

Sensor for Measuring Chain Wear

Adam G, Tianhao L, Corbin K, Kai R

Table of Contents:

Table of Contents:	2
Section 1 - Overview	6
1.1 - Executive Summary:	6
1.2 - Team Contacts and Protocols:	6
1.3 - Gap Analysis:	8
1.4 - Timeline/Proposed Timeline:	9
1.5 - References and File Links:	10
References:	10
File Links:	10
1.6 - Revision Table:	10
Section 2 - Impacts and Risks	11
2.1 - Design Impact Statement:	11
2.2 - Risks:	12
2.3 - References and File Links:	14
References:	14
2.4 - Revision Table	14
Section 3 - Top-Level Architecture	16
3.1 - Block Diagram:	16
3.2 - Block Descriptions:	17
3.3 - Interface Definitions:	21
3.4 - References and File Links:	25
3.5 - Revision Table:	25
Section 4 - Block Validations	26
4.1 - Power Converter:	26
4.1.1 - Description:	26
4.1.2 - Design:	27
4.1.3 - General Validation:	29
4.1.4 - Interface Validation:	29
4.1.5 - Verification Process:	32
4.1.6 - References and File Links	33
4.1.7 - Revision Table	34
4.2 - Power Filter:	35
4.2.1 - Description:	35
4.2.2 - Design:	35
4.2.3 - General Validation:	37
4.2.4 - Interface Validation:	37
4.2.5 - Verification Process:	39
4.2.6 - References and File Links	40

4.2.7 - Revision Table	41
4.3 - Enclosure:	42
4.3.1 - Description:	42
4.3.2 - Design:	42
4.3.3 - General Validation:	43
4.3.4 - Interface Validation:	44
4.3.5 - Verification Process:	45
4.3.6 - Reference and File Links	46
4.3.7 - Revision Table	46
4.4 - Software/Code:	47
4.4.1 - Description:	47
4.4.2 - Design:	47
4.4.3 - General Validation:	51
4.4.4 - Interface Validation:	51
4.4.5 - Verification Process:	52
4.4.6 - Reference and File Links	53
4.4.7 - Revision Table	54
4.5 - CAN Controller:	55
4.5.1 - Description:	55
4.5.2 - Design:	55
4.5.3 - General Validation:	58
4.5.4 - Interface Validation:	58
4.5.5 - Verification Process:	61
4.5.6 - Reference and File Links	62
4.5.7 - Revision Table	62
4.6 - Microcontroller:	63
4.6.1 - Description:	63
4.6.2 - Design:	63
4.6.3 - General Validation:	65
4.6.4 - Interface Validation:	66
4.6.5 - Verification Process:	68
4.6.6 - Reference and File Links	68
4.6.7 - Revision Table	69
4.7 - Chain Measurement Sensor:	70
4.7.1 - Description:	70
4.7.2 - Design:	70
4.7.3 - General Validation:	72
4.7.4 - Interface Validation:	72
4.7.5 - Verification Process:	76
4.7.6 - Reference and File Links	76
4.7.7 - Revision Table	76

4.8 - Final PCB:	77
4.8.1 - Description:	77
4.8.2 - Design:	77
4.8.3 - General Validation:	79
4.8.4 - Interface Validation:	79
4.8.5 - Verification Process:	81
4.8.6 - Reference and File Links	82
4.8.7 - Revision Table	82
Section 5 - System Verification Evidence	83
5.1 - Universal Constraints:	83
5.1.1 - The system may not include a breadboard:	83
5.1.2 - The final system must contain a student designed PCB:	83
5.1.3 - All connections to PCBs must use connectors:	84
5.1.4 - All power supplies in the system must be at least 65% efficient:	84
5.1.5 - The system may be no more than 50% built from purchased 'modules':	85
5.2 - System Requirements:	86
5.2.1 - PCB:	86
5.2.1.1 - Project Partner Requirement:	86
5.2.1.2 - Engineering Requirement:	86
5.2.1.3 - Verification Process:	86
5.2.1.4 - Testing Evidence:	86
5.2.2 - Power Input:	87
5.2.2.1 - Project Partner Requirement:	87
5.2.2.2 - Engineering Requirement:	87
5.2.2.3 - Verification Process:	87
5.2.2.4 - Testing Evidence:	87
5.2.3 - Project Partner Checkoff:	87
5.2.3.1 - Project Partner Requirement:	87
5.2.3.2 - Engineering Requirement:	87
5.2.3.3 - Verification Process:	87
5.2.3.4 - Testing Evidence:	88
5.2.4 - Robust Enclosure:	88
5.2.4.1 - Project Partner Requirement:	88
5.2.4.2 - Engineering Requirement:	88
5.2.4.3 - Verification Process:	88
5.2.4.4 - Testing Evidence:	89
5.2.5 - Sensor Accuracy:	89
5.2.5.1 - Project Partner Requirement:	89
5.2.5.2 - Engineering Requirement:	89
5.2.5.3 - Verification Process:	89
5.2.5.4 - Testing Evidence:	89

5.2.6 - Signal Output:	89
5.2.6.1 - Project Partner Requirement:	89
5.2.6.2 - Engineering Requirement:	90
5.2.6.3 - Verification Process:	90
5.2.6.4 - Testing Evidence:	90
5.2.7 - System Cost:	90
5.2.7.1 - Project Partner Requirement:	90
5.2.7.2 - Engineering Requirement:	90
5.2.7.3 - Verification Process:	90
5.2.7.4 - Testing Evidence:	91
5.2.8 - Water Resistance:	91
5.2.8.1 - Project Partner Requirement:	91
5.2.8.2 - Engineering Requirement:	91
5.2.8.3 - Verification Process:	91
5.2.8.4 - Testing Evidence:	91
5.3 - References and File Links:	91
5.4 - Revision Table:	91
Section 6 - Future Recommendations	93
6.1.1. - Technical recommendations:	93
6.1.2. - Global impact recommendations	94
6.1.3. - Teamwork recommendations	94
6.2 - Project Artifact Summary with Links	94
6.3 - Presentation Materials	95

Section 1 - Overview

1.1 - Executive Summary:

The goal of this project is to create a chain elongation monitor for the Hyster-Yale Group that will automatically detect a worn lift chain and notify an operator to replace said chain. Chains, especially lifting chains, need to be frequently inspected to ensure they meet specifications. A 3% increase in a chain's length can reduce its lifting capacity by 15% [1]. This can become a dangerous hazard if out-of-spec chains are not caught.

Currently, this inspection is either done by hand or by expensive third party devices, thus the creation of a cheap electronic alternative is very desirable. We plan to use three pairs of IR LEDs and IR receivers that will act as our sensors. These pairs will be situated in such a way that the IR "beam" will break when a chain link passes the sensor and reconnect in the gap between chain links. One pair will be used to measure the speed, while the other two will measure the "stretch" of the chain by recording the time difference between activations and calculating a length measurement.

In addition to the sensors, the device will consist of a power circuit to convert a Forklift's 24V power into 5Vs, a CAN communication circuit, and a microcontroller to calculate the elongation based on the sensors and send those measurements via the CAN Circuit to the Hyster Vehicle. We chose an ATtiny814 for our microcontroller as it is a relatively small and cheap microcontroller and is capable of performing necessary functions, such as basic calculations and time measurement.

Our project partner has specified that the device needs to meet four requirements:

- 1) The device needs to measure chain "stretch" (wear) within $\pm 5\%$ of the actual measurement.
- 2) The device, specifically the electronics, need to cost less than \$15 or less at a production quantity of 15,000/yr
- 3) The device need to be robust enough to survive abrasive (high chain wear) applications
- 4) The device needs to communicate using the CAN communication protocol

At the end of our project, we were able to meet 3 of the 4 project partner requirements with the accuracy requirement being the one that was not met. This is mostly due to not having enough time to finish experimenting and debugging the code. With better time management and planning, we would have been able to finish the development of our code and could have met the final requirement.

1.2 - Team Contacts and Protocols:

Team Contacts:

Kai Roy, royka@oregonstate.edu

Tianhao Lin, lintia@oregonstate.edu
 Adam Grzelewski, grzelewa@oregonstate.edu
 Corbin Krecklow, kreckloc@oregonstate.edu

Team Roles:

Kai Roy - Sensor specialist/PCB designer
 Tianhao Lin - CAN Controller programmer/Microprocessor specialist
 Adam Grzelewski - Power converter/filter designer
 Corbin Krecklow - Programmer/Enclosure designer

Protocols:

Topic	Protocol	Standard
Taking Breaks	To prevent burn out, we will take mandatory breaks.	A break must be taken after two hours of work.
Communication and Meetings	We will use Discord for general communication i.e research documents, virtual meetings between group members, etc. We will use email and text for more important/immediate communication	We will have at least two group meetings a week, one physically in Dear 211 and one during/after the Friday Lecture. We will communicate any change in availability for meetings through the aforementioned communication systems. Otherwise it is expected that each member will be on time for every meeting
Task Management	We will use ClickUp as our main task/project management software to keep track of what needs to be done, what is currently being worked on, and what has been completed	Every meeting we will discuss the tasks we have been working on, and updating the ClickUp with in Progress notes and editing tasks status.
Work Distribution/Group Roles	General Weekly Roles: <ul style="list-style-type: none"> ● Note taker Permanent Roles: <ul style="list-style-type: none"> ● ClickUp Organizer (Kai) ● Partner Point of Contact (Corbin) 	Weekly Roles will be defined during the Monday Meetings Permanent Roles will have tasks defined to be completed during the week. If needed, the group will help the individual accomplish their project goal.
Work Quality	We will not submit anything for a group grade that we would not feel comfortable submitting for ourself	Work standards should be maintained from personal work to group work. If work quality from an individual is found lacking, a

		conversation will be had to determine if there are external factors at work and necessary assistance will be provided.
Work Submission Deadlines	Start work early and be proactive about deadlines. However, submitting an assignment on the due date is acceptable.	We will maintain a group agenda/calendar that is frequently updated and accessed group-wide.
Vacations/Sickness	Sickness is an unavoidable factor that can hinder progress. Vacations should be planned in advance and must notify the rest of the group so that work does not fall behind.	We will notify other members of the group of vacation plans two weeks in advance and, in the case of sickness immediately.

Table 1.2.1 - Table of the Team Protocols

Throughout the course of our project, we met with our project partner on average at least once a month over zoom meetings, though a few members of the team were able to attend an in-person tour of the facility at Hyster in January. Our project partner was able to directly assist us by providing advice and additional information as requested throughout the project development period and during the testing phase. He was able to provide a “new” and “used” chain for the purposes of testing the sensor as well as a frequency generator for testing the PCB.

1.3 - Gap Analysis:

The reason our project exists is because lift chains are in constant use for raising and lowering loads in a lift truck (forklift) mast and should be replaced when they reach a certain amount of wear. The rate of wear is dependent on environment and maintenance, so it’s difficult to state a specific time interval for replacement. Being able to easily detect this wear and communicate it to the operator via the display would help ensure chains are replaced at the appropriate service interval for the customer application.

Some requirements that we need to fulfill for this project include measuring chain stretch with an accuracy of +/- 5%, robust enough to survive high chain wear applications, output a 5V analog or digital signal, and cost no more than 10 dollars for a production quantity of 15,000/yr [2].

According to USA Roller Chain and Sprockets, a chain wear gauge prevents downtime by alerting you when the chain needs to be replaced, prevents financial cost and stress involved with downtime, ease of mind having a monitor check the chain for you, and the convenience of plugging in a monitor directly from the box with no calibration, programming, etc [3]. Our end users and stakeholders will be only Hyster and no other affiliates.

1.4 - Timeline/Proposed Timeline:

TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION	PCT OF TASK COMPLETE	Fall Term														
						1	2	3	4	5	6	7	8	9	10					
Term 1																				
Team work Definition and Documentation	Team	9/26/22	10/21/22	25	100%															
Contact Partner	Corbin K	10/3/22	10/14/22	11	100%															
Section 1 Documentation	All	10/10/22	10/14/22	4	100%															
Microcontroller Research + Software	Tianho L	10/7/22	10/21/22	14	100%															
Research/Create Chain Measuring Methods	All	10/24/22	10/28/22	4	100%															
Section 2 Documentation	All	10/31/22	11/2/22	2	75%															
Designing Prototype	All	10/31/22	11/18/22	18	0%															
Visit Hyster	All	11/7/22	11/11/22	4	0%															
Tech Demo work	All	11/7/22	12/9/22	32	80%															
Sensor Research	All	11/7/22	11/18/22	11	50%															
Finalize Sections 1 & 2	All	11/14/22	11/18/22	4	100%															
Prototype Solutions and Ideas	All	11/14/22	11/25/22	11	20%															
Testing Prototype	All	11/14/22	12/9/22	25	0%															

Fig. 1.4.1 Gantt Chart for Term 1 of ECE 44x

TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION	PCT OF TASK COMPLETE	Winter Term														
						1	2	3	4	5	6	7	8	9	10					
Term 2																				
Preliminary System Testing	All	1/9/23	2/24/23	45	0%															
Testing Prototype Solutions	All	1/16/23	1/27/23	11	0%															
Order PCB	All	1/30/23	2/10/23	10	0%															
3D Print Chassis	All	1/30/23	2/10/23	10	0%															
System-Level Testing	All	2/6/23	3/17/23	41	0%															

Fig. 1.4.2 Gantt Chart for Term 2 of ECE 44x

TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION	PCT OF TASK COMPLETE	Spring Term														
						1	2	3	4	5	6	7	8	9	10					
Term 3																				
Final Fixes	All	4/3/23	4/28/23	25	0%															
Project Poster	All	5/1/23	5/12/23	11	0%															
Closing Documentation	All	5/1/23	6/9/23	38	0%															

Fig. 1.4.3 Gantt Chart for Term 3 of ECE 44x

1.5 - References and File Links:

References:

- [1] “The importance of measuring leaf chain for wear,” Leaf Chain, <https://www.leafchain.com/knowledge-hub/the-importance-of-measuring-leaf-chain-for-wear> (accessed May 13, 2023).
- [2] EECS Project Portal, “Lift Chain Stretch Monitor,” *eecs.oregonstate.edu*, [Online]. Available: <https://eecs.oregonstate.edu/capstone/submission/pages/viewSingleProject.php?id=7vaCla2dTeWGxa9d>. [Accessed Oct. 3, 2022].
- [3] USA Roller Chain and Sprockets, “Digital 12B Roller Chain Wear Gauge | Chain Monitoring System,” *usarollerchain.com*, [Online]. Available: <https://www.usarollerchain.com/Digital-12B-Roller-Chain-Wear-Gauge-p/4947-40008851-ccm-12b.htm>. [Accessed Oct. 3, 2022].

File Links:

44x Gantt Chart:

https://docs.google.com/spreadsheets/d/1VC-Uo-bYTamijoSm4TDXBItY_ATqIW1UT4FlpQXtkyl/edit?usp=sharing

1.6 - Revision Table:

Revision Number	Revision Date	Description	Revision Author
1	10/14	Initial Revision	Kai, Corbin, Adam, Tianhao
2	5/13	Update Executive Summary to reflect the current state of the project	Kai, Corbin

Section 2 - Impacts and Risks

2.1 - Design Impact Statement:

Our choice of how to best use our project has its equal trade offs. We considered failures that can occur to a forklift chain other than chain elongation/wear, and the safety hazards such as injury and possible death if a chain failure occurs [3]. Also important to consider is the welfare impacts of creating our own device versus continuing to use a third party device for chain elongation/wear monitoring. Certain communities could be affected from mining during the manufacturing process of our design [4]. Furthermore, research into the environmental impact of creating semiconductors revealed that our potential carbon footprint will be deeper than we previously believed [5]. We also discussed how Hyster's current sensors come from expensive manufacturers and how our design will impact not only Hyster, but the rival manufacturing companies that will no longer see some business due to our design. Finally, we discussed the trend of forklift production going down due to the influence of covid-19 and how this could affect the production of chain sensors as well [6].

Some serious considerations with public health, safety, and welfare impacts is using open source versus closed source code and other safety hazards that occur with forklift chains besides chain elongation. The recommended use of code for this project will be using closed source software because you will have to sign an NDA (Non-Disclosure Agreement) with Hyster-Yale which provides you software from Hyster-Yale except the code cannot be shared with the public or any other drafts and designs relating to any of their products. We also recommend investigating if you have spare time on the project to account for protruding or loose pins, plate cracking, misalignment due to worn pins, and broken links because these cases can cause harm worse than chain elongation. We recommend adding additional sensors to detect protruding or loose pins, plate cracking, and misalignment.

Some serious considerations with cultural and social impacts is limiting the use of mining for resources to make our design because of the pollution it causes on the environment and the displacement of aboriginal communities where mining is taking place. We recommend to continue using these materials as there are not too many alternatives if not at all but to limit the amount of waste that is used to produce concept models and the final product. Also, to recognize communities that are being affected by mining activities.

Some serious considerations with environmental impacts is the manufacturing and disposal of electronic devices and components. The recommendation here is to limit the amount of electronic waste since manufacturing will continue by planning ahead on schematic designs and simulations to verify that designs will work when produced to reduce the amount of electronic waste that will occur when producing prototypes and final design. Reducing the amount of waste will also limit the amount of errors in the design process.

2.2 - Risks:

Risk ID	Risk Description	Risk Category	Risk Probability	Risk Impact	Performance Indicator	Action Plan
R1	Team member needs to take time for personal matters unexpectedly	Organizational	Med	Med	Team member sends short notice of absence	<ul style="list-style-type: none"> - Gather time-sensitive materials that the team member had - List work team member had to complete and make plan to delay/adjust timelines - Redistribute work for expected time of absence
R2	Unable to order parts	Organizational	Low	High	Lack of parts	<ul style="list-style-type: none"> - Look for parts that match existing footprints - If no parts exist, redesign board and potential splice in a part temporarily to continue work/testing -
R3	Prototype gets water damaged during field test	Technical	Med	Med	Electronics board gets fried	<ul style="list-style-type: none"> - Ensure that we have a second/backup device and/or spare parts ready to swap in. - Bring any necessary tools for making quick repairs or replacements on hand during field tests.
R4	PCB breaks	Technical	Med	Med	The PCB is damaged or there are logic errors	<ul style="list-style-type: none"> - Check for errors and try to fix if possible - Order a new PCB board
R5	Programming error	Technical	High	Low	Encounter some unexpected bugs when	<ul style="list-style-type: none"> - Debug by consulting the relevant datasheets or references. - Seek help from someone familiar with the relevant programming language.

					writing the program	
R6	Enclosure breaks	Technical	Low	Med	Enclosure damage due to friction with the chain	<ul style="list-style-type: none"> - As funds permit, use enclosures made of better materials. - Reduce friction by changing the shape of the design.
R7	Insufficient sensitivity of the sensor	Technical	Low	Med	Unable to accurately measure the length of the chain	<ul style="list-style-type: none"> - Choose a sensor with higher sensitivity. - Design different computing methods to suit the sensor.
R8	Documentation/schematics are corrupted or lost	Organizational	Low	High	Some documents or schematics are lost due to any unexpected reasons	<ul style="list-style-type: none"> - Make sure to back up all important documents. - For lost electronic data, some software can be used to try to recover them.

