The Angular Position Project

Amber Haley, Jenna Soto, and Benjamin Williamson

Dept. of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* — The aim of this project is to obtain angular position from a dynamic test object, then transmit the position wirelessly to a PLC for interpretation. Sponsor requirements restricted customized component usage so extensive research was done on standardized components and how to interface them to produce a system in its entirety. This system is robust enough to meet and exceed all requirements established and could be easily integrated into a [redacted] control system in the future.**

***Index Terms* — Angular position, Encoder, On-board, Way-side, Z-wave, Microcontroller.**

I. Introduction

The project is to capture angular position and transmit that information wirelessly to a PLC. The Angular Position Project (AP Project) uses an on-board encoder attached to the shaft of the test object to read the test object's linear angular position. Once the angular position is obtained, the encoder feeds that information to an on-board microcontroller for processing. The microcontroller will take in the analog data, run it through its algorithms to decide if the test object is operating in an "in range" mode or "out of range mode". Then the microcontroller gives the data to an on-board transmitter. The transceiver packages the data then sends it wirelessly to a wayside transceiver. From the transceiver, the data is sent directly to the way-side PLC.

In addition to the wireless signal and range, the PLC will have three other inputs. The stuck button will simulate if the test object is in a fixed position or stuck position. The polling button will simulate movement of the test object as if it was progressing around a [redacted] track. The [redacted] key button will simulate some [redacted] personnel physically inserting a key into a turn lock to reset the [redacted] stop request error. As each piece of data enters the PLC, the PLC program will decide if a wireless signal was received, if the test object is in or out of range, is stuck or not stuck and if a [redacted] stop request should be issued. Once the PLC makes the decision, it will illuminate the appropriate light. There will be a light for signal received, in range, out of range, stuck, not stuck and [redacted] stop request. Each polling point will reset the data in the PLC and therefore reset all the lights except for the [redacted] stop request. The only way to reset the [redacted] stop request light is if the abnormality has been fixed and the [redacted] key button has been pressed; then the ride stop request light will reset. The on-board components are powered by a rechargeable battery. The battery sends the power through a voltage monitor then to a voltage regulator to step down the power for component protection. The power through a battery monitor, which will monitor the battery level at all times. The battery tester sends the power to the on-board microcontroller. The microcontroller then provides the power to the encoder and the transceiver. The way-side microcontroller will be powered from a power supply that is plugged directly into the wall. The PLC will be powered directly from the wall. The input stuck button, the input polling button and the input [redacted] key button will be powered by a power supply. All output lights will be powered directly from the PLC output module.

II. REQUIREMENT SPECIFICATIONS

The initial requirements of the sponsored project will be to: accurately transmit and receive data, reliably transmit and receive data, be supported for PLC I/O, standardized not customized parts, light weight, potential use in dynamic situations/environments, vibration tolerant, fail safe, relatively small, water resistant, and function in Florida weather (ride operational conditions) such as sunshine, water resistant, heat, fog, windy, light, dark, etc.

III. HARDWARE BLOCK DIAGRAM

The general concept, as illustrated with **Error! Reference source not found.**, is that there will be on-board components that send wireless data to way side components. The on-board components and way-side components will work together to create an entire system of checks and balances. The on-board box is physically on the test object and the way side box can be located anywhere not on the test object such as the side of a track [redacted] ride[redacted]

Fig. 1. System Block Diagram

The ICD in **Error! Reference source not found.** begins to look inside the boxes listed in the block diagram in **Error! Reference source not found.**. The ICD shows the different sections of the on-board configurations and the way side configurations. In visual terms, you easily see how many different sections are being encompassed in the overall design. The ICD also lists the wire sizes of the wires between the different sections and shows the flow of data and power.

