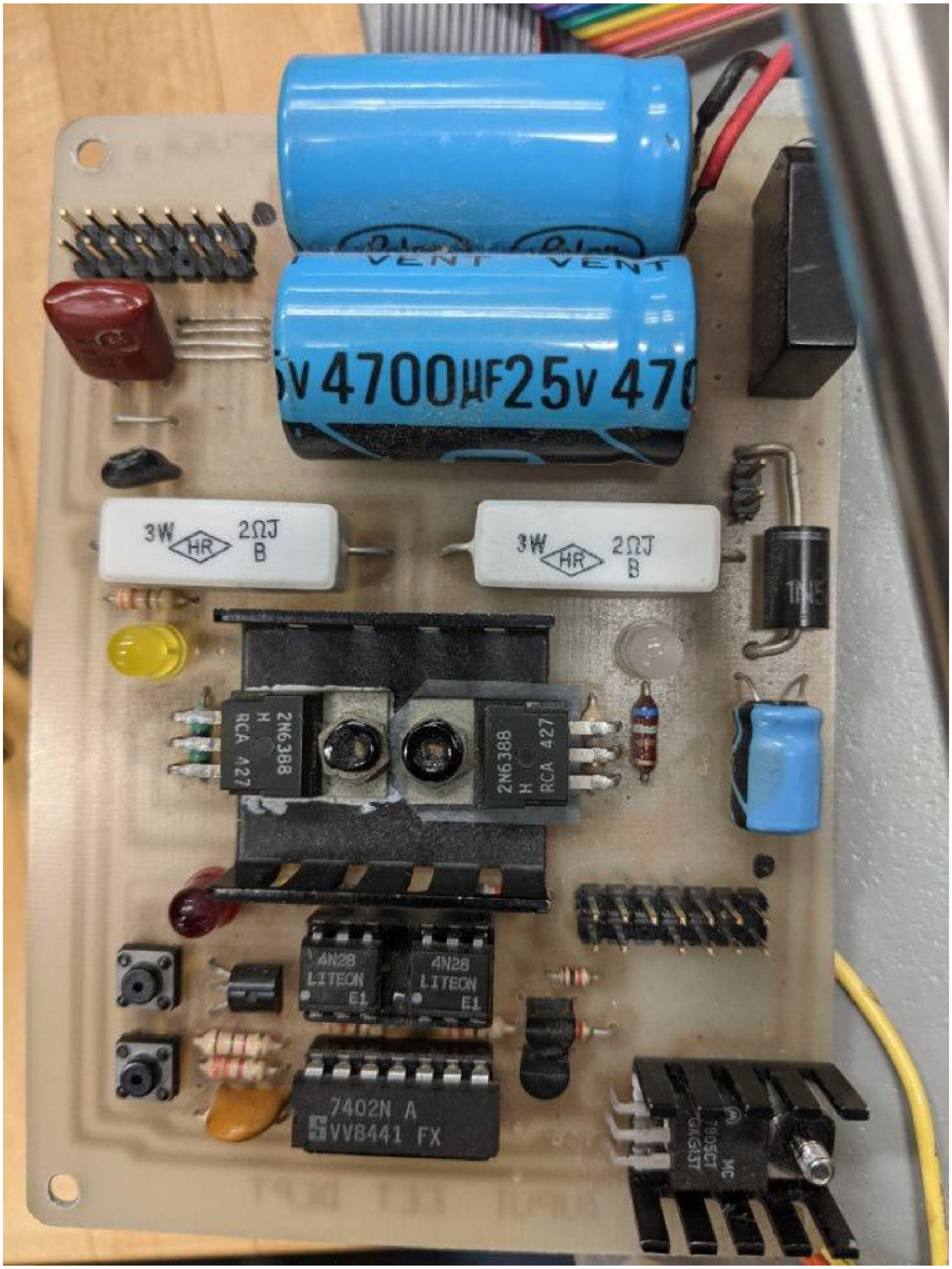
Responsibilities - logic board	Underasting how data transferring between the power supply and the logic board Identify microcontroller type	Underasting how data transferring between the power supply and the logic board Identify microcontroller type	Underasting how data transferring between the power supply and the logic board Identify microcontroller type
Responsibilities - block diagram	Create a block diagram for system connections.	Create a block diagram for system connections.	N/A
Responsibilities - Timeline	N/A	Create Timeline for the project	Create Timeline for the project
Responsibilities - Matrix	Tradeoff analysis	N/A	Tradeoff analysis

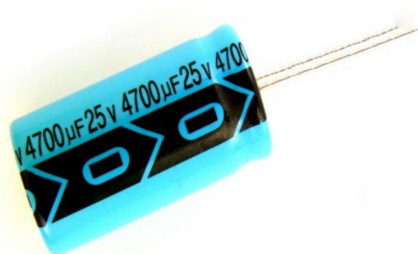
Phase 2 user roles and responsibilities:

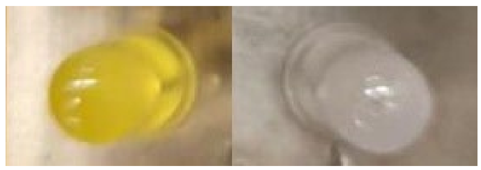
Name	**Fuad**	**Jake**
Team project roles	Circuit analyzer System documenter Researcher	Main Programmer Control(PID)

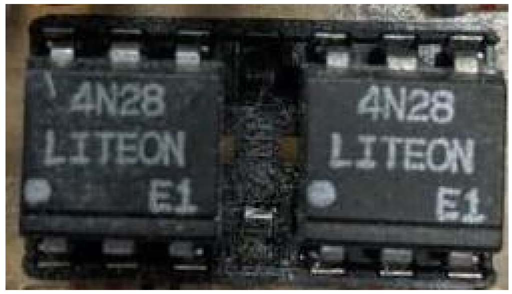
POWER SUPPLY COMPONENTS IDENTIFICATION

GIVEN CIRCUIT:

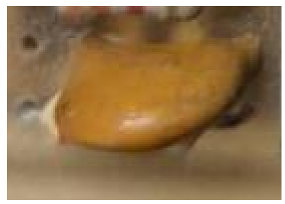
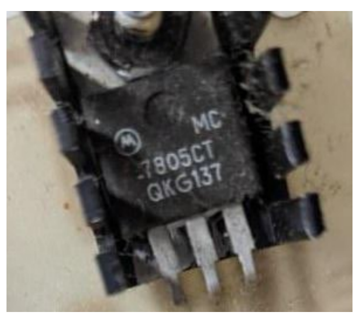


Sr. No.	Part Name	Rating	Part Pic	Product Source
1	Capacitor	4700 uF 25 V		https://www.ebay.co.uk/itm/10pcs-Xicon-4700uF-25v-Radial-Electrolytic-Capacitor-/200832100565
2	Capacitor (104 K)	100 nF		https://www.alibaba.com/product-detail/JXWS3-CBB-capacitor-CL21-250V-104K_60785276061.html?spm=a2700.7724857.normal_offer.d_title.514d37270YDZGL
3	Thermistor (NTC 5D-11)	5Ω 4A		https://www.amazon.com/uxcell-Thermistor-Resistors-Current-Temperature/dp/B07KZYPYJG/ref=sr_1_2?crid=15XOD7F1TFH88&dchild=1&keywords=ntc+5d-11&qid=1634303563&sprefix=NTC+5%2Caps%2C430&sr=8-2
4	Resistor	2Ω 3W		https://www.ebay.com/itm/2-ohm-3-Watt-10-Wire-Wound-Power-Resistor-NOS-New-Old-Stock-QTY-10-ea-D56-/152974628278?_ul=IL

5	Diode (1N5408)	3A		https://www.digikey.com/en/products/detail/fairchild-semiconductor/1N5408/12610864
6	LED's	3V 20 mA		https://www.amazon.com/MCIGICM-Circuit-Assorted-Science-Experiment/dp/B07PG84V17?th=1
7	Bipolar Power Transistor (2N6388)	10 A		https://www.digikey.com/en/products/detail/onsemi/2N6388/605522
8	Capacitor	10 uF 25 V		https://www.pcboard.ca/10uf-25v-aluminum-electrolytic-capacitor.html

9	Optocoupler (4N28)	10 mA 30 V		https://www.digikey.com/en/products/detail/lite-on-inc/4N28/3198886
10	Resistor	2.2 kΩ		
11	Resistor	240 Ω		
12	Resistor	8.2k Ohms 5%		