Table 2.2.1 - Table of Potential risks and their probability and impacts

2.3 - References and File Links:

References:

- [1] Leavitt Machinery, "Leavitt Basics: How to Measure Forklift Forks and Chains for Wear", *Leavitt Machinery*, Dec. 20, 2018. [Online Video]. Available: https://www.youtube.com/watch?v=3qw-x4oK_7I, [Accessed Oct. 26, 2022].
- [2] Rod Reinholdt, "How To Implement An Effective Chain-Wear Monitoring Program". QCTL Inc., 2022. Available: <https://www.reliableplant.com/Read/28623/chain-wear-monitoring>. [Accessed Oct. 26, 2022].
- [3] Hodson. Jim, "How to Inspect Forklift Chains and Know When to Replace Them" *tmhnc.com*, Mar. 14, 2017. [Online]. Available: <https://www.tmhnc.com/blog/how-to-inspect-forklift-chains-and-know-if-forklift-chain-is-worn-out>. [Accessed Nov. 4, 2022].
- [4] Gibson, Ginger, and Jason Klinck. "Canada's resilient north: the impact of mining on aboriginal communities." *Pimatisiwin* 3.1 (2005): 116-139. Available: <https://caid.ca/JAICH2005v3n1p115.pdf> [Accessed Dec. 2, 2022].
- [5] Belton. Padraig, "The computer chip industry has a dirty climate secret" *theguardian.com*, Sep. 18, 2021. [Online]. Available: <https://www.theguardian.com/environment/2021/sep/18/semiconductor-silicon-chips-carbon-footprint-climate>. [Accessed Oct. 31, 2022].
- [6] Data Bridge Market Research, "North America Forklift Truck Market - Industry Trends and Forecast to 2027" *databridgemarketresearch.com*, Jun, 2020. [Online]. Available: <https://www.databridgemarketresearch.com/reports/north-america-forklift-truck-market>. [Accessed Nov. 4, 2022].

2.4 - Revision Table

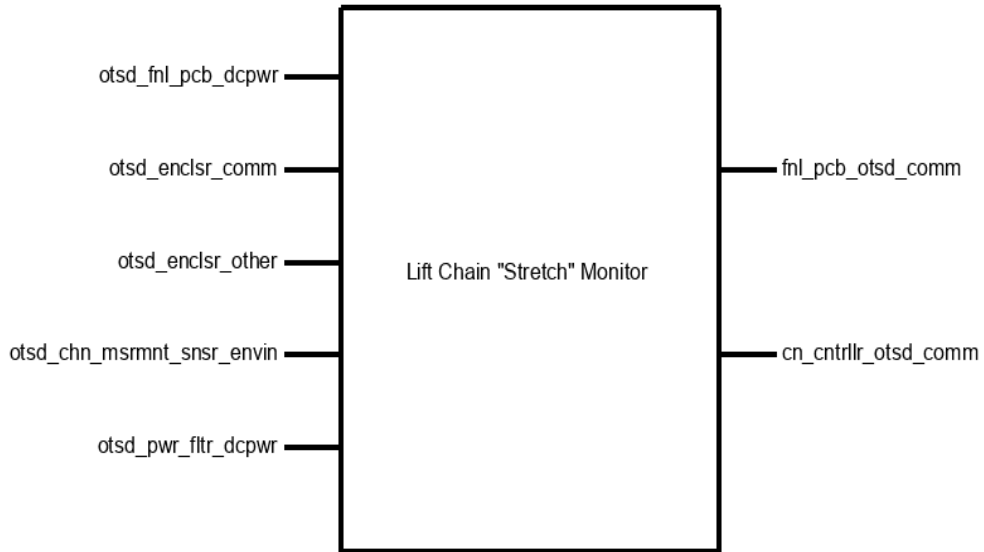
Revision Number	Revision Date	Description	Revision Author
1	11/4	Initial Revision	Kai, Corbin, Adam, Tianhao
2	11/18	Updated Risk Table	Kai, Corbin,

3	4/28	Included Design Impact Statement	Adam, Corbin
4	5/14	Modified documentation and added citation	Corbin

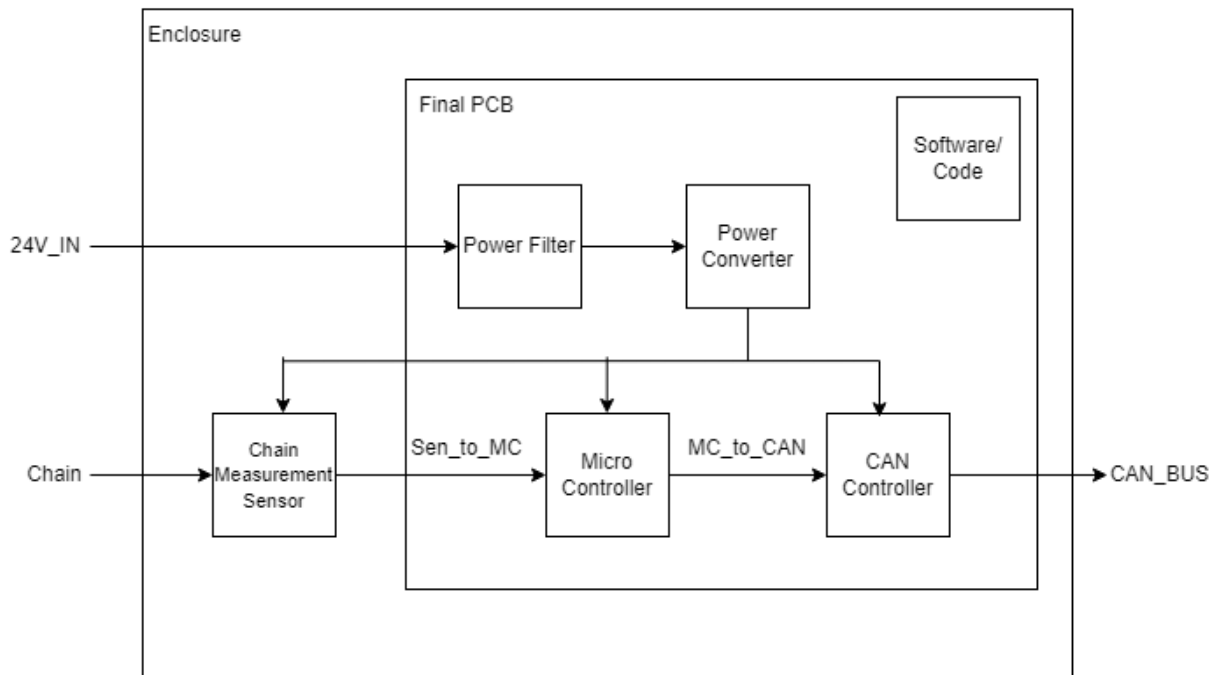
Section 3 - Top-Level Architecture

3.1 - Block Diagram:

Black Box Diagram:



Block Diagram:



3.2 - Block Descriptions:

Block: Power Converter

Role: Adam Grzelewski

The power converter block is based on an application schematic from the MC34063A-E datasheet to be used as a step-down converter. This block accomplishes the goal of reducing voltage and increasing current from 25V to 5V and current up to 500mA. The goal is to make the voltage and current levels safe for our microcontroller, CAN controller, and sensors to be used properly without overheating. The block contains the MC34063A/E chip itself, IN5819 Zener diode, three resistors (two are external to control the voltage output), three capacitors, and one inductor.

This block fulfills a few of our system requirements which include power input, power output, and assists in meeting signal output. Our power input requirement is defined by our project partner as, "Outputs an analog (5v) or digital signal (CAN or other serial protocol signal)." The engineering requirement is defined as, "The system will operate on 24VDC with up to 4Vpp of noise. The system will supply a 5V output at up to 1A peak and no more than .1V ripple." This block meets this requirement by taking the given 24VDC from the forklift battery and will step-down the voltage to 5V. Once again, this is done so that the voltage levels for our microcontroller, CAN controller, and sensors operate at a safe level without overheating. The requirement for 1A peak current may not be met and the requirement may be changed at a later date because 1A peak current is a substantial amount of current for our devices and may damage some of the subsystems. Our power output has the same project partner requirement as the previous requirement. Our engineering requirement however, is defined differently as, "The system will supply a 5V output at up to 500mA peak and no more than .1V ripple." This block meets this requirement because the schematic used is designed to maintain 5V and 500mA output. The signal output requirement has the same project partner requirement as before but with an engineering requirement defined as, "The system will communicate with the Hyster Forklift's electronics using a 5V digital serial signal." This block partially meets this requirement because of the conversion of 24V to 5V. However, this block is not responsible for communicating with Hyster's electronics.

How the block will be validated is by running a demo test in a lab where we will set interface properties that this block will have to meet which will be provided in detail in the following sections. Running a demo in the lab will serve as a form of validation because there I can replicate the forklift battery using a DC power supply and load to read the output current and voltage. I will also provide video evidence of my demo in case the live demo does not perform to standard.

Block: Power Filter

Role: Adam Grzelewski

The power filter block is based on the LC filter schematic from [Digikey's online LC filter calculator](#) to be used as a first order LC filter. This block accomplishes the goal of reducing

frequency down to a fixed value of about less than 16kHz. This is to ensure that our high noise level of frequency is reduced so that our microcontroller and sensors can operate effectively. The block contains one inductor and one capacitor.

This block contributes to a couple of our system requirements which are sensor accuracy and signal output. The LC filter contributes to sensor accuracy because the sensors need to be able to take the measurements it reads and be able to send a signal to the microcontroller to interpret and calculate chain elongation. If the input signal is noisy, the microcontroller will not be able to interpret the data and have false negatives or positives. How the filter contributes to signal output is the same with sensor accuracy except it involves the CAN network and how a noisy signal can have incorrect data sent back to the forklift which can create false negatives if the chain is elongated and the CAN controller would fail to warn the forklift operator that the chain is elongated.

How the block will be validated is by running a demo test in the lab where we will set interface properties that this block will have to meet which will be provided in detail in the following sections. Running a demo in the lab will serve as a form of validation because there I can replicate the forklift battery using a function generator and DC power supply with a load/oscilloscope to read the output current, voltage, and frequency to verify that I can hold a signal to below 16kHz. I will also provide video evidence of my demo in case the live demo does not perform to standard.

Block: Enclosure

Role: Corbin Krecklow

A device may be able to function without an enclosure, but it is exceptionally fragile without a container to ensure all components have consistent connections, orientations, and security. The enclosure for this project will be a durable Pelican Case, created and distributed by the Pelican company. This ensures that our enclosure will be consistent, professionally made, and strong enough to protect our fragile components from the dangerous environment they will have to be in. Additionally, a piece of acrylic (see Figure 4.3.2) will be laser cut to allow for the mounting of the PCB and the guiding of wires through the walls of the enclosure, which will have holes cut into it.

Block: Software/Code

Role: Corbin Krecklow

An electronic device intended for something as precise as measuring a 3% deviation of chain length needs to, at the very least, perform some math and make some decisions (i.e. determining whether or not to send a particular signal). To achieve this, our project will implement a microprocessor containing downloaded software to interpret its inputs and decide what outputs to be delivered. The microprocessor aspect of the project can at times be difficult to distinguish from the software portion, largely because the two are so heavily intertwined. They are, however, different and significant enough to merit their separation as two components

of the project because what is a brain without thoughts and what are thoughts without a way to act on them? As such, this document will refer to signals being received by the microcontroller as signals being received by the software because the only way the software can perform calculations or carry on in a useful manner is by receiving the information provided to and by the microcontroller.

The inputs for this project that are most important to the software will be the outputs of the sensor modules. The sensors used for this project will be infrared break beam sensors. Their beams will be “threaded” through the gaps of the chain. When links of the chain pass over them, chain speed and length may be calculated by the microprocessor from the “yes I am receiving the beam” and “no I am not receiving the beam” signals.

The information gleaned about the chain’s integrity by the calculations performed by the software is meaningless unless acted on. Thus, if a stretch of longer than 3% is detected, the microprocessor will output a signal to the CAN controller that initiates a signal transmission to the operating forklift.

Block: CAN Controller

Role: Tianhao Lin

The can block is built with one of the most practical stand-alone CAN controllers, MCP 2515 as the core. This module achieves the goal of communicating with Hyster Forklift’s electronics via 5 V digital serial signal. After achieving this goal, we can connect our device to the forklift with only two wires, so that the forklift can receive any information accurately. In this block, in addition to MCP 2515, we will also use TJA 1051 as the transceiver of the block to transmit or receive information transmitted through the CAN bus, an 8M Hz crystal oscillator as a bit rate regulator, an LED as an indicator, and some resistors and capacitors to stabilize the voltage and current of the part. In addition, we will also add two terminal block connectors to allow the output of the signal to adapt to more kinds of wires.

This module mainly meets the requirements of signal output. Our project partner defines the signal output as "Outputs an analog (5 V) or digital signal (CAN or other serial protocol signal)". This means that the voltage range of the electrical signal will be between 0 V and 5 V while making the output signal accurate enough. After receiving data from the MCU block via SPI protocol, this module will convert these data into two strings of signal data, namely can_h and can_l, and transmit them through CAN bus. According to the requirements, the voltage of can_h of this block will be between 2.5 V and 4.5 V, and the voltage of can_l will be between 0.5 V and 2.5 V. In order to achieve this effect, the voltage converter (another block) of the device will provide a voltage of 5 V and a current of at least 16 mA to this block. In addition, the CAN protocol requires that the nominal bit rates of the transmitter and receiver are consistent, so we will use an additional crystal oscillator to achieve the corresponding NBR. After the block meets all the above requirements, the device can access its CAN bus at any position of the forklift to transmit the calculation results of the sensor. It is also worth mentioning that the CAN bus will only be used to transmit electrical signals, so it will not have any current, and it cannot be used as a voltage source.

All the above information and some interface properties not mentioned will be introduced in detail in the following sections, and their verified solutions will be mentioned.

Block: Microcontroller

Role: Tianhao Lin

The Microcontroller block to be used in this project is the ATtiny 814. The module will be responsible for collecting and analyzing signals from the sensor, as well as controlling the operation of the signal transmitter. With the ATtiny 814, the project aims to achieve certain targets, such as measuring chain "stretch" within +/- 5% of the actual measurement, costing less than \$10 at a production quantity of 15,000/yr, and outputting an analog (5V) or digital signal (CAN or other serial protocol signal).

However, since most of these requirements will be implemented in other modules, the Microcontroller can assist these blocks through written code or additional components to better meet the set targets. For instance, by adding an external crystal oscillator and some code, the Microcontroller block can accurately calculate the data obtained from the sensor. Similarly, it can also control the output content and bit rate of the transmission module through code.

The ATtiny 814 has 14 ports, which include the Vin port, GND port, 4 spi ports, 1 UPDI port, 2 ports for crystal oscillator, and 5 ports for sensors. Among these ports, the spi interface will be used to connect with the CAN controller, while the UPDI will be used to write the program into the microcontroller. Additionally, the crystal oscillator interface will be used to connect an 8MHz oscillator as an external timer.

Three of the five sensor interfaces will be connected with pull-resistors in parallel to improve the accuracy of the sensor. By doing this, the Microcontroller block can accurately calculate the sensor data to ensure that the project meets its targets.

It is important to note that the price of the entire Microcontroller should not exceed \$1. Since the ATtiny 814 costs only about \$0.75, it leaves room for selecting a better power filter or sensor, among other things. This means that under the limit of \$10, the project can make use of a better sensor or power filter to enhance its performance.

In the following sections, all the information presented above will be discussed in greater detail, along with their verified solutions. This will provide a clearer understanding of the project and how it can be implemented. Overall, the Microcontroller block is an essential component of the project that will enable it to achieve its targets. Through the use of additional components and written code, it can work seamlessly with other modules to ensure the success of the project.

Block: Chain Measurement Sensor

Role: Kai Roy

The purpose of this document is to describe the (Chain Measurement) Sensor Block of the ECE44x Chain Elongation Monitor project to 4th-year ECE students with the intention of these

students being able to build and verify the block without further research. The Sensor block is implemented using three IR Breakbeam sensors. This document provides an overview of the overall block function including interface properties and a schematic, verification for the design in the form of a step-by-step testing process, and support for the validity of the design in the form of outside research and numerical justification addressing individual properties.

One sensor is placed in the 'middle' middle of the device and will be for measuring the speed of the Chain, while two sensors will be placed on the outer edges of the device and will be used for measuring/detecting the elongation of the chain.

The measurement of speed can be done by measuring the time difference between sensor activation as a chain link crosses the two central sensors and calculating the speed with that delta and the chain link length. For the elongation measurement/monitoring, the outer two sensors are placed with a specific distance between them, X amount of chain links. If positioned correctly, the two sensors will activate simultaneously with a chain that is not elongated and activates with some time delta with a chain that is elongated. This time delta coupled with the calculated speed can provide an elongation measurement.

The idea behind this block is to do the physical testing of the sensors, not calculate the actual measurements. This will assist in the implementation and testing of the "Sensor Accuracy" system requirement by testing the rising and falling times of the digital signal.

Block: Final PCB

Role: Kai Roy

The Final PCB contains most of the circuits for all of the electronics blocks of the system. These circuits include the Power Filter, Power Converter, Microcontroller, and CAN Controller Blocks. This means that all of the outward-facing interfaces for these blocks, such as the sensor, power, and CAN connectors, are the interfaces for this block.

To keep the system compact and to reduce the cost of the PCB, the system requirement associated with this block is the one that limits the size of the PCB to 6 in². The choice to put nearly all of the circuits onto one PCB guarantees that the Universal constraints of having a "student-designed PCB" and "no more than 50% built from purchased modules" are met.

3.3 - Interface Definitions:

Name	Properties
otsd_fnl_pcb_dcpwr	<ul style="list-style-type: none"> ● Inominal: 450mA ● Ipeak: 600mA ● Vmax: 24V ● Vmin: 15V
otsd_enclsr_comm	<ul style="list-style-type: none"> ● Other: Wires coming going into/coming from case shall be sheathed and should wire directly into the case ● Other: Enclosure will still be able to close while allowing for exterior wires to reach inner components ● Other: Enclosure will have a transparent side to allow for visual access to interior
otsd_enclsr_other	<ul style="list-style-type: none"> ● Other: Enclosure shall fit within a 20cm x 20cm x 20cm cube (excluding wiring) ● Other: Enclosure shall be able to be easily opened for access to inner components ● Other: Enclosure shall not fracture after being dropped from 6ft
otsd_chn_msrmnt_snsr_envin	<ul style="list-style-type: none"> ● Other: Min Speed the sensor needs to detect: 0m/s ● Other: Detect a break in between chain links ● Other: Max Speed the sensors needs to detect: 3m/s
otsd_pwr_fltr_dcpwr	<ul style="list-style-type: none"> ● Inominal: 450mA ● Ipeak: 600mA ● Vmax: 27V ● Vmin: 7V

sftwracd_mrcntrllr_code	<ul style="list-style-type: none"> ● Other: Logic-Level: 1 - Chain needs replacement ● Other: Logic-Level: 0 - Chain is fine ● Other: Polling Loop Statement will continuously transmit data
fnl_pcb_otsd_comm	<ul style="list-style-type: none"> ● Messages: The Chain detection result ● Protocol: CAN ● Vmax: 4.5V ● Vmin: 0.5V
mrcntrllr_sftwracd_code	<ul style="list-style-type: none"> ● Other: Logic-Level 0 : Chain is blocking sensor ● Other: Transport Protocol Initialization Situation will ensure communication/transmission ● Other: Logic-Level 1 : Chain is not blocking sensor
mrcntrllr_cn_cntrllr_comm	<ul style="list-style-type: none"> ● Messages: Logic 1 for when the chain does need to be replaced ● Messages: Logic 0 for when the chain does not need to be replaced ● Messages: team number: 15
enclsr_fnl_pcb_mech	<ul style="list-style-type: none"> ● Other: Enclosure shall contain the PCB with room to spare for the CAN controller
chn_msrmnt_snsr_fnl_pcb_dsig	<ul style="list-style-type: none"> ● Logic-Level: Logic 1: Vmax = 5V ● Logic-Level: Logic 1: Vmin = 3.5V ● Logic-Level: Logic 0: Vmax = 1.5V ● Logic-Level: Logic 0: Vmin = 0V

chn_msrmnt_snsr_mrcntrlr_dsig	<ul style="list-style-type: none"> ● Logic-Level: Logic 1: Vmax = 5V ● Logic-Level: Logic 1: Vmin = 3.5V ● Logic-Level: Logic 0: Vmax = 1.5V ● Logic-Level: Logic 0: Vmin = 0V
cn_cntrlr_otsd_comm	<ul style="list-style-type: none"> ● Vmax: 2.5V for CAN_L ● Vmax: 4.5V for CAN_H ● Vmin: 2.5V for CAN_H ● Vmin: 0.5V for CAN_L
pwr_cnvtrtr_mrcntrlr_dcpwr	<ul style="list-style-type: none"> ● Inominal: 40mA ● Ipeak: 200mA ● Vmax: 5V ● Vmin: 4.5V
pwr_cnvtrtr_chn_msrmnt_snsr_dcpwr	<ul style="list-style-type: none"> ● Inominal: 60mA ● Ipeak: 90mA ● Vmax: 5.0V ● Vmin: 4.5V
pwr_cnvtrtr_cn_cntrlr_dcpwr	<ul style="list-style-type: none"> ● Inominal: 16mA ● Ipeak: 92.5mA ● Vmax: 5.5V ● Vmin: 2.7V
pwr_filtr_pwr_cnvtrtr_dcpwr	<ul style="list-style-type: none"> ● Inominal: 450mA ● Ipeak: 600mA ● Other: Cutoff Frequency: < 16kHz ● Vmax: 27V ● Vmin: 7V

3.4 - References and File Links:

3.5 - Revision Table:

Revision Number	Revision Date	Description	Revision Author
1	3/15	Initial Revision	Kai, Corbin, Adam, Tianhao
2	5/14		

Section 4 - Block Validations

4.1 - Power Converter:

4.1.1 - Description:

The power converter block is based on an application schematic from the [MC34063A-E datasheet](#) to be used as a step-down converter. This block accomplishes the goal of reducing voltage and increasing current from 25V to 5V and current up to 500mA. The goal is to make the voltage and current levels safe for our microcontroller, CAN controller, and sensors to be used properly without overheating. The block contains the MC34063A/E chip itself, IN5819 Zener diode, three resistors (two are external to control the voltage output), three capacitors, and one inductor.