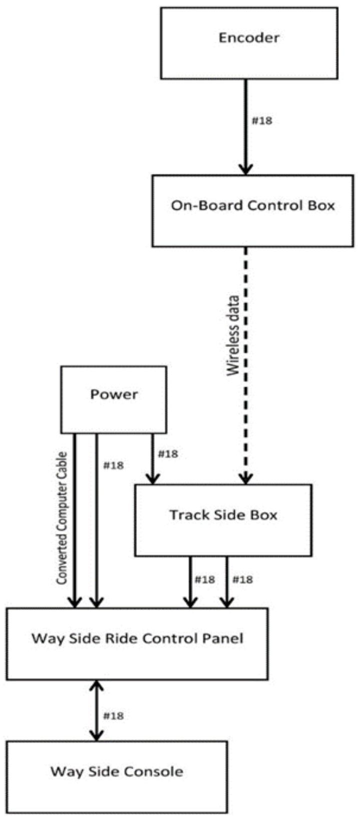

Fig. 2. Integrated Circuit Design

IV. ON-BOARD HOUSING ASSEMBLY DETAILS

The on-board components will be powered by a 22000mAh battery at 11.1V. The battery will be constantly tested by a battery tester to check on the voltage difference between the wires to show the battery life in terms of amp-hours or watt-hours. The connection between the battery and battery tester will be a XT60 connector. Then the power is distributed to a voltage regulator to step down the voltage to 5V as shown in Fig. 3. The power goes through a circuit breaker as an added layer of security for the other on-board components. The 5V sent through the circuit breaker will go to a micro-USB connection which will attach to the microcontroller which will give power to the Z-Wave module and the encoder.

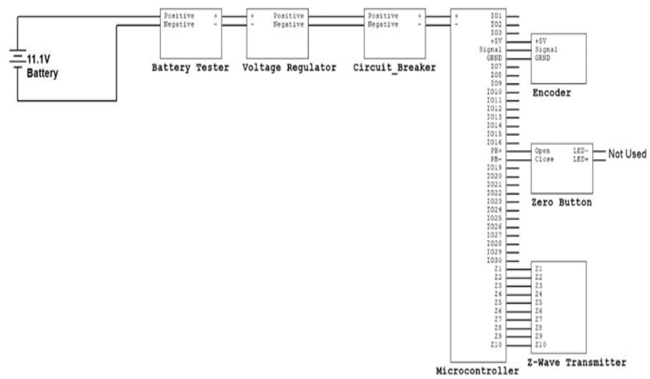

Fig. 3. On-Board Schematic

A shafted absolute encoder will attach to the test objects shaft by an encoder coupling. The shaft of the encoder will be ¼” and the coupling chosen will be for ¼” to ¼” mounting. The encoder will take read the angular position and send out the data in analog form to the on-board microcontroller. The microcontroller has a set of pins that are GPIO’s, General Purpose Input/Output, and take in and send out values as either high or low values. These high and low values are determined by the input voltage. One GPIO pin on the on-board MCU will be used for the data from the encoder. The MCU will then use an algorithm to convert the encoder data in to Boolean values for the PLC. Afterwards, the MCU will use a secondary algorithm to ensure that the converted data represents the original data correctly. Then these converted values will be transferred to the Z-Wave module on the on-board MCU and sent way-side wirelessly.

V. WAY-SIDE HOUSING ASSEMBLY DETAILS

The way-side MCU has an attached Z-Wave module that will recognize the wireless signal being transmitted by the on-board Z-Wave module and receives the signal after confirming the source and strength of the signal. The module will then reverse the encapsulation process that the transmitter used to access the data and transfer it to the way-side microcontroller. The MCU will then take the transferred data and send through an output pin to the PLC as shown in **Error! Reference source not found.**.

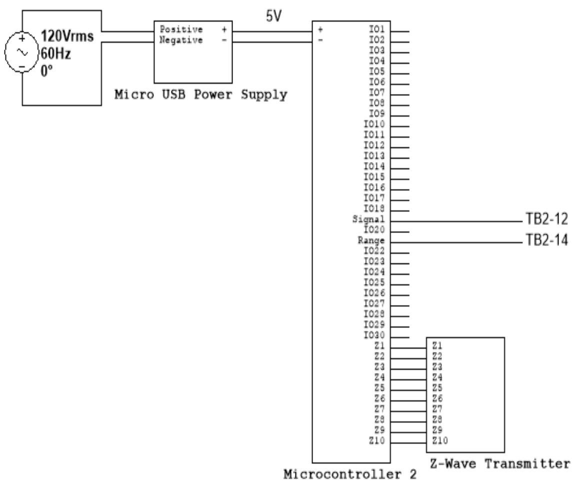

Fig. 4. Way-Side Box Schematic

The output truth table for the way-side receiving module and microcontroller is illustrated in **Error! Reference source not found.**. The X in the truth table means the MCU will bypass this data and will output low for the signal output.