13	Resistor	10 kΩ 0.5 W		
14	Resistor	47 Ω 0.5 W		
15	Resistor	8.2 kΩ 0.5 W		
16	Transistor (2N3904)	200 mA 40 V		https://www.digikey.com/en/products/detail/fairchild-semiconductor/2N3904/11528113

17	Push Button Switch	12 V		https://www.amazon.com/QTEATAK-Momentary-Tactile-Button-Switch/dp/B07VSNN9S2/ref=sr_1_12?dchild=1&keywords=Slide+Switch+and+Push+Button&qid=1634304829&sr=8-12
18	Capacitor (Ceramic Disc)	47 pF 50 V		https://www.jameco.com/z/DC47-Jameco-Valuepro-Capacitor-Ceramic-Disc-47pF-50V-20-_15512.html
19	Logic Gate IC (7402N)	27 uA 5 V		https://www.digikey.com/en/products/detail/texas-instruments/SN7402N/1575110?s=N4IgTCBcDaIKwE44FoDKA5A7AFgAxnRAF0BfIA
20	Linear Voltage Regulator	2.2 A 5 V		https://eu.mouser.com/ProductDetail/onsemi/MC7805CT?qs=%252B9%2Fcbd0IE0RnZFyGavC%252BVw%3D%3D

High Level Design:

1. Introduction

In this project, we will be repairing a broken Inverted Pendulum for the ENT department for teaching purposes. We will first troubleshoot the broken Inverted Pendulum and find out parts on the Inverted Pendulum that are not operating normally. After our assessments on all the components, we will take further decisions on how to fix those unoperated parts, replace broken parts, or implement software programing for system control to work. We will outline our test plans and conduct them to identify our project's performance against our project's goal which is to fully repair the Inverted Pendulum with a new software system.

2. Scope

Major parts that will need to be replaced are the microcontroller and the two serial communication ports and cables (DB9 and DB25). The two serial communication ports will be replaced with more modern communication ports such as USB. Also, smaller mechanical parts such as the cart tracking gears might need replacement as they got worn by time.

3. Referenced Documents

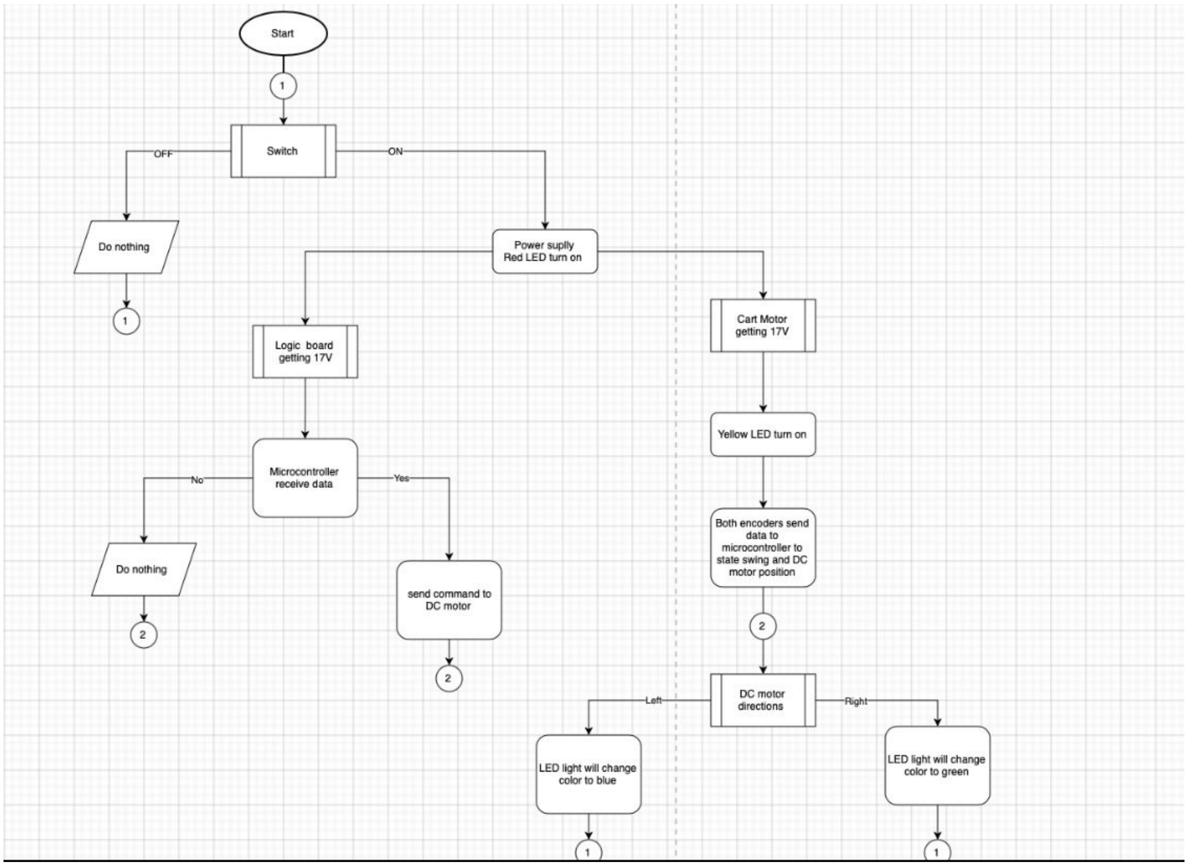
Linear flexible inverted Pendulum	https://www.quanser.com/products/linear-flexible-inverted-pendulum/#overview	The source gave us a brief overview of what a Linear flexible inverted pendulum is and what are the specifications in the inverted pendulum and also some other related information.
Linear servo control lab	https://www.scribd.com/document/462128842/Quanser-Linear-Servo-Control-Systems-Brochure-online-1	This pdf document includes really detailed information about the inverted pendulum and equipment required for the inverted pendulum to work.

AT22LV10- Integrated Circuit	http://ww1.microchip.com/downloads/en/DeviceDoc/doc0190.pdf	This document introduced some basic characteristics of AT22LV10(L).
MMC2001 Reference Manual	https://www.nxp.com/docs/en/reference-manual/MMC2001DDRM.pdf	It is a user manual for MMC2001 and some basic introductions about MMC2001
CMD- 2001 Rev. C User Manual	http://www.ece.ualberta.ca/~cmpe490/documents/axiom/Axiom_CMD2001_Manual.pdf	This document is a user manual for CMD-2001. It includes some features of CMD2001.

4. System- Wide Design Decisions

We will implement a system solution that includes a power supply that can generate electricity for the microcontroller and the cart motor. In addition, the program will have a microcontroller that will meet both the hardware and software requirements, because the program will have a microcontroller that will communicate with the power supply and the cart motor through a cable to give instructions to the whole system. We will use a newer microcontroller to replace the old no longer operational microcontroller. We will install a new microcontroller on the basic platform which is the same as the general installation location of the current microcontroller. The new microcontroller contains a processor that can process and respond to commands sent by the code. It achieves this by using its central processing unit to interpret the data received from its I/O peripherals. The microcontroller needs to have input/ output pins, analog input (hardware serial port), USB connections and a reset button. We will replace the two serial communication ports with more modern communication ports such as USB. These RS232 ports are used to program the current microcontroller from the PC and possibly the host PC and user interface to send system start and stop commands. Smaller mechanical parts might also need to be replaced as they will be worn by time. Such as the wear of cart tracking gears might result in cart housing to skip on the track rail. In phase 2, We have decided to customize a shield for the microcontroller. This microcontroller shield will reduce the voltage of the power supply from 17V to 9V. We also decided to use a remote control for safety, to protect the user from being hit by the pendulum swing.

