Design and Implementation of an Overhead Transmission Line Position Tracking Device

by
Braedon Borschawa
Kevin Lamothe
Mark Rabena
Eric St-Onge

Final report submitted in partial satisfaction of the requirements for the degree of Bachelor of Science in Electrical and Computer Engineering in the Faculty of Engineering of the University of Manitoba

Faculty and/or Industry Supervisors:
Dr. Greg Bridges
Mr. Dean Reske

Submitted on March 17th, 2017

© Copyright by Braedon Borschawa, Kevin Lamothe, Mark Rabena and Eric St-Onge, 2017
Abstract

As required by the environmental provisions for the Manitoba Hydro Bipole III transmission line, its electric and magnetic fields must be measured and recorded for analysis and design validation. Both electric and magnetic fields are a function of position relative to the measurement points, thus requiring the position of each conductor to be measured. The position measurements are also required to ensure that the horizontal sway of each conductor remains within safe operating limits.

A Transmission Line Position Tracker (TLPT) was designed to monitor the position of overhead electric transmission lines in all weather conditions. The system is outfitted with 4 main subsystems: an enclosure to protect from environmental conditions, a motor-drive system to rotate the laser through its field of view, a laser measurement unit and electronic circuitry to operate, control and gather sensor data from the various systems within the enclosure. Software was written to operate each subsystem and communicate the data gathered to a central data acquisition unit over a Controller Area Network (CAN) bus.

Unit testing was performed on each subsystem to verify they met the operational requirements and when assembled as a completed system would meet or exceed the performance metrics. Software test programs were used to verify reliable CAN bus communication under ideal conditions and in an environment with high levels of corona discharge. Mechanically, the TLPT was tested under ideal and nominal loading conditions to confirm mechanical robustness of all moving parts.

Once set up and powered on, the TLPT is capable of continually scanning, measuring, and communicating in an environment of noise induced from constant corona discharge from the High Voltage Direct Current (HVDC) transmission line above. Due to the operational lifetime required for the TLPT, the system is capable of indicating faults with the rotary position encoder, Universal Laser Sensor (ULS), CAN bus and one user-defined error indicator, indicating maintenance is required.
The success of the TLPT cannot be solely attributed to the authors. The project would not have been possible without the time, knowledge and experience from each but not limited to the following individuals.

- Mr. Dean Reske - Supplied numerous components and materials in-kind on behalf of Manitoba Hydro as well as technical advice throughout the project.
- Dr. Ahmad Byagowi - Provided insight and advice on the mechanical aspect of our design.
- Dr. Behzad Kordi - Allowed testing at the Stanley Pauley High Voltage testing Facility to verify the robustness of our communication system.
- Dr. Greg Bridges - Provided technical advice and guidance throughout the project.
- Mr. Daniel Card - Provided feedback and advice on our coursework that was helpful for us throughout the whole project.

Contributions from each author are listed in Table 1. A o indicates a partial contribution while a ● indicated a lead role.

<table>
<thead>
<tr>
<th></th>
<th>Braedon Borschawa</th>
<th>Kevin Lamothe</th>
<th>Mark Rabena</th>
<th>Eric St-Onge</th>
</tr>
</thead>
<tbody>
<tr>
<td>Laser Unit Configuration</td>
<td>●</td>
<td></td>
<td></td>
<td>o</td>
</tr>
<tr>
<td>Circuit Design</td>
<td>o</td>
<td>●</td>
<td>o</td>
<td></td>
</tr>
<tr>
<td>PCB Design</td>
<td></td>
<td>o</td>
<td>●</td>
<td></td>
</tr>
<tr>
<td>PCB Assembly</td>
<td></td>
<td></td>
<td></td>
<td>●</td>
</tr>
<tr>
<td>PCB Testing</td>
<td></td>
<td></td>
<td></td>
<td>o</td>
</tr>
<tr>
<td>Human Machine Interface</td>
<td></td>
<td>●</td>
<td></td>
<td>o</td>
</tr>
<tr>
<td>Software Development</td>
<td>o</td>
<td></td>
<td>●</td>
<td></td>
</tr>
<tr>
<td>Motor Drive</td>
<td>●</td>
<td>o</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Enclosure Design</td>
<td>o</td>
<td></td>
<td>●</td>
<td></td>
</tr>
<tr>
<td>Mechanical Assembly</td>
<td>o</td>
<td>●</td>
<td>o</td>
<td></td>
</tr>
<tr>
<td>Environmental Protection</td>
<td>●</td>
<td>o</td>
<td></td>
<td>o</td>
</tr>
</tbody>
</table>
Contents

1 Introduction ........................................... 12
  1.1 System Overview .................................... 12
  1.2 Performance Metrics ................................. 14

2 Universal Laser Sensor ................................ 15
  2.1 Overview ........................................... 15
  2.2 Measurement Modes ................................. 15
  2.3 Viewing Window .................................... 16

3 Power Supply ........................................ 21
  3.1 Overview ........................................... 21
  3.2 Design and Component Selection .................... 21
  3.3 Implementation & Validation ......................... 24

4 Rotary Position Encoder ............................... 26
  4.1 Overview ........................................... 26
  4.2 Decision Reasoning ................................ 26
  4.3 Implementation & Validation ......................... 27

5 Motor & Drive ......................................... 29
  5.1 Overview ........................................... 29
  5.2 Motor & Controller ................................ 29
    5.2.1 Design ........................................ 29
    5.2.2 Implementation & Validation .................... 30
5.3 Motor Drive Assembly ........................................ 31
5.3.1 Design .................................................. 31
5.3.2 Implementation & Validation ............................ 33

6 Temperature Management ................................. 35
6.1 Design & Decision Reasoning ............................... 35
6.2 Implementation & Validation ............................... 36

7 Main Control Unit ........................................... 38
7.1 Microcontroller .............................................. 39
7.1.1 Decision Reasoning ..................................... 39
7.2 Communications ............................................. 40
7.2.1 Design & Decision Reasoning ......................... 40
7.2.2 Implementation & Validation ......................... 41
7.3 Temperature Control ........................................ 42
7.3.1 Design & Decision Reasoning ......................... 42
7.3.2 Implementation & Validation ......................... 44
7.4 Light Emitting Diode (LED) Control ..................... 45
7.4.1 Design & Decision Reasoning ......................... 45
7.4.2 Implementation & Validation ......................... 47
7.5 Firmware .................................................. 47
7.5.1 Design & Decision Reasoning ......................... 47
7.5.2 Implementation .......................................... 49

8 Data Receiver ................................................ 50
8.1 Microcontroller .............................................. 50
8.2 Communications .................................................. 51
  8.2.1 Design & Decision Reasoning ................................ 51
  8.2.2 Implementation & Validation ................................ 51
8.3 Firmware ........................................................... 52
  8.3.1 Design & Decision Reasoning ................................ 52
  8.3.2 Implementation ................................................. 54

9 Data Transmission .................................................... 55
  9.1 Design & Decision Reasoning .................................... 55
  9.2 Implementation & Validation .................................... 55

10 Enclosure ............................................................. 57
  10.1 Overview ......................................................... 57
  10.2 Frame ............................................................. 57
     10.2.1 Design ...................................................... 57
     10.2.2 Implementation & Validation .............................. 59
  10.3 Environmental Protection ...................................... 59
     10.3.1 Design ...................................................... 59
     10.3.2 Implementation & Validation .............................. 60

11 Conclusion .......................................................... 62
  11.1 Future Recommendations ....................................... 62
     11.1.1 Adjustable Motor Speed .................................. 62
     11.1.2 Temperature Management ................................. 62
     11.1.3 System Portability ......................................... 63
A Electrical Schematics 66

B Firmware Code 70

B.1 Main Control Unit 70

B.2 Data Receiver 77
List of Figures

1.1 System Block Diagram ........................................ 13
1.2 The insides of the TLPT ...................................... 13

2.1 Laser separator .................................................... 18
2.2 Measurement difference at window angle of 30° ............... 19
2.3 Measurement difference at window angle of 35° ............... 19
2.4 Measurement difference with an increased height ............ 20

3.1 Power supply block diagram .................................... 21
3.2 Power supply wiring schematic ................................ 22
3.3 Power supply circuit .............................................. 24
3.4 Power Printed Circuit Board (PCB) layout .................... 25
3.5 Calculation of the ampacity of the Power PCB traces ........ 25

5.1 Motor & Controller .............................................. 30
5.2 Motor-Drive assembly render .................................. 32
5.3 Motor-drive assembly ............................................ 33
5.4 Signal testing on slip ring, rotating & stationary ............ 34

6.1 Location of the heating element inside the TLPT enclosure 37

7.1 The Main Control Unit (MCU), installed in the TLPT ....... 38
7.2 Nucleo-F303K8 pin-out[18] ..................................... 40
7.3 CAN controller and supporting circuitry ....................... 41
7.4 PCB layout of the MCU’s CAN hardware ...................... 42
7.5 MCU’s CAN temperature management circuitry .................................. 43
7.6 PCB layout of the MCU’s temperature control hardware ......................... 44
7.7 The TLPT’s Human-Machine Interface (HMI) ........................................ 45
7.8 LED driver circuit ............................................................................... 46
7.9 Hardware timer circuit ........................................................................ 46
7.10 PCB layout of the MCU’s LED control hardware ................................... 47
7.11 The MCU’s firmware initialization sequence ........................................ 48
7.12 The MCU’s firmware main loop .......................................................... 49
8.1 The Data Receiver ............................................................................... 50
8.2 The Data Receiver (DR) circuit schematic ............................................. 51
8.3 The DR PCB layout ............................................................................. 52
8.4 Trigonometry of the line position calculations; not drawn to scale. ........ 53
8.5 The DR’s firmware program ................................................................. 54
9.1 The TLPT in the High Voltage (HV) lab, ready for stress testing ............ 56
10.1 Arched frame design .......................................................................... 58
10.2 Rectangular frame design ................................................................... 58
10.3 Rectangular frame realized ................................................................. 59
10.4 Enclosure panel sealing ...................................................................... 61
A.1 MCU hardware timer schematic (scanning indicator) ............................ 66
A.2 MCU LED drivers schematic ............................................................... 66
A.3 The MCU PCB layout ......................................................................... 67
A.4 MCU top-level schematic .................................................................... 68
A.5 TLPT wire harness schematic . . . . . . . . . . . . . . . . . . . . . . . . . 69
List of Tables

1  Project and Report Contributions ............................................. 2

1.1 System performance metrics ................................................. 14

2.1 Distance measurements at 16m ........................................... 17
2.2 Distance measurement at 20m ............................................ 17
2.3 Distance measurement at 40m ............................................ 18