TABLE I
Output Truth Table for Way-Side Receiver

Signal	Received Data	Signal Output Pin	Range Output Pin
Received	TRUE	HIGH	HIGH
Received	FALSE	HIGH	LOW
Not Received	TRUE	LOW	X
Not Received	FALSE	LOW	X

The PLC will have five inputs as shown in **Error! Reference source not found.**. The two digital signals from the way-side MCU, the stuck button, the polling button and the [redacted] key button. The digital signal will determine if the test object is in range or out of range and if a signal was sent or not. The stuck button will determine if the test object is stuck or not stuck. The polling button will simulate different polling points as if the test object was moving around a track. The [redacted] key button will simulate a key being inserted into a turn lock which will reset the [redacted] stop request signal if the abnormality in the operation has been corrected.

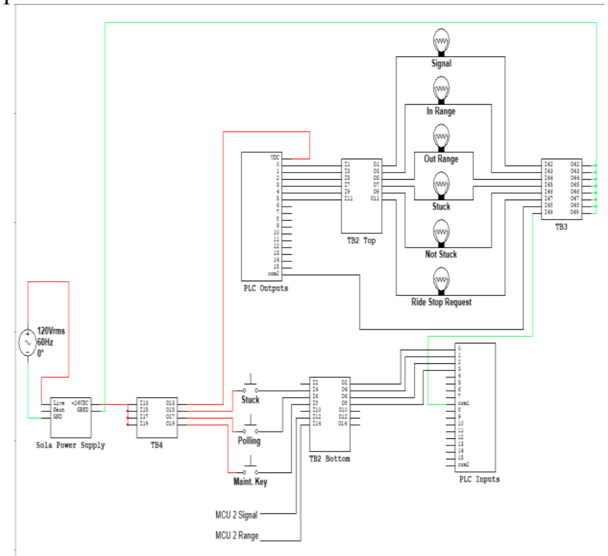

Fig. 5. Way-Side I/O Schematic

The PLC will interpret the data and follow the truth in **Error! Reference source not found.** Input and Table 3: PLC Truth Table Output for each case.

TABLE 2
PLC Truth Table Inputs

	Signal Received	In Range Signal	Out of Range Signal	Stuck Button	Not Stuck Button	Operation
1)	Yes	Yes	No	No	Yes	Normal
2)	No	X	X	X	X	Abnormal
3)	Yes	Yes	No	Yes	No	Abnormal
4)	Yes	No	Yes	No	Yes	Abnormal
5)	Yes	No	Yes	Yes	No	Abnormal

TABLE 3
PLC Truth Table Outputs

	In Range Light	Out Range Light	Stuck Light	Not Stuck Light	[redacted] Stop Request	Signal Light
1)	Yes	No	No	Yes	No	Yes
2)	No	No	No	No	Yes	Blink
3)	Yes	No	Yes	No	Yes	Yes

4)	No	Yes	No	Yes	Yes	Yes
5)	No	Yes	Yes	No	Yes	Yes

The PLC power supply will receive its power directly from the wall as illustrated in **Error! Reference source not found.**. The input buttons going into the PLC will be powered from a separate power supply as shown in **Error! Reference source not found.**. The same power supply will also power the PLC output module which provides power to the output lights.

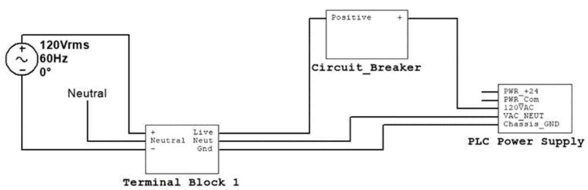

Fig. 6. PLC Power Supply Schematic

VI. Microcontroller Coding

The coding for the microcontroller will be spilt up by what the on-board MCU will accomplish and what the way-side MCU will accomplish. To start, the on-board MCU is the device that will be determining if data given by the encoder is in range or out of range. This will be done by first by pressing a button which will zero the data being given to it by the encoder. The zeroing is started by storing the incoming encoder data in to three different variables that are the initial encoder data, the previous encoder data, and the current encoder data. Then the incoming encoder data is subtracted by the initial encoder data which will provide the value that goes in to current encoder data.