5. System Architectural Design



5.1 System Components

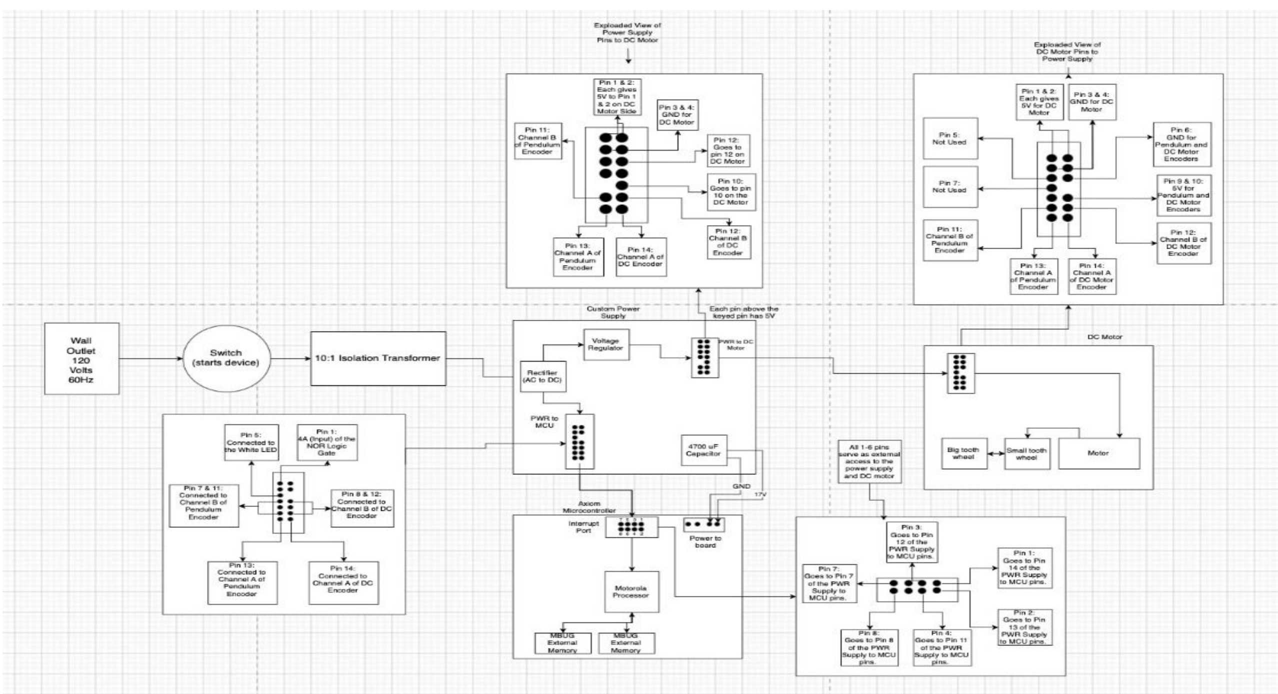
- Power supply: used to convert electric current from a source to the correct voltage, current, and frequency to power the load.
- Switch: used to turn on or turn off the flow of the electrical current in the program.
- DC motor: used to convert electrical energy into mechanical energy.
- Encoders: used to detect the motor's rotation.
- Microcontroller: receiving signals and controlling a singular function in a device.
- Plug: Allow electrical equipment to connect to the electrical grid.
- Shield for microcontroller: to reduce the voltage of power supply
- Remote control: for safety use

5.2 Concept of Execution

As the diagram shows, electricity will flow to the transformer as soon as the outlet is plugged into the wall. Then the switch controls to turn on or turn off the flow of the electrical current in the program. If the switch turns off, then the whole program is not doing anything. However, if the switch turns on, the power supply will power up both the cart motor and the logic board. The cart motor will receive power right after the switch turns on the power from the power supply and the encoder of the DC motor will start to give measurement. The DC motor will move left and right as it receives order, the inverted pendulum will start to swing as well. The encoder of the DC motor will keep measuring and it will send data to the microcontroller, also the encoder of the swing will send data of the swing position to the microcontroller and the microcontroller will give command to the DC motor inverted pendulum until Inverted Pendulum is in a balanced position.

5.3 Interface Design

Current system diagram showing all pins that are used and describe each pin's role in the whole system.



Low Level Design

1. Introduction

In this project, our goal is to get the broken Inverted Pendulum fully repaired and operate again. We will decide which part needs to be repaired or replaced and what software programming is needed for system control to work by judging if all the parts of the Pendulum operate normally. This project will require students to repair a digital control for an inverted pendulum system which can keep the Inverted pendulum in a balanced position. The current brian that controls the inverted pendulum in our project is the microcontroller, specifically the Motorola MMC2001 Mcore Microcontroller. After full system analysis, the system power supply is operational and appears the microcontroller requires replacement. Also, USB communication connections and a program are needed to bring this system back into operation.

In our phase 2 project, we are using an Arduino 2560 microcontroller for this project due to the availability of the I/O pins in Arduino 2560 microcontroller. This microroncoller has more than 6 I/O pins so if one pin burns out, we can use another pin instead of replacing the whole microcontroller. The original voltage from the power supply was 17V, but the highest voltage the microcontroller can take is 12V. So, we have designed a shield for the microcontroller which is going to have a voltage regulator and the voltage regulator will reduce the voltage to 9V. We also have built a safety remote control to prevent the user from being hit by the Pendulum swing. The remote control has three push buttons which are push buttons to control start/stop, push button to move the cart motor left direction, and push button to move the cart motor right direction. The remote control is connected to the project directly on the microcontroller shield through a 20 inches long ribbon cable. Also, the motor control is using four pins, including three digital pins and one ground pin.

2. Scope

The scope of this project is to completely restore the Inverted Pendulum provided to us. The goal is to restore the Inverted Pendulum to working condition. Major parts that will need to be replaced are the microcontroller and the two serial communication ports and cables (DB9 and DB25). The two serial communication ports will be replaced with more modern communication ports such as USB.

In the phase 2 project, we have decided to customize a shield for the microcontroller to reduce the voltage of the power supply. We decided to add a remote control to ensure safety. Remote control has three push buttons, one is to control the cart motor move left, one is to control the cart motor move right, and one is to control the start/stop of the project.

3. Referenced documents

Linear flexible inverted Pendulum	https://www.quanser.com/products/linear-flexible-inverted-pendulum/#overview	The source gave us a brief overview of what a Linear flexible inverted pendulum is and what are the specifications in the inverted pendulum and also some other related information.
Linear servo control lab	https://www.scribd.com/document/462128842/Quanser-Linear-Servo-Control-Systems-Brochure-online-1	This pdf document includes really detailed information about the inverted pendulum and equipment required for the inverted pendulum to work.
AT22LV10- Integrated Circuit	http://ww1.microchip.com/downloads/en/DeviceDoc/doc0190.pdf	This document introduced some basic characteristics of AT22LV10(L).
MMC2001 Reference Manual	https://www.nxp.com/docs/en/reference-manual/MMC2001DDRM.pdf	It is a user manual for MMC2001 and some basic introductions about MMC2001
CMD- 2001 Rev. C User Manual	http://www.ece.ualberta.ca/~cmpe490/documents/axiom/Axiom_	This document is a user manual for CMD-2001. It includes some features of CMD2001.

	CMD2001_Manual.pdf	
MMC- 2001 Processor	https://www.nxp.com/docs/en/fact-sheet/MMC2001PB.pdf	This document specifically introduced about the processor of MMC2001.