3.1 Voltage and current requirements of TLPT components ............... 22

4.1 Decision matrix grading scale ............................................ 26
4.2 Rotary position encoder decision matrix ................................ 27

5.1 Direct Current (DC) motor loading data ................................ 31
5.2 Motor-drive assembly part description .................................. 32

6.1 Operating temperature ranges of components .......................... 35

7.1 Microcontroller decision matrix ......................................... 39

9.1 HV environment data transmission test results ......................... 56

10.1 Enclosure material decision matrix ..................................... 60
Acronyms

AC  Alternating Current
AWG  American Wire Gauge
CAD  Computer-Aided Design
CAN  Controller Area Network
DAQ  Data Acquisition
DC  Direct Current
DIP  Dual In-Line Package
DR  Data Receiver
ESD  Electrostatic Discharge
HMI  Human-Machine Interface
HV  High Voltage
HVDC  High Voltage Direct Current
IDE  Integrated Development Environment
IEEE  Institute of Electrical and Electronics Engineers
IMU  Inertial Measurement Unit
LED  Light Emitting Diode
LTI  Laser Technology, Inc.
MCU  Main Control Unit
MOSFET  Metal-Oxide-Semiconductor Field-Effect Transistor
PCB  Printed Circuit Board
PPM  Pulses Per Measurement
PRF  Pulse Rate Frequency
RMS  Root Mean Square
RPM  Revolutions Per Minute
SPI  Serial Peripheral Interface
SPST  Single Pole Single Throw
TLPT  Transmission Line Position Tracker
TOF  Time of Flight
ULS  Universal Laser Sensor
Chapter 1

Introduction

The purpose of this project is to design and implement an overhead transmission line position tracking device, referred to herein as the Transmission Line Position Tracker (TLPT). The TLPT is designed to remain under the Bipole III transmission line on a testing site for an operating life of 2 years to continually measure the transmission line height and horizontal sway, thereby fulfilling environmental licensing requirements for Manitoba Hydro. The TLPT was designed to rotate a Universal Laser Sensor (ULS) distance measurement device to scan a $110^\circ$ field of view while collecting angle measurements using a rotary position encoder. The data is then sent over a Controller Area Network (CAN) bus to a data acquisition system, physically located in a trailer on the testing site, at a distance of approximately 70m from the TLPT. All of the TLPT’s electronics are powered by a custom power supply and supported by a steel frame enclosure to protect from the elements.

1.1 System Overview

Early design work of the TLPT involved breaking the overall device into sub-systems outlining the major functional components needed to meet the performance metrics. As can be seen in fig. 1.1, the enclosure sub-system encloses the ULS and supporting electronics, protecting them from the elements while providing a physical structure to mount the motor-drive unit, power supply system, MCU and wiring between components. Figure 1.2 shows a top-view of the TLPT electronics and components.
Figure 1.1: System Block Diagram

Figure 1.2: The insides of the TLPT
1.2 Performance Metrics

The metrics outlined in table 1.1 were used to quantitatively evaluate the performance of the TLPT. This table compares the proposed performance metrics to the measured performance metrics of the completed device to ensure each sub-system functions as designed and that the device meets Manitoba Hydro’s needs. It should be noted that the two unmet performance metrics have been renegotiated with our client; they are to be confirmed at a later date.

Table 1.1: System performance metrics

<table>
<thead>
<tr>
<th>Performance Metric</th>
<th>Proposed Value</th>
<th>Measured Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Laser Measurement Angle</td>
<td>-55°..+55°</td>
<td>-60°..+60°</td>
</tr>
<tr>
<td>Time for One Full Scan</td>
<td>≤5 Seconds</td>
<td>3 Seconds</td>
</tr>
<tr>
<td>Distance to Detect Transmission Line</td>
<td>10..40m</td>
<td>10..40m</td>
</tr>
<tr>
<td>Must Detect All Conductors</td>
<td>6 Conductors</td>
<td>TBD</td>
</tr>
<tr>
<td>Position Error</td>
<td>≤1%</td>
<td>0.27%</td>
</tr>
<tr>
<td>Operating Temperature Range</td>
<td>-30°..+60°C</td>
<td>TBD</td>
</tr>
<tr>
<td>Data Transmission Distance</td>
<td>≥70m</td>
<td>100m</td>
</tr>
</tbody>
</table>
Chapter 2
Universal Laser Sensor

2.1 Overview

The ULS made by Laser Technology, Inc. (LTI) is the distance measurement device of the TLPT. The ULS is a laser-based device that operates based on the Time of Flight (TOF) principle, utilizing a 905nm infrared wavelength. The ULS has two main lenses on the front face, one for transmission and one for reception. It also has a red laser pointer for a visual indication of its line of sight. There are two communication ports on the back, the configuration and universal output ports. The configuration port is used to provide power as well as input and output data to and from the ULS, whereas the universal port can only be used for power and data output. The configuration port utilizes an RS-232 interface. The universal output port outputs data over RS-232, RS-485 or a 4-20mA current loop. For the TLPT, only the configuration port is used since the MCU configures the ULS upon initialization.

2.2 Measurement Modes

The ULS has four different operating modes for different applications: Averaging Mode, Detection Mode, Last Target Mode and Binning Mode. Within each measurement mode there are several parameters which will affect how the measurement is taken. Certain parameters are used for multiple or all measurement modes. Pulse Rate Frequency (PRF) determines the rate at which the ULS takes measurements and must be set for all modes. The power level can also be adjusted for each mode with higher power used to penetrate dust and fog and low power to avoid reflections.

Averaging Mode utilizes a series of pulses and detections then averages them to determine the distance. That is, the output data rate is equal to the Pulses Per Measurement (PPM) divided by the PRF. For the purposes of this project, averaging mode was used for initial testing, typically at low PPM and medium to high PRF as the transmission lines are relatively small at far distances. However, when measuring continuous output data in this mode with no background to detect, the output returns an error. The numerous outputs also require messages to be continuously read by the MCU, which resulted in data loss due to a baud rate bottleneck.

Detection Mode has two different operational modes: Relative Detection Mode and Absolute Detection Mode. Detection Mode outputs the distance reading to a target if it is within a user defined trip point. The difference between Relative and Absolute Detection Mode is what the trip point is relative to. The trip point in Relative
Detection Mode is a user defined distance from a constant background behind the target, while for Absolute Detection Mode, the trip point is a user defined distance from the ULS. Since there is no constant background behind a transmission line, Absolute Detection Mode was utilized. Once the target is detected and a measurement is output, the ULS then waits for a set number of false measurements to reset. This is ideal for the TLPT as it eliminates the baud rate issue and numerous meaningless measurements/errors from the continuous output of averaging mode.

The last two operating modes, last target and binning have specific roles. Last target is used in low visibility conditions where a target may be difficult to detect and has lowered accuracy. Binning mode is utilizes multiple scans to detect objects at greater distances than the transmission lines. In testing, these modes proved to be of little use for the TLPT.

### 2.3 Viewing Window

Initially, a narrow opening was to be created in the top of the enclosure through which the ULS would scan for the overhead transmission lines. To validate this method, testing was conducted using two solid panels to simulate the cover of the enclosure with the space between them being the viewing window. The ULS was set to constantly output data measurements by using Averaging Mode. While observing the data measurements, one side of the cover was slowly moved towards blocking the path of the laser until the cover began to interfere with the measurements. The cover was then moved back until readings returned to their expected values. This was then repeated for the other side of the cover. This test showed that for the ULS to operate without getting erroneous data from interference due to the enclosure, an opening of at least 4.5cm was required. As such, a physical viewing window is necessary to allow the ULS to scan for the transmission lines while keeping the enclosure sealed from the elements.

In choosing the material for the window, considerations were made for light transmission, durability, cost, and ease of use. The three main materials considered due to availability were acrylic, polycarbonate and glass. Glass has the highest optical transmission but is the least impact resistant and the most difficult to cut and maintain. Polycarbonate is the most durable, but is the most expensive and has the lowest optical transmission, which also degrades over time. Acrylic is inexpensive, durable, easy to cut and maintain and has good optical transmission. Acrylic was chosen for these reasons.

Testing was done to determine the amount of error in distance measurement added by the presence of acrylic. Error was considered to be the difference in average measurements with and without the acrylic. The speed of light in acrylic is found using eq. (2.1) where the index of refraction of acrylic is $n \approx 1.49[1]$ and the accepted speed of light in a vacuum is $c = 299792458 \frac{m}{s}$.
\[ \nu = \frac{c}{n} = 201202992 \frac{m}{s} \] (2.1)

With the speed of light in acrylic known and a thickness of 2.5mm used, the percent change in the TOF of a measurement can be found using eq. (2.2). For this case, a 16m measurement is used where 15.9975m is in air and 0.0025m is in acrylic.

\[
\% \text{ Change} = \frac{15.9975}{c} + \frac{0.0025}{16} \cdot \frac{201202992}{c}
= 1.00007656 \%
\] (2.2)

New Distance = 16 \times 1.00007656
= 16.00122 m

At a thickness of 2.5mm, the acrylic adds 1.22mm of error to a distance reading of 16m. To test this, the ULS measured a target at 16, 20 and 40 meters. The ULS was set to averaging mode, 500 PRF and one pulse per measure such that it would output 500 measurements per second. The data was gathered for at least six seconds, or 3000 measurements, per scenario. For each scenario, measurements were taken without the acrylic, with the acrylic parallel to the face of the laser and with the acrylic 45° to the face of the laser. The results can be found in tables 2.1 to 2.3.

Table 2.1: Distance measurements at 16m

<table>
<thead>
<tr>
<th></th>
<th>No Window</th>
<th>Parallel Window</th>
<th>Angled Window</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg</td>
<td>16.0068</td>
<td>16.0056</td>
<td>16.0043</td>
</tr>
<tr>
<td>Min</td>
<td>15.945</td>
<td>15.977</td>
<td>15.971</td>
</tr>
<tr>
<td>Max</td>
<td>16.017</td>
<td>16.019</td>
<td>16.043</td>
</tr>
</tbody>
</table>

Table 2.2: Distance measurement at 20m

<table>
<thead>
<tr>
<th></th>
<th>No Window</th>
<th>Parallel Window</th>
<th>Angled Window</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg</td>
<td>20.0231</td>
<td>20.0164</td>
<td>20.0133</td>
</tr>
<tr>
<td>Min</td>
<td>19.995</td>
<td>19.984</td>
<td>19.976</td>
</tr>
<tr>
<td>Max</td>
<td>20.033</td>
<td>20.051</td>
<td>20.048</td>
</tr>
</tbody>
</table>
Table 2.3: Distance measurement at 40m

<table>
<thead>
<tr>
<th></th>
<th>No Window</th>
<th>Parallel Window</th>
<th>Angled Window</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg</td>
<td>40.0178</td>
<td>40.0184</td>
<td>40.0184</td>
</tr>
<tr>
<td>Min</td>
<td>39.981</td>
<td>39.97</td>
<td>39.982</td>
</tr>
<tr>
<td>Max</td>
<td>40.053</td>
<td>40.039</td>
<td>40.054</td>
</tr>
</tbody>
</table>

To prevent reflections, two methods were tested. As suggested by the ULS user’s manual[2], a physical separator was designed and 3D-printed which helps to prevent reflections from being detected by the receiving lens. The separator can be seen in fig. 2.1.