The next part of the process to be coded is the algorithms to determine the in range and out of range values. The first algorithm will be a set of if-else statements that will determine if the current encoder data is either in-range, out of range, or has jumped a significant amount. The MCU will start off by looking for a jump in data by comparing the current encoder data to the previous encoder data value. If a jump in data is found, the current encoder data will be set to 2 and sent to the value comparison code. If there is no jump the MCU will proceed to the next statement. The MCU will transfer the current encoder data to the previous encoder data. If the current encoder value is in-range, the current encoder data will be set to true. If not, the current encoder value will be set to false. The second algorithm will be case statements that follow the same method as the first algorithm. The only difference between the second and first algorithm is that when the second detects a big jump it will set the current encoder data to 3 instead of 2. The values from these two algorithms will be compared to determine if there was a failure. No failure means that they are the same and one of the values will be sent to the Z-Wave module, encapsulated, and transmitted. A failure will increment a counter and while that counter is less than 3, the MCU will request new data from the encoder. As soon as the failure counter reaches 3, the MCU will make the Z-Wave module send a signal to the way-side microcontroller to cause a signal not received flag to set.

The way-side microcontroller has a simpler code that will require it to send respective data through the signal output pin and range output pin. The signal output pins value is determined by two things. The first being if the Z-Wave modules for both on-board and way-side are communicating with each other. The second being from the failsafe's stop receiving signal flag from the on-board microcontroller. The code for both the on-board and way-side Z-Wave modules are provided by the software that comes with them.

VII. PLC Coding

The language used to code the PLC will be ladder logic, as most PLC use this language as a standard. The PLC will have 5 inputs. The wireless signal received, range signal, stuck button, polling button and the [redacted] key button. The first three inputs will be used to illuminate the PLC output lights according to the truth table in **Error! Reference source not found.** Inputs. The PLC output lights are in range, out range, stuck, not stuck, [redacted] stop request and signal. Each light will illuminate according to the truth table in **Error! Reference source not found.**: PLC Truth Table Outputs.

The polling button is to simulate the test object moving around a track passing another sensor to gather the next batch of data. When the polling button is pressed, it will reset all the illuminated lights except the [redacted] stop request light. Then wait 10 seconds before it pulls the next batch of data. The [redacted] key button is to simulate a technician inserting a key into a turn lock. This feature was an added addition to the design for [redacted] purposes. The [redacted] stop request light will stay illuminated until the maintenance key button is pressed **and** the condition causing the operation to be outside of normal is corrected. If the [redacted] key button is pressed without the abnormal condition being correct, the [redacted] stop request light will stay illuminated. The PLC program will have a fail-safe procedure built in that will confirm the information it received is accurate. If somehow the PLC gets conflicting information when it receives the signal, then the PLC will blink the light of the input that is giving incorrect information. For example, if the PLC received the test object was stuck and not stuck in the same data package, the stuck and not stuck lights would blink indicating where the misinformation came from. The [redacted] stop request light will illuminate to ensure that the [redacted] is not operating in an abnormal mode.

VIII. Component Selection

After extensive research on all the possibilities for each component needed the parts have been chosen. Each component listed below was the best choice for this project. Once they type of part needed next was to find who can supply the part and in the least amount of time for shipping and least cost. The following components below will have the specification, and which supply we choose to go with. For each supplier we did try to stick with a distribution in the United States to help with to minimize time if shipping back parts is needed.

A. Battery

The final battery that was decided for this project was a 22000mAh battery at 11.1 volts designed by MaxAmps. The battery is a lithium polymer that can tolerate up to 140°F until thermal runaway occurs. Below 14°F, the shorter the run times which slows down the reaction within the battery, which can cause failure. The battery has a 40C rating for its discharging and a 5C fast charge capability. The power wires that come with the battery is a 12-gauge wire that will have a XT60 connector to help with anti-spark when unplugging and plugging in the battery to the microcontroller. With the capacity chosen will give about 30 hours of run time before recharging will be needed or replacement. In Fig. 7, shows the battery that will be used. The **Error! Reference source not found.** gives the rest of the specifications and ratings for this battery from this supplier that we also took into consideration when deciding on this supplier.

(MaxAmps, n.d.)

Fig. 7. MaxAmps 11.1V Battery.

TABLE 4
LiPo Battery Specification from Supplier

Specification	Rating
Type	Lithium Polymer (LiPo)
Cell(s)	3S – 3 cells in series
Capacity	22,000mAh
Volts	11.1V
Charge Capability	5C
Rating	40C
Warranty	Lifetime
Wire	12awg
Charging Connector	JST-XH
Plug	XT60

B. Encoder

We were able to locate a miniature absolute magnetic regular shaft encoder from US Digital show in **Error! Reference source not found.**. This encoder produces an analog output with a 10-bit resolution. This means each output corresponds to .351° which will exceed the accuracy needed for this design. The miniature encoder is .48” in diameter and is so small, the weight is negligible. The encoder is small, but robust in nature.