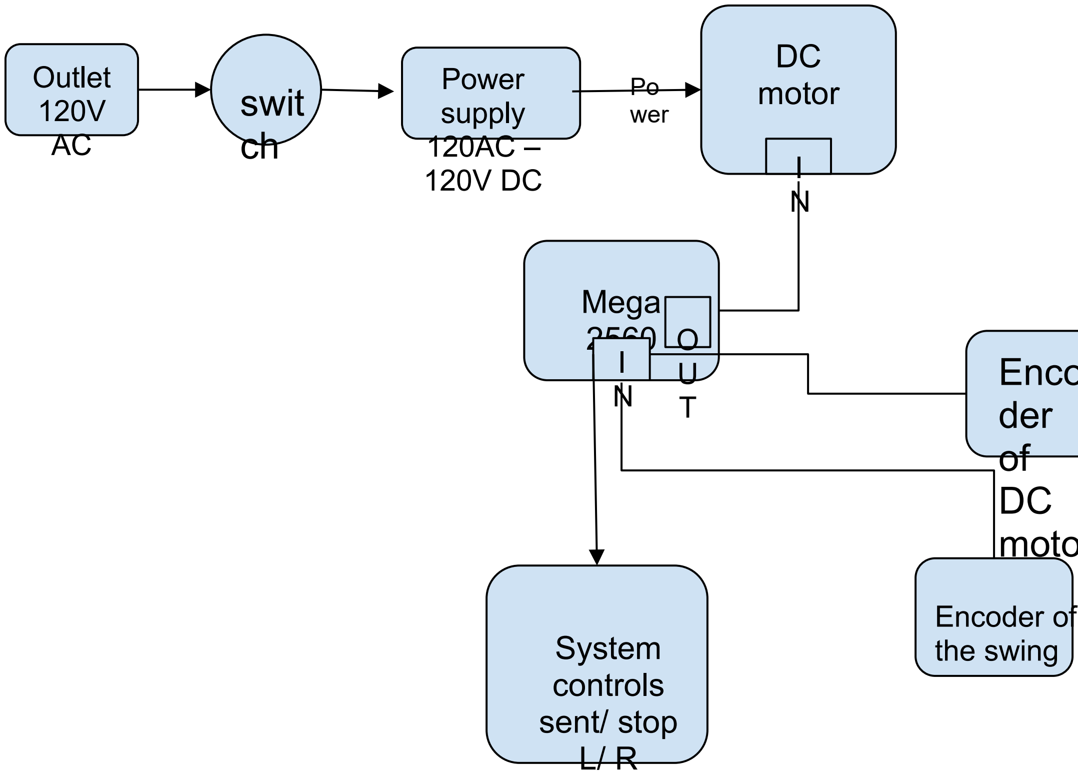
4. **System- Wide Design Decisions**

As we mentioned the scope of this project is to fully restore an Inverted Pendulum that was provided to us. The objective is to repair the Inverted Pendulum to working order. In order to meet our goal which is to get the broken Inverted Pendulum fully repaired and operate again, we will decide which components need to be replaced or need to be repaired and what software programming is needed for system control to work. After examining all the components, we will implement a system solution which includes the working power supply that will generate system wide power and power for the microcontroller. The core of our current project is the microcontroller interface and software. The microcontroller will need to be able to interface with the power supply and the cart motor by cables in order to send/ receive the instruction to the system. The microcontroller needs to be capable of having a USB port in order to download the software control logic program. The chosen microcontroller will also be able to receive and send data to process encoder data and control the DC motor.

Major parts that we will keep will be the power supply, DC motor, rod, cart mounting platform and base, track rails, power cable. ON/OFF switch and encoders. The parts that will need to be replaced are the microcontroller and the two serial communication ports, cables (DB9 and DB25) and software.

In our phase 2 project, we decided to make some changes, so the parts we will keep will be the power supply, rod, cart mounting platform and base, track rails, power cable, ON/OFF switch, one encoder for the Pendulum swing, one encoder for DC motor. The parts that will need to be replaced are the microcontroller and the two serial communication ports, cables (DB9 and DB25) with a modern USB port and software. We also will add a remote control and a shield for the microcontroller.

5. **System Architectural Design**



Hardware Block Diagram

5.1 System Components

- Power supply: used to convert electric current from a source to the correct voltage, current, and frequency to power the load.
- Switch: used to turn on or turn off the flow of the electrical current in the program.
- DC motor: used to convert electrical energy into mechanical energy.
- Encoders: used to detect the motor's rotation.
- Microcontroller: receiving signals and controlling a singular function in a device.
- Plug: Allow electrical equipment to connect to the electrical grid.

Phase 2:

- Power supply: used to convert electric current from a source to the correct voltage, current, and frequency to power the load.
- Stepper motor: used to convert electrical energy into mechanical energy.
- Stepper driver: receive the command from the microcontroller and then send the order to the stepper motor in order to make it left and right.

- Encoder for the Pendulum swing: used to detect the swing’s position.
- System controls: remote control has a start/ stop button and left/ right push button to control the cart motor’s directions.
- Shield for microcontroller: to reduce the voltage of power supply

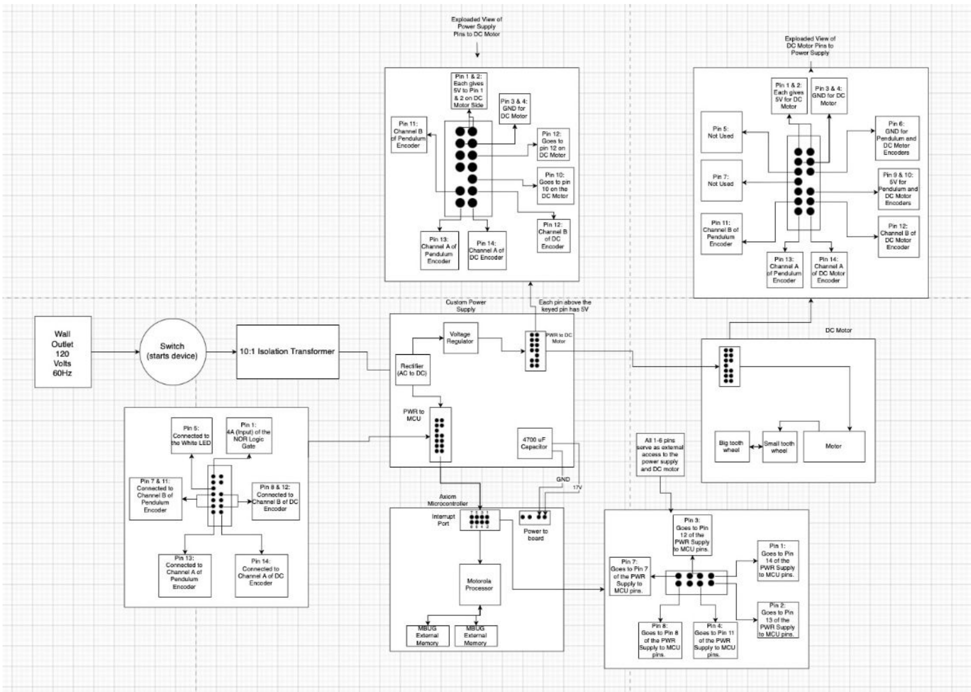
5.2 Concept of Execution

We first use a power supply to convert electric current from a source to the correct voltage, current, and frequency to power the load and make the whole program able to run. Then we have the switch which controls the flow of the electrical current on and off in the program. The cart motor and logic board will be powered up if there is electrical current in the whole program. Encoders used to detect the motor’s rotation. The cart motor includes two encoders which are the encoder of the DC motor and the encoder of the swing. Encoder of the DC motor used to take measurements and send signals to the DC motor in order to make it move left and right. The encoder of the swing will send the data of the swing position to the microcontroller. The microcontroller used to receive signals and control the functions of the inverted pendulum until it is balanced.

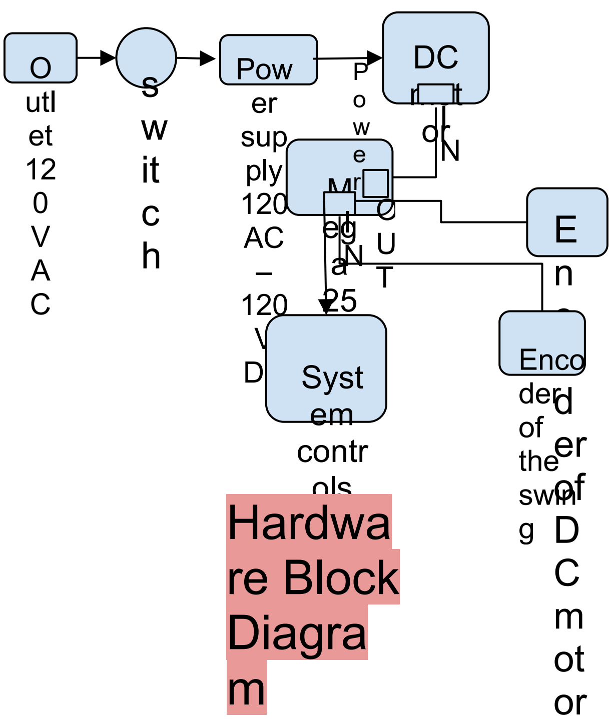
In phase 2, we first use a power supply to convert electric current from a source to the correct voltage, current, and frequency to power the load and make the whole program able to run. Then we have the switch which controls the flow of the electrical current on and off in the program. The cart motor and logic board will be powered up if there is electrical current in the whole program. The microcontroller will send the order to the DC motor in order to make it move left or right. The encoder of the Pendulum swing will be used to detect the swings’ position. The encoder of the swing will send the data of the swing position to the microcontroller. The microcontroller used to receive signals and control the functions of the inverted pendulum until it is balanced.