In addition, the window was angled to reduce the risk of glare. To test these methods, the ULS was set to Averaging Mode, at 500 PRF and one PPM such that it would output 500 measurements per second. Data was collected while the laser was rotating with no window, and then with the window at various angles. The results of the test can be seen in figs. 2.2 and 2.3, where peaks represent a discrepancy between the baseline no-window measurements compared to the with-window measurements.
It can be seen that the addition of the acrylic window narrows the viewing angle while also causing reflections at ± 30° from the vertical until the window is angled at a minimum of 35°. These tests were done with and without the laser separator and it was determined that the laser separator provided no reduction in reflections, so it was removed. To increase the viewing angle, the height of the laser and motor drive assembly was raised. As seen in fig. 2.4, the viewing angle increased from 100° to 120°.
Figure 2.4: Measurement difference with an increased height
Chapter 3

Power Supply

3.1 Overview

The TLPT will be powered from the only available source, which is a standard 120VAC 60Hz outlet located 70m away at the test site. The power supply utilizes a step down transformer, and a power PCB that has a full bridge rectifier and regulator to provide the various components of the TLPT with constant 12VDC over the entire range of current draw requirements. The high-level block diagram of the complete power supply and distribution can be seen in fig. 3.1 below.

Figure 3.1: Power supply block diagram

3.2 Design and Component Selection

To design a power supply that is sufficient for all of the TLPT’s components, their voltage and current requirements were considered as shown in table 3.1. It can be seen that most components can operate on 12VDC\(^1\) and the total current draw for all components is 2.5A.

\(^1\)Components operating on less than 12V are powered from the microcontroller
Table 3.1: Voltage and current requirements of TLPT components

<table>
<thead>
<tr>
<th>Component</th>
<th>Voltage Requirement</th>
<th>Current Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motor Controller</td>
<td>6-24V nominal, 30V max</td>
<td>100mA maximum</td>
</tr>
<tr>
<td>Human Machine Interface</td>
<td>1.85V nominal, 2.5V max</td>
<td>27mA maximum</td>
</tr>
<tr>
<td>ULS</td>
<td>12-24V nominal</td>
<td>170mA typical</td>
</tr>
<tr>
<td>Heating Element</td>
<td>12-24V nominal</td>
<td>0.625..1.25A</td>
</tr>
<tr>
<td>Microcontroller</td>
<td>3.3-12</td>
<td>300mA maximum</td>
</tr>
<tr>
<td>Motor</td>
<td>6-12V</td>
<td>540mA nominal, 20A stall</td>
</tr>
<tr>
<td>Encoder</td>
<td>4.5-5.5V, 5V typical</td>
<td>8mA typical, 10mA maximum</td>
</tr>
<tr>
<td>Fan</td>
<td>12VDC</td>
<td>120mA typical</td>
</tr>
</tbody>
</table>

The power supply was designed taking into account the required voltage and current to operate the TLPT. The power supply is comprised of a power cable, transformer, bridge rectifier, and voltage regulator. It was decided for the rectifier and regulator to be implemented as a PCB while the transformer would remain external to the PCB due to its size. Prior to the transformer is the main power connector, power switch, and fuse, as can be seen from fig. 3.2 (extracted from the full wire harness schematic in fig. A.5).

![Figure 3.2: Power supply wiring schematic](image)

In order to carry power from the test-site to the TLPT, a power cable of at least 70m long had to be selected. Due to the considerable distance, the voltage drop across the power cable must be minimized to continually power the ULS under different loading conditions. Based on eq. (3.1), it was determined that a conductor size of 18AWG running a load current of 0.5A will only result in a 1.4V voltage drop. The Belden 5301FE was then chosen as an economical option that meets this size requirement and is also shielded.[3]
\[ V_{\text{drop}} = I_{\text{wire}} \cdot R_{\text{wire}} \]  
\[ = 0.5 \text{A} \cdot 2 \cdot 70 \text{m} \cdot 20.95 \frac{\text{n} \Omega}{\text{m}} \]  
\[ = 1.4 \text{V} \]  

The main power connector is a C13-type connector, chosen to encourage user-familiarity with the TLPT prior to use, as the commonly-used NEMA 5-15 connector is not compatible with the design. The main power switch is a Single Pole Single Throw (SPST) pushbutton with a current rating of 16A at 125VAC and an IP54 ingress protection rating.[4]  

The transformer is a 120V RMS:12V RMS step-down Triad Magnetics transformer (TCT50-08E07AB) with a maximum power output rating of 50VA and maximum output current rating of 4.17A[5]. This transformer was selected based on its cost, availability and its ability to meet the power requirements outlined in table 3.1. The primary side of the transformer is fused in order to protect the electrical components of the TLPT from an overcurrent scenario. Since the transformer has a secondary-side current of 4.17A, then the maximum primary-side current is around 417mA. As such, the primary side of the transformer (and therefore the entire electrical system of the TLPT) is fused with a 0.417A fuse with an \( I^2t \) rating of 2.55A^2t[6]. This fuse was selected such that it would trip within 10s of a 0.5A current draw as suggested by the transformer data sheet [5].  

The secondary side of the transformer connects to the Power PCB, on which is found the remainder of the power supply circuitry, shown in fig. 3.3. The bridge rectifier (KBU6D) is used to rectify the 12VAC voltage from the transformer into an unregulated DC voltage, and was chosen due to its low 1V voltage drop and 6A rectified current rating[7].  

The capacitor labeled C1 is used to reduce voltage ripple of the rectified voltage from the bridge rectifier. The value of this capacitor was calculated using eq. (3.2) for a peak-to-peak voltage ripple of no more than 3V with a peak load current of 4.17A. Based on common capacitor values, a capacitance of 22mF was chosen as the smoothing capacitor. The capacitor increased the average voltage of the unregulated DC side to above 14.5VDC.  

\[ C = \frac{I}{f \cdot \Delta V} \]  
\[ = \frac{4.17 \text{A}}{60 \text{Hz} \cdot 3 \text{V}} \]  
\[ = 0.02317 \text{F} \]  

The final component in the power supply is to regulate the voltage to 12VDC. This was deemed necessary as an unregulated power supply would fluctuate due to varying...
load currents. The LT1084CT linear voltage regulator was chosen due to its low dropout voltage of 1.3VDC at 5A, high output current of 5A and output voltage of 12VDC[8]. The low dropout voltage is beneficial as a secondary protection in the scenario that the peak to peak voltage ripple is more than the 3V. However, linear regulators produce excessive heat as a result of energy loss and must be cooled. From the regulator data sheet[8], the thermal resistance \( R_T = 241 \) C/W would produce 12°C above the ambient temperature. In order to aid in temperature regulation, a fan was installed to circulate air in the enclosure and keep the devices in their operating temperature ranges which is discussed in section 6.2.

### 3.3 Implementation & Validation

The power supply subsystem is required to provide a peak current of 4.17A while maintaining a regulated voltage of 12VDC. The power supply was tested using a Programmable DC Electronic Load. The internal resistance of the load was varied such that it would draw a minimum of 2A to a maximum of 4.17A, simulating the minimum and maximum loading conditions of the TLPT. Upon completion of testing, it was found the power supply was successful at maintaining 12VDC at the designed loading.

Physical implementation of the Power PCB is shown as a 2-dimensional CAD render in fig. 3.4 with physical dimensions of 2.2” x 2.2”. All components were placed on the top layer to have the bottom layer as a ground plane. The traces are 80 mil wide in order to accommodate the power supply’s maximum current of 4.17A. The Saturn PCB Toolkit was used to verify that an 80 mil wide trace is suitable for up to 5.76A of current, as shown in fig. 3.5. Testing points were intentionally added to the PCB to help with design validation. Mounting holes were chosen to be 3mm in diameter to
allow the use of M3 standoffs. Mate-n-Lok connectors were chosen because of their voltage and current ratings of 600 VDC and 9.5A, respectively[9].

Figure 3.4: Power PCB layout

Figure 3.5: Calculation of the ampacity of the Power PCB traces
Chapter 4

Rotary Position Encoder

4.1 Overview

The rotary position encoder is mounted to the motor drive shaft and is used to
determine the angle at which a distance measurement occurred. It is a CUI AMT20
absolute encoder with 12-bit resolution and a position accuracy of 0.2 degrees, and uses
an Serial Peripheral Interface (SPI) interface to receive commands and output
responses[10].

4.2 Decision Reasoning

An absolute rotary encoder is a device which measures angular position in discrete
steps relative to a programmable zero point reference. The primary factors considered
when selecting a rotary encoder were cost, communication protocol, resolution,
accuracy, size, and mounting. Encoders from Baumer (G1M2H), BEI (CHO5),
Broadcom (AEAT-6012), CUI (AMT-20) and Dynapar (AD35) were considered as seen
in table 4.2. The decision matrices in this report utilize the rating system seen in
table 4.1.

Table 4.1: Decision matrix grading scale

<table>
<thead>
<tr>
<th>Grade</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exceptional</td>
<td>5</td>
</tr>
<tr>
<td>Good</td>
<td>4</td>
</tr>
<tr>
<td>Average</td>
<td>3</td>
</tr>
<tr>
<td>Adequate</td>
<td>2</td>
</tr>
<tr>
<td>Poor</td>
<td>1</td>
</tr>
</tbody>
</table>
Table 4.2: Rotary position encoder decision matrix

<table>
<thead>
<tr>
<th></th>
<th>G1M2H</th>
<th>CHO5</th>
<th>AEAT-6012</th>
<th>AMT-20</th>
<th>AD35</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Protocol</td>
<td>5</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>Resolution</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>Accuracy</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Size</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>Mounting</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>Rank</td>
<td>18</td>
<td>20</td>
<td>18</td>
<td>24</td>
<td>22</td>
</tr>
</tbody>
</table>

Absolute encoders have resolutions of $X$ bits, where $2^X$ represents the number of positions per revolution. By researching absolute encoders and contacting suppliers, it was determined that encoders with greater than 12-bit resolution were far more expensive and beyond budgetary constraints. The CUI AMT20 was the perfect balance between cost and performance. The accuracy of the rotary encoder, which is given as an angle $\theta = 0.2^\circ$, can be represented in terms of distance from the actual transmission line to the detected location. To do this, simple trigonometry was used to derive eq. (4.1) where $d$ is the distance along the arc of a circle of radius $r$ from the laser to the transmission line.

$$d = \sqrt{2r^2 - \cos(\theta)(2)(r)^2} \quad (4.1)$$

At accuracy of $0.2^\circ$, results in a measurement accuracy of 5.6cm when the transmission line is 16m from the ULS and 11.34cm when the transmission line is 32.5m from the ULS.