This block fulfills a few of our system requirements which include power input, power output, and assists in meeting signal output. Our power input requirement is defined by our project partner as, "Outputs an analog (5v) or digital signal (CAN or other serial protocol signal)." The engineering requirement is defined as, "The system will operate on 24VDC with up to 4Vpp of noise. The system will supply a 5V output at up to 1A peak and no more than .1V ripple." This block meets this requirement by taking the given 24VDC from the forklift battery and will step-down the voltage to 5V. Once again, this is done so that the voltage levels for our microcontroller, CAN controller, and sensors operate at a safe level without overheating. The requirement for 1A peak current may not be met and the requirement may be changed at a later date because 1A peak current is a substantial amount of current for our devices and may damage some of the subsystems. Our power output has the same project partner requirement as the previous requirement. Our engineering requirement however, is defined differently as, "The system will supply a 5V output at up to 500mA peak and no more than .1V ripple." This block meets this requirement because the schematic used is designed to maintain 5V and 500mA output. The signal output requirement has the same project partner requirement as before but with an engineering requirement defined as, "The system will communicate with the Hyster Forklift's electronics using a 5V digital serial signal." This block partially meets this requirement because of the conversion of 24V to 5V. However, this block is not responsible for communicating with Hyster's electronics.

How the block will be validated is by running a demo test in a lab where we will set interface properties that this block will have to meet which will be provided in detail in the following sections. Running a demo in the lab will serve as a form of validation because there I can replicate the forklift battery using a DC power supply and load to read the output current and voltage. I will also provide video evidence of my demo in case the live demo does not perform to standard.

4.1.2 - Design:

The black box is included with the design section to show a broad overview of what is exactly going into and out of the block as shown in Figure 1. Power filter is the only input that will be going into the power converter block which is supposed to filter out voltage and current spikes that occur when a battery is initially booted up. This is a different spike than what will be presented later in the interface properties section. Outputs are the microcontroller, sensors, and CAN controller. The microcontroller is used to interpret the signals and measurements coming from the sensors to calculate chain elongation. Sensors are IR lasers that measure point to point using a break beam to determine position and time to measure speed. CAN controller is the primary method to communicate with the forklift to transmit messages back to the forklift operator informing them if the chain is elongated therefore unsafe or no elongation and safe to use.

Schematic is based on the datasheet as presented in the description section and schematic is shown in Figure 2. This is included to show what is exactly used in the schematic and what changes have been made from the datasheet model versus the model created in the schematic and PCB design program KiCad. Notable differences are the diodes because the original diode from the datasheet is an outdated product and is no longer in production. Original diode is replaced with an equivalent diode that is shown in Figure 2. Also, the original schematic is backwards horizontally (pin 1 side and 8 side need to switch spots on the chip) and current schematic in KiCad fixes that issue.

PCB is based on the datasheet as presented in the description section and PCB is shown in Figure 3. PCB is included to demonstrate what the schematic is expected to look like when converted into a PCB. Schematic in Figure 2 is almost identical to the schematic in the datasheet so Figure 3 will be the recommended PCB layout for the final design.

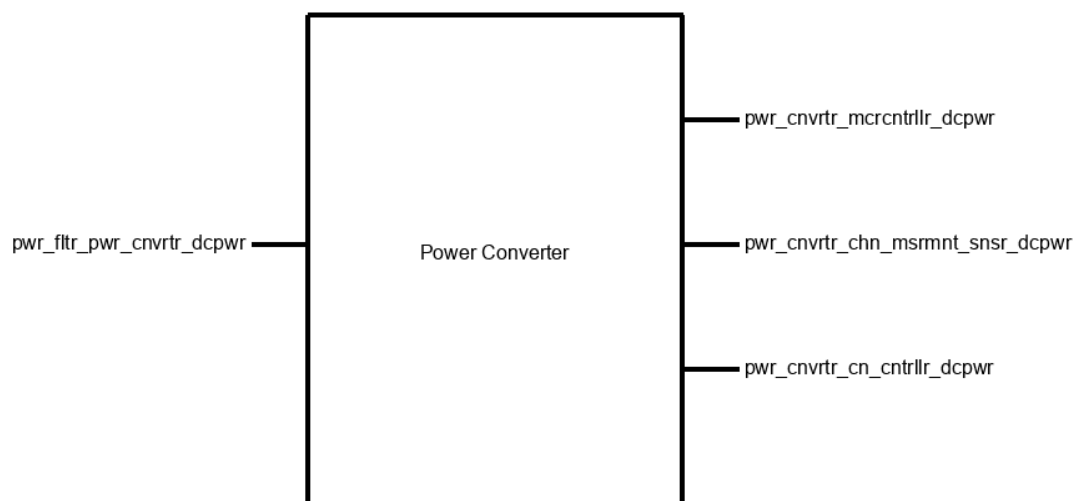


Figure 4.1.1: Black Box for Power Converter

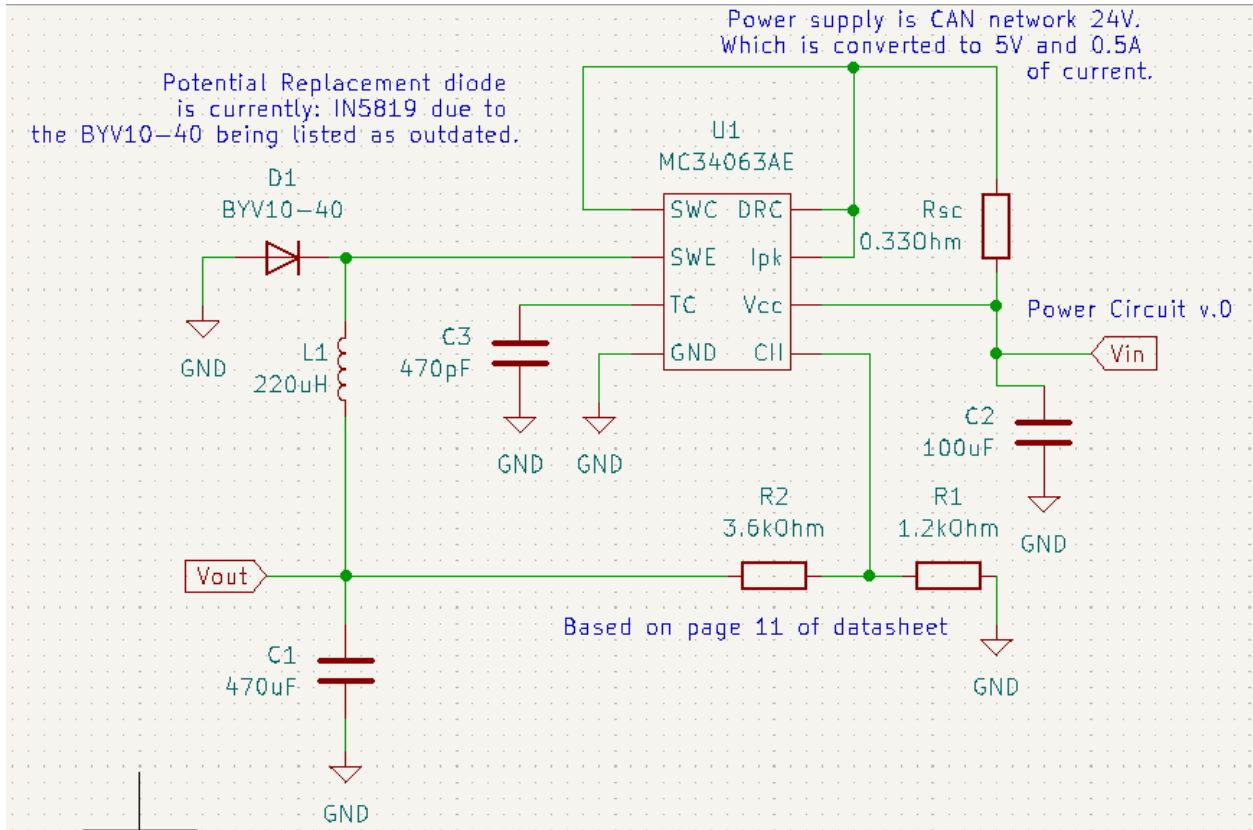


Figure 4.1.2: Power Converter Block Schematic

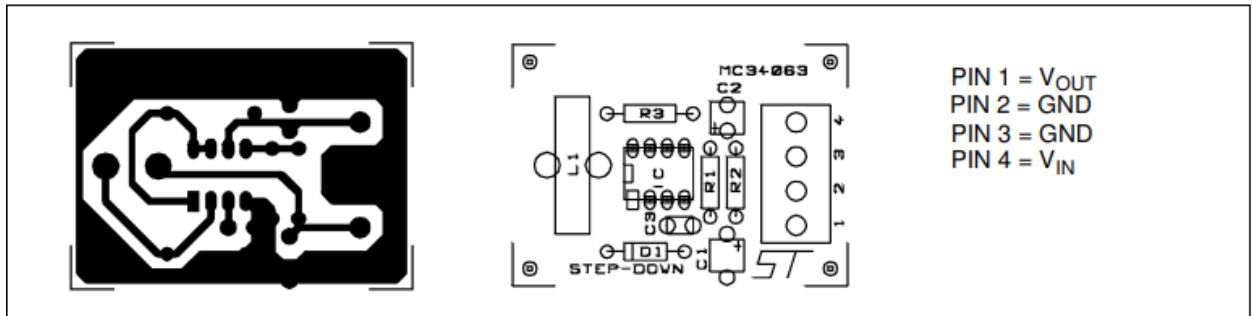


Figure 4.1.3: Power Converter Block PCB

Note that Pwr_filtr_pwr_cnvtr_dcpwr is our input 24 volt CAN network wire from the forklift battery and pwr_cnvtr_mrcntrlr_dcpwr, pwr_cnvtr_chn_msrmt_snsr_dcpwr, pwr_cnvtr_cn_cntrlr_dcpwr has a voltage output of 5 volts and 500 mA going into the microcontroller, CAN controller, and sensors input power.

4.1.3 - General Validation:

This block is based on the application schematic from the [MC34063A-E datasheet](#). The chip is rated to 50V input and output current rating of 1.5A so it can handle volatile voltages from the forklift of input voltages of 27V or potentially higher. Also, the output current using a buck converter design will reduce the voltage and increase current to 5V and 500mA.

I chose this particular design because it was affordable to meet our requirement of the whole design cost (including all blocks) being under 10 dollars since the chip itself is priced at about 1 dollar. However, the other expenses of passive components such as resistors, capacitors, inductors, and diodes total the amount to about 3 dollars which is not too expensive in terms of the total amount for the whole design. I also wanted a buck converter to step down voltage and that was the only way I could accomplish that without having to purchase a voltage regulator.

The resistors in the buck converter circuit act as a load, zener diode acts as forward current, capacitors as discharging current, and the inductor used to store current.

PCB for the buck converter is provided in the datasheet and will follow the recommended layout on figure 16 of the datasheet.[2]

4.1.4 - Interface Validation:

Interface validation will cover what will be demonstrated in the video and live demo of this block and the sections cover why the values are significant and what important details are included with each interface. The interfaces covered are the input and three outputs from Figure 1.

Interface Property	Why is this interface this value?	Why do you know that your design details <u>for this block</u> above meet or exceed each property?
---------------------------	--	---

Pwr_fltr_pwr_cnvtr_dcpwr (Input)

Vmax: 27V	Forklift battery is a 24V battery and start-up voltage is taken into account to make the value about 27V and 7V for our original minimum 15V. (should expect a voltage spike of about 2-3V during start-up)	MC34063A-E datasheet , page 11: Expected input voltage is 24V. Max input voltage is 50V in the datasheet page 5. Start-up voltage is between 1.5-2.1V in the datasheet page 7.
-----------	---	--

Vmin: 7V	Start-up voltage makes the value 7V instead of 17V. Value was found from testing in the lab. 7V can be converted to 5V with little current draw.	MC34063A-E datasheet , page 11: Expected input voltage is 24V. Start-up voltage is between 1.5-2.1V in the datasheet page 7.
Ipeak: 600mA	Based on the schematic in the datasheet.	MC34063A-E datasheet , page 5: Maximum driver collector current is 100mA. Switch current can take 1.5A max Inominal would be 150mA (1 magnitude smaller).
Inominal: 450mA	Nominal current typically operates at a magnitude lower than Ipeak for a buck converter. Value is arbitrary because the datasheet does not have Inominal for the driver collector current.	MC34063A-E datasheet , page 5: Maximum driver collector current is 100mA. Switch current can take 1.5A max Inominal would be 150mA (1 magnitude smaller).

Pwr_cnvtr_mrcntrlr_dcpwr (output microcontroller)

Vmax: 6V	This is the required voltage for our microcontroller to function properly without failure due to high voltage levels.	ATtiny datasheet , page 516: Maximum power supply voltage is 6V.
Vmin: 1.8V	This is the minimum required voltage our microcontroller can take as input voltage.	Vin for the ATtiny can range from 1.8-5.5V, (ATtiny datasheet , page 517).
Ipeak: 200mA	This is the maximum current into the VDD pin.	ATtiny datasheet , page 516: Maximum current into a VDD pin is 200mA.

Inominal: 20mA	Nominal current typically operates at a magnitude lower than Ipeak. Value is arbitrary because the datasheet does not have Inominal for a VDD pin	ATtiny datasheet , page 516: Maximum current into a VDD pin is 200mA.
----------------	---	---

Pwr_cnvtr_chn_msrmt_snsr_dcpwr (output sensor)

Vmax: 5.0V	Max operating voltage of the sensor is 5.5V, but we will only be supplying a max of 5.0V. Maximum voltage is not listed in the datasheet or product description.	C2916 datasheet , page 1: Operating voltage is between 4.5-5.0V.
Vmin: 4.5V	Minimum operating voltage is 4.5V.	C2916 datasheet , page 1: Operating voltage is between 4.5-5.0V..
Ipeak: 90mA	Emitter current draw is between 10mA at 3.3V and 20mA at 5.5V. Including the startup current of 10mA. We have three sensors so 10mA + 20mA = 30mA, 30 x 3 = 90mA.	C2916 product description : Emitter current draw is 10mA at 3.3V and 20mA at 5.5V. C2916 datasheet , page 1: Start-up current is 10mA.
Inominal: 60mA	Emitter current draw is between 10mA at 3.3V and 20mA at 5.5V. Including the startup current of 10mA. We have three sensors so 10mA + 10mA = 20mA, 20 x 3 = 60mA.	C2916 product description : Emitter current draw is 10mA at 3.3V and 20mA at 5.5V. C2916 datasheet , page 1: Start-up current is 10mA.

Pwr_cnvtr_cn_cntrlr_dcpwr (output CAN controller)

Vmax: 5.5V	Maximum supply voltage is 5.5V. Absolute maximum voltage is 7V	MCP2515 datasheet , page 73 and 74: VDD max is 7V and operating voltage is between 2.7-5.5V
Vmin: 2.7V	Minimum supply voltage is 2.7V.	MCP2515 datasheet , page 74: Operating voltage is between 2.7-5.5V

I _{peak} : 92.5mA	CAN transceiver supply current maximum is 2.5mA in silent mode (this is the mode we will run in consistently), LED peak current is 80mA, and CAN controller operating current is 10mA. So, 2.5 + 80 + 10 = 92.5mA. Note that there is no peak current found for the CAN controller.	TJA1051 datasheet , page 2: supply current ranges from 1-2.5mA in silent mode. LTST-C191KRKT datasheet , page 3: Peak forward current is 80mA. MCP2515 datasheet , page 74: Operating current is 10mA.
I _{nominal} : 16mA	CAN transceiver supply current minimum is 1mA in silent mode (this is the mode we will run in consistently), LED nominal current is 5mA at 5V with a 1k ohm resistor in series, and CAN controller operating current is 10mA. So, 1 + 5 + 10 = 16mA.	TJA1051 datasheet , page 2: supply current ranges from 1-2.5mA in silent mode. LTST-C191KRKT datasheet , page 3: Operating current for a LED with 5V input and a 1k ohm resistor is $V = IR$ where $I = 5mA$. MCP2515 datasheet , page 74: Operating current is 10mA.

Table 4.1.1: Interface Properties

4.1.5 - Verification Process:

- Verifying V_{max}, V_{min}, I_{peak}, and I_{nominal} Pwr_filtr_pwr_cnvtrr_dcpwr (input)
 - Set up prototype board with schematic provided from the [MC34063A-E datasheet](#). ****Important Note**** (Must read): Use Figure 2 for reference because the original schematic in Figure 1 has pins backwards.
 - Plug connectors to power supply and DC load.
 - Connect supply to V_{cc} and GND.
 - Connect load to GND and V_{out}.
 - Set load to 500mA and set channel 1 on the power supply to 27V.
 - Inspect the power supply that 27V is being properly supplied to the circuit and there is no short circuit on the supply side (no current going in about 0A).
 - Inspect load that 500mA and 5V is being displayed on the load screen.
 - Shut off load and supply to let circuit components cool down.
 - Repeat steps 1.5-1.8 for V_{min} and I_{nominal} (set supply voltage and current to 7V/10mA set load to 10mA). Note, I_{nominal} values must hold their value for 30 seconds.
 - Note: if connections are not long enough, use alligator clips as needed or other connector types.
- Verifying V_{max}, V_{min}, I_{peak}, and I_{nominal} Pwr_cnvtrr_mrcntrlr_dcpwr, Pwr_cnvtrr_chn_msrmnt_snsr_dcpwr, Pwr_cnvtrr_cn_cntrlr_dcpwr (all outputs)
 - Set up prototype board with schematic provided from Figure 2 (if not set up already).

- Repeat steps 1.2-1.4 from part 1.
- Set load to 500mA and set channel 1 on the power supply to 24V. (Ipeak for all outputs total to about 480mA).
- Use multimeter, oscilloscope, or load to verify that Vout is 5V
- Inspect load that 500mA is being displayed on the load screen
- Shut off load and supply to let circuit components cool down.
- Repeat steps 2.3-2.6 for Vmin (output voltage 3.3V) and hold Inominal (100mA) for 30 seconds to see if circuit can hold the required voltage before failing. Inominal will be 96mA in this case so set load to 100mA.