6. Detailed Hardware Design

6.1 Schematic and operation



Phase 2:



6.2 Bill of materials

ITEM NO.	NAME/ PART NO.	OTY.	DESCRIPTION	Link	Cost
1	Plastic tracking gears for motor	1	Track rail for the cart motor. (wheels)	https://www.mouser.com/ProductDetail/OSEPP-Electronics/LS-00022?qs=rrS6PyfT74ddIyCjVaoW0w%3D%3D	$ 7.65
2	Microcontroller (Arduino mega 2560)	2	Control the Inverted pendulum	https://www.mouser.com/ProductDetail/Arduino/A000067?qs=gMoqXxk%2F%2F5apEOlL63oO8Q%3D%3D	$ 40.30
3	Ribbon cables	1	Communication cables	https://www.digikey.com/en/products/detail/assmann-wsw-components/AWG28-14%2FG%2F300/686330?utm_adgroup=Flat%20Ribbon%20Cables&utm_source=google&utm_medium=cpc&utm_campaign=Shopping_Product_Cables%2C%20Wires_NEW&utm_term=&utm_content=Flat%20Ribbon%20Cables&gclid=Cj0KCQjw0PWRBhDKARIsAPKHFGhnCKi8R2zf9oeGm1bZ7XhnfIxOX4ODg1To7z9yhphSQq-L02-QoFkaAr6WEALw_wcB	$ 3.82
4	14 pin connector (female)	3	Connector	https://www.digikey.com/en/products/detail/molex/0430251400/531405?utm_adgroup=Connectors%2C%20Interconnects&utm_source=google&utm_medium=cpc&utm_campaign=Shopping_Suppli	$ 0.83

				er_Molex_0900_Co-op&utm_term=&utm_content=Connectors%2C%20Interconnects&gclid=Cj0KCQjw0PWRBhDKARIsAPKHFGikGgj4SQC_flp-B4lwI6j-EaK_UBbHiNiDt5-e85SyUMgaoktilgEaArjmEALw_wcB	
5	Linear Voltage Regulator (L7809CV)	2	Decrease voltage	https://www.digikey.com/en/products/detail/stmicroelectronics/L7809CV/585969?utm_adgroup=PMIC%20-%20Voltage%20Regulators%20-%20Linear&utm_source=google&utm_medium=cpc&utm_campaign=Shopping_Product_Integrated%20Circuits%20%28ICs%29_NEW&utm_term=&utm_content=PMIC%20-%20Voltage%20Regulators%20-%20Linear&gclid=Cj0KCQjw0PWRBhDKARIsAPKHFGg-qobQlIn05QwRVCrS9rq0ik0AJH3agjJWRn4xQrOmkrYxN0YBxvgaAgDBEALw_wcB	$ 0.63
6	Heat Sink	1	Reduce heating	https://www.mouser.com/ProductDetail/Aavid/577102B00000G?qs=cR7Qvk1oinlekh9MFs6NUA%3D%3D&mgh=1&gclid=Cj0KCQjw0PWRBhDKARIsAPKHFGh0cBW4S5tmFm5i6oXFfnEmWYCQ5Rx7ySCmUypxJmBeo5NepABEPZsaAtcCEALw_wcB	$ 0.75
7	Capacitor (470 uF 25 V)	2	Clear noise	https://www.digikey.com/en/products/detail/kemet/ESH477M025AG4AA/7423185?utm_adgroup=Capacitors&utm_source=googl	$ 0.37

				e&utm_medium=cpc&utm_campaign=Shopping_Supplier_KEMET&utm_term=&utm_content=Capacitors&gclid=Cj0KCQjw0PWRBhDKARIsAPKHFGiWWviDheT3bnf9pH23OXJ9_xhOjJHaGISIFeLilFMW4x4ZIVQjnDEaAsINEALw_wcB	
8	Capacitor 100 uF	2	Clear noise	https://www.digikey.com/en/products/detail/kemet/ESK107M025AC3AA/2712507?utm_adgroup=Capacitors&utm_source=google&utm_medium=cpc&utm_campaign=Shopping_Supplier_KEMET&utm_term=&utm_content=Capacitors&gclid=Cj0KCQjw0PWRBhDKARIsAPKHFGhaqn3t-s3kvXB_YuAXLRvSmrbSI-3XO_04I00LLcMd_Uc8LgsvQkgaAlenEALw_wcB	$ 0.23
9	Push Button Switch	6	remote control	https://www.mouser.com/ProductDetail/Grayhill/95C06C3RAT?qs=ls7QRyWmRk4CrjuiuVECkg%3D%3D&mgh=1&gclid=Cj0KCQjw0PWRBhDKARIsAPKHFGhiFIyVTwEGSVkdtWAs0v3fvzw9ZqNE_vMKILGc3DZs0FcsBBWhTEwaAj4yEALw_wcB	$ 0.47
10	90-degree header pins	4	Remote control and MCU shield	https://www.digikey.com/en/products/detail/amphenol-icc-(fci)/68016-204HLF/1878459?utm_adgroup=Connectors%20and%20Interconnects&utm_source=google&utm_medium=cpc&utm_campaign=Shopping_Supplier_Amphenol%20FCI_0609_Co-	$ 0.41

				op&utm_term=&utm_content=Connectors%20and%20Interconnects&gclid=CjwKCAjwrfCRBhAXEiwAnkmKmaxuvCNeGF_IIwzUDK-K03WJO1cPZDXmPHCYO1xdEggeu6DHF7dfJRoCKHAQAvD_BwE	
11	4-PIN CONNECTOR W/HEADER,	2	Wire connecter	https://www.mouser.com/ProductDetail/Adafruit/4722?qs=hd1VzrDQEGh7o1NiF1si3A%3D%3D&mgh=1&gclid=Cj0KCQjw0PWRBhDKARIsAPKHFGgzTRTTKmXNriIbDMUQ9uSQxE30v2rjBIY6sYxxVKirTtjX8lMg-hQaAleBEALw_wcB	$ 0.95

7. Detailed Software Design

7.1 Main Program

STEP 1: Plug the outlet into the wall, and electricity will flow to the transformer.
STEP 2: Use the switch to control the flow of the electrical current in the program. (When the switch turns off, then the whole program is not doing anything. However, when the switch turns on, the whole program will be powered up.)
STEP 3: Once when the switch turns on, the power supply will power up both the cart motor and the logic board.
STEP 4: The encoder of the DC motor will start to give measurement.
STEP 5: The DC motor will move left and right as it receives order, and the inverted pendulum will start to swing as well.
STEP 6: The encoder of the DC motor will keep measuring and it will send data to the microcontroller, also the encoder of the swing will send data of the swing position to the microcontroller.
STEP 7: Microcontroller will give command to the DC motor until the Inverted Pendulum is in a balanced position. It will still need to send/receive data commands to “hold” the pendulum in an upright position.

As the diagram shows, electricity will flow to the transformer as soon as the outlet is plugged into the wall. Then the switch controls to turn on or turn off the flow of the electrical current in the program. If the switch turns off, then the whole program is not doing anything. However, if the switch turns on, the power supply will power up both the cart motor and the logic board. The cart motor will receive power right after the switch turns on the power from the power supply and the encoder of the DC motor will start to give measurement. The DC motor will move left and right as it receives order, the inverted pendulum will start to swing as well. The encoder of the DC motor will keep measuring and it will send data to the microcontroller, also the encoder of the swing will send data of the swing position to the microcontroller and the microcontroller will give command to the DC motor inverted pendulum until Inverted Pendulum is in a balanced position.

Phase 2:
Functions:

GetPendulumPosition: This will obtain the pendulum encoder position using interrupts and convert that value to degrees from 0 to 360. This function will run within the other specified functions to update and use the pendulum positions.

MoveCarriage: This function will move the carriage left or right. It will handle giving the appropriate "steps" to the stepper driver. It will receive an input for direction and speed.

Swing: This function will swing the pendulum from the resting position.

HoldUpright: This function will balance the pendulum upright and will run once the Swing function gets the pendulum to a certain threshold.

Process:

Wait for the Start button to be pressed.

Once pressed, get the encoder positions, and run the swing function.

Once the pendulum reaches the specified vertical angle, the HoldUpright function will run.

The HoldUpright function will continue to run and correct for any disturbances.

If the system is disturbed outside the specified vertical angle, the swing function will run until the angle is met.

The HoldUpright function will then run again.

The program will stop when the Start/Stop button is pressed at any time.