### 4.3 Implementation & Validation

All interactions with the encoder are done by the MCU using an SPI interface. The two primary commands that can be sent to the encoder are set zero point and read position. The set zero point command is used to calibrate the encoder prior to its first use. This command sets the encoder’s current position as ”zero degrees”, storing it in non-volatile memory. Subsequent position readings are then given, using the read position command, as a numerical value ranging from 0 to 4096, relative to this zero point.

For the purposes of the TLPT, the rotary encoder was calibrated such that the zero point is level with the ground, i.e. zero degrees is the position at which the ULS is pointing perfectly horizontally. This leveling of the ULS was done using the Ivensense
MPU-6500 Inertial Measurement Unit (IMU) and an online interfacing tool provided by Webkay[11]. Using the relative angle of the IMU it was possible to position the ULS to 0.0° by fixing the IMU to the top of the ULS and positioning the rotating assembly to the desired position. The set zero point command was then sent to the encoder at that point.

To verify the general functionality of the encoder, an example program was developed to continuously send the read position command and display the result in a computer terminal. It was confirmed that the encoder outputs a value from 0 to 4096 dependent on its rotary position, as expected. If the encoder was turned beyond the 4096th position, the value rolled over back to 0. Rotation in both clockwise and counter-clockwise directions posed no difficulties.
Chapter 5
Motor & Drive

5.1 Overview

The ULS is required to rotate to scan the field of view to detect each set of conductors on the Biople III transmission line. To rotate the ULS a small DC motor with a planetary gear set rotates a shaft at 21.69 Revolutions Per Minute (RPM). A 12-circuit slip ring provides a means to communicate with the ULS and provide power to the device as it rotates. An absolute position encoder is fixed to the motor shaft to provide accurate angle measurements to the MCU to calculate the vertical distance to the ground of the transmission line, details regarding the position encoder can be found in chapter 4. All parts are fastened to an aluminum channel which is mounted to the frame of the TLPT.

5.2 Motor & Controller

5.2.1 Design

Initial designs called for the ULS to scan the 110° field of view in an oscillating motion, scanning 55° on either size of the vertical axis. This motion was to be performed by a stepper motor and controlled by the MCU. After consulting with Dr. Ahmad Byagowi and brainstorming methods to control the motion and measure its position, it became apparent that this design posed some significant drawbacks. An oscillating motion would introduce slop in the planetary gears as the direction of rotation is changed.

To eliminate the undesirable effects of an oscillating motion, continuously rotating the ULS in one direction at a constant speed was determined to be the best alternative. Using a 12V brushed DC motor with an attached planetary gear set, the ULS is continuously rotated at an easily controllable speed. The advantages that a geared DC motor offers are easy control, ample torque for starting the rotation and low power draw. The motor selected also has dual internal ball bearings ensuring smooth, efficient operation over the lifetime of the TLPT. As seen in fig. 5.1, the 12VDC motor and the SyRen 10A motor controller are both mounted to the aluminum channel.
The SyRen motor controller was chosen because of its low cost, economical availability, compact size and ease of implementation. The SyRen motor controller was used on previous ECE 4600 Capstone projects and was an in-kind donation by the University of Manitoba student branch of Institute of Electrical and Electronics Engineers (IEEE). The motor controller has a 6-24VDC input range and is capable of 10A continuous output. The motor controller is more than capable of powering and controlling the DC motor chosen and has many operating modes. Using the selectable Dual In-Line Package (DIP) switches any of the 18 controllable input modes can be selected [12]. For the purposes of this project, operating mode 2 was chosen, where a 0V-5VDC analog signal corresponds to a 0-100% motor speed. Operating mode 1 was chosen because of the simplicity to interface with the MCU while leaving the other operating modes available for future improvements. An additional benefit of the SyRen 10A is its on-board overcurrent, thermal, and short circuit protection, making it the ideal motor controller for the TLPT. [12]

5.2.2 Implementation & Validation

The simplicity of the motor-drive assembly made it possible to test the design in stages. Prior to connecting the ULS mechanically or electrically to the assembly, the motor and controller were first tested to confirm they were operating according the manufacturers specifications and to ensure that the MCU could control the motor as desired.

To validate the motor and controller, the DC motor needed to be tested under nominal loading conditions to determine if the attached planetary gear set was operating as designed and if the two bearings supporting the rotating shaft were defective or not.
The motor was first tested under no load with the ULS not attached, then with the ULS attached. The data from this test can be seen in table 5.1. Both tests were performed at the nominal rotational speed of 32 RPM and for a total duration of one hour each with the measured current draw recorded every minute and averaged over the duration of the test. From these tests, it can be concluded the DC motor and motor controller are operating as designed and that the ULS does not add significant load to the motor-drive assembly, thus ensuring the DC motor will continue to work over the required lifetime of the TLPT.

<table>
<thead>
<tr>
<th>Loading Condition</th>
<th>Rated Current</th>
<th>Measured Current</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Load, 32RPM</td>
<td>540mA</td>
<td>420mA</td>
</tr>
<tr>
<td>Full Load, 32RPM</td>
<td>1.5A</td>
<td>534mA</td>
</tr>
</tbody>
</table>

### 5.3 Motor Drive Assembly

#### 5.3.1 Design

Using fig. 5.2 and table 5.2 for part descriptions it can be seen that the ULS is capable of rotating about the axis of the 1” shaft (Part-9). For the TLPT the 1” shaft rotates in one direction at a continuous speed by the DC motor (Part-1). The slip ring (Part-6) allows for the power and signal wires of the ULS to pass through the shaft as it rotates and connects to the microprocessor. An absolute position encoder (Part-2) is used to determine the position of the ULS, discussed further in chapter 4. The motor-drive shaft coupler and shaft-ULS mounting coupler (Part-4 and Part-7, respectively) were designed to be manufactured using 3D printing technology when research into similar components determined they could not be purchased within our budget. These parts were printed using ABS plastic.

In order to power and communicate to the ULS as it rotates, a connection method was required that would not induce noise into the circuits and remain operational in the -30 to +60°C temperature range. A 12-circuit slip ring (LPT025) produced by Jinpat Electronics Ltd. was determined to be the preferred connection method for the TLPT. The slip ring has features such as a 1” inner diameter, IP54 ingress protection rating, and 1mΩ electrical noise induced by the slip ring contacts.[13] Because of the importance of the slip ring to the operation of the TLPT, the characteristics previously discussed were carefully considered and verified, as the remaining components of the motor-drive assembly were dependent on the slip ring used.
Table 5.2: Motor-drive assembly part description

<table>
<thead>
<tr>
<th>Part Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DC Motor with Planetary Gear Set</td>
</tr>
<tr>
<td>2</td>
<td>CUI Absolute Position Encoder</td>
</tr>
<tr>
<td>3</td>
<td>Motor - Drive Shaft Coupler</td>
</tr>
<tr>
<td>4</td>
<td>1” Shaft - Motor Coupler</td>
</tr>
<tr>
<td>5</td>
<td>1” Bearing</td>
</tr>
<tr>
<td>6</td>
<td>12 Circuit Slip Ring</td>
</tr>
<tr>
<td>7</td>
<td>Shaft - ULS Mounting Coupler</td>
</tr>
<tr>
<td>8</td>
<td>ULS</td>
</tr>
<tr>
<td>9</td>
<td>1” Aluminum Shaft</td>
</tr>
<tr>
<td>10</td>
<td>Mounting Channel</td>
</tr>
</tbody>
</table>

The mounting hardware consists of off-the-shelf threaded fasteners. The two bearings (Part-5) share the same mounting configuration and fastening method as the mount for the DC motor (Part-1), simplifying the number and type of fasteners required. Fastening the ULS to the shaft-ULS mounting coupler (Part-7) and the mounting plate followed a similar methodology.

All parts were selected based on their availability, budgetary considerations, and most importantly, their robustness to continued use over the expected life of the TLPT. One important performance metric is the operating temperature range of the device as a
whole. Each part in the motor-drive assembly had to meet mechanical longevity requirements but also able to work reliably in the TLPT’s operating temperature range of -30 to +60°C.

### 5.3.2 Implementation & Validation

The completed motor drive assembly can be seen in fig. 5.3. All wiring was fastened to the aluminum channel using zip ties and heat shrink tubing to protect the wiring on sharp edges of the aluminum channel and aluminum shaft. To ensure that all fasteners will remain in place throughout the lifetime of the TLPT, thread locking fastening compound was used on all threaded fasteners. With a breakaway torque rating of 34Nm, vibrations induced by mechanical motion will not be strong enough to loosen any fasteners on the motor-drive assembly.[14]

![Figure 5.3: Motor-drive assembly](image)

To further validate the motor-drive assembly, testing on the slip ring consisted of transmitting a square wave of 5V\text{pk-pk} at 115.2kHz to simulate the data transmission baud rate of 115.2 kilobaud to the ULS. The test was performed with the slip ring both stationary and rotating at 27 RPM to validate the manufacturers claim of no detectable noise being introduced at rotational speeds below 100 RPM. As can be see in fig. 5.4, the square wave appears almost identical with almost no detectable noise for both the stationary and rotating scenarios. Clearly from this test, it can be concluded that digital communications will not be affected by noise induced by the slip ring rotation and the ULS will communicate at its intended baud rate through the slip ring.
The rotational speed of the motor was also verified, for which the rotary position encoder was used. This was done using an index pulse output that the encoder produces once per turn. Observing the index pulse with a logic analyzer during normal operation of the motor showed the time between pulses to be 2.766 seconds. This gives a motor rotational speed of

\[
\frac{1}{2.766 \text{ sec/rotation}} \cdot 60 \text{ sec} = 21.69 \text{ RPM} \quad (5.1)
\]

Finally, the motor-drive assembly was tested as a complete sub-system. While rotating at the intended operating speed of 21.69 RPM, the ULS was powered and made to output data to a personal computer. The purpose of this test was to ensure that all mechanical and electrical connections were made properly and to confirm the motor-drive assembly functions as the two previously discussed tests indicated it would. The assembly was observed during continuous operation for one hour while measuring current to the motor and monitoring communications from the ULS to ensure the sub-system was ready to be integrated as a completed system. For the duration of this test, communication with the ULS was maintained and the motor maintained the average loading current shown in table 5.1.
Chapter 6

Temperature Management

The TLPT will be under the effects of extreme temperature variations since it will be operating outdoors. Temperatures can vary from close to -50°C in the winter to +40°C in the summer. The need for a heating source in particular becomes apparent when looking at the operating temperature ranges of TLPT’s main components, as shown in table 6.1. In order to help keep the components of the TLPT within their rated operating temperatures, a heating element (HP05-1/08-24) and cooling fan (CHA8012BS-TA) were installed in the TLPT’s enclosure.