4.1.6 - References and File Links

[1] Microchip, "ATtiny214/414/814", *Microchip Technology Inc*, 2017. [Online]. Available:<https://ww1.microchip.com/downloads/en/DeviceDoc/40001912A.pdf>, [Accessed Feb. 10, 2023].

[2] STMicroelectronics, "DC-DC Converter Control Units Datasheet", *STMicroelectronics*, 2020. [Online]. Available:<https://www.st.com/content/ccc/resource/technical/document/datasheet/03/f9/c4/3d/7f/eb/4c/5e/CD00001232.pdf/files/CD00001232.pdf/jcr:content/translations/en.CD00001232.pdf>, [Accessed Jan. 17, 2023].

[3] Adafruit, "HD-DS25CM-3MM", *Adafruit*, 2021. [Online]. Available:<https://cdn-shop.adafruit.com/product-files/2167/C2916.pdf>, [Accessed Jan. 20, 2023].

[4] Microchip, "Stand-Alone CAN Controller with SPI Interface", *Microchip Technology Inc*, 2019. [Online]. Available:<https://ww1.microchip.com/downloads/en/DeviceDoc/MCP2515-Stand-Alone-CAN-Controller-with-SPI-20001801J.pdf>, [Accessed Feb. 10, 2023].

[5] NXP Semiconductors, "TJA1051 High-speed CAN transceiver Product data sheet", *NXP Semiconductors*, 2017. [Online]. Available:<https://www.mouser.com/datasheet/2/302/TJA1051-3083392.pdf>, [Accessed Feb. 10, 2023].

[6] Adafruit, "IR Break Beam Sensors with Premium Wire Header Ends - 3mm LEDs", *Adafruit*, 2021. [Online]. Available:<https://www.adafruit.com/product/2167>, [Accessed Feb. 10, 2023].

[7] Liteon Optoelectronics, "SMD LED Product Data Sheet LTST-C191KRKT", *Lite-on Technology Corporation*, 2022. [Online]. Available:<https://optoelectronics.liteon.com/upload/download/DS22-2000-223/LTST-C191KRKT.PDF>, [Accessed Feb. 10, 2023].

4.1.7 - Revision Table

Revision Number	Revision Date	Description	Revision Author
1	1/20/23	Initial Revision of section 4.1	Adam Grzelewski
2	2/10/23	Updated Description, Design, Interface Properties, Verification Plan, and References and File Links sections.	Adam Grzelewski
3	2/11/23	Block validation proofread and made minor grammar changes.	Adam Grzelewski
4	3/14/23	Updated Interface property input voltage minimum to 7V.	Adam Grzelewski

4.2 - Power Filter:

4.2.1 - Description:

The power filter block is based on the LC filter schematic from [Digikey's online LC filter calculator](#) to be used as a first order LC filter. This block accomplishes the goal of reducing frequency down to a fixed value of about less than 16kHz. This is to ensure that our high noise level of frequency is reduced so that our microcontroller and sensors can operate effectively. The block contains one inductor and one capacitor.

This block contributes to a couple of our system requirements which are sensor accuracy and signal output. The LC filter contributes to sensor accuracy because the sensors need to be able to take the measurements it reads and be able to send a signal to the microcontroller to interpret and calculate chain elongation. If the input signal is noisy, the microcontroller will not be able to interpret the data and have false negatives or positives. How the filter contributes to signal output is the same with sensor accuracy except it involves the CAN network and how a noisy signal can have incorrect data sent back to the forklift which can create false negatives if the chain is elongated and the CAN controller would fail to warn the forklift operator that the chain is elongated.

How the block will be validated is by running a demo test in the lab where we will set interface properties that this block will have to meet which will be provided in detail in the following sections. Running a demo in the lab will serve as a form of validation because there I can replicate the forklift battery using a function generator and DC power supply with a load/oscilloscope to read the output current, voltage, and frequency to verify that I can hold a signal to below 16kHz. I will also provide video evidence of my demo in case the live demo does not perform to standard.

4.2.2 - Design:

The black box is included with the design to show a broad overview of what is exactly going into and out of the block as shown in Figure 1. An outside power source (forklift battery) is the only input that will be going into the power filter block which is supposed to filter the high frequency (>16kHz) down to less than 16kHz to deal with excess noise from the battery. Only output is the power converter. The power converter will take the high 24 volts from the battery and reduce it down to a safe 5 volts to prevent any sensors, microcontroller, or other components from getting damaged.

Schematic is based on the Digikey reference as presented in the description section and the schematic is shown in Figure 2. This is included to show what is exactly used in the schematic and what changes have been made if any from the reference on Digikey versus the model created in the schematic and PCB design program KiCad. PCB directly copies the schematic from KiCad and is shown in Figure 3.

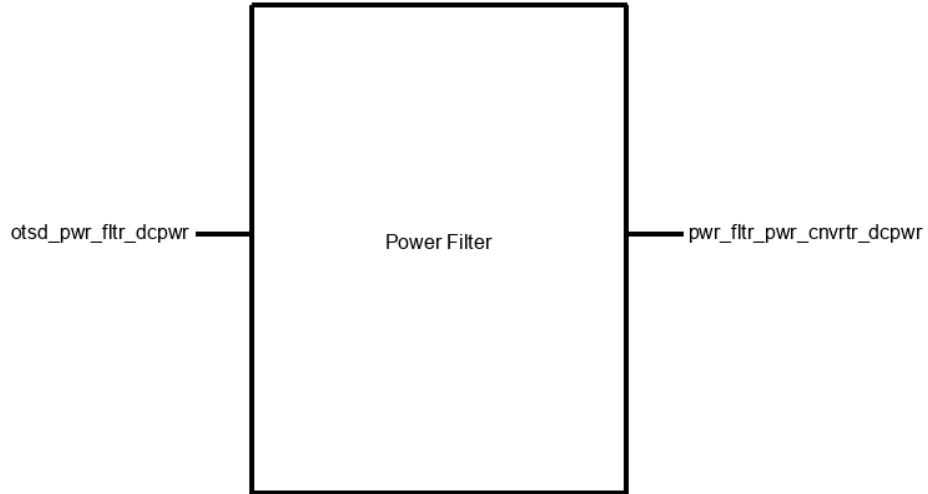


Figure 4.2.1: Black Box for Power Filter

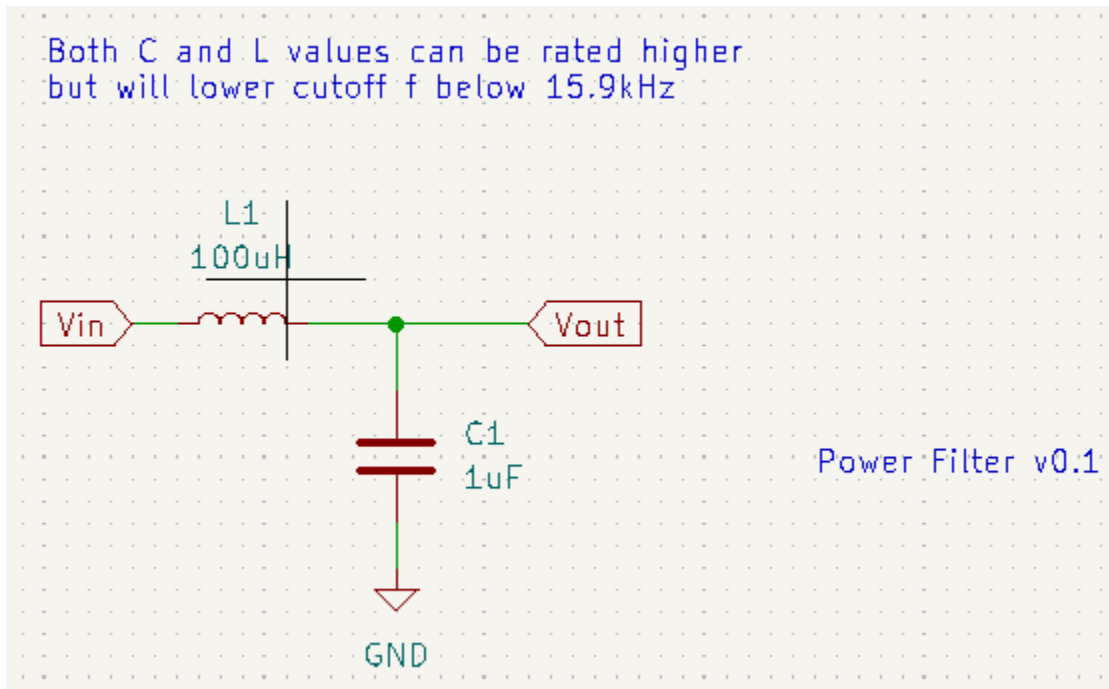


Figure 4.2.2: Power Filter Block Schematic

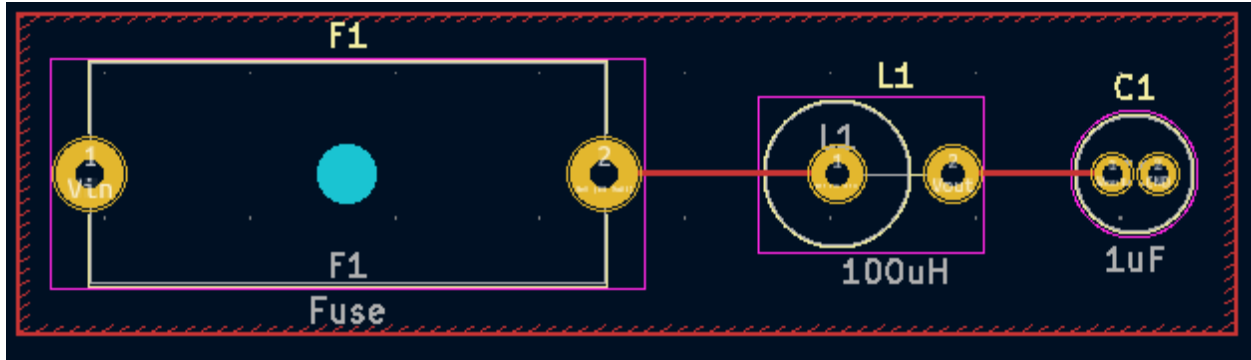


Figure 4.2.3: Power Filter Block PCB

Note that Otsd_pwr_ftr_dcpwr is our input 24 volt CAN network wire from the forklift battery and pwr_ftr_pwr_cvrtr_dcpwr is the filtered 16kHz output to the power filter.

4.2.3 - General Validation:

This block is based on [Digikey's online LC filter calculator](#). The capacitor is rated to 50V and the inductor is rated for 632mA so it can handle volatile voltages from the forklift with input voltages of 27V and input current peak of about 600mA on startup. Also, the output frequency using a LC filter design will limit the frequency to less than 16kHz.

I chose this particular design because these types of filters are power filters and are designed to handle high voltage and current (27Vmax/600mA peak). Cost is inexpensive compared to the power converter with the inductor costing \$1.35 and the capacitor \$0.10 making the total \$1.45. This makes both blocks priced at \$4.45 which is still under the total design cost of being under 10 dollars. However, there are other blocks that still need to be included (6 others) that need to be priced. I also could not find cheaper components because I need power inductors and capacitors that could deal with high voltage and current coming from the forklift's 24V battery. The capacitor acts as discharging current and the inductor to store current. PCB is created in KiCad and is shown in Figure 3.

4.2.4 - Interface Validation:

Interface validation will cover what will be demonstrated in the video and live demo of this block and the sections cover why the values are significant and what important details are included with each interface. The interfaces covered are the input and output from Figure 1.

Interface Property	Why is this interface this value?	Why do you know that your design details <u>for this block</u> above meet or exceed each property?
--------------------	-----------------------------------	--

Otsd_pwr_fltr_dcpwr (Input)

Vmax: 27V	Forklift battery is a 24V battery and start-up voltage is taken into account to make the value about 27V and 7V for our original minimum 15V. (should expect a voltage spike of about 2-3V during start-up)	Capacitor datasheet , page 1: Rated max DC voltage is 50V. Inductor datasheet , page 3: Max current is 632mA.
Vmin: 7V	Start-up voltage makes the value 7V instead of 17V. Value was found from testing in the lab. 7V can be converted to 5V with little to no current draw.	Capacitor datasheet , page 1: Rated max DC voltage is 50V. Inductor datasheet , page 3: Max current is 632mA.
Ipeak: 600mA	Forklift expected peak current is around 600mA on startup.	Capacitor datasheet , page 1: Rated max DC voltage is 50V. Inductor datasheet , page 3: Max current is 632mA.
Inominal: 450mA	Input current from the forklift is between 300-600mA so the nominal would be around 450mA in this case.	Capacitor datasheet , page 1: Rated max DC voltage is 50V. Inductor datasheet , page 3: Max current is 632mA.

Pwr_fltr_pwr_cnvtr_dcpwr (output power converter)

Vmax: 27V	Forklift battery is a 24V battery and start-up voltage is taken into account to make the value about 27V and 7V for our original minimum 15V. (should expect a voltage spike of about 2-3V during start-up) (no conversions made same output)	Capacitor datasheet , page 1: Rated max DC voltage is 50V. Inductor datasheet , page 3: Max current is 632mA.
Vmin: 7V	Start-up voltage makes the value 7V instead of 17V. Value	Capacitor datasheet , page 1: Rated max DC voltage is 50V. Inductor

	was found from testing in the lab. 7V can be converted to 5V with little or no current draw. (no conversions made same output)	datasheet , page 3: Max current is 632mA.
I _{peak} : 600mA	Forklift expected peak current is around 600mA on startup. (no conversions made same output)	Capacitor datasheet , page 1: Rated max DC voltage is 50V. Inductor datasheet , page 3: Max current is 632mA.
I _{nominal} : 450mA	Input current from the forklift is between 300-600mA so the nominal would be around 450mA in this case. (no conversions made same output)	Capacitor datasheet , page 1: Rated max DC voltage is 50V. Inductor datasheet , page 3: Max current is 632mA.
Other: Cutoff Frequency: <16kHz	Frequency cutoff is less than 16kHz to deal with excess noise from the forklift battery, ensure sensor accuracy, and accurate CAN network communication between the forklift and our system.	$F_{lc} = 1/(2\pi\sqrt{LC}) = 1/(2\pi\sqrt{(100\cdot 10^{-6}\cdot 1\cdot 10^{-6})}) = 15.9\text{kHz}$ (L = 100uH, C = 1uF)

Table 4.2.1: Interface Properties

4.2.5 - Verification Process:

Verification plan is a step-by-step procedure to test the interface properties in the lab and is designed such that somebody who knows nothing about this block would be able to follow the same steps and arrive at the same results as the person who designed the block.

1. Verifying V_{max}, V_{min}, I_{peak}, and I_{nominal} Otsd_pwr_fldr_dcpwr (input)
 - 1.1. Set up prototype board with schematic provided from [Digikey's online LC filter calculator](#).
 - 1.2. Plug connectors to power supply and DC load.
 - 1.3. Connect supply to V_{cc} and GND.
 - 1.4. Connect load to GND and V_{out}.
 - 1.5. Set load to 600mA and set channel 1 on the power supply to 27V.

- 1.6. Inspect the power supply that 27V is being properly supplied to the circuit and there is no short circuit on the supply side (no current going in about 0A).
- 1.7. Inspect load that 600mA and 27V is being displayed on the load screen.
- 1.8. Shut off load and supply to let circuit components cool down.
- 1.9. Repeat steps 1.5-1.8 for V_{min} and $I_{nominal}$ (set supply voltage and current to 7V/450mA set load to 450mA). Note, $I_{nominal}$ values must hold their value for 30 seconds.
- 1.10. Note: if connections are not long enough, use alligator clips as needed or other connector types.
2. Verifying V_{max} , V_{min} , I_{peak} , $I_{nominal}$, and Frequency Pwr_fldr_pwr_cnvtrr_dcpwr (output)
 - 2.1. Set up prototype board with schematic provided from Figure 2 (if not set up already).
 - 2.2. Repeat steps 1.2-1.4 from part 1.
 - 2.3. Set load to 600mA and set channel 1 on the power supply to 27V.
 - 2.4. Use multimeter, oscilloscope, or load to verify that V_{out} is 27V
 - 2.5. Inspect load that 600mA is being displayed on the load screen
 - 2.6. Shut off load and supply to let circuit components cool down.
 - 2.7. Repeat steps 2.3-2.6 for V_{min} (output voltage 7V) and hold $I_{nominal}$ (450mA) for 30 seconds to see if circuit can hold the required voltage before failing.
 - 2.8. Connect a probe to oscilloscope.
 - 2.9. Disconnect connectors from DC power supply and load and plug new connectors from the function generator to V_{in} and GND.
 - 2.10. Plug probe to V_{out} and GND.
 - 2.11. Set function generator to sine wave, 10 volts amplitude, and frequency to greater than 16kHz.
 - 2.12. Verify on oscilloscope that output signal is less than 16kHz.

4.2.6 - References and File Links

- [1] Digikey Electronics, "Low Pass/High Pass Filter Calculator", *Digikey Electronics*, 2023. [Online]. Available: <https://www.digikey.com/en/resources/conversion-calculators/conversion-calculator-low-pass-and-high-pass-filter>, [Accessed Feb. 20, 2023].
- [2] Würth Elektronik, "WCAP-ATG5 Aluminum Electrolytic Capacitors Datasheet", *Würth Elektronik eiSos GmbH & Co. KG*, 2020. [Online]. Available: <https://www.mouser.com/datasheet/2/445/860020672005-3099424.pdf>, [Accessed Mar. 14, 2023].
- [3] Signal Transformer, "SFAP1 Series Fixed Axial Power Inductors", *Signal Transformer*, 2022. [Online]. Available: https://www.mouser.com/datasheet/2/643/ds_ST_SFAP1_series-3078205.pdf, [Accessed Mar. 14, 2023].

4.2.7 - Revision Table

Revision Number	Revision Date	Description	Revision Author
1	3/14/23	Initial Revision	Adam Grzelewski
2	3/15/23	Revision Table created	Adam Grzelewski
3	3/16/23	Explained why filter needed to be below 16kHz. Proofread document.	Adam Grzelewski

4.3 - Enclosure:

4.3.1 - Description:

A device may be able to function without an enclosure, but it is exceptionally fragile without a container to ensure all components have consistent connections, orientations, and security. The enclosure for this project will be a durable Pelican Case, created and distributed by the Pelican company. This ensures that our enclosure will be consistent, professionally made, and strong enough to protect our fragile components from the dangerous environment they will have to be in. Additionally, a piece of acrylic (see Figure 4.3.2) will be laser cut to allow for the mounting of the PCB and the guiding of wires through the walls of the enclosure, which will have holes cut into it.

4.3.2 - Design:

The type of case we will be using for this project will assure the safety and stability of our electronic components. The enclosure will protect and secure the PCB and CAN controller. The enclosure will also allow for wires to enter and leave. The casing will prevent the more delicate components from adverse interactions with their environs such as moisture or abrasive contact and will have a lid to allow for modifications after the interior components have been installed.