Pseudo Code:

```
// Phase connections of the encoders
Encoder Pendulum(2, 3);  // (A, B)

void setup()  // here to setup the inputs and outputs
{
//  pinMode(StarStop, INPUT_PULLUP);
}

Void Loop() //This is the main function that runs continuously
{
        If Start/Stop is pressed then continue
        Run the Swing Function
        Check if the pendulum is vertical enough to run the HoldUpright function
        If not, continue to run the swing function
        If it is, run the HoldUpright Function
        Continue to run the HoldUpright function unless it is disturbed past the vertical
        threshold.
        Then run the swing function
        If the stop is pressed anytime, stop the functions and wait for another press.

}

GetPendulumPosition function
{
  newPend = Pendulum.read();  //Get the updated values of the encoder.
```

```
 if (newPend != positionPend) // If the values has changed
 {
   positionPend = newPend;
   if(positionPend >4095)   // If the Pendulum counts are over, reset the 0
     Pendulum.write(0); //reset to 0

   if(positionPend < -4095) // If the pendulum counts are under, Reset to 0
     Pendulum.write(0);  // reset to 0

   if(positionPend > 0) // if running in the positive direction...
      pendDegrees = map(positionPend, 0, 4095, 0, 359);  //Map to degrees from 0 to 359

   if(positionPend < 0) // if running in the negative direction…
      pendDegrees = map(positionPend, 0, -4095, 359, 0); /map the opposite way.

Then return the position in degrees
}

Swing Function()
{
Move the carriage to one side
When the pendulum reaches the max value of degrees, then move back the opposite
direction.
}

HoldUpright Function()
```

{

Calculate the absolute number of degrees away from 180 = P. This is the multiplier.

X = P * Constant. This will provide a proportional value for speed. The further the number is from 180, the faster it moves.

If the position is less than 180 degrees then MoveCarriage Function(go Right, at speed x)

If the position is more than 180 degrees then run the MoveCarriage Function(go Left, at speed x)

}

7.2 Variables

- PWR to MCU
- Axiom Microcontroller Interrupt Port
- PWR to DC Motor
- DC Motor
- Switch
- Swing encoder
- DC motor encoder
- Red LED light for power supply getting powered
- Yellow LED for cart motor getting 17V
- Directional LED light for DC motor

Phase 2:

- Outlet 120V AC
- Power supply
- Switch
- Mega 2560
- DC motor
- Encoder of swing
- Encoder of DC motor
- Remote control
- Shield of microcontroller

7.3 I/O Assignments

PWR to MCU

- Pin 1: 4A (Input) of the NOR Logic Gate
- Pin 5: Connected to the White LED
- Pin 7 & 11: Connected the Channel B of Pendulum Encoder
- Pin 13: Connect to Channel A of Pendulum Encoder
- Pin 14: Connected to Channel A of DC Encoder
- Pin 8 & 12: Connected to Channel B of DC Encoder
- Pin 2, 3, 4, 6, 9, 10, 12: not used

Axiom Microcontroller Interrupt Port

- Pin 1 Goes to Pin 14 of the PWR Supply to MCU pins
- Pin 2: Goes to Pin 13 of the PWR Supply to MCU pins
- Pin 3: Goes to Pin 12 of the PWR Supply to MCU pins
- Pin 4: Goes to Pin 11 of the PWR Supply to MCU pins
- Pin 7: Goes to Pin 7 of the PWR Supply to MCU pins
- Pin 8: Goes to Pin 8 of the PWR Supply to MCU pins
- Pin 5 & 6: not used

PWR to DC Motor

- Pin 1 & 2: Each gives 5V to Pin 1 & 2 on DC Motor Slide
- Pin 3 & 4: GND for DC motor
- Pin 10: Goes to pin 10 on the DC Motor
- Pin 11: Channel B of Pendulum Encoder
- Pin 12: Goes to pin 12 on DC motor
- Pin 12: Channel B of DC Encoder
- Pin 13: Channel A of Pendulum Encoder
- Pin 14: Channel A of DC Encoder
- Pin 5, 6, 7, 8 & 9: not used

DC Motor

- Pin 1 & 2: Each gives 5V for DC Motor
- Pin 3 & 4: GND for DC Motor
- Pin 5: not used
- Pin 6: GND for Pendulum and DC Motor Encoders
- Pin 7: not used
- Pin 9 & 10: 5V for Pendulum and DC Motor Encoders
- Pin 11: Channel B of Pendulum Encoder
- Pin 12: Channel B of DC Motor Encoder
- Pin 13: Channel A of Pendulum Encoder
- Pin 14: Channel A of DC Motor Encoder

In phase 2:

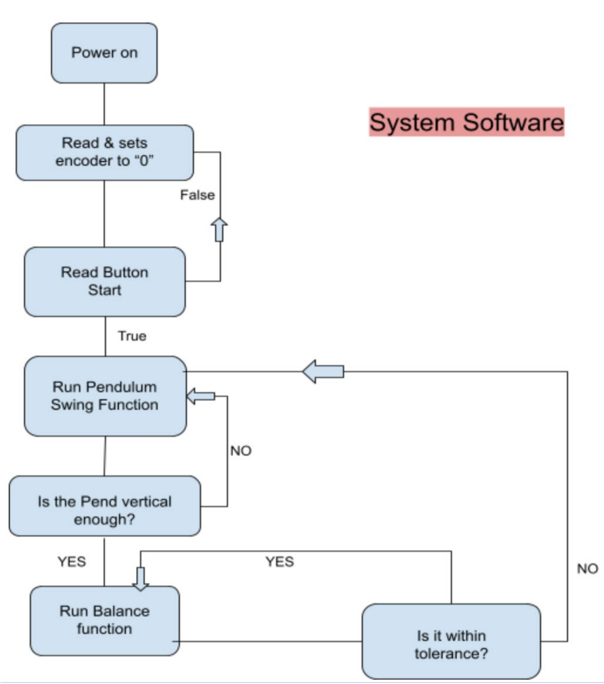
MCU to remote controls:
Pin 26: goes to remote control left direction
Pin 24: goes to remote control right direction
Pin 22: goes remote control to start/ stop
Pin 20: goes to GND

MCU to swing Encoder:
Pin 19: goes to channel A
Pin 18: goes to to channel B

MCU to DC encoder:
Pin 21: goes to channel A
Pin20: goes to channel B

MCU to DC motor:
Pin 8: goes Dir-
Pin 9: goes to Dir +

Software flowchart:
Phase 2:



Test Methodology & Test Results

Overview

The inverted Pendulum system is an example system in physics, dynamics, and control field. It is widely used as a benchmark for testing control algorithms. The center of mass of the Inverted Pendulum is about its pivot point and the Inverted Pendulum can be hung stably in its inverted position by using the control system to monitor the angle of the rod and moving the pivot point horizontally back below the center of gravity when the rod starts to fall over to maintain its balance.

The ENT department uses an inverted pendulum for teaching purposes. However, the Inverted pendulum is not working sufficiently and needs to be repaired. In our project, the goal is to get the broken Inverted Pendulum fully repaired and operate again

Identification

This document provides the detailed procedures which need to be excused by the lab in order to inspect whether all the components within the Inverted Pendulum are working normally and which one needs to be replaced or repaired to make the broken Inverted Pendulum operate again. After testing procedure, the modifications have been made in the system after finding some parts that are not working are the microcontroller, power supply and the encoder of the swing. We also added a shield for the microcontroller in order to power up the microcontroller from the shield instead of other resources, and a remote control in consideration of the user’s safety. The microcontroller needed to be completely replaced because it is unable to function at all, instead, we use Arduino mega. We test the encoder of the swing by the oscilloscope, in the oscilloscope we are able to see channel A and channel B waveforms and observe if they are giving signals. We also see how the encoder counts by running the code and see how it operates. We connect all the wires and power it up and run the code to see if the order has been given to the hardware and make it move in order to test the microcontroller’s function. For the power supply, there is small rust on the power supply and the circuit was shorted. We cleaned rust and the power supply returned functioning well. We also reheated the circuit and resoldering all components.

Testing process

This document will include all the detailed processes needed to be executed in the lab in order to meet the goal of this project which is to get the Inverted Pendulum fully operational again. Follow the detailed steps below to test all the components within an Inverted Pendulum to identify whether each component is working normally or not. The end result is to be able to find out all the damaged components that cause the Inverted Pendulum not working normally by testing.

We have tested the encoder of the swing by the oscilloscope in order to observe the channel A and channel B, we also run it through the code, to see if it is operating correctly. It appears that the encoder of the swing is operating fine, and it counts from 0 degree to 360 degree.

We replace the old microcontroller with the new microcontroller, Arduino Mega and run the system to see if the whole system is operating correctly. It appears that the whole system is working well and follows the order which was sent from the microcontroller.

The power supply is functioning well after we cleaned the rust and reheated the components because the circuit was shorted.

We added a shield for the microcontroller in order to power up the microcontroller. The power supply is giving 17V out, however, the new microcontroller can only take up to 12V so the shield includes a voltage regulator which can bring the voltage down to 9- 12V.

Remote controller was added for the safety of the user, it includes three push buttons which are: start/stop button, left direction and right direction button. We connected it to the microcontroller and observed how it behaves, it was working fine.

This was done in Phase I.
Again, you will need test procedures for how your modification/restoration (inclusion of updated parts) are working. As well as how the software is working.