Table 6.1: Operating temperature ranges of components

<table>
<thead>
<tr>
<th>Component Name</th>
<th>Operating Temperature Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>ULS</td>
<td>-30.. + 60°C</td>
</tr>
<tr>
<td>Rotary Encoder</td>
<td>-40.. + 125°C</td>
</tr>
<tr>
<td>DC Motor</td>
<td>-10.. + 60°C</td>
</tr>
</tbody>
</table>

6.1 Design & Decision Reasoning

The cooling fan is a Superred CHA8012BS, selected due to economical availability and low current draw. The heating element is a DBK HP05-1/08-24 PTC element, with an operating voltage range of 12 to 30V and maximum power output of 15W[16]. This element was chosen due to its small size, low cost, simple operation, and adequate power output. The required power output for the element was determined by calculating the length time needed to heat up the air inside the enclosure by 50°C.\(^1\) The parameters required for these calculations are the specific heat of air, \(C = 1.006 \frac{kJ}{kg \cdot C}\), the density of air, \(0.036 \frac{kg}{ft^3}\), the volume of the enclosure as \(5ft^3\), and \(\Delta T = 50°C\).

\[
\text{Mass of air} = \text{Density} \cdot \text{Volume} = 0.036 \frac{kg}{ft^3} \cdot 5ft^3 = 0.18 \text{ kg}
\]

\(^1\)These calculations are assuming an adiabatic enclosure.
\[
\text{Energy} = \text{Specific heat} \cdot \text{Mass of air} \cdot \Delta T \\
= \frac{1.006 \text{ kJ}}{\text{kg} \cdot \text{C}} \cdot 0.18 \text{kg} \cdot 50^\circ \text{C} \\
= 9.054 \text{ kJ}
\]

This means that with a 15W heating element, it would take 10 minutes to increase the temperature inside the enclosure from -50°C to 0°C. Setting a minimum temperature limit (section 7.3) of -10°C, should increase the temperature to 0°C within 2 minutes.

### 6.2 Implementation & Validation

Through unit testing of the various components of the TLPT, it was found that the only component in need of cooling was the voltage regulator of the power supply. As such, the fan was mounted directly behind the power supply to blow turbulent air over the entire PCB. Since the voltage regulator is operational anytime the TLPT is powered, the fan was wired directly to the 12V unregulated source to ensure adequate cooling at all times.

As shown in fig. 6.1, the heating element is mounted directly to the aluminum mounting channel of the motor-drive assembly, in close proximity to the DC motor, which is the most temperature-sensitive component of the system. This was done to ensure that the motor remains within its operation range while taking advantage of the channel’s thermal conductivity to dissipate the heat throughout the enclosure. In addition, the fan of the power supply assembly has the dual effect of circulating the air inside the enclosure, further aiding in heat distribution.
The heating element has one terminal connected directly to the 12V power supply. The other terminal is connected to a control circuit on the MCU, which is described in section 7.3. Since the element’s maximum power output is 15W, then the maximum current through its wiring is $15 \div 12 = 1.25\text{A}$, which is well within the current carrying capability of the 18 American Wire Gauge (AWG) wire used[17].

The effectiveness of the heating element was intended to be tested by operating the TLPT inside an environmental test chamber. However, there was no opportunity to do so prior to the writing of this report, testing will be scheduled with Manitoba Hydro at a later date.
Chapter 7

Main Control Unit

The MCU is the central "brain" of the TLPT, overseeing all aspects of its day-to-day operation. At the heart of the MCU is an ST Nucleo-F303K8 microcontroller. The MCU also includes a temperature sensor for temperature management, a CAN transceiver for transmitting measurement data, and LED drivers for controlling the HMI. In addition, the MCU employs RS-232 and SPI protocols to interface with the ULS and rotary encoder, respectively. The MCU was implemented as a 2-layer PCB of dimensions 3.3 x 2.8 inches using Altium Designer, shown in fig. 7.1.

![The MCU, installed in the TLPT](image)

Figure 7.1: The MCU, installed in the TLPT

The following sections will only show select portions of the MCU’s circuitry and PCB layout. The full schematics and layout can be found in appendix A.
7.1 Microcontroller

7.1.1 Decision Reasoning

A few different microcontrollers were considered before settling on the Nucleo-F303K8. First, the microcontroller was required to meet the following basic criteria:

- RS-232 port for a connection with the ULS
- CAN capability for transmitting measurements to the DR
- SPI capability for reading from the rotary position encoder
- Some communications port for a temperature sensor

Then, a decision matrix (table 7.1) was created, using the grading weights of 1 to 5 from table 4.1 to indicate performance in that area.

Table 7.1: Microcontroller decision matrix

<table>
<thead>
<tr>
<th></th>
<th>Arduino</th>
<th>ST Nucleo</th>
<th>mbed</th>
<th>Switch Science</th>
<th>Texas Instruments</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Due</td>
<td>F303K8</td>
<td>LPC1768</td>
<td>LPC824</td>
<td>MSP432P401</td>
</tr>
<tr>
<td>Cost</td>
<td>2</td>
<td>5</td>
<td>1</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>Size</td>
<td>3</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>Clock Speed</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Memory</td>
<td>5</td>
<td>3</td>
<td>5</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>Communications</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>Support</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td><strong>Rank</strong></td>
<td><strong>24</strong></td>
<td><strong>25</strong></td>
<td><strong>23</strong></td>
<td><strong>17</strong></td>
<td><strong>20</strong></td>
</tr>
</tbody>
</table>

The Nucleo-F303K8 seemed naturally poised to play the part. Not only did it achieve the highest score in the decision matrix, it also met all of the technical requirements of the project without any unneeded features and with minimal external components required. Figure 7.2 shows the pin-out of the Nucleo F303K8.
7.2 Communications

7.2.1 Design & Decision Reasoning

The main task of the MCU’s Nucleo microcontroller is to connect and communicate with the different devices that it must monitor and control. The DS1621+ temperature sensor and AMT-20 rotary encoder do not require any additional hardware to communicate with the Nucleo. These components are discussed in chapters 4 and 6, respectively. The TLPT includes two devices which did not require any additional hardware beyond what is already included on the Nucleo. These are the DS1621+ temperature sensor and the AMT20 rotary encoder, which are discussed in sections 6 and 4, respectively.

The Nucleo also includes the hardware needed for the RS-232 interface to the ULS, with one minor exception. Preliminary testing of the ULS showed that its RS-232 signals are inverted. This was resolved by simply adding a dual-channel logic inverter (ON Semiconductor NL27WZU04DTT1G), so that both the RX and TX signals get inverted. Inverting the signals in software was also considered, although this proved
difficult due to the need to access and modify low-level registers from the very high-level mbed programming environment, which is discussed in section 7.5.

With regard to CAN communication, the Nucleo includes a CAN controller but not a CAN transceiver. As such, an MCP2562 CAN transceiver was used, along with some supporting circuitry. The MCP2562 was selected due to prior familiarity and because it includes an internal level shifter which allows for a bus voltage of 5V while interfacing with the Nucleo’s 3.3V signals.

Figure 7.3 shows the CAN transceiver circuitry. The bus termination resistor (R1) is an RC2012F121CS with a ±1% value tolerance in order to improve the accuracy of the bus impedance. D1 is a CAN bus protector\(^1\) to add additional protection from Electrostatic Discharge (ESD) and transients, since the TLPT will be operating in an electrically noisy environment.

\[\text{Figure 7.3: CAN controller and supporting circuitry}\]

### 7.2.2 Implementation & Validation

Since the Nucleo already includes the necessary hardware for RS-232, SPI, and I²C communications, there was little to no implementation or validation to be done in that regard.

The physical implementation of the CAN circuit is shown as a 2-dimensional Computer-Aided Design (CAD) render in fig. 7.4. The general approach for the MCU PCB layout was to keep as many features as possible on the top layer, in order to leave the bottom layer open for a ground plane. The two CAN traces were kept as close as possible to one another to aid in the effectiveness of the differential signals against the risk of crosstalk. All traces are 20 mil wide, with the minimum spacing between traces set to 10 mil.

\(^1\)More precisely, a dual bidirectional transient voltage suppressor device
The validation of CAN bus circuitry is a matter of transmitting data between the MCU and DR, which is covered in chapter 9.

# 7.3 Temperature Control

## 7.3.1 Design & Decision Reasoning

The MCU maintains the temperature inside the TLPT’s enclosure with the help of a temperature sensor, heating element, and driver circuitry, shown in fig. 7.5. The temperature sensor and driver circuitry are located directly on the MCU PCB, while the heating element is external to the MCU. More information on the heating element can be found in chapter 6.

The temperature sensor is a Maxim DS1621 with an integrated thermostat control output. Using an I²C interface, the DS1621 is programmed with a threshold temperature by the Nucleo during system initialization.

According to a Texas Instruments application report[19], the minimum acceptable pull-up resistance for an I²C bus is found by
\[ R_{p\text{ (min)}} = \frac{V_{cc} - V_{OL\text{ (max)}}}{I_{OL}} \] (7.1)

where \( V_{OL\text{ (max)}} \) is the maximum logical low output voltage, and \( I_{OL} \) is the logical low output current. From the I\(^{2}\)C specification [20], \( V_{OL\text{ (max)}} = 0.4 \text{V} \) and \( I_{OL} = 3 \text{mA} \). We then get

\[ R_{p\text{ (min)}} = \frac{5 - 0.4}{0.003} = 1533\Omega \] (7.2)

The difference between small and large pull-up resistor values is response time and power consumption [19]. Since the TLPT is powered from a direct 120VAC source, power consumption is not a large concern, and so it was preferred to attempt to maximize response time. A pull-up resistor value of 2k\(\Omega\) ensures fast response times without risking too low of a resistance due to unpredictable variances such as manufacturing tolerances and reduced resistance resulting from temperature changes.