(US Digital Products, 2018)

Fig. 8. US Digital Miniature Absolute Magnetic Shaft Encoder.

C. Encoder Shaft

The regular shaft encoder is designed with a solid shaft to be inserted into a bore that accepts that size shaft. This project is to prove a concept that could be used in the future, so it is not known if the future shaft will have a bore or not. Taking this into consideration, we decided to include an encoder coupling. This coupling is designed to fit on top of the encoder shaft and connect it to another solid shaft. This gives the design more flexibility of what type of shaft as well as the diameter of the shaft that can be used in the future. There are a few different types of couplings to choose from. There are magnetic and nonmagnetic, rigid or flexible just to name a few. We felt a magnetic coupling could possibly interfere with the encoder information so to err on the cautious side, we chose a nonmagnetic coupling. Rigid or flexible couplings would both work for our application but to increase the flexibility of the design, we chose a flexible coupling. The flexible coupling will help with run-out putting excessive loads on the encoder. The encoder has a ¼" shaft so one side of the coupling must fit a ¼" shaft. There are several other sizes to pick from for the other side of the coupling so for simplicity reasons, we chose a ¼" bore size. We located a ¼" to ¼" flexible coupling from Encoder Products Company, as shown in **Error! Reference source not found.**, for a reasonable price.

(Encoder Couplings , 2018)

Fig. 9. Encoder Company Flexible Coupling

D. Microcontroller

The microcontroller that was decided on for use was the Raspberry Pi 2 Model B as shown in **Error! Reference source not found.**. There will be two of these in use for the overall project and one of them will be used to program the PCB for the test object's control system and display device. The first Microcontroller will have a push-button, a Z-Wave module, and an encoder attached to it. The first MCU will also be powered by the on-board battery. The second will be powered by a regular ac to dc power supply with a microUSB attachment. The second will have a Z-Wave module and two wires running from two different IO pins to the PLC. The MCUs maximum operating voltage is 5 Volts and Amperage of 2 Amps. The microprocessors on them operate at 900 MHz and have a 1 GB of RAM on-board for volatile memory use.

(Allied Electronics & Automation, 2018)

Fig. 10. Raspberry Pi 2 Model B

E. Z-Wave Module

The Z-Wave.Me RaZberry, **Error! Reference source not found.**, is the module that will be on both of the microcontrollers that enable the ability for the devices to communicate wirelessly. The module has a 10-pin female connector attached to it that allow it to communicate with the microcontroller it is attached to. This device is the reason why the Raspberry Pi 2 was chosen as the desired microcontroller. The other microcontrollers did not offer a way to use the specified wireless communication method. The module is powered by the MCU, has a 32 KB EEPROM to flash a program to, and is low-power compared to the microcontroller.

(Z-Wave.Me, 2017)

Fig. 11. RaZberry Z-wave Plus GPI Card

F. Push Button

This Adafruit 16mm illuminated pushbutton, **Error! Reference source not found.**, was chosen for the zeroing button that will be wired to the microcontroller. The function this button serves is shorting a GPIO pin with a

3.3 Volt pin on the microcontroller. This will allow the microcontroller to initialize the encoder value being fed in to it. The LED on the pushbutton does not need to be powered for it to work but is an added benefit if it ever needs to be utilized. This button will always be operated manually at the start of the test procedure for the project.

(Adafruit, 2018)