Test Plan- hardware

Test 1- Power supply

Testing power supply and power flow in the whole system

Step	Actions	Expected Results	Observed Results
1	Plug the port of the power supply to the wall	The power supply powers up	Power supply getting power.
2 Switch	Turn on and off switch to power the power supply	Power supply react to switch on and off	When the switch turns on, the red LED lights up.
3 Cart motor	Power up the cart motor	Power goes out from power supply to cart motor	Yellow LED will turn on when the cart motor powers up.
4 Logic board	Power up the logic board	Power goes out from the power supply through the shield to the logic board.	Voltmeter result shows 9-12V goes to the logic board.

Test state:

- Result: 4 Passed, 0 Failed
- Tested by: Fuad Aljadani
- Comments: we tested the power supply and all the components that power flows to, and it turns out that no component is having trouble receiving power. It proves

that the power supply is working fine.

Test 2- Cart motor

Testing cart motor and components within the cart motor

4

Step	Actions	Expected Results	Observed Results
1 DC motor	Press two push buttons manually on the power supply which controls the DC motor going left or right.	LEDs will light up different colors based on the different directions the DC motor goes.	Different colors light up when the DC motor goes different directions.
2 Encoder for the DC motor	Use voltmeter to detect voltages Use oscilloscope to observe channel A & B Run through the code and see if the encoder is operating correctly.	• The voltmeter will detect voltage. • The waveform will be observed • The system is counting correctly.	• Voltmeter detected 5V and the DC motor position was stated. • Oscilloscopes detect the waveforms for channel A & B • The system is counting correctly.
3 Encoder for the swing	Use voltmeter to detect voltages Use oscilloscope to observe channel A & B	• The voltmeter will detect voltage. • The	• Voltmeter detected 5V and the swing position

	Run through the code and see if the encoder is operating correctly.	waveform will be observed • The system is counting correctly.	being stated. • Oscilloscopes detect the waveforms for channel A & B • The system is counting correctly.

Test state:

- Result: 3 Passed, 0 Failed
- Tested by: Fuad Aljadani
- Comments: We tested the cart motor and the different components within the cart motor, The result shows that all of them are working fine.

Test 3- Logic board

Testing the logic board

Step	Actions	Expected Results	Observed Results
1 Microcontroller	Run the whole system.	The system receives the signal, and the Inverted pendulum will swing.	The inverted pendulum will stand up at 180 degrees.

2 Shield of microcontroller	Reduce the voltage	The voltage in this system will be reduced and the program will run when the power is up.	The program was running well, and the voltage has been reduced, and the microcontroller has been powered up.
3 Remote control	To control the project in distance	When push button on the remote control been pressed, the program reacts to different push buttons	The cart motor was able to move left or right based on the push buttons and the program can start and stop by the remote control.

Test state:

- Result: 3 Passed, 0 Failed
- Tested by: Fuad Aljadani.
- Comments: we tested the microcontroller, and the result met the expectation. The Microcontroller was able to play its role in the system. The microcontroller was able to send the signal and make the inverted pendulum swing, and the voltage in the program has been reduced, the microcontroller has been powered up and the whole program was working well. The remote control can control the program by pressing different push buttons which can prevent the users from being hit by the pendulum swing to ensure safety.

Test 4- Inverted Pendulum parts

Testing Inverted Pendulum parts

Step	Actions	Expected Results	Observed Results
1 Track and trail	Star the DC motor and observe the movement of the DC motor wheel on the track rail.	The Wheel moves smoothly on the track rail.	The wheel was able to move smoothly within the safety limits of the track rail.
2 Cart motor	run the cart motor on the track rail and observe its movements.	The cart motor can run stably on the track rail	The cart motor can run stably on the track rail
3 Swing pendulum	Run the inverted Pendulum and observe the swing.	The swing will swing until it gets to a balanced position.	The swing goes left and right without hindrance.

Test state:

- Result: 3 Passed, 0 Failed
- Tested by: Fuad Aljadani
- Comments: We tested the track rail, cart motor, and the swing pendulum by running the system and seeing their movements. The result shows that the track rail, cart motor and the swing pendulum are all working efficiently without problem.

Conclusions, Recommendations, and Individual Assessment

Objective

In this project, we will be repairing a broken Inverted Pendulum for the ENT department in order to use for teaching purposes. In order to meet our goal which is to get the broken Inverted Pendulum fully repaired and operate again, we need to judge whether all the parts in the Inverted Pendulum are operating normally. Then we will decide which components need to be replaced or need to be repaired and what software programming is needed for system control to work.

Introduction

The inverted Pendulum system is an example system in physics, dynamics, and control field. The center of mass of the Inverted Pendulum is about its pivot point and the Inverted Pendulum can be hung stably in its inverted position by using the control system to monitor the angle of the rod and moving the pivot point horizontally back below the center of gravity when the rod starts to fall over to maintain its balance. So, the purpose of this project is to repair a digital control for an inverted pendulum system which can keep the Inverted pendulum in a balanced position.

Procedure

After our assessment on all the components, we will implement a system solution which includes a power supply that will generate power for the microcontroller and the cart motor. Also, this program will have a microcontroller that will meet the requirement for both hardware and software as this program will have a microcontroller that is able to communicate with the power supply and the cart motor by cables in order to give the instruction to the system. We outlined our test plans and conducted them to identify our project's performance against our project's goals which is to get the broken Inverted Pendulum fully repaired and operate again with a new software system. We have also figured out that the brain that controls the inverted pendulum is the microcontroller. We need to replace the old Motorola MMC2001 Mcore Microcontroller to a new microcontroller. With the help of the professor, we found out that the Inverted Pendulum could work manually by doing an experiment of disconnecting the cable from the logic board. After all the above observations in the system, we proved that this system could work manually and only needs a new microcontroller, USB communication connections and a program to bring this system back into operation. The hardware is working fine, except the weighted cart needs more weights to allow it to travel along the rails more smoothly and the motor gear that contacts the rail needs to be replaced.

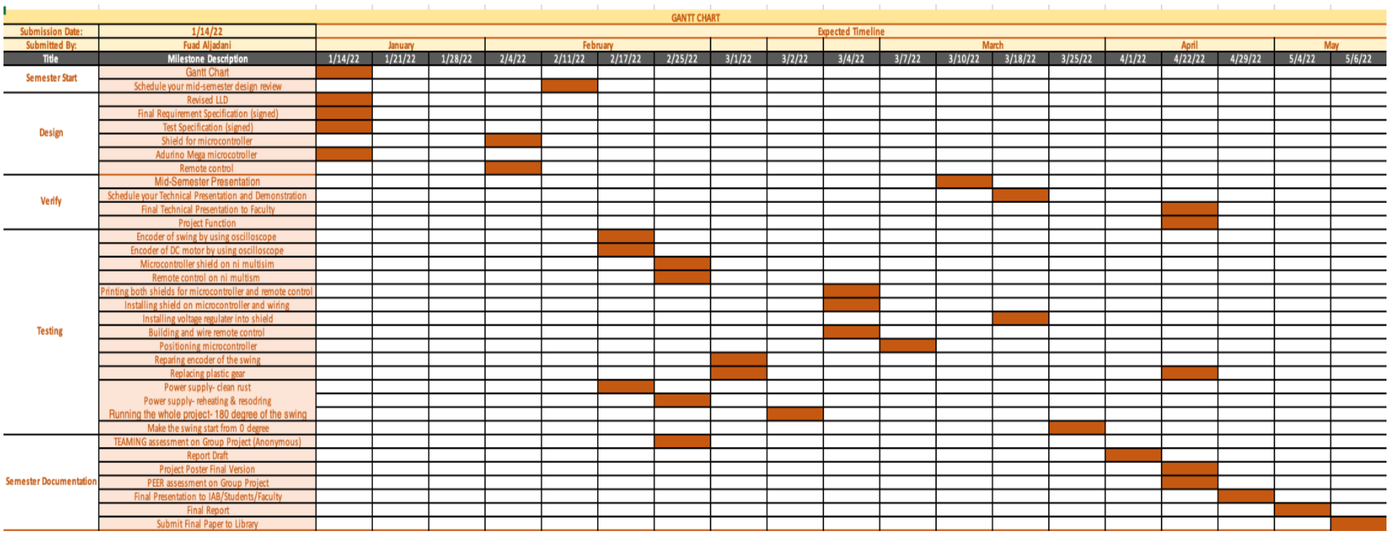
Conclusion

By doing this Inverted pendulum project, I have learned a lot of new information and improved many of my skills. I set plans and troubleshooted all the components in the Inverted

Pendulum in order to find the main problem that causes the Inverted Pendulum to not work. I have determined all the parts in the Inverted Pendulum. I figured out all the unbroken parts in the Inverted Pendulum that can be kept. I have also figured out which part is broken and needs to be repaired or replaced. Not only do I have a better understanding about this project, I think my ability to think logically when it comes to fixing and repairing things also improved as I have experience on where to start when a project is signed.