Once the DS1621 has been programmed, any time it senses a temperature that is lower than the threshold temperature, its thermostat control pin (TOUT) automatically goes high. This in turn sends the Metal-Oxide-Semiconductor Field-Effect Transistor (MOSFET) (Q1) into saturation mode, thereby grounding the coil of the relay (K1), which thus grounds the floating terminal of the heating element, turning it on.\(^{2}\) Once the DS1621 senses that the temperature has increased sufficiently, TOUT

\(^{2}\)It should be noted that the relay K1 is not actually needed; the heating element could instead be directly driven by the Q1. However, the heating element was initially intended to be powered from an Alternating Current (AC) source, in which case the isolation of a relay would have been necessary.
goes low, turning the heating element off. More precisely, the DS1621 actually stores two different threshold temperatures: one for turning TOUT on and one for turning TOUT off. As such, any desired amount of hysteresis can be programmed by the user.

For the purposes of the TLPT, the lower and upper thresholds were set to -5°C and 0°C, respectively. These thresholds were selected based on the minimum operating temperature of the DC motor (-10°C), keeping in mind that the sensor reads ~5°C higher than the actual air temperature (as mentioned in section 6.2) and that the motor itself warms up during operation.

### 7.3.2 Implementation & Validation

The physical implementation of the temperature control circuit is shown as a 2-dimensional CAD render in fig. 7.6. As previously mentioned in section 7.2.2, keeping components and traces on the top layer allowed for a solid ground plane on the bottom layer. The temperature sensor is physically located directly underneath the Nucleo, in between its two mounting headers. This was done as a space-saving measure, however it was later determined that such an enclosed location for a temperature sensor is not ideal, as it reduces measurement precision of the actual air temperature.

![Figure 7.6: PCB layout of the MCU’s temperature control hardware](image)

All traces in this circuit are 20 mil wide, with the exception of the 12V and heating element connections, which are 30 mil wide for an additional factor of safety. Since the heating element’s maximum power output is 15W, the maximum current through these traces is $15 \div 12 = 1.25A$. The Saturn PCB Toolkit (previously shown in fig. 3.5) was used to verify that 30 mil wide traces are suitable for up to 3.06A of current.

The functionality of the temperature control circuit was verified by measuring the electrical continuity of the relay contacts as the air temperature around the sensor changed. The test was performed by heating the sensor from room temperature rather than cooling it to below freezing, since it is easier to produce heat than to remove it. The temperature sensor was configured with a high temperature threshold of 30°C and
a low temperature threshold of 28°C. At room temperature, the contacts of the relay were measured to be continuous, i.e. the heating element would be on, as intended for a "cold" scenario. The temperature sensor was then gradually heated with the help of a hot air gun. By continuing to measure the continuity of the relay’s contacts while observing the sensor’s temperature output, it was seen that the relay opened once the sensor’s temperature reached 30°C, i.e. the heating element would be turned off. The sensor was then allowed to cool back to room temperature, and the relay closed again when the sensor’s temperature read as 28°C. This procedure confirmed that the circuit functions as intended.

7.4 LED Control

7.4.1 Design & Decision Reasoning

The MCU includes a set of low-side driver circuits used to control a set of five indicator LEDs which serve as the TLPT’s HMI, which is shown in fig. 7.7. The conditions indicated by these LEDs are ULS error, rotary encoder error, CAN error, actively scanning, and one miscellaneous indicator that is reserved for troubleshooting purposes.

![Figure 7.7: The TLPT’s HMI](image)

Figure 7.8 shows one of the five LED driver circuits from the MCU. This is a simple low-side driver circuit using an N-type MOSFET. The two resistors connected to the gate (pin 4) of the MOSFET are to reduce ripple and discharge the gate when the control signal is not present. The control signal (MISC LED CTRL, in this case) is a 3.3V signal generated by the Nucleo and toggled in software. The drain (pins 5 & 6) of

---

Note that there are actually six LEDs on the HMI, but one of them is a power indicator, which is hardwired to the TLPT’s power supply and requires no additional control circuitry.
the MOSFET are connected to the cathode of the LED it controls. The anode of the LED is hardwired to a 12VDC source through a resistor. As such, the LED is switched on and off based on whether or not its cathode is grounded through the MOSFET. This simple operation, as well as low component cost, is the reason why low-side drivers were chosen over high-side drivers.

One of these indicators also relies on a hardware timer circuit, shown in fig. 7.9, in order for it to continuously blink, indicating that the TLPT is actively scanning for transmission lines. In this case, the control signal for the LED driver is the output of the timer circuit rather than a software-controlled signal from the Nucleo. The software-controlled signal (SCN INDC CTRL in fig. 7.9) is instead used to turn on the timer circuit.

This is a widely-known circuit using a 555 timer operating in Astable Mode, resulting in an oscillating output on pin 3. The frequency of the oscillation is given by eq. (7.3).
The result is that the scan indicator LED blinks with a frequency of approximately 3Hz. While the same result could be achieved through software, it would have required multi-tasking, reducing the Nucleo’s effectiveness at its main task of interpreting and transmitting measurements. This hardware-based design allows the Nucleo to initiate the blinking once during initialization, and not have to worry about turning stopping it again until some error occurs.

### 7.4.2 Implementation & Validation

Figure 7.10 shows the PCB layout of the LED drivers and hardware timer. Once again we see the ground plane on the bottom layer. Passive components were placed in neat rows with ordered designators for easy identification and quick assembly. The largest currents running through these circuits are the LEDs’ at 44mA. All of the traces are 20 mil in width.

![Figure 7.10: PCB layout of the MCU’s LED control hardware](image)

### 7.5 Firmware

#### 7.5.1 Design & Decision Reasoning

Firmware for the MCU was written in the mbed environment, which is a web-based Integrated Development Environment (IDE) that compiles C++ code from a
computer’s web browser. The included mbed libraries provide high-level support for compatible hardware, abstracting away most of the low-level aspects of microcontroller hardware. The major disadvantage of mbed is that the IDE does not support step-by-step debugging. However, it was decided that this flaw was worthwhile in exchange for the ability to get sub-systems up and running much more quickly. In addition, the mbed platform is very well supported, which lead to very few difficult bugs throughout the development process.

The MCU’s firmware can be split into two main portions: initialization and operation. The initialization flowchart is shown in fig. 7.11. The initialization sequence is responsible for initializing all necessary objects and variables, and configuring the ULS and temperature sensor. While all of these configuration settings are stored in non-volatile memory, they are nonetheless repeated upon powering the TLPT for redundancy.

![Figure 7.11: The MCU’s firmware initialization sequence](image)

The operation portion of the firmware follows a continuous loop that collects raw data from the ULS and rotary encoder, as shown in fig. 7.12. The first step is to wait for a
distance measurement from the ULS, which only arrives when a target is detected. If no errors occur, the measurements are transmitted to the DR.

![Flowchart](image)

Figure 7.12: The MCU’s firmware main loop

### 7.5.2 Implementation

See appendix B for full firmware code implementation.
Chapter 8

Data Receiver

The DR is closely related to the MCU in that it is also comprised of an ST Nucleo F303K8 microcontroller and a CAN transceiver. The DR serves as the interface between the TLPT and the database. The DR reads in measurements from the CAN bus (sent by the MCU) and converts them to Cartesian coordinates before passing them on to the database for storing.

Figure 8.1: The Data Receiver

8.1 Microcontroller

For simplicity, it was decided that the DR would make use of the same Nucleo microcontroller as the MCU. See section 7.1 for information on it.
8.2 Communications

8.2.1 Design & Decision Reasoning

The DR uses the same CAN hardware as the MCU, which was discussed in section 7.2.2. For completeness, the circuit schematic of the DR is shown in fig. 8.2.

As mentioned above, the DR’s primary function is to process and transmit data to a database. This may be done either directly through a USB connection from the DR to the computer, or through a Data Acquisition (DAQ) module, which then sends the data to the database. A 4-pin connector was used to breakout some general I/O pins to leave the possibility of transmitting measurement data to the DAQ module, either as digital or analog signals. The final method of communication from the DR to the database has been left to the discretion of our client and is outside of the scope of this project.

Figure 8.2: The DR circuit schematic

8.2.2 Implementation & Validation

Figure 8.3 shows the PCB layout of the DR as a 2-dimensional CAD render. Similar to the other PCBs of the TLPT, a ground plane was applied to the bottom layer. Some of the CAN hardware was placed on the bottom in order to help minimize the size of the DR.
8.3 Firmware

8.3.1 Design & Decision Reasoning

Similar to the MCU, the DR’s firmware was also developed with the mbed platform, as discussed in section 7.5.

The processing of measurements is being done by the DR rather than the MCU in order to leave the MCU focused on pre-processing and filtering measurements received from the ULS.

With each data transmission, the DR receives a TOF distance from the ULS to the transmission line (in millimeters) and a value of 0 to 4096 from the rotary encoder representing the angle at which the distance measurement occurred. The first calculation performed by the DR is to convert this angle value to degrees, using eq. (8.1).

\[
\theta = \frac{x \cdot 360}{4096} \quad (8.1)
\]

Once the DR has the angle in degrees, it then applies the sine and cosine trigonometric functions to calculate the horizontal and vertical distances of the conductor relative to the TLPT, that is

\[
\text{Height} = \sin(\theta) \cdot \text{TOF distance} \quad (8.2)
\]
\[
\text{Width} = \cos(\theta) \cdot \text{TOF distance} \quad (8.3)
\]
Since the TLPT is to be mounted on a raised platform\(^1\), the height of this platform is added to the calculated height, along with the height from the top of the platform to the lens of the ULS. Figure 8.4 shows a simplified diagram of this scenario.

\(^1\)The exact height of this platform was not known at the time of this writing but is expected to be around 1 meter.
Figure 8.5 shows a flowchart of the DR’s firmware program that was describe above.

![Flowchart of DR's firmware program](image)

Figure 8.5: The DR’s firmware program

### 8.3.2 Implementation

See appendix B.2 for full firmware code implementation.
Chapter 9

Data Transmission

The data collected by the TLPT is to be transmitted to the DR over a CAN bus over a distance of approximately 70m. This long transmission distance combined with the risk of electrical noise from the HV transmission lines introduces a considerable risk of interference in data transmissions. As a result, the method of data transmission was carefully considered and incrementally tested.

9.1 Design & Decision Reasoning

The two data transmission methods that were considered were CAN bus and optical fibre. Both options met the technical requirements of transmission distance, transmission speed, and noise immunity. However, optical fibre cable is considerably more costly than CAN cable, making CAN the data transmission method of choice for this project.

In order to reduce costs further, it was decided that a shielded CAT5 ethernet cable (Vericom MBW5M-04004DN) would be used rather than standard CAN cable. The 100Ω characteristic impedance of this CAT5 cable [21] is sufficiently close to the 120Ω impedance specified by the CAN standard that this was not expected to interfere with data transmission. The cable’s shielding is maintained both outside and inside the TLPT enclosure, and is grounded only at a single end in order to filter out environmental noise from the data signals.