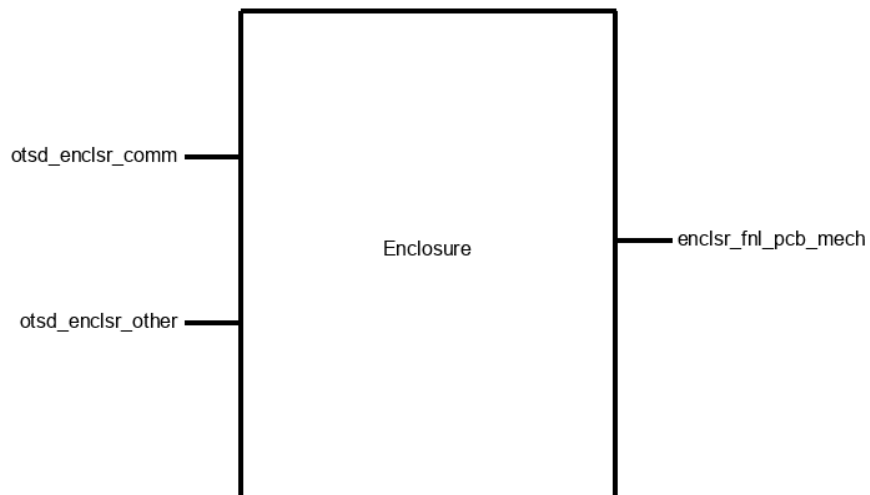


Figure 4.3.1: Black Box Diagram

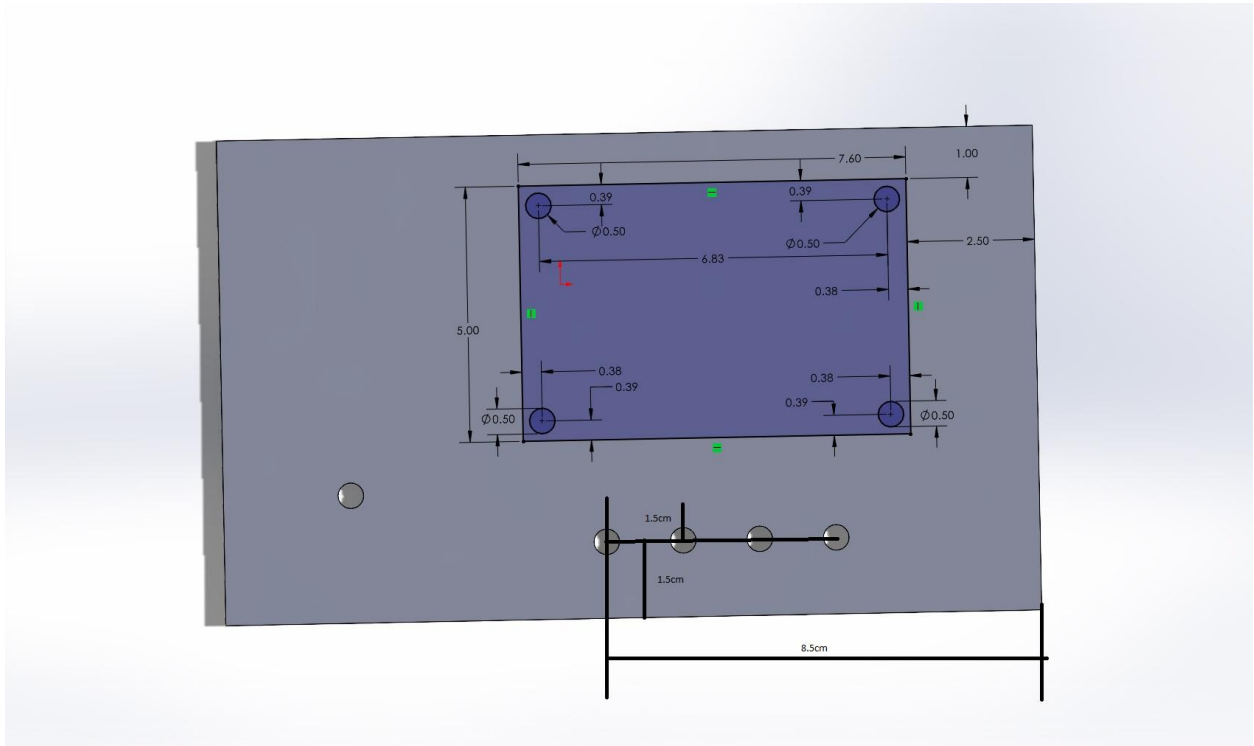


Figure 4.3.2: CAD model of Acrylic cutout (Overall rectangle is 16 cm x 9.5 cm; Purple rectangle is where PCB will be mounted)

4.3.3 - General Validation:

The choice of a pelican case is validated by the need for a sturdy enclosure that will hold the electronic components without too much extra space inside, while being pliable enough to be drilled into. Pelican cases are a thick and sturdy plastic, which makes it not only water and shatter resistant, but also able to be cut into with a drill bit.

4.3.4 - Interface Validation:

Interface Property

Why is this interface this value?

Why do you know that your design details for this block above meet or exceed each property?

otsd_enclsr_comm : Input

Other: Enclosure will still be able to close while allowing for exterior wires to reach inner components	Wires need to be able to leave the enclosure and not interfere with the closing of the lid.	We were able to close the lid while holding a wire between the lid and enclosure as the lid was shut completely. This will be further met when holes are drilled into the enclosure
Other: Enclosure will have a transparent side to allow for visual access to interior	The lid should be transparent to allow for an observer to look in and ensure the electronic components are in good health.	We have the box and it has a clear lid
Other: Wires coming going into/coming from case shall be sheathed and should wire directly into the case	The wires leaving the enclosure should not tangle or interfere with their surroundings	We have ordered and received wires that are “pre-sheathed” and will be able to connect to each other seamlessly

otsd_enclsr_other : Input

Other: Enclosure shall be able to be easily opened for access to inner components	An enclosure that permanently seals is helpful only in the cases of <u>extreme</u> security or durability. Our project will likely need to be accessed many times after it is initially placed in the box	We have the box and it has a hinged lid that opens and closes easily
---	---	--

Other: Enclosure shall not fracture after being dropped from 6ft	Durability is an essential aspect of an enclosure that will be in an environment like our intended use. If the box can withstand a drop of this height, it is indeed extremely durable	We have the box and have dropped it from 6ft without it fracturing
Other: Enclosure shall fit within a 20cm x 20cm x 20cm cube (excluding wiring)	The entire system should be relatively small to be able to be easily mounted to a forklift. These dimensions are “small” for the purposes of our project	We have the box and have measured it; it does indeed fit within these dimensions

enclsr_fnl_pcb_mech : Output

Other: Enclosure shall contain the PCB with room to spare for the CAN controller	This requirement, admittedly, is redundant because the CAN controller is on the PCB itself. Otherwise, it is important all relevant components can fit within the enclosure	The PCB fits inside the box with room to spare - we have both the box and PCB and have placed the PCB inside to verify
--	---	--

Table 4.3.1: Interface Properties

(the box on the right of the final row “is outlined in black” according to google docs. Apologies for its invisibility)

4.3.5 - Verification Process:

otsd_enclsr_comm Input Verification:

- Enclosure will still be able to close while allowing for exterior wires to reach inner components
 - This can be verified by placing wires the thickness of those we will be using (or simply using those wires) between the lid and the case and attempting to close it. This test was performed and indeed the case was still able to close. This will be further possible when we have drilled holes in the enclosure, ensuring the wires do not come close to interacting with the lid’s ability to close
- Enclosure will have a transparent side to allow for visual access to interior
 - This can be verified by placing something inside the enclosure, closing the lid, and seeing that, yes, the object can still be seen
- Wires coming going into/coming from case shall be sheathed and should wire directly into the case
 - This can be verified by using sheathing on the wires entering or leaving the case. The appropriate wires have been bought for this purpose already.

otsd_enclsr_other Input Verification:

- Enclosure shall be able to be easily opened for access to inner components
 - This can be verified by opening and closing the lid and feeling how easy, and additionally breezy, it is to do so
- Enclosure shall not fracture after being dropped from 6ft
 - This can be verified by measuring out 6ft from the ground and dropping the enclosure from that point, then seeing if the box sustains fractures or other damage
- Enclosure shall fit within a 20cm x 20cm x 20cm cube (excluding wiring)
 - This can be verified by measuring the dimensions of the box itself. If all three of the length, width, and height are below 20cm, then it will indeed fit into a 20cm cube

enclsr_fnl_pcb_mech Output Verification:

- Enclosure shall contain the PCB with room to spare for the CAN controller
 - This can be verified by measuring out a piece of cardboard or other object that is the same size as our PCB (3in x 2in) and placing it within the enclosure to see if it fits without taking up the entire volume of the box

4.3.6 - Reference and File Links

4.3.7 - Revision Table

Revision Number	Revision Date	Description	Revision Author
1	3/15	Initial Revision of section 4.4	Corbin

4.4 - Software/Code:

4.4.1 - Description:

An electronic device intended for something as precise as measuring a 3% deviation of chain length needs to, at the very least, perform some math and make some decisions (i.e. determining whether or not to send a particular signal). To achieve this, our project will implement a microprocessor containing downloaded software to interpret its inputs and decide what outputs to be delivered. The microprocessor aspect of the project can at times be difficult to distinguish from the software portion, largely because the two are so heavily intertwined. They are, however, different and significant enough to merit their separation as two components of the project because what is a brain without thoughts and what are thoughts without a way to act on them? As such, this document will refer to signals being received by the microcontroller as signals being received by the software because the only way the software can perform calculations or carry on in a useful manner is by receiving the information provided to and by the microcontroller.

The inputs for this project that are most important to the software will be the outputs of the sensor modules. The sensors used for this project will be infrared break beam sensors. Their beams will be “threaded” through the gaps of the chain. When links of the chain pass over them, chain speed and length may be calculated by the microprocessor from the “yes I am receiving the beam” and “no I am not receiving the beam” signals.

The information gleaned about the chain’s integrity by the calculations performed by the software is meaningless unless acted on. Thus, if a stretch of longer than 3% is detected, the microprocessor will output a signal to the CAN controller that initiates a signal transmission to the operating forklift.

4.4.2 - Design:

To ensure the chain being measured is within safe levels of wear, the device will calculate a running average of the time between signals from the IR sensor, equating the small time interval the sensor is interrupted to the length of each link of the chain. This method of measurement poses some difficulty due to differing possible speeds, so the chain links measured will, instead of being compared to some remembered universal variable “Chain_Length” or a similar method, the average of the current sample (samples being separated by an interval of no change in signal from the IR sensor) will be calculated on a running basis and, should a particular measure be higher than the calculated average by more than 3%, the code will determine that it is necessary the microprocessor sends a signal to the CAN controller to let it “know” the chain needs to be replaced and a signal on the CAN network needs to be sent.

A brief analogy of CAN signal networks is a landline with as many callers as there are signal-capable components all trying to talk to one, attentive central boss. Each of the callers listens to the line to know if another caller is communicating, determines the priority of their own

information, and waits until their appropriate turn to deliver their message to the boss. This may sound inefficient and like important information may end up being put to the wayside if something has to wait for their turn for too long, but luckily all the callers are extremely fast talkers and the boss processes information extremely quickly because these CAN networks operate with a very, very high baud/bit rate.

An aspect of the running average method that will have to be considered is that additional weight/scrutiny will be placed on the first few measurements gathered by the device. As such, a way to wait until at least 3 samples have been collected before the average begins to be calculated may be implemented.

Another downside to the averaging method of measurement is that our device will likely be mounted to the center mast of the forklift and, as such, will only be able to measure one section of the overall chain. The upside to this downside is that this particular section is a very high-stress portion of the chain and, as such, will benefit from being closely monitored.

One edge case that will be particularly difficult to work around is a situation where speed is abruptly adjusted, such as an increase or decrease in chain velocity or the situation of constant acceleration - especially an abrupt reverse in direction of the chain. This will be addressed by having the IR sensor also determine the relative speed of the chain, which *will* be done by using a universal chain_length constant. The speed determined here cannot be used in the math itself, but can be used to know when it is time to start a new sample pool if change in speed is measured. The speed determined cannot be directly used because if it was every link of chain would come out to the "perfect and healthy" length:

Speed = 1 known link length (universal/schematic value) / time it took to pass beams

Link length = time it took this particular link to pass beam * Speed (link / time) = universal length

Figures 1 and 2 below are a black box diagram and abstract flow chart respectively. Figure 1 shows that the actual processing of the data from the IR sensor is performed using the code, but that the actual transmission/receiving of the data is performed in the microprocessor, which it is downloaded onto. Figure 2 shows the more realistic flow of data/signals - contacting the microprocessor, which "thinks" about input with the code brain, then sends a signal to the CAN controller based on its conclusions. Pseudocode is also included below, which is a skeleton/intention of what will be uploaded onto the microprocessor.

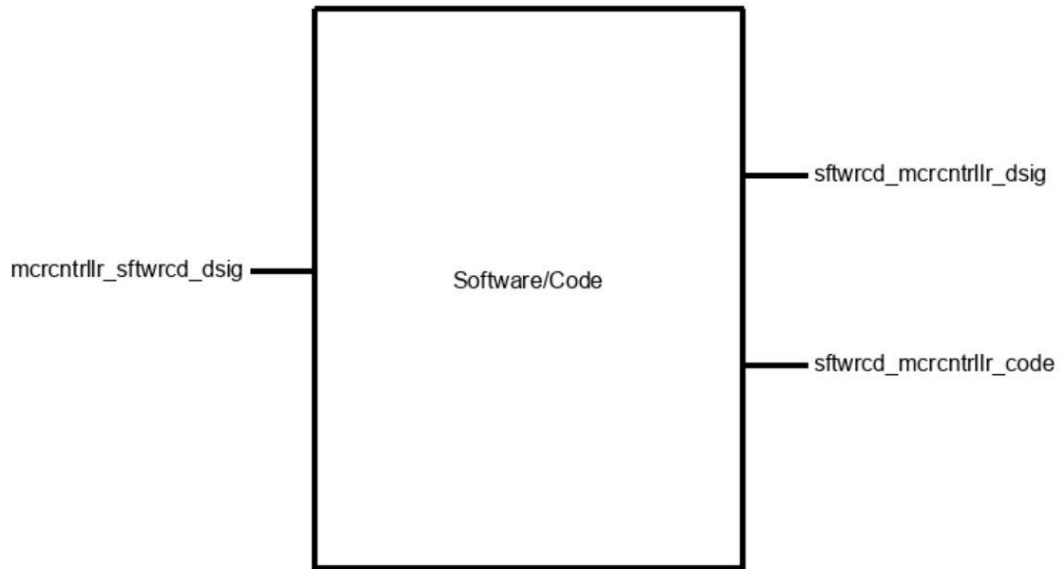


Figure 4.4.1: Black box diagram for Software/Code

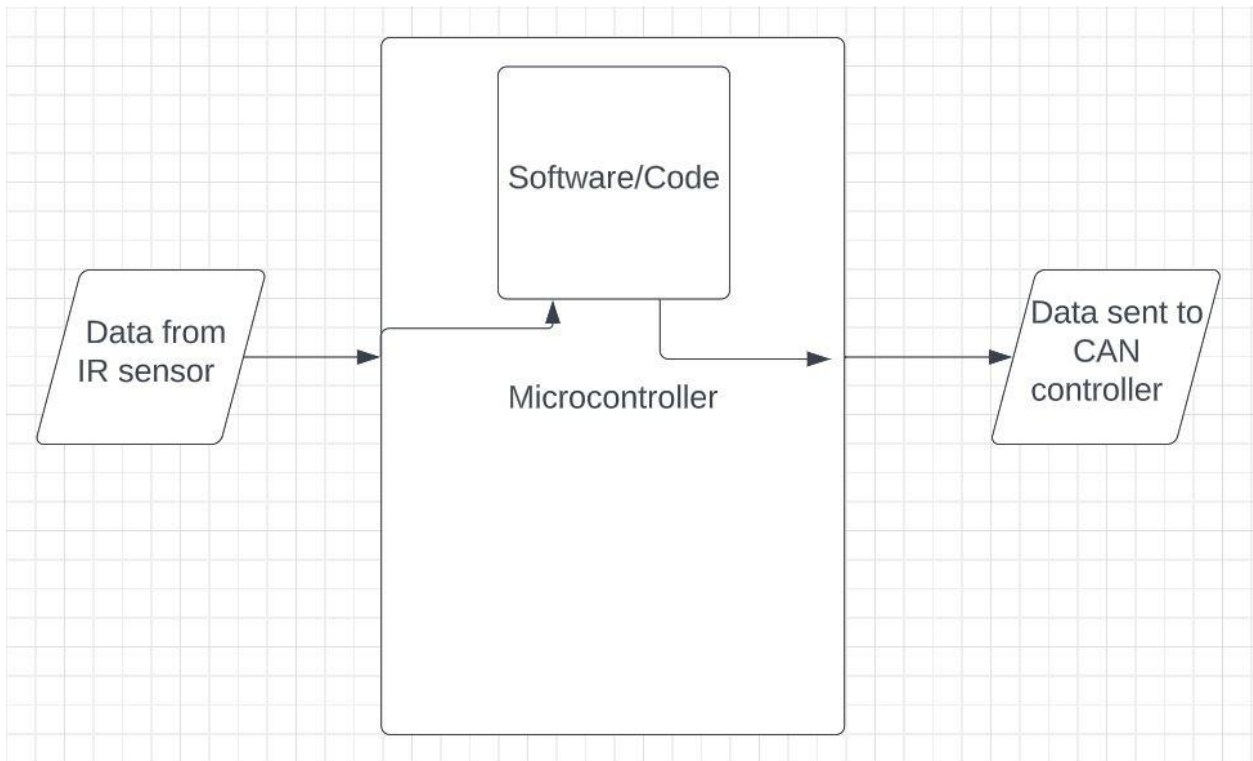


Figure 4.4.2: Abstract flow diagram for physical path(s) of data

Pseudocode:

```
include <necessary libraries>
```

```

setup{

    define an array with 500 spaces //(likely be scaled down or up
    based on the needs determined from testing)

    access and configure the clock/timer within microprocessor

    define variable rightNow = 0 //(current time)

    define variable beamYes = 0 //(input signal from IR)

    define variable Time1 = 0 //(last time IR beam was "seen")

    define variable Time2 = 0 //(next most recent "sighting" time
    of beam)

    define variable Length //( "length" of most recent link)

    //It will likely prove necessary to write a code that
    "harvests" the information present on the pin connected to the IR
    output instead of simply treating the information as passively
    available

}

loop{

    start timer

    if (beamYes==1 & Time1 == 0){

        Time1 = rightNow //basically for the first time the thing
        starts

    }

    if (beamYes==1 && Time1!=0){

        Time2 = rightNow

        Length = Time2 - Time1

        if (Length>3seconds){

            //this is the situation where the lift stopped

        }

    }

}

```

```

    put Length into next available slot of the table

    perform mean operation with all current contents of the
    table

    while(numberBeingLookedAt!=0){ //while looking at
    non-empty

        compare all numbers currently in table to the mean

        if (numberBeingLookedAt is >3% larger than average){

            Send an Uh Oh signal to CAN controller

        }

    }

    Time1 = Time2

}

Back to the start of the loop!

}

```

4.4.3 - General Validation:

The design presented above meets the needs of the system and this block because the sole intention of the device we are designing is to determine if a chain is within a safe length limit and, as such, will be performing a lot of mathematical calculations but will not be doing any other fancy tricks. This means the bulk of the programming will focus on calculations and communication with other components of the device. It is important to keep things simple, especially in engineering and when interacting with other components. However, the existence of the edge cases and their inevitable nature with the current setup means that, perhaps in the future with another more elaborate design or with a different team continuing this, implementations could be made in the code and overall design to consider the current state of the motors. However, as this project overall is a proof of concept for Hyster, we will not worry too much about making it bullet-proof in terms of dealing with every possible outcome. As the concept and design for the code stand, it will fulfill the role set out for it well enough.

4.4.4 - Interface Validation:

Interface Property Why is the interface this value?