Appendixes

Ghatta chart



Code:

```
/*
    Inverted Pendulum Senior Project Spring 2022

    Programmer: Charles "Jake" Masheck
    Consulting Programmer: Nikolas Mahaffey

    Date: 04/21/2022
    IUPUI ENT ECET Department - Location of Pendulum

    Goal: The problem/purpose is to start from the resting position.  The pendulum
    then will swing up to the vertical position.  The pendulum will then hold its
    vertical position and compensate for disturbances.

    Arduino Mega
    Encoders CPR = 4096
```

```
    Carriage Far LEFT to Far RIGHT 0 -> 34360  Half = 17180
    Pendulum  CCW   0 -> 4096    Half = 2048

    Mega Interupt Pins 2, 3, 18, 19, 20, 21
    Avoid using pins with LEDs attached
*/


#include <Encoder.h> //Found in the Arduino Library Installer Version 1.4.2 by Paul Stoffregen
#include <PID_v1.h> //Found in the Arduino Lebrary Installer Version  1.2.0 by Brett Beauregard


const int pendulumCPR = 4095;  //Must subtract by 1 from encoder CPR
const int trackMin = 14000; //15000
const int trackMax = 17000; //20000

const int jogSpeed = 75;   //Set the speed that the cariage moves when the jog buttons are
pressed  (0-255)
const int swaySpeed = 175; //Set the speed that the cariage moves to invert the pendulum (0-
255)

// Buttons
const int StartButton = 40;
const int RightButton = 36;
const int LeftButton = 38;

// Motor control pins
const int motorLeft = 9;
const int motorRight = 8;

// Phase connections of the encoders
Encoder Pendulum(20, 21);  // (A, B) 20 21
Encoder Carriage(18, 19); // (A, B) 18 19

//Running Variable
bool shouldRun = false;
int LeftRightToggle = 0;

// PID Configuration
double setpoint = 179.28; //This is the vertical position of the pendulum.

//Read the project documentation on circuitdigest.com to learn how to set these values
double Kp = 800; //Proportional   TUNE 1st
```

```
double Ki = 1.50; //Integral      Tune 3rd
double Kd = .5;//Derivative       Tune 2nd


double input, output;
PID pid(&input, &output, &setpoint, Kp, Ki, Kd, DIRECT);  //setup the PID control IOs



void setup() {

  //Buttons
  pinMode(StartButton, INPUT_PULLUP);
  pinMode(RightButton, INPUT_PULLUP);
  pinMode(LeftButton, INPUT_PULLUP);

  //Motor Controls
  pinMode(motorLeft, OUTPUT);
  pinMode(motorRight, OUTPUT);

  //PID Setup
  pid.SetMode(AUTOMATIC);
  pid.SetSampleTime(10);
  pid.SetOutputLimits(-255, 255);

}

long positionPend  = 100;  // set to 100 allows for the program to detect change/initial position
long positionCarr  = 100;

void loop() {

  ////////////////////////////////////////////////  Convert Pendulum Encoder Counts to
Degrees  ////////////////////////////////////////////////
  static float pendDegrees = 0;
  static long newPend, newCarr;

  newPend = Pendulum.read();
  newCarr = Carriage.read();

  if (newPend != positionPend || newCarr != positionCarr)
  {
    positionPend = newPend;
    positionCarr = newCarr;
```

```
    if (positionPend > pendulumCPR)
    {
      Pendulum.write(0);    // Reset if the pendulum goes all the way around in the positive
direction
    }

    if (positionPend < (-1*pendulumCPR))
    {
      Pendulum.write(0);  // Reset if the pendulum goes all the way around in the negative
direction
    }

    if (positionPend > 0)
    {
      pendDegrees = map(positionPend * 10, 0, (pendulumCPR *10), 0, 3590); // Count up and
map the values to degrees in the CC direction
    }

    if (positionPend < 0)
    {
      pendDegrees = map(positionPend * 10, 0, (pendulumCPR *-10), 3590, 0);  //Count down
and map the values to degrees in the CW direction
    }

    pendDegrees = pendDegrees / 10.0;  // Divide by 10 to get degrees From 0 to 360 with
decimal resolution
  }

  ////////////////////////////////////////////////////////////  Detect Fall  ///////////////////////////////////////////////////

  static float lastDegrees;   // The last position of the pendulum in degrees
  static int isFalling = 0;   //Falling status indicator
  static int swayStatus = 0;  //Indicates if the pendulum should be in "Swing Mode"

  isFalling = 0;


  //delay(10);

  if (LeftRightToggle == 0 && ((lastDegrees - pendDegrees) > 1)) //falling from the RIGHT
  {
    //delay(5);
    isFalling = 1;
```

```
}

if (LeftRightToggle == 1 && ((lastDegrees - pendDegrees) < -1)) //falling from the LEFT
{
  //delay(5);
  isFalling = 1;
}

lastDegrees = pendDegrees;  // Record the last position


///////////////////////////////////////////////////////////////////////////////////  Invert and
Balance  ////////////////////////////////////////////


if (digitalRead(StartButton) == LOW)
{
  shouldRun = !shouldRun; //toggle state

  while (digitalRead(StartButton) == LOW)
  {
    ; // Do nothing when the button is held
  }
  delay(200); //Debounce
}

if (shouldRun) {

  //////////  Go Right Swing   /////////////////////

  if (LeftRightToggle == 0 && swayStatus == 0)
  {
    if (positionCarr < trackMax && swayStatus == 0)
    {
      digitalWrite(motorLeft, LOW);
      analogWrite(motorRight, swaySpeed);
    }

    if (positionCarr > trackMax)
    {
      digitalWrite(motorLeft, HIGH);
      digitalWrite(motorRight, HIGH);  // Stop the carriage when it reaches the MAX position

      if (isFalling == 1)
```

```
    {
      LeftRightToggle = 1;   //When the pendulum starts to fall change direction trigger
    }
  }
}

///////////// Go to LEFT Swing  /////////////////

if (LeftRightToggle == 1 && swayStatus == 0)
{
  if (positionCarr > trackMin)
  {
    digitalWrite(motorRight, LOW);
    analogWrite(motorLeft, swaySpeed);
  }

  if (positionCarr < trackMin)
  {
    //stop the motor
    digitalWrite(motorRight, HIGH);
    digitalWrite(motorLeft, HIGH);
    if (isFalling == 1)
    {
      LeftRightToggle = 0;
    }
  }

}


////////////////////////////////////////   Balance  ///////////////////////////////

if (pendDegrees <= 185 && pendDegrees >= 175)  // If the pendulum is +-5 degrees from 180
{
  swayStatus = 1;

  input = pendDegrees; //set the input of the PID to the pendulum degrees

  pid.Compute(); //Compute the PID output control values


  if (output > 0) //If the PID output is positive
  {
    digitalWrite(motorLeft, LOW);
```

```
    analogWrite(motorRight, output);  // Go to the RIGHT
   }
   else if (output < 0)  //If the PID output is negative
   {
    digitalWrite(motorRight, LOW);
    analogWrite(motorLeft, output * -1);
   }
  }

  else
  {
   swayStatus = 0;  //Set the sway status to 0 if the pendulum falls outside the +-7 degrees
from 180
  }

 }  // end of button pressed IF



///////////////////////  JOG  ////////////////////////////////////////////


 else if (digitalRead(RightButton) == LOW) // Jog RIGHT
 {
  digitalWrite(motorLeft, LOW);
  analogWrite(motorRight, jogSpeed);  //Travel at the jog speed
 }
 else if (digitalRead(LeftButton) == LOW) // JOG LEFT
 {
  digitalWrite(motorRight, LOW);
  analogWrite(motorLeft, jogSpeed); //Travel at the jog speed
 }
 else
 {
  digitalWrite(motorLeft, LOW);  //If no button presses are detected, turn off the motors
  digitalWrite(motorRight, LOW);
 }

}
```

Presentations slides:

Midterm slides

Technical presentation: in process

Weekly progress reports

https://drive.google.com/drive/folders/1pwKLm40sQjXSwlvUXmnmXRUU3XdY0PWx?usp=sharing