9.2 Implementation & Validation

CAN bus data transmission was initially validated with the help of a sample program by transmitting dummy messages back and forth between the MCU and DR across a cable of 1m in length. This test program was left to run continuously for 30 minutes with no occurrence of transmission errors. The same test was repeated with cable lengths of 10m and then 100m, with both being successful.

Once a full integration of the TLPT’s sub-systems was completed, the full system was stress tested by operating the TLPT while in close proximity to electrical corona discharge in the University of Manitoba High Voltage Laboratory. As can be seen from fig. 9.1, the 100m long CAN cable was coiled up and placed directly underneath a HV source to incur the maximum effect of the corona.
The purpose of this test was to verify the consistency of data transmissions in two different scenarios: without corona discharge activity and with corona discharge activity in the range of 18 to 25kV. The TLPT was given a target located approximately 750mm away and 90° from horizontal and allowed to operate for a few minutes for each scenario. Table 9.1 shows the average, minimum, and maximum recorded values based on 90 data points. The test was determined to be successful since the values transmitted were relatively consistent in both cases, confirming the noise immunity of the overall system.

Table 9.1: HV environment data transmission test results

<table>
<thead>
<tr>
<th></th>
<th>No Corona</th>
<th></th>
<th>With Corona</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Distance (mm)</td>
<td>Angle (degrees)</td>
<td>Distance (mm)</td>
<td>Angle (degrees)</td>
</tr>
<tr>
<td>Avg.</td>
<td>793</td>
<td>87.59</td>
<td>772</td>
<td>87.96</td>
</tr>
<tr>
<td>Min.</td>
<td>588</td>
<td>84.90</td>
<td>631</td>
<td>84.90</td>
</tr>
<tr>
<td>Max.</td>
<td>957</td>
<td>92.02</td>
<td>954</td>
<td>91.93</td>
</tr>
</tbody>
</table>

The variation in measured distances seen in Table 9.1 is a result of time constraints which did not allow for the ULS to be properly configured for the given target, as discussed in chapter 2.
Chapter 10
Enclosure

10.1 Overview

The purpose of the frame and enclosure is to provide a rigid structure for the mounting of electronics and motor-drive assembly as well as provide environmental protection in all weather conditions. Both the frame and enclosure were designed using 3D CAD software to ensure all sub-systems will fit within the frame and to shorten manufacturing time. The approach taken to designing the physical components allowed the TLPT to be designed and manufactured with little material wasted during manufacturing.

10.2 Frame

10.2.1 Design

The frame remains static in all operating conditions and must provide a rigid structure to securely fasten all components to while meeting the performance metric of allowing the ULS to have at least 110° field of view. Many iterations of the frame were conceived to allow for the most compact design possible while meeting all required performance metrics. Initial designs were created using SolidWorks 3D CAD software to visualize all components and ensure that the ULS could rotate freely. Using SolidWorks allowed for many iterations to be created and tested virtually without adding significant time or cost to prototype each design.

The first iteration of the frame can be seen in fig. 10.1. An arched design was chosen to give a field of view of 150°, greatly exceeding the performance metric of 110°. A prototype was created using wood to mount the ULS and rotate by hand to verify the field of view was as designed. The arched design shown in fig. 10.1 was intended to have a very narrow opening cut into the enclosure for the ULS to measure through as discussed in chapter 2. It was quickly determined the arched design would not work as the window would have to be continuous curved piece of transparent plastic or glass to avoid reflections from the ULS.
From the aforementioned testing, it was determined that a simple rectangular frame would allow for the proper field of view while still allowing a window for the ULS to measure through. Measuring 22” x 24” x 12” (LxWxH), the redesigned frame met all of the required performance metrics while providing adequate room for the remaining sub-systems.
10.2.2 Implementation & Validation

The frame was manufactured using square steel tube welded at each joining surface. Care was taken to ensure that the frame remained square by clamping each steel tube and allowing time for each weld to cool to avoid warping. In fig. 10.3, the frame is shown with both bottom pieces of wood bolted on, the two side panels of the enclosure and the motor drive assembly in the initial testing stages. The inner piece of wood allows for mounting of all electronics while the outer pieces are part of the enclosure. Validation of the rectangular frame consisted of comparing measurements of the physical frame to the calculated dimensions from the 3D CAD model.

Figure 10.3: Rectangular frame realized

10.3 Environmental Protection

10.3.1 Design

Environmental protection of the TLPT includes protection from water, wind, snow, dust, and wildlife. Since the TLPT will be operated year-round, it must be designed to withstand all weather conditions. Along with protecting electrical and mechanical components, the enclosure provides a point to mount handles to move the device to move the device and a HMI panel to notify the user of the current state of operation of the TLPT.

Different methods to provide environmental protection from the elements were
investigated to meet the performance metrics. The grading weights from table 4.1 were used to systematically determine which enclosure type would be most suitable for our application. After researching different materials to manufacture the enclosure from it was determined that wood, steel sheet metal, aluminum sheet metal, and acrylic plastic sheets were the four most appealing choices. Each material has its own strengths and weaknesses with respect to the grading criteria. In order to quantitatively evaluate each material a decision matrix shown in table 10.1 was created to rank each material.

Table 10.1: Enclosure material decision matrix

<table>
<thead>
<tr>
<th></th>
<th>Wood</th>
<th>Steel</th>
<th>Aluminum</th>
<th>Acrylic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Availability</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>Manufacturability</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Ease of Implementation</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Thermal Insulation</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>Electric Conductivity</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td><strong>Rank</strong></td>
<td><strong>24</strong></td>
<td><strong>13</strong></td>
<td><strong>16</strong></td>
<td><strong>18</strong></td>
</tr>
</tbody>
</table>

10.3.2 Implementation & Validation

To secure the enclosure to the frame, grade 5, 0.25” bolts were fastened through the wooden enclosure and metal frame, as shown in section 10.3.2. To ensure that a water resistant seal was created, a bead of outdoor rated silicone caulking was placed along the outer face of the frame. When the enclosure panel is fastened to the frame it compresses the caulking to fill any voids when the wood panel. The same fastening procedure was repeated for each of the five sides of the frame. To seal the lid to the frame a thick bead of caulking was applied and once dry would be compressed as the lid is fastened the frame creating a seal. Silicon sealant was used on the AC power entry, CAN bus connection, HMI panel, and the acrylic window that the ULS measures through. In order to preserve the integrity of the wood panels over the lifetime of the device, acrylic latex outdoor paint was applied to the enclosure panels. Section 10.3.2 shows how the silicone sealant was applied to the frame prior to fastening an enclosure panel. Extra sealant was applied on the inside to ensure a water resistant seal as shown in Figure 10.4.
Figure 10.4: Enclosure panel sealing
Chapter 11

Conclusion

The project was a success in that the critical performance metrics were met as outlined in table 1.1. The TLPT is capable of measuring a field of view measuring 110° once every 3 seconds. As the ULS rotates through its field of view, it can successfully measure distances 10m to 40m away all while recording the angle at which each distance measurement occurred with a position accuracy of 0.27%. The TLPT has been tested to successfully transmit data up to 100m away while under the effects of corona from a transmission line as required by the project scope.

Two important performance metrics which require further validation are the operating temperature range of -30°C to +60°C and the ability to detect each of the six conductors of the transmission line. All components included in the TLPT are either rated for the proposed temperature range or are can be kept within their individual rated temperatures through the TLPT’s heat management system, but testing to validate this claim remains to be accomplished. Testing on the ULS has successfully confirmed objects smaller than a single conductor used on the Bipole III transmission line can be detected, however field testing of the device under the Bipole III line remains to be completed.

11.1 Future Recommendations

11.1.1 Adjustable Motor Speed

It would be possible in future iterations of the TLPT to include adjustable motor speed control to accommodate a variety of specific operating conditions. The recommended approach is to design a new motor speed control circuit between the MCU and motor controller. Rather than sending the 3.3V motor control signal from the MCU directly to the motor controller, this control signal would instead be used to activate the motor speed control circuit. The motor speed control circuit would send the actual motor control signal to the motor controller, which could be made to be adjustable from 0 to 5V with a potentiometer.

11.1.2 Temperature Management

The current temperature management system uses only a single temperature sensor located in a small area which is not necessarily representative of the actual air temperature throughout the enclosure. A more sophisticated temperature management
systems would allow for improved confidence in reliable operation throughout a wider temperature range. It is recommended to consider expanding the temperature management system to an array of temperature sensors. In this way, a more accurate reading of the enclosure’s air temperature can be achieved by taking the average of the temperature readings, or each sensor can be used to control individual heating elements for different areas of the enclosure.

11.1.3 System Portability

Another possible improvement that was identified was to make the TLPT more portable, so that it can be used to track any transmission line without the need for an on-site trailer. This could be done by including internal memory on the MCU to log system data and positional measurements locally, leaving them to be retrieved at a later time. The need for an AC power source could be alleviated by powering the TLPT with batteries or solar energy.
References


Chapter A

Electrical Schematics

Full schematics of the power PCB and DR are shown in ?? and ??, respectively.