Why do you know that your design details for this block above meet or exceed each property?

sftwrcd_mrcntrlr_code : Output

Other (Memory size): Program size must be less than 4KB	The ATtiny814 has a 8KB reprogrammable Flash memory to store code/programming	The data sheet for the ATtiny214/414/ <u>814</u> lists this value [2]
---	---	---

sftwrcd_mrcntrlr_dsig : Output

Logic-Level: 1 (2.8 - 5.5V): Chain needs replacement	When the software determines an unacceptable level of lengthening, a HIGH signal will be sent	The data sheet for the ATtiny214/414/ <u>814</u> lists this value [2]
Logic-Level: 0 (0V): Chain is fine	When the software determines the chain is within an acceptable length, a LOW will be sent	The data sheet for the ATtiny214/414/ <u>814</u> lists this value [2]

mrcntrlr_sftwrcd_dsig : Input

Logic-Level: 0A - Chain in the way	When the beam is unable to be detected by the receiver, a 0 in the form of no current will be sent to the microprocessor and then to the code from the microproc.	The sensor can be configured to be active high or active low [1]
Logic-Level: 100mA - Chain not in the way	When the beam is able to be detected, a 1 in the form of 100mA (max) will be received	The sensor can be configured to be active high or active low [1]

Table 4.4.1: Interface Properties

4.4.5 - Verification Process:

For sftwrcd_mrcntrlr_code Output:

1. Other (Memory size): Program size must be less than 8KB

Admittedly, this parameter will not be as necessary to test so much as to avoid. Should we wish to test it, we may simply attempt to upload a 8.1KB or larger file onto the device and see if it

accepts the too-large file.

For `sftwrcd_mrcntrlr_dsig` Output:

1. Logic-Level: 1 (2.8 - 5.5V): Chain needs replacement

Because this is more a parameter of the microcontroller, I will set up a condition to test that there *is* a non zero voltage output when the “send uh oh” condition is met. Because we don’t have a system to rotate a test chain at a specific speed quite yet, I will replicate the passing of a chain with the pushing of a button. I will change the error percentage to ten times the current 3% to accommodate for the fact that I am a mere human and cannot press a button with an exact frequency and would likely deviate by 3% entirely on accident in the testing process. The delivery of a “1” will be observed by the illumination of an LED.

2. Logic-Level: 0 (0V): Chain is fine

This can be tested by setting up the same button-pressing test conditions as above, but by pressing the button with as close-to-consistent time between presses as possible and checking that the output of the used microprocessor (arduino uno for the demonstration/testing purposes) is “0” for the duration and that the LED remains off.

For `mrcntrlr_sftwrcd_dsig` Input:

1. Logic Level: 0A - Chain in the way

I/We will test that the software understands a ~0A input as Logic 0 using a current generator and the software implemented into an arduino uno. A “0” will mean the associated LED should turn on.

2. Logic Level: 100mA - Chain not in the way

This will be tested in the exact same way as the Logic Level 0 above - with an LED in a breadboard and current generator, except this time the LED should turn off when the 100mA signal is received.

4.4.6 - Reference and File Links

- [1] Adafruit Industries, “Ir break beam sensor with premium wire header ends - 5mm leds,” *adafruit industries blog RSS*, 10-Dec-2014. [Online]. Available: <https://www.adafruit.com/product/2168>. [Accessed: 20-Jan-2023].
- [2] ATmega, “ATtiny214/414/814 AVR® Microcontroller with Core Independent Peripherals and picoPower® Technology.” Microchip, Los Angeles, 14-Jan-2017. <http://ww1.microchip.com/downloads/en/DeviceDoc/40001912A.pdf>

4.4.7 - Revision Table

Revision Number	Revision Date	Description	Revision Author
1	3/15	Initial Revision of section 4.4	Corbin

4.5 - CAN Controller:

4.5.1 - Description:

The can block is built with one of the most practical stand-alone CAN controllers, MCP 2515 as the core. This module achieves the goal of communicating with Hyster Forklift's electronics via 5 V digital serial signal. After achieving this goal, we can connect our device to the forklift with only two wires, so that the forklift can receive any information accurately. In this block, in addition to MCP 2515, we will also use TJA 1050 as the transceiver of the block to transmit or receive information transmitted through the CAN bus, an 8M Hz crystal oscillator as a bit rate regulator, an LED as an indicator, and some resistors and capacitors to stabilize the voltage and current of the part. In addition, we will also add two terminal block connectors to allow the output of the signal to adapt to more kinds of wires.

This module mainly meets the requirements of signal output. Our project partner defines the signal output as "Outputs an analog (5 V) or digital signal (CAN or other serial protocol signal)". This means that the voltage range of the electrical signal will be between 0 V and 5 V while making the output signal accurate enough. After receiving data from the MCU block via SPI protocol, this module will convert these data into two strings of signal data, namely can_h and can_l, and transmit them through CAN bus. According to the requirements, the voltage of can_h of this block will be between 2.5 V and 4.5 V, and the voltage of can_l will be between 0.5 V and 2.5 V. In order to achieve this effect, the voltage converter (another block) of the device will provide a voltage of 5 V and a current of at least 16 mA to this block. In addition, the CAN protocol requires that the nominal bit rates of the transmitter and receiver are consistent, so we will use an additional crystal oscillator to achieve the corresponding NBR. After the block meets all the above requirements, the device can access its CAN bus at any position of the forklift to transmit the calculation results of the sensor. It is also worth mentioning that the CAN bus will only be used to transmit electrical signals, so it will not have any current, and it cannot be used as a voltage source.

All the above information and some interface properties not mentioned will be introduced in detail in the following sections, and their verified solutions will be mentioned.

4.5.2 - Design:

The figure 1 below is the black box of the can controller block. It can be seen that this block has two inputs and one output. Among them, mrcntrlr_cn_cntrlr_comm is an input, which is the information given to the can controller block by the MCU block. After the MCU has processed all the data, it will send the result information to the block through the SPI protocol. This input will consist of four parts, CS, SO, SI, and SCK. Another input, pwr_cnvtr_cn_cntrlr_dcpwr, is the voltage input from the power converter to the can controller block. Typically, the voltage input is 5 V and the current is about 16 mA. Depending on the temperature of the environment, this

number may change slightly. `cn_cntrlr_otsd_comm` is the only output that will be connected to the external device, Hyster Forklift's electronics. As mentioned in the previous section, this output will be sent to the device through can protocol using `can_h` and `can_l`.

The schematic of the can controller block is shown in figure 2, which contains all the components that will be used, as well as all input pins and output pins. Among them, TJA 1050 is similar to TJA 1051, but TJA1050 is currently a more widely used transceiver, so this block will use it as a transceiver. Also the resistors connected to the LEDs are variable, depending on the brightness of the LEDs that the project partner expects, but it cannot be lower than 70 ohms or the LED may be broken. It is also worth mentioning that J2 will place the terminal block connector, so our device can adapt to most wires.

Figure 3 shows the PCB of the can controller block. This figure shows the general appearance of the block, including the location and routing of components. Since PCB is converted from schematic, their principles are roughly the same.

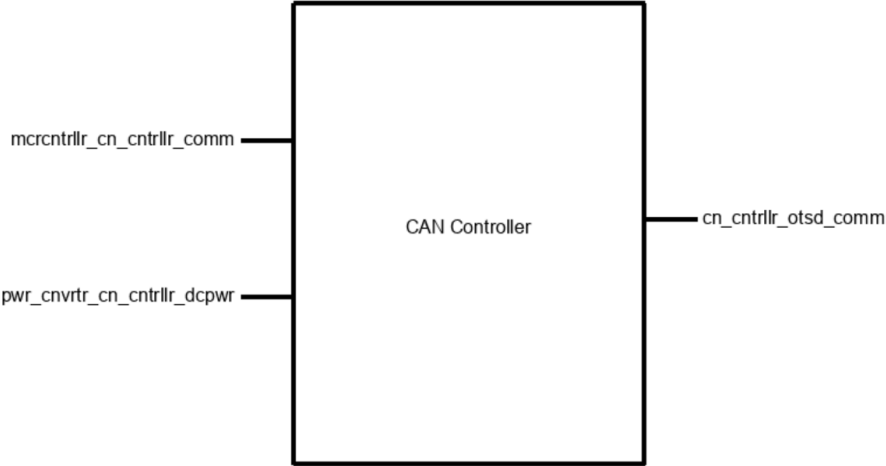


Figure 4.5.1. Black Box of Block

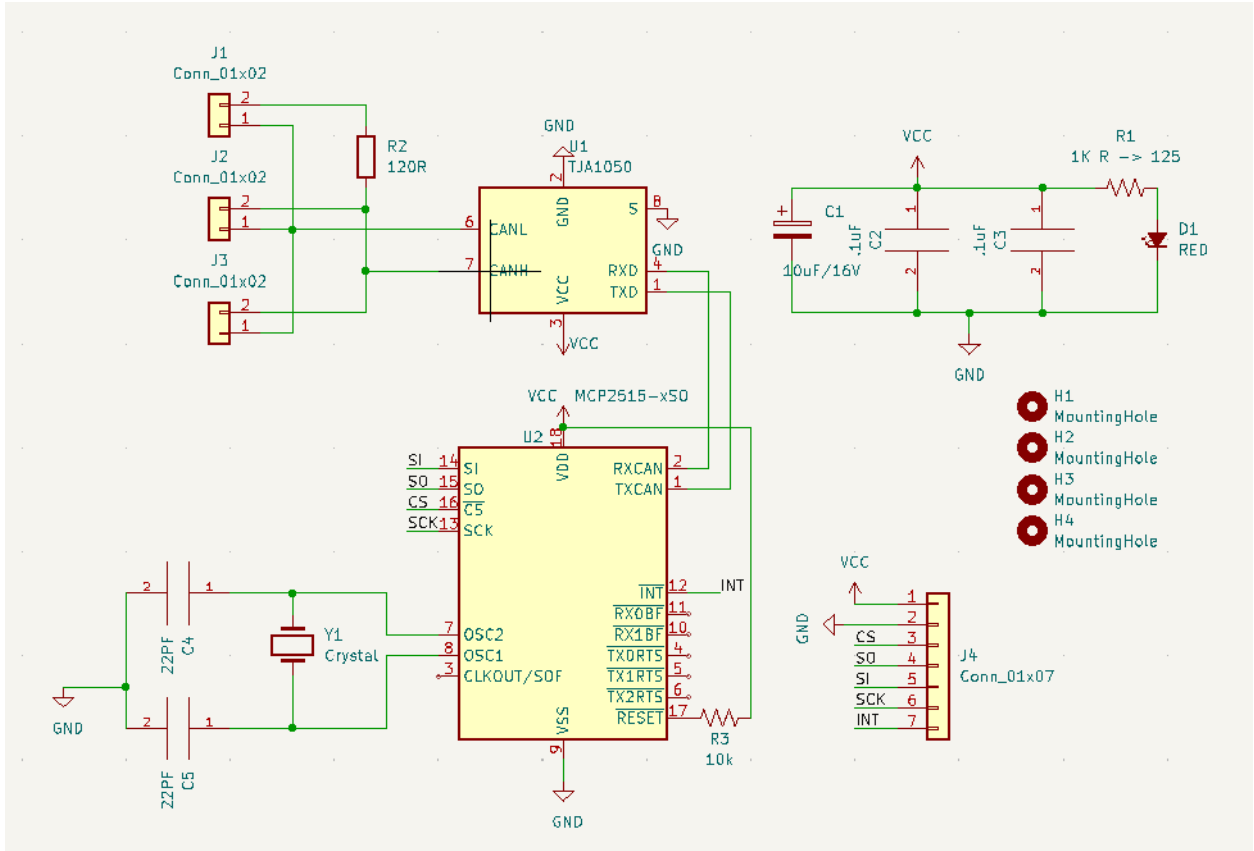


Figure 4.5.2: CAN Controller Block Schematic

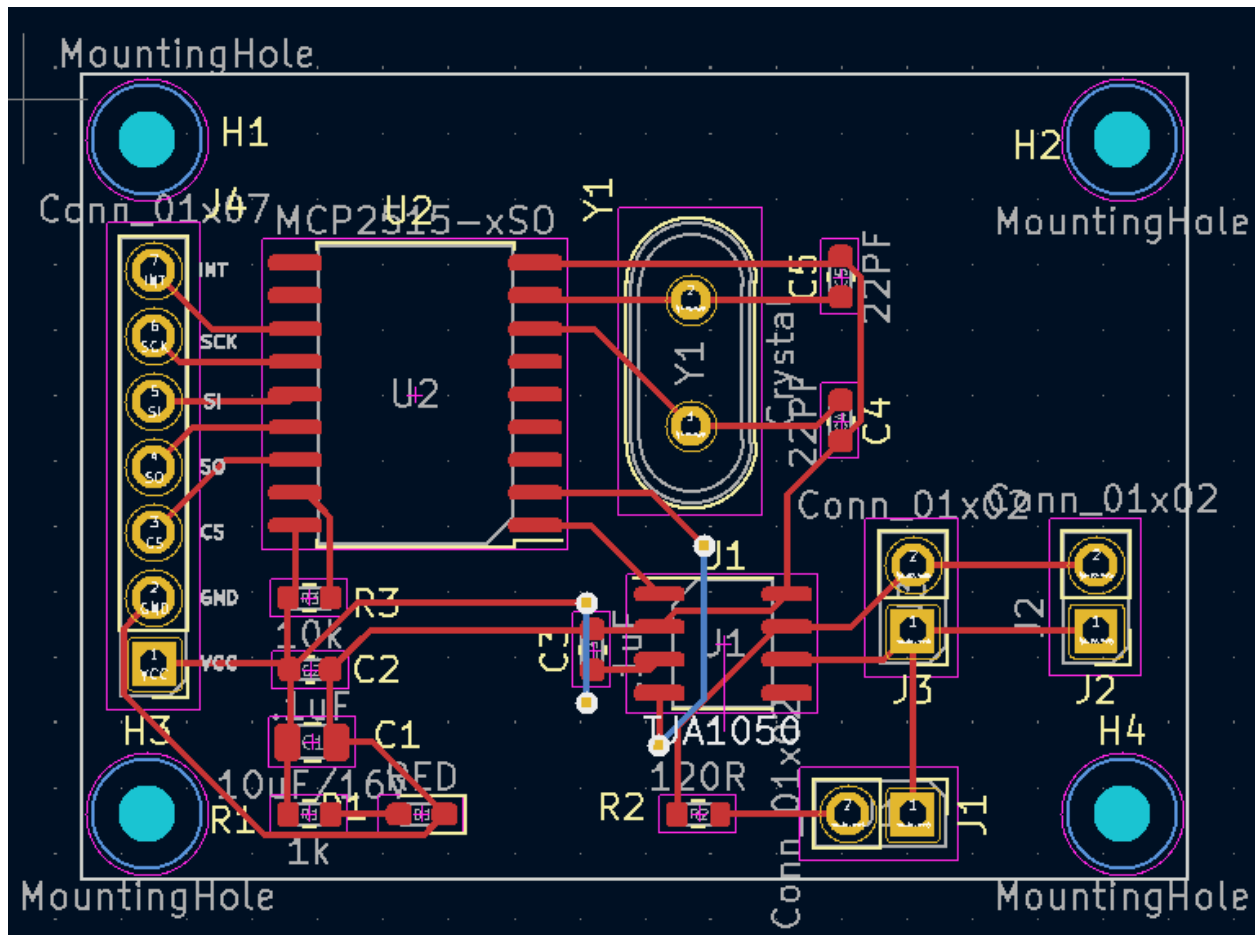


Figure 4.5.3: CAN Controller Block PCB

Notes:

CS, SO, SI, SCK are the input (SPI) for this block, `mrcntrlr_cn_cntrlr_comm`

VCC is the voltage input for this block, `pwr_cnvtr_cn_cntrlr_dcpwr`

CANH and CANL are the output for this block, `cn_cntrlr_otsd_comm`

4.5.3 - General Validation:

One of the main reasons for using MCP 2515 is that it has a SPI protocol interface and it can convert information into CAN protocol signals, so it can accurately receive and translate the information obtained from MCU. For the nominal bit rate required by the project partner, we can achieve it through an external crystal oscillator. Compared to other options, this design is cheap and effective.

Compared with other similar CAN transceivers, TJA 1050 is more widely used, which helps Hyster electronics to receive information timely and accurately. This equipment is required to be at a very low cost, so in many cases, cheapness is a high-priority advantage.

The capacitors near the power supply can stabilize the voltage obtained from the power converter. Through testing, it is found that the MCP 2515 seems to get about 2.7 V voltage through the SPI interface, which means that the voltage at VCC may be unstable, so this design is necessary. In addition, the LED light can quickly detect whether the can controller block is correctly connected to the voltage source.

4.5.4 - Interface Validation:

The Interface Validation section will list all input and output interface properties and how they are obtained, including through experiments or by consulting the datasheet. In addition, the listed content will also be displayed in the video demonstration. The lower table will contain two input interfaces' properties and one output interface's properties, as shown in figure 1.

Interface Property	Why is the interface this value?	Why do you know that your design details <u>for this block</u> above meet or exceed each property?
---------------------------	---	---

mrcntrlr_cn_cntrlr_comm : Input

Messages: team number: 15	The CAN protocol will require each transmitted signal to have an ID, so the receiver can identify the identity of the transmitter and make the correct response (receive or ignore). Our group number is 15, so we set the ID of the signal to 15 (HEX).	The receiver can correctly receive the hexadecimal ID (15) and respond accordingly.
Messages: Logic 0 for when the chain	The receiver needs to receive information about whether the	When the MCU needs to transmit the logic 0 signal to the receiver,

does not need to be replaced	chain needs to be replaced. When the signal received by the receiver is 0, it means that the forklift does not need to replace the chain.	the receiver can correctly receive the signal.
Messages: Logic 1 for when the chain does need to be replaced	The receiver needs to receive information about whether the chain needs to be replaced. When the signal received by the receiver is 1, it means that the forklift needs to replace the chain.	When the MCU needs to transmit the logic 1 signal to the receiver, the receiver can correctly receive the signal.

Pwr_cnvtrr_cn_cntrlr_dcpwr: input

Vmax: 5.5 V	The maximum voltage that both MCP2515 and TJA1051 can withstand is 5.5 V.	MCP2515 datasheet page 74 and TJA1051 datasheet page 2 show the max supply voltage, and both are 5.5 V
Vmin: 2.7 V	The minimum voltage that MCP 2515 can work on is 2.7 V.	MCP2515 datasheet page 74 says the operating voltage is between 2.7 V to 5.5 V..
Ipeak: 92.5mA	In silent mode, TJA 1051 supply maximum current is 2.5 mA, MCP 2515 operating current is 10 mA, and LED peak forward current is 80 mA. All the above components are connected in parallel so the total peak current is 92.5 mA.	TJA1051 datasheet , page 2: supply current ranges from 1-2.5mA in silent mode. LTST-C191KRKT datasheet , page 3: Peak forward current is 80mA. MCP2515 datasheet , page 74: Operating current is 10mA.
Inominal: 16mA	In silent mode, TJA 1051 supply nominal current is 1 mA, MCP 2515 operating current is 10 mA, and LED current is 5 mA since the supply voltage is 5 V and there is a 1K ohm resistor in series with it. All the above components are connected in	TJA1051 datasheet , page 2: supply current ranges from 1-2.5mA in silent mode. LTST-C191KRKT datasheet , page 3: Operating current for a LED with 5V input and a 1k ohm resistor is $V = IR$ where $I = 5\text{mA}$. MCP2515 datasheet , page 74: Operating current is 10mA.

	parallel so the total peak current is 16 mA.	
--	--	--

cn_cntrlr_otsd_comm : Output

Vmax: 4.5V for CAN_H	In silent mode, with 5 V input, the can_h signal is from 2.5 V to 4.5 V(on oscilloscope), so the Vmax should be 4.5 V.	TJA1051 datasheet page 8 shows CAN high pin's output voltage will be from 2.75 V to 4.5 V, which means the maximum voltage output will be 4.5 V.
Vmin: 2.5V for CAN_H	In silent mode, with 5 V input, the can_h signal is from 2.5 V to 4.5 V(on oscilloscope), so the Vmin should be 2.5 V.	TJA1051 datasheet page 8 shows CAN high pin's output voltage will be from 2.75 V to 4.5 V. In the actual 2.5 V situation, the receiver can receive the signal normally.
Vmax: 2.5V for CAN_L	In silent mode, with 5 V input, the can_l signal is from from 0.5 V to 2.5 V (on oscilloscope), so the Vmax should be 2.5 V.	TJA1051 datasheet page 8 shows CAN low pin's output voltage will be from 0.5 V to 2.25 V. In the actual 2.5 V situation, the receiver can receive the signal normally.
Vmin: 0.5V for CAN_L	In silent mode, with 5 V input, the can_l signal is from from 0.5 V to 2.5 V (on oscilloscope), so the Vmin should be 0.5 V.	TJA1051 datasheet page 8 shows CAN low pin's output voltage will be from 0.5 V to 2.25 V, which means the minimum voltage output will be 0.5 V.