Fig. 12. Illuminated Push button

G. PLC

When we approached [redacted] about our choice of PLC type, size and manufacturer, we were told they have some spare Allen Bradley PLCs in their lab and would allow us to use one of theirs for cost effective purposes. After comparing the different CPU modules available, we chose the Allen Bradley SLC 5/05 1747-L553. The SLC 5/05 has a quicker scan time than older models and faster bit execution. The L553 is a 64K memory that will be more than enough for our programming needs. Once the CPU was chosen, we had to verify the proper power supply module was available for the SLC 5/05. Fortunately, the proper power supply was on hand which is an Allen Bradley 1746 SLC System, Power Supply - Rack Mount, 120/240 VAC, 5A. This means we can plug it directly into a 120V wall outlet without the need for a power supply in between. Next was deciding what I/O modules were needed. The input coming into the PLC is a digital signal from the microcontroller that means the input card must be digital. There is going to be four inputs into the card so a minimum of an eight-point input card was needed. The input points in an Allen Bradley are packaged in eights. The input is only supposed to receive information and does not need to power any externals therefore a sinking module was the best option. The 1746-IB16 input module was available which covers all the requirements. The 1746-IB16 is a sinking digital module with sixteen available input points. Even though we need only eight points, it is always a good idea to have spares on hand in case the design requires more in the future. Lastly an output module was needed. The PLC would be powering the external lights as outputs, so we needed a sourcing module. There are six different lights that need to illuminate therefore we needed at least an eight-point output module. The signal being sent to the light is to be either on or off, which is a digital signal, so we needed a digital module. We were able to locate a 1746-OB16 output module in the lab. The 1746-OB16 output module is a digital sourcing module with sixteen available output points. Once again, even though only eight points are needed, it is always good to have spares on hand. The chassis that was available in the lab has slots for the power supply, the CPU, the input module, the output module and one other module. Purely for aesthetic purposes and possible future use, we decided to add in a communications module to complete the chassis. The module available is a MVI46-MCM which we will not be using for this project.

IX. Safety Aspect

The encoder ensures the position is maintained when a power outage occurs, this ensures an accurate position is always maintained. The MCU will also store this maintained position in non-volatile memory storage which will ensure that the data coming in from the encoder is correct. Once the angular position enters the on-board microcontroller, the data will be converted and tested as follows **Error! Reference source not found.**.

Fig. 13. On-Board Failsafe Code Flow Diagram

The first algorithm will be a set of if-else statements that will determine if the given value from the encoder is either an in-range angle or out of range angle. The MCU will set a new value to true if the encoder value is in-range, or else the new value will be false because the original value was out of range. The second algorithm will be case statements based on the same encoder data the first algorithm uses. Case one will check if the encoder value is an in-range angle and if it is a new value is set to true. Case two will check if the encoder value is an out of range angle and if it is a new value is set to false. The values from these two algorithms using the same original data from the encoder will be compared. If they are the same, one of the values will be sent to the Z-Wave module, encapsulated, and transmitted. **Error! Reference source not found.** shows the truth table of how the MCU uses both algorithms to determine whether or not to send the data.

TABLE 5
Truth Table for Algorithm Comparison Output

Encoder Data	Primary Algorithm	Secondary Algorithm	Comparison
In Range	TRUE	TRUE	TRUE
In Range	TRUE	FALSE	FALSE
Out of Range	FALSE	TRUE	FALSE
Out of Range	FALSE	FALSE	TRUE

The way-side microcontroller will receive the signal. When the inputs enter the PLC for interpretation, the PLC will ensure the information received is accurate shown in **Error! Reference source not found.**. If there is misinformation, then the PLC will blink the light of the input that is giving incorrect information.

Fig. 14. PLC Fail Safe Tree

X. Budget

We anticipate the project budget rough order of magnitude to be $10,000. This project will be funded by [redacted]. The UCF part of the project's budget will be roughly $250 and was not funded by [redacted]. In **Error! Reference source not found.** is the breakdown of the cost for each component that will need to be bought. There are other components that are needed for this project, but [redacted] already had in the lab to use.

TABLE 6
Cost of Parts

Part	Qty	Cost
5V Voltage Regulator	1	$8.89
Battery	1	$399.99
SKYRC iMAX B6AC	1	$54.55
Connector – XT60	1	$9.56
WattMeter-Bare	1	$48.99
Encoder	1	$55.50
Encoder connector	1	$8.30
Encoder Coupling	1	$50.00
Micro USB Power	2	$9.00
Microcontroller	2	$34.99
Zeroing push button	1	$1.95
Z-Wave Module	2	$49.99
Total		$725.71

XI. Conclusion

As [redacted] advance in speeds and technology, the need for this angular position accuracy needs to be increased. This project proves that the concept is not only possible but that it can also meet all the sponsor design requirements. The ultimate goal will be to incorporate this project into a [redacted] control system.

Acknowledgement

The authors wish to acknowledge Tina Haley, Jillian Bryant and Roland Bulnes for their guidance in our design reviews. The authors also wish to acknowledge Justin Brock and Chris Russell for their extensive help with the PLC setup and troubleshooting.