Figure A.1: MCU hardware timer schematic (scanning indicator)

Figure A.2: MCU LED drivers schematic
Figure A.3: The MCU PCB layout
Figure A.4: MCU top-level schematic
Figure A.5: TLPT wire harness schematic
Chapter B
Firmware Code

B.1 Main Control Unit

main.cpp

```cpp
#include "CANnucleo.h"
#include "mbed.h"
#include "pinAssignments.h"
#include "laser.h"
#include "encoder.h"

const int PCBAUD = 921600; // Should be considerably higher than LASERBAUD
const int LASERBAUD = 115200; // 115200 is the maximum reliable baud rate
const int ENCDRFREQ = 1000000; // default was 1MHz
const int CANFREQ = 125000;
const float SHORTGATE = 500; // Filter out anything closer than 5000mm
const float LONGGATE = 50000; // Filter out anything beyond 50000mm
const unsigned int TX_ID = 0x101;

Laser laser;
int laserMeasurement;
float encoderAngle;
CANnucleo::CANMessage rxMsg;
CANnucleo::CANMessage txMsg;

/* PROTOTYPES */
void init();
void flushSerialBuffer();

int main()
{
    init();

    // Objects have to be declared and instantiated here
    // Start scanning
    // pc.printf("Start scanning\n");
    laser.sendCmd("$GO\"r\")
    scanIndicator = 1;
    motorControl = 1;

    while (1)
    {
```
```
laserMeasurement = laser.getDistance();

if (laserMeasurement != -1)
{
    encoderAngle = encoder.getAngle();
    errLedLzr = 0;
    scanIndicator = 1;
    if (encoderAngle != -1)
    {
        errLedEncdr = 0;

        if (! (laserMeasurement < SHORTGATE) && !(laserMeasurement > LONGGATE)) // check gates
        {
            txMsg.clear(); // clear Tx message storage
            txMsg.id = TX_ID;
            txMsg << laserMeasurement; // append first data item
            txMsg << encoderAngle; // append second data item (total data length must be <= 8 bytes!)
            pc.printf("Angle: %d\n", encoderAngle);
            if (can->write(txMsg)) // transmit message
            {
                errLedCan = 0;
                //pc.printf("CAN message sent\n");
            }
            else
            {
                // CAN Error
                errLedCan = 1;
                //pc.printf("Transmission error\n");
            }
        }
        else
        {
            // Unexpected encoder error
            errLedEncdr = 1;
        }
    }
    else
    {
        // Unexpected laser error
        errLedLzr = 1;
        scanIndicator = 0;
    }
}

void init()
{
    pc.baud(PCBAUD);
    pc.printf("Main init...\n");
    motorControl = 0;
    scanIndicator = 0;
    errLedCan = 0;
    errLedMtr = 0;
    errLedLzr = 0;
    errLedEncdr = 0;
    laserMeasurement = 0;
    encoderAngle = 0;
```cpp
void flushSerialBuffer(void)
{
    char char1 = 0;
    while (pc.readable()) {
        char1 = pc.getc();
    }
}

Laser.cpp

#include <stdlib.h>
#include "mbed.h"
#include "laser.h"

Laser::Laser(int LASERBAUD, Serial* pc, DigitalOut* scanIndicator,
              DigitalOut* errLedLzr) : serialPort(PA9, PA10)
{
    laserPC = pc;
    // laserPC->printf("Initialize laser...\n");
    ledScan = scanIndicator;
    ledErr = errLedLzr;
    *ledScan = 0;
    *ledErr = 0;
    inputBuffer = (char*) malloc(24); // Largest possible message size is 24 bytes
    serialPort.baud(LASERBAUD);
    initLaser();
}

void Laser::cmd(char* command, char* expectedResponse)
{
    sendCmd(command);
    // laserPC->printf("Response: %s\n", inputBuffer);
    if (strcmp(inputBuffer, expectedResponse) != 0)
    {
        *ledScan = 0;
        *ledErr = 1;
        // laserPC->printf("LZR CMD ERR: sent %s and received %s\n", command,
        // inputBuffer);
        // flushSerialBuffer();
        // exit(EXIT_FAILURE);
    }
}

void Laser::sendCmd(char* command)
{
    if (serialPort.writeable())
    {
        serialPort.printf(command);
        getResponse();
    }
```
Checks the laser's serial port for a response
Puts the result in the inputBuffer variable

```cpp
void Laser::getResponse()
{
    memset(inputBuffer, '\0', 24);
    responseChar = 0;
    responseAck = false;

    while (!responseAck)
    {
        if (serialPort.readable())
        {
            responseChar = serialPort.getc();
            if (responseChar == '$')
            {
                int index = 0;
                while (responseChar != '\r')
                {
                    inputBuffer[index] = responseChar;
                    index++;
                    responseChar = serialPort.getc();
                }
                responseAck = true;
            }
        }
    }
}
```

```cpp
int Laser::getDistance()
{
    distance = -1;
    char distanceChars[8];
    int i;

    getResponse();
    // laserPC->printf("Response: %s\n", inputBuffer);
    if (strcmp(inputBuffer, "$ER,4") != 0) // If a target is found, i.e.
        // not looking at open sky
        { // Remove the first 4 chars from the response to get numerical
            for (i = 4; i < strlen(inputBuffer); i++)
            {
                distanceChars[i-4] = inputBuffer[i];
            }
            distanceChars[i] = '\0';
            distance = strtol(distanceChars, NULL, 10); // Convert measured
            // value to an int
        }
    else if (strcmp(inputBuffer, "$ER,4") == 0) // If no target is found, i.e.
        // looking at open sky
        {
            distance = 999; // A value of 999 indicates open sky (design
            // decision)
        }
```
// If any other unexpected error occurs, then –1 is returned by default
return distance;
}

/**
Sets laser parameters according to what the ULS GUI does when you
connect to the laser.
Shouldn't technically be required since settings are stored in non-
volatile memory
But errors often occur while taking measurements if this is skipped
*/

void Laser::initLaser()
{
  cmd("$ST\r", "$OK"); // Stop ULS in case it was already in measurement
  //cmd("$BR,0,115200\r", "$OK");
  //serialPort.baud(115200);
  cmd("$US\r", "$US,1"); // Check laser status
  // DETECTION MODE INIT
  cmd("$MM,3\r", "$OK"); // Set measurement mode
  cmd("$SU,1\r", "$OK"); // Set measure units to meters
  cmd("$PL,0\r", "$OK"); // Set power level to high
  cmd("$FL,0\r", "$OK"); // Disable cooperative target
  cmd("$TE,0\r", "$OK"); // Disable RS-485 bus termination
  cmd("$EG,4\r", "$OK"); // Enable short gate for some reason
  cmd("$CG,0\r", "$OK"); // Disable gates
  cmd("$OF,0.000\r", "$OK"); // Set offset
  cmd("$MP,0\r", "$OK"); // Minimum pulse width rejection
  cmd("$XP,6000\r", "$OK"); // Pulse width rejection
  cmd("$PF,4500\r", "$OK"); // Set PRF
  cmd("$LA,1\r", "$OK"); // Set to absolute detection mode
  cmd("$TP,1.000\r", "$OK"); // Set trip point
  cmd("$STT,2BF20\r", "$OK"); // Set trip timeout
  cmd("$SMX,20\r", "$OK"); // Set max false pulses
  cmd("$CT,20\r", "$OK"); // Set number of valid pulses before a
  signal change can occur
  cmd("$FT,500\r", "$OK"); // Set flyer trap
  cmd("$TB,0\r", "$OK"); // Disable time between events
  cmd("$CO,1\r", "$OK"); // Enable continuous mode
  cmd("$MA,0\r", "$OK"); // Set to idle on power up
  //cmd("$PT,1\r", "$OK"); // Turn on pointer
}

void Laser::flushSerialBuffer(void)
{
  char char1 = 0;
  while (laserPC->readable())
  {
    char1 = laserPC->getc();
  }
}
Encoder.cpp

```cpp
#include <stdlib.h>
#include "mbed.h"
#include "encoder.h"

DigitalOut encdrSel(PA_4);
Encoder::Encoder(int ENCDRFREQ, Serial* PC, DigitalOut* errLedEncdr) : spi(PA_7, PA_6, PA_5)
{
    encdrPC = PC;
    //encdrPC->printf("Initialize encoder...

    ledErr = errLedEncdr;
    *ledErr = 0;
    angle = 0;
    encdrSel = 1;
    spi.format(8,0);
    spi.frequency(ENCDRFREQ);
}

int Encoder::getAngle()
{
    angle = -1;
    int response = 0;
    int msb = 0;
    int lsb = 0;
    bool gotAngle = false;

    //encdrPC->printf("Getting angle...

    while (!gotAngle)
    {
        cmd(0x10, &response); // Request position
        response = cmd(0x00, &response); // Get response from position

        while (response == 0xA5)
        { // While encoder not ready, send no operation command
            cmd(0x00, &response);
        }

        if (response == 0x10)
        { // Receiving 0x10 back means the encoder is ready, and the next two bytes will be the position
            msb = cmd(0x00, &response);
            lsb = cmd(0x00, &response);

            angle = (msb << 8) | (lsb & 0xff);
            gotAngle = true;
        }

        return angle;
    }

int Encoder::cmd(int msg, int* response)
{
encdrSel = 0;
*response = spi.write(msg);
wait_us(10); // Delay might have to be increased if bus frequency is lowered
encdrSel = 1;
return *response;

void Encoder::flushSerialBuffer(void)
{
    char char1 = 0;
    while (encdrPC->readable())
    {
        char1 = encdrPC->getc();
    }
}
B.2 Data Receiver

```c
#include "CANnucleo.h" //CANnucleo only supports up to version 127 of mbed
#include "mbed.h"
#include "pinAssignments.h"

/* CONSTANTS */
const int PCBAUD = 921600;
const unsigned int RX_ID = 0x101;
const int ENCODER_RESOLUTION = 4096;
const float GND_TO_LZR_HEIGHT = 1500; // in ULS units, currently mm

/* VARIABLES */
CANnucleo::CANMessage rxMsg;
CANnucleo::CANMessage txMsg;
volatile bool msgAvailable;
int distance;
int angleRaw;
float angle;
DigitalOut receivingMsg(LED1); // LED blinks to indicate a msg is being processed
float xPosition;
float yPosition;

/* PROTOTYPES */
void onMsgReceived();
void toggleOutput(DigitalOut output);

int main()
{
    pc.baud(PCBAUD);
    CANnucleo::CAN* can = new CANnucleo::CAN(PA11, PA12); // (RX, TX)
    can->frequency(125000);
    can->attach(&onMsgReceived); // attach 'CAN receive-complete' interrupt handler
    msgAvailable = false;
    receivingMsg = 0;

    // Start reading
    pc.printf("Start reading\n");

    while (1)
    {
        if (msgAvailable)
        {
            msgAvailable = false;
            can->read(rxMsg);
            toggleOutput(receivingMsg);

            if (rxMsg.id == RX_ID)
            {
                toggleOutput(receivingMsg);
                rxMsg >> distance; // extract first data item
                rxMsg >> angleRaw; // extract second data item
                angle = (float)(angle*360) / (float)ENCODER_RESOLUTION;
                // Convert raw angle value to degrees
                toggleOutput(receivingMsg);
            }
        }
    }
}
```
if (angle > 90)
{
    // The detected conductor is on the "secondary" side
    angle = 180 - angle;
}
toggleOutput(receivingMsg);

// Calculate X-Y position
yPosition = sin(angle) * distance;
toggleOutput(receivingMsg);
xPosition = (cos(angle) * distance) + GND_TO_LZR_HEIGHT;
toggleOutput(receivingMsg);

// Transmit position
pc.printf("%f\n", xPosition);
toggleOutput(receivingMsg);
pc.printf("%f\n", yPosition);
toggleOutput(receivingMsg);

receivingMsg = 0;
}
}

void onMsgReceived()
{
    msgAvailable = true;
}

void toggleOutput(DigitalOut output)
{
    if (output == 1)
    {
        output = 0;
    }
    else
    {
        output = 1;
    }
}