Table 4.5.1: Interface Properties

4.5.5 - Verification Process:

1. Verifying 3 messages mrcntrlr_cn_cntrlr_comm (input)
 - a. Solder two CAN controller PCBs, one MCU PCB, and prepare one Arduino UNO board.

- b. Connect one CAN controller to an MCU via SPI lines, and connect another CAN to the UNO board via SPI lines.
 - c. Connect two CAN controllers together via CAN bus.
 - d. Set the UNO board as a receiver to receive signals, and show results on the serial monitor.
 - e. Set MCU as transmitter, and send messages to the receiver. The message should include the three properties mentioned above: ID, logic 0, and logic 1.
 - f. Check whether the receiver can show all messages on the serial monitor correctly.

2. Verify the Vmax and Vmin pwr_cnvtrr_cn_cntrlr_dcpwr (input)
 - a. Repeat steps from 1.a to 1.e.
 - b. Disconnect the power cable between the transmitter's MCU and CAN, and use a DC voltage source to supply voltage to CAN.
 - c. Try to provide a voltage of 5.5 V to the CAN controller, and input information from the transmitter after 30 seconds to see if the receiver can receive this part of the information correctly.
 - d. Try to provide a voltage of 4.5 V to the CAN controller, and input information from the transmitter after 30 seconds to see if the receiver can receive this part of the information correctly.
 - e. Provide a voltage of 5 V to the CAN controller, and check whether the input current is 16 mA.
 - f. Provide a voltage of 5 V to the CAN controller, and set load to 92.5 mA, then check whether the CAN controller still works after 30 seconds.

3. Verifying Vmax, and Vmin cn_cntrlr_otsd_comm (output)
 - a. Repeat steps from 1.a to 1.e.
 - b. Connect CANH to the oscilloscope, and observe the high and low of CANH respectively. These two values correspond to Vmax and Vmin of CANH respectively
 - c. Connect CANL to the oscilloscope, and observe the high and low of CANL respectively. These two values correspond to Vmax and Vmin of CANL respectively

4.5.6 - Reference and File Links

[1] Microchip, "AVR® Microcontroller with Core Independent Peripherals and picoPower® Technology", *Microchip Technology Inc.*, 2017. [Online].

Available: <http://ww1.microchip.com/downloads/en/DeviceDoc/40001912A.pdf> , [Accessed Jan. 19, 2023].

[2] Microchip, “Stand-Alone CAN Controller with SPI Interface”, *Microchip Technology Inc.* , 2019. [Online]. Available: <https://ww1.microchip.com/downloads/en/DeviceDoc/MCP2515-Stand-Alone-CAN-Controller-with-SPI-20001801J.pdf>, [Accessed Jan. 19, 2023].

[3] NXP, “TJA1051 High-speed CAN transceiver”, NXP USA Inc., 2017. [Online]. Available: <https://www.digikey.com/htmldatasheets/production/616039/0/0/1/tja1051.html>, [Accessed Jan. 19, 2023].

[4] Liteon Optoelectronics, “SMD LED Product Data Sheet LTST-C191KRKT”, *Lite-on Technology Corporation*, 2022. [Online]. Available: <https://optoelectronics.liteon.com/upload/download/DS22-2000-223/LTST-C191KRKT.PDF>, [Accessed Feb. 10, 2023].

4.5.7 - Revision Table

Revision Number	Revision Date	Description	Revision Author
1	1/20	Initial Revision	Tianhao
2	5/50	Modified the description about the transceiver	Tianhao

4.6 - Microcontroller:

4.6.1 - Description:

The Microcontroller block to be used in this project is the ATtiny 814. The module will be responsible for collecting and analyzing signals from the sensor, as well as controlling the operation of the signal transmitter. With the ATtiny 814, the project aims to achieve certain targets, such as measuring chain "stretch" within +/- 5% of the actual measurement, costing less than \$10 at a production quantity of 15,000/yr, and outputting an analog (5V) or digital signal (CAN or other serial protocol signal).

However, since most of these requirements will be implemented in other modules, the Microcontroller can assist these blocks through written code or additional components to better meet the set targets. For instance, by adding an external crystal oscillator and some code, the Microcontroller block can accurately calculate the data obtained from the sensor. Similarly, it can also control the output content and bit rate of the transmission module through code.

The ATtiny 814 has 14 ports, which include the Vin port, GND port, 4 spi ports, 1 UPDI port, 2 ports for crystal oscillator, and 5 ports for sensors. Among these ports, the spi interface will be used to connect with the CAN controller, while the UPDI will be used to write the program into the microcontroller. Additionally, the crystal oscillator interface will be used to connect an 8MHz oscillator as an external timer.

Three of the five sensor interfaces will be connected with pull-resistors in parallel to improve the accuracy of the sensor. By doing this, the Microcontroller block can accurately calculate the sensor data to ensure that the project meets its targets.

It is important to note that the price of the entire Microcontroller should not exceed \$1. Since the ATtiny 814 costs only about \$0.75, it leaves room for selecting a better power filter or sensor, among other things. This means that under the limit of \$10, the project can make use of a better sensor or power filter to enhance its performance.

In the following sections, all the information presented above will be discussed in greater detail, along with their verified solutions. This will provide a clearer understanding of the project and how it can be implemented. Overall, the Microcontroller block is an essential component of the project that will enable it to achieve its targets. Through the use of additional components and written code, it can work seamlessly with other modules to ensure the success of the project.

4.6.2 - Design:

The figure 1 below shows the black box of the Microcontroller block. It can be seen that this block has three inputs and two outputs. First of all, `chn_msrmt_snsr_mcrntrlr_dsig` is the input from the sensor block, which is an electrical signal and contains information about the movement of the chain. The logic 0 in the electrical signal represents the chain is blocking the sensor, whereas logic 1 represents the chain does not block the sensor. After transmitting the

information from sensor to the code block through `mrcntrlr_sftwrcd_code` output, code block will calculate the result and send result back to the Microcontroller through `sftwrcd_mrcntrlr_code` input. The result will be a logic number and an ID number. Among them, logic 0 means that the wear degree of the chain is low and does not need to be replaced, whereas logic 1 means that the chain needs to be replaced, and ID number is our group number 15, which is used to allow the receiver to identify the source of the signal. After that, the Microcontroller block will transmit information to the CAN controller block through `mrcntrlr_cn_cntrlr_comm` output. The `pwr_cnvtr_mrcntrlr_dcpwr` input is the power supply to provide 4.5 V to 5 V DC voltage to the Microcontroller.

The figure 2 shows the schematic of the Microcontroller, which contains all the components that will be used, as well as all the input and output pins. The Microchip we choose is the ATtiny 814, which has a low price but has relatively enough ability to complete all tasks we want it to achieve. To meet the computing needs, we added an 8 million Hz crystal oscillator. Compared with internal, this performance will be better. In addition, we also have some capacitors to stabilize the voltage, and three parallel pull up resistors to improve the stability of the sensor.

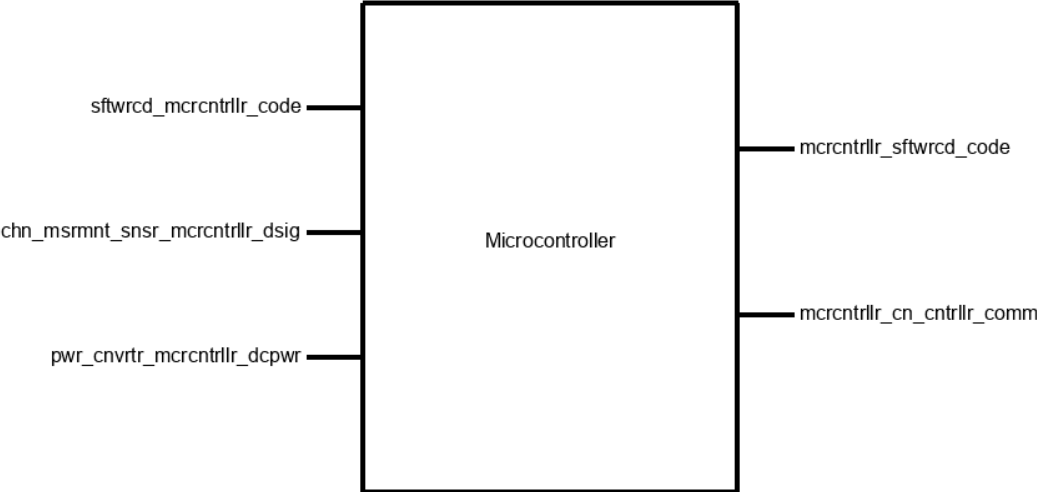


Figure 4.6.1: Black Box of Microcontroller

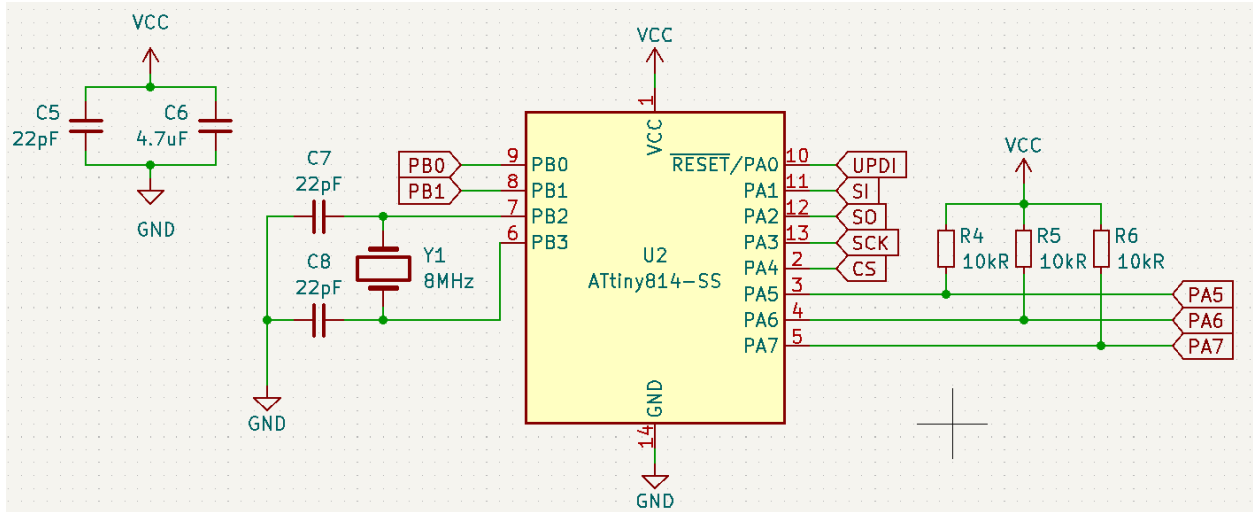


Figure 4.6.2: Microcontroller Block Schematic

Notes:

PA5, PA6, PA7 are the input for this block, `chn_msrmnt_snsr_mcrntrllr_dsig`

SI, SO, SCK, CS are the output for this block, `mcrntrllr_cn_cntrllr_comm`

UPDI is used to program the Microcontroller

PB0 and PB1 are used as a routine test pin

4.6.3 - General Validation:

Compared with ATtiny series chips, other commonly used chips, such as ATmega328P or ESP 32, undoubtedly have larger memory and better performance, but they are often more expensive. ATtiny series chip is a cheap choice, it can meet signal reading, calculation, and control SPI protocol and CAN protocol, so it is a better choice to choose this series of chips. ATtiny814 has 8192 bytes (8K) program storage space and 14 pins, which can just meet the requirements of the code and the number of pins. Based on these considerations, we chose this chip. The only problem is that the internal oscillator of ATtiny 814 is unstable, which means that we may not be able to limit the error value of the calculation results, so we consider adding an external oscillator to meet the calculation error (5%) requirements.

In terms of sensors, we found that a parallel pull-up resistor can increase the accuracy of the sensor. Since we may use existing sensors instead of self-made ones to reduce possible risks

(such as self-made sensors may be unstable or have higher errors), we directly added three pull-up resistors to the sensor pins.

4.6.4 - Interface Validation:

The Interface Validation section will list all input and output interface properties and how they are obtained, including through experiments or by consulting the datasheet. In addition, the listed content will also be displayed in the video demonstration. The lower table will contain two input interfaces' properties and one output interface's properties, as shown in figure 1.

Interface Property	Why is this interface this value?	Why do you know that your design details <u>for this block</u> above meet or exceed each property?
sftwrcd_mrcntrlr_code : Input		
Other: Logic-Level: 0 - Chain is fine	When the software determines the chain is within an acceptable length, a LOW will be sent	The data sheet for the ATtiny214/414/814 lists this value [2]
Other: Polling Loop Statement will continuously transmit data	Every time when softcode is going to transmit data, it is able to display "in loop" to indicate that it is transmitting data.	According to the CAN library[1], CAN is able to display any information during sending data.
Other: Logic-Level: 1 - Chain needs replacement	When the software determines an unacceptable level of lengthening, a HIGH signal will be sent	The data sheet for the ATtiny214/414/814 lists this value [2]
mrcntrlr_sftwrcd_code : Output		
Other: Transport Protocol Initialization Situation will ensure communication/transmission	When the software is going to transmit data, it will first check the connection between receiver and	According to the CAN library [1], CAN will check the connection status and return it.

	transmitter and then will return the connection status	
Other: Logic-Level 1 : Chain is not blocking sensor	When the chain is blocking the sensor, the sensor will send a high voltage signal to the pin so that the softcode will get logic 1.	According to the IR break beam sensor datasheet [3], the sensor will return a high voltage signal when the sensor does not receive any infrared rays.
Other: Logic-Level 0 : Chain is blocking sensor	When the chain is not blocking the sensor, the sensor will send a low voltage signal to the pin so that the softcode will get logic 0.	According to the IR break beam sensor datasheet [3], the sensor will return a low voltage signal when the sensor receives an infrared ray.

mcrctrlr_cn_cntrlr_comm : Output

Messages: Logic 0 for when the chain does not need to be replaced	The receiver needs to receive information about whether the chain needs to be replaced. When the signal received by the receiver is 0, it means that the forklift does not need to replace the chain.	When the MCU needs to transmit the logic 0 signal to the receiver, the receiver can correctly receive the signal.
Messages: team number: 15	The CAN protocol will require each transmitted signal to have an ID, so the receiver can identify the identity of the transmitter and make the correct response (receive or ignore). Our group number is 15, so we set the ID of the signal to 15 (HEX).	The receiver can correctly receive the hexadecimal ID (15) and respond accordingly.
Messages: Logic 1 for when the chain does need to be replaced	The receiver needs to receive information about whether the chain needs to be replaced. When the signal received by the	When the MCU needs to transmit the logic 1 signal to the receiver, the receiver can correctly receive the signal.

	receiver is 1, it means that the forklift needs to replace the chain.	
--	---	--

chn_msrmnt_snsr_mrcntrlr_dsig : Input

Logic 0: Vmin = 0V.	According to the datasheet [2], $0.3 \times V_{DD}$ is the maximum voltage an I/O pin can handle to be considered a LOW logic, and -0.2V is the minimum voltage. Since V_{DD} will be $V_{CC}(5V)$, these values are calculated to be 1.5V and -0.2V.	As long as the sensor has enough time to change the output logic levels, i.e. 2ms, then the logic levels will fall between the range.
Logic 0: Vmax = 1.5V.	According to the datasheet [2], $0.3 \times V_{DD}$ is the maximum voltage an I/O pin can handle to be considered a LOW logic, and -0.2V is the minimum voltage. Since V_{DD} will be $V_{CC}(5V)$, these values are calculated to be 1.5V and -0.2V.	As long as the sensor has enough time to change the output logic levels, i.e. 2ms, then the logic levels will fall between the range.
Logic 1: Vmin = 3.5V	According to the datasheet [2], $V_{DD} + 0.2V$ is the maximum voltage an I/O pin can handle to be considered a HIGH logic and $0.7 \times V_{DD}$ is the minimum voltage. Since V_{DD} will be $V_{CC}(5V)$, these values are calculated to be 5.2V and 3.5V.	As long as the sensor has enough time to change the output logic levels, i.e. 2ms, then the logic levels will fall within the range.
Logic 1: Vmax = 5V	According to the datasheet [2], $V_{DD} + 0.2V$ is the maximum voltage an I/O pin can handle to be considered a HIGH logic	As long as the sensor has enough time to change the output logic levels, i.e. 2ms, then the logic levels will fall within the range.

	and $0.7 \times V_{DD}$ is the minimum voltage. Since V_{DD} will be $V_{CC}(5V)$, these values are calculated to be 5.2V and 3.5V.	
--	--	--

pwr_cnvtr_mrcntrlr_dcpwr : Input

Vmax: 6V	This is the required voltage for our microcontroller to function properly without failure due to high voltage levels.	ATtiny datasheet , page 516: Maximum power supply voltage is 6V.
Vmin: 1.8V	This is the minimum required voltage our microcontroller can take as input voltage.	Vin for the ATtiny can range from 1.8-5.5V, (ATtiny datasheet , page 517).
Ipeak: 200mA	This is the maximum current into the VDD pin.	ATtiny datasheet , page 516: Maximum current into a VDD pin is 200mA.
Inomial: 20mA	Nominal current typically operates at a magnitude lower than Ipeak. Value is arbitrary because the datasheet does not have Inomial for a VDD pin	ATtiny datasheet , page 516: Maximum current into a VDD pin is 200mA.

Table 4.6.1: Interface Properties

4.6.5 - Verification Process:

1. Verifying 3 messages mrcntrlr_cn_cntrlr_comm (output)
 - a. Solder two CAN controller PCBs, one MCU PCB, and prepare one Arduino UNO board.
 - b. Connect one CAN controller to an MCU via SPI lines, and connect another CAN to the UNO board via SPI lines.
 - c. Connect two CAN controllers together via CAN bus.
 - d. Set the UNO board as a receiver to receive signals, and show results on the serial monitor.
 - e. Set MCU as transmitter, and send messages to the receiver. The message should include the three properties mentioned above: ID, logic 0, and logic 1.

- f. Check whether the receiver can show all messages on the serial monitor correctly.
2. Verifying 3 properties `sftwrcd_mrcntrlr_code` : Input
 - a. Prepare one PCB board with sensors, one CAN receiver, one Arduino UNO board, one good chain, and one bad chain.
 - b. Program PCB and UNO board. Connect PCB with CAN receiver and Arduino UNO board.
 - c. After both the PCB and the UNO board are powered on, PCB will start sending information to the UNO board. During this process, see whether PCB is able to return “in loop” information.
 - d. Use the sensor to test the two chains separately, and check whether the UNO board can receive the correct chain status information.
3. Verifying 3 properties `mrcntrlr_sftwrcd_code` : Output
 - a. Prepare one PCB board with sensors, one CAN receiver, one Arduino UNO board, one good chain, and one bad chain.
 - b. Program PCB and UNO board. Connect PCB with CAN receiver and Arduino UNO board.
 - c. When both the PCB and the UNO board are powered on, check whether the serial monitor of the UNO board shows the correct CAN bus connection.
 - d. Cover any sensor with your finger, and check whether the serial monitor of the UNO board can display the correct occlusion result.
4. Verifying 4 properties `chn_msrmnt_snsr_mrcntrlr_dsig` : Input
 - a. Prepare one PCB board with sensors, one CAN receiver, one Arduino UNO board, one good chain, and one bad chain.
 - b. Program PCB and UNO board. Connect PCB with CAN receiver and Arduino UNO board.
 - c. Let the sensor output voltages of different magnitudes by blocking and other methods, and record the pin with a voltmeter to obtain a critical voltage of 1 or 0. Through this method we can find the highest and lowest voltages of output 1 and output 0..
5. Verifying 4 properties `pwr_cnvtr_mrcntrlr_dcpwr` : Input
 - a. Prepare one PCB board with sensors, one CAN receiver, one Arduino UNO board, one good chain, and one bad chain.
 - b. Program PCB and UNO board. Connect PCB with CAN receiver and Arduino UNO board.
 - c. Change the input voltage of the microcontroller to 6V and run for 30s. During the time, check the input current and see whether it is about 200 mA.
 - d. Change the input voltage of the microcontroller to 1.8V and run for 30s.
 - e. Set input voltage to 5V and check the nominal current.

4.6.6 - Reference and File Links

[1] Pillar1989, “Seeed_Arduino_CAN”, *Github*, 2020. [Online].

Available: https://github.com/Seeed-Studio/Seeed_Arduino_CAN/tree/old [Accessed: 21-Jan-2023].

[2] "Microchip technology." [Online]. Available: <https://ww1.microchip.com/downloads/en/DeviceDoc/40001912A.pdf> [Accessed: 21-Jan-2023].

[3] A. Industries, "Ir break beam sensors with premium wire header ends - 3mm leds," *adafruit industries blog RSS*. [Online]. Available: <https://www.adafruit.com/product/2167>. [Accessed: 20-Jan-2023].

4.6.7 - Revision Table

Revision Number	Revision Date	Description	Revision Author
1	1/20	Initial Revision	Tianhao
2	5/14	Revise properties description, verification process and reference links	Tianhao

4.7 - Chain Measurement Sensor:

4.7.1 - Description:

The purpose of this document is to describe the (Chain Measurement) Sensor Block of the ECE44x Chain Elongation Monitor project to 4th-year ECE students with the intention of these students being able to build and verify the block without further research. The Sensor block is implemented using three IR Breakbeam sensors. This document provides an overview of the overall block function including interface properties and a schematic, verification for the design in the form of a step-by-step testing process, and support for the validity of the design in the form of outside research and numerical justification addressing individual properties.

One sensor is placed in the 'middle' middle of the device and will be for measuring the speed of the Chain, while two sensors will be placed on the outer edges of the device and will be used for measuring/detecting the elongation of the chain.

The measurement of speed can be done by measuring the time difference between sensor activation as a chain link crosses the two central sensors and calculating the speed with that delta and the chain link length. For the elongation measurement/monitoring, the outer two sensors are placed with a specific distance between them, X amount of chain links. If positioned correctly, the two sensors will activate simultaneously with a chain that is not elongated and activates with some time delta with a chain that is elongated. This time delta coupled with the calculated speed can provide an elongation measurement.

The idea behind this block is to do the physical testing of the sensors, not calculate the actual measurements. This will assist in the implementation and testing of the "Sensor Accuracy" system requirement by testing the rising and falling times of the digital signal.

4.7.2 - Design:

The sensor block has two inputs and two outputs. The first input is from the Power Regulator block, providing power for the sensors to function. The second input interface is a chain whose properties can be determined whenever an individual chain link breaks the IR beam of the sensor. This means the "measuring" is just detecting the break and changing the logic level of the sensor's output. The two outputs are practically one and the same. Both interfaces deal with the digital signal being sent from the sensor to the microcontroller.



Figure 4.7.1: Black Box Image of the Sensor Block

Since three sensors will be needed to make the proper measurements later, all three sensors are included in this block. Below is a “wiring” diagram for this.

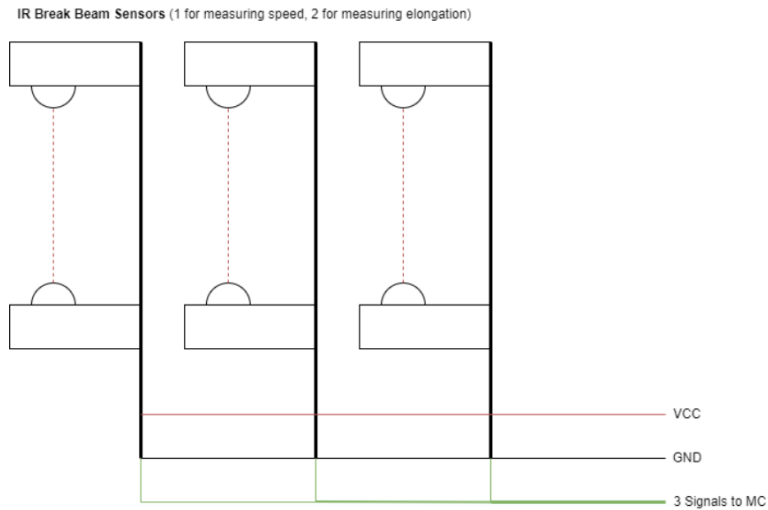


Figure 4.7.2: Schematic / Wiring Diagram of the Sensor Block

To assist with the testing, I made a quick CAD of a mount that will keep the two parts of the sensor facing each other. This version will not be the final design used for the complete device but it will be useful in the development of the enclosure.

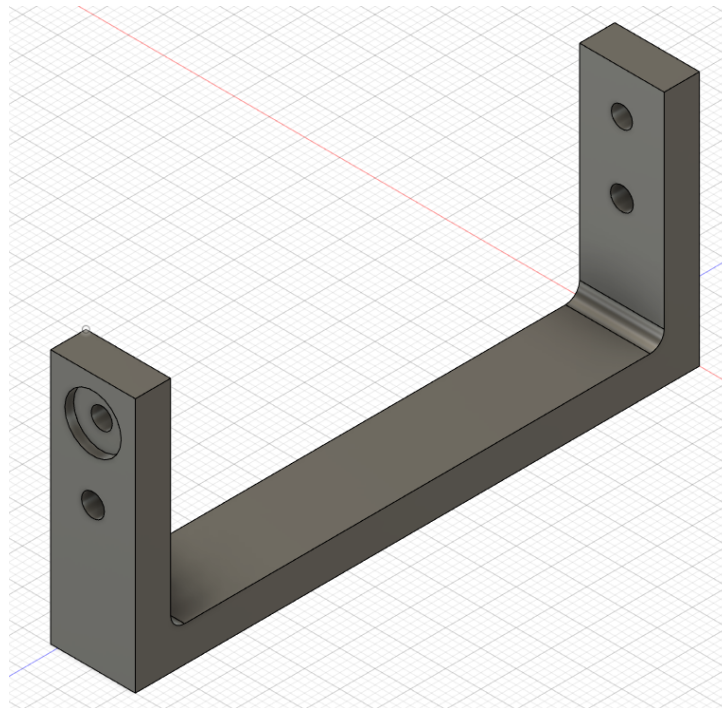


Figure 4.7.3: Image of the CAD Sensor Bracket

Interface	Properties
otsd_chn_msrmt_snsr_envin	Other: Detect a break in between chain links Other: Max Speed 3m/s Other: Min Speed 0m/s
pwr_cvrtr_chn_msrmt_snsr_dcpwr	V_{max} : 5.5V ^[1] V_{min} : 4.5V $I_{nominal}$: 60mA (20mA/Sensor) ^[1] I_{peak} : 90mA (30mA/Sensor) ^[1]
chn_msrmt_snsr_mrcntrlr_dsig	Fall Time: Max 2 ms ^[1] Logic-Level: Logic 1: Between 3.5V and 5.2V ^[2] Logic-Level: Logic 0: Between -0.2V and 1.5V ^[2] Rise Time: Max 2ms ^[1]
chn_msrmt_snsr_fnl_pcb_dsig	Fall Time: Max 2 ms Logic-Level: Logic 1: Between 3.5V and 5.2V Logic-Level: Logic 0: Between -0.2V and 1.5V Rise Time: Max 2ms

Table 4.7.1: Interface Properties

4.7.3 - General Validation:

The idea for using break beam sensors comes from a visual analysis of existing chain elongation monitors that exist, specifically the monitor from IWIS [3]. There seem to be four sensors that seem to shine an IR light or laser through the gaps that exist between chain links. The goal was to replicate this device as best as possible, though, with little information about the actual function, most information is assumed.

4.7.4 - Interface Validation:

otsd_chn_msrmt_snsr_envin

Interface Property	Why is this interface this value?	Why do you know that your design details <u>for this block</u> above meet or exceed each property?
Other: Detect a break in between chain links	The core functionality of the sensors is to detect when a chain link passes the sensor, breaking the IR Beam.	Chains only have gaps between the chain links a break beam sensor should be able to detect a gap or a chain link.

Interface Property	Why is this interface this value?	Why do you know that your design details <u>for this block</u> above meet or exceed each property?
Other: Max Speed 3m/s	The project partner is the Hyster-Yale Group which focuses on the design of Forklifts. This value was given by the project partner.	According to the datasheet for the IR Breakbeam sensor, the sensor has a response time of less than 2ms, and the shortest distance on the chain is 6.5mm, this means the maximum speed that the sensor can function under is 3.25 m/s.
Other: Min Speed 0m/s	The minimum chain speed is set to this value as the slowest the chain will ever move is when the chain is at rest i.e 0m/s	If the chain is not moving, the State of the beam (broken and unbroken) does not change, this fact can be used to determine that the chain is not moving.

Table 4.7.2: Interface Property Validations for the Chain (Input) Interface

pwr_cnvtr_chn_msrmt_snsr_dcpwr

Interface Property	Why is this interface this value?	Why do you know that your design details <u>for this block</u> above meet or exceed each property?
Vmax: 5.0V	Max operating voltage of the sensor is 5.5V, but we will only be supplying a max of 5.0V. Maximum voltage is not listed in the datasheet or product description.	C2916 datasheet , page 1: Operating voltage is between 4.5-5.0V.
Vmin: 4.5V	Minimum operating voltage is 4.5V.	C2916 datasheet , page 1: Operating voltage is between 4.5-5.0V..
Ipeak: 90mA	Emitter current draw is between 10mA at 3.3V and 20mA at 5.5V. Including the startup current of 10mA. We have three sensors so $10\text{mA} + 20\text{mA} = 30\text{mA}$, $30 \times 3 =$	C2916 product description : Emitter current draw is 10mA at 3.3V and 20mA at 5.5V. C2916 datasheet , page 1: Start-up current is 10mA.

	90mA.	
Inominal: 60mA	Emitter current draw is between 10mA at 3.3V and 20mA at 5.5V. Including the startup current of 10mA. We have three sensors so $10\text{mA} + 10\text{mA} = 20\text{mA}$, $20 \times 3 = 60\text{mA}$.	C2916 product description : Emitter current draw is 10mA at 3.3V and 20mA at 5.5V. C2916 datasheet , page 1: Start-up current is 10mA.

Table 4.7.3: Interface Property Validations for the VCC (Input) Interface

chn_msrmnt_snsr_mrcntrllr_dsig

Interface Property	Why is this interface this value?	Why do you know that your design details <u>for this block</u> above meet or exceed each property?
Logic 0: $V_{\min} = 0\text{V}$.	According to the datasheet [2], $0.3 \times V_{\text{DD}}$ is the maximum voltage an I/O pin can handle to be considered a LOW logic, and -0.2V is the minimum voltage. Since V_{DD} will be $V_{\text{CC}}(5\text{V})$, these value calculates to be 1.5V and -0.2V .	As long as the sensor has enough time to change the output logic levels, i.e. 2ms, then the logic levels will fall between the range.
Logic 0: $V_{\max} = 1.5\text{V}$.	According to the datasheet [2], $0.3 \times V_{\text{DD}}$ is the maximum voltage an I/O pin can handle to be considered a LOW logic, and -0.2V is the minimum voltage. Since V_{DD} will be $V_{\text{CC}}(5\text{V})$, these value calculates to be 1.5V and -0.2V .	As long as the sensor has enough time to change the output logic levels, i.e. 2ms, then the logic levels will fall between the range.
Logic 1: $V_{\min} = 3.5\text{V}$	According to the datasheet [2], $V_{\text{DD}} + 0.2\text{V}$ is the maximum voltage an I/O pin can handle to be considered a HIGH logic and $0.7 \times V_{\text{DD}}$ is the minimum voltage. Since V_{DD} will be $V_{\text{CC}}(5\text{V})$, these value calculates to be 5.2V and 3.5V .	As long as the sensor has enough time to change the output logic levels, i.e. 2ms, then the logic levels will fall within the range.

Logic 1: Vmax = 5V	According to the datasheet [2], $V_{DD} + 0.2V$ is the maximum voltage an I/O pin can handle to be considered a HIGH logic and $0.7 \times V_{DD}$ is the minimum voltage. Since V_{DD} will be $V_{CC}(5V)$, these value calculates to be 5.2V and 3.5V.	As long as the sensor has enough time to change the output logic levels, i.e. 2ms, then the logic levels will fall within the range.
--------------------	--	--

Table 4.7.4: Interface Property Validations for the Sensors_to_MC Interface

chn_msrmt_snsr_fnl_pcb_dsig

Interface Property	Why is this interface this value?	Why do you know that your design details for this block above meet or exceed each property?
Logic 0: Vmin = 0V.	According to the datasheet [2], $0.3 \times V_{DD}$ is the maximum voltage an I/O pin can handle to be considered a LOW logic, and $-0.2V$ is the minimum voltage. Since V_{DD} will be $V_{CC}(5V)$, these value calculates to be 1.5V and $-0.2V$.	As long as the sensor has enough time to change the output logic levels, i.e. 2ms, then the logic levels will fall between the range.
Logic 0: Vmax = 1.5V.	According to the datasheet [2], $0.3 \times V_{DD}$ is the maximum voltage an I/O pin can handle to be considered a LOW logic, and $-0.2V$ is the minimum voltage. Since V_{DD} will be $V_{CC}(5V)$, these value calculates to be 1.5V and $-0.2V$.	As long as the sensor has enough time to change the output logic levels, i.e. 2ms, then the logic levels will fall between the range.
Logic 1: Vmin = 3.5V	According to the datasheet [2], $V_{DD} + 0.2V$ is the maximum voltage an I/O pin can handle to be considered a HIGH logic and $0.7 \times V_{DD}$ is the minimum voltage. Since V_{DD} will be $V_{CC}(5V)$, these value calculates to be 5.2V and 3.5V.	As long as the sensor has enough time to change the output logic levels, i.e. 2ms, then the logic levels will fall within the range.

Logic 1: $V_{max} = 5V$	According to the datasheet [2], $V_{DD} + 0.2V$ is the maximum voltage an I/O pin can handle to be considered a HIGH logic and $0.7 \times V_{DD}$ is the minimum voltage. Since V_{DD} will be $V_{CC}(5V)$, these value calculates to be 5.2V and 3.5V.	As long as the sensor has enough time to change the output logic levels, i.e. 2ms, then the logic levels will fall within the range.
-------------------------	--	--

Table 4.7.5: Interface Property Validations for the Sensors_to_MC Interface

4.7.5 - Verification Process:

To verify the power/voltage based interfaces, I will use a variable power supply to vary the voltage levels and measure the output voltage of the signal wire. If the output voltage falls within the correct ranges, then the test is a pass. This will have the secondary effect of verifying the output properties if I let the output signals sit for an extended period of time. Using the power supply will also show me if the current properties are being met. If the current is set to I_{peak} and never exceeds $I_{nominal}$ then I will consider the current properties met.

To verify the Chain Sensing Properties, I will wire the sensors to a set of LEDs to visually show the sensors being activated. The sensors will then be moved over a chain to show that the sensor is able to detect the chain link and gaps between chain links, proving the first property. The second and third properties will be verified by attaching the sensor to a test rig consisting of a motor with a small piece of wood attached to it. The wood will pass through the sensor and by varying the speed, I can show that the sensor continues to function at 0m/s and 3m/s. A digital tachometer will be used to verify the speed of the test rig.

4.7.6 - Reference and File Links

- [1] A. Industries, "Ir break beam sensors with premium wire header ends - 3mm leds," *adafruit industries blog RSS*. [Online]. Available: <https://www.adafruit.com/product/2167>. [Accessed: 20-Jan-2023].
- [2] "Microchip technology." [Online]. Available: <https://ww1.microchip.com/downloads/aemDocuments/documents/MCU08/ProductDocuments/DataSheets/ATtiny212-214-412-414-416-DataSheet-DS40002287A.pdf>. [Accessed: 21-Jan-2023].
- [3] "Digital 40 roller chain WEAR GAUGE: Chain Monitoring System," *www.usarollerchain.com*. [Online]. Available: <https://www.usarollerchain.com/Digital-40-Roller-Chain-Wear-Gauge-p/4947-40008897-cm-40.htm>. [Accessed: 20-Jan-2023].

4.7.7 - Revision Table

Revision Number	Revision Date	Description	Revision Author
1	1/20	Initial Revision	Kai
2	3/14	Update with new interface properties and validations	Kai
3	5/14	Updated the Interface Properties and Verification Processes to match the finalized properties. Updated References and File Links	Kai

4.8 - Final PCB:

4.8.1 - Description:

The Final PCB contains most of the circuits for all of the electronics blocks of the system. These circuits include the Power Filter, Power Converter, Microcontroller, and CAN Controller Blocks. This means that all of the outward-facing interfaces for these blocks, such as the sensor, power, and CAN connectors, are the interfaces for this block.

To keep the system compact and to reduce the cost of the PCB, the system requirement associated with this block is the one that limits the size of the PCB to 6 in². The choice to put nearly all of the circuits onto one PCB guarantees that the Universal constraints of having a “student-designed PCB” and “no more than 50% built from purchased modules” are met.

4.8.2 - Design:

This block is focused on the design of the PCB and not the design of the schematic as the circuits were created/provided from the other blocks. The only schematic changes come from the combination of these separate circuits into one.

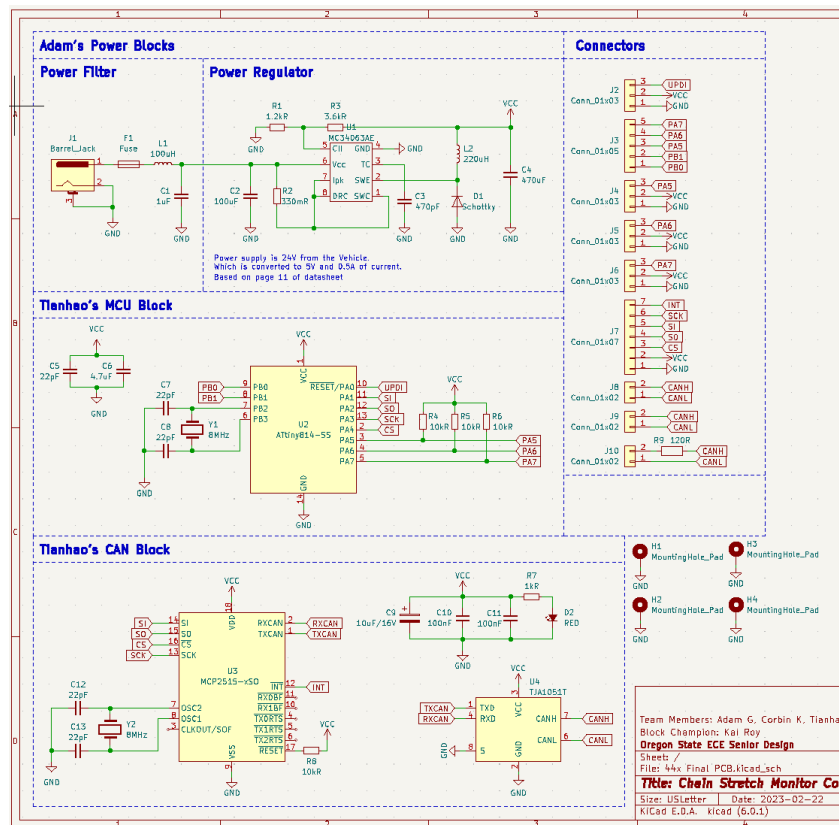


Figure 4.8.1: Schematic of the Final PCB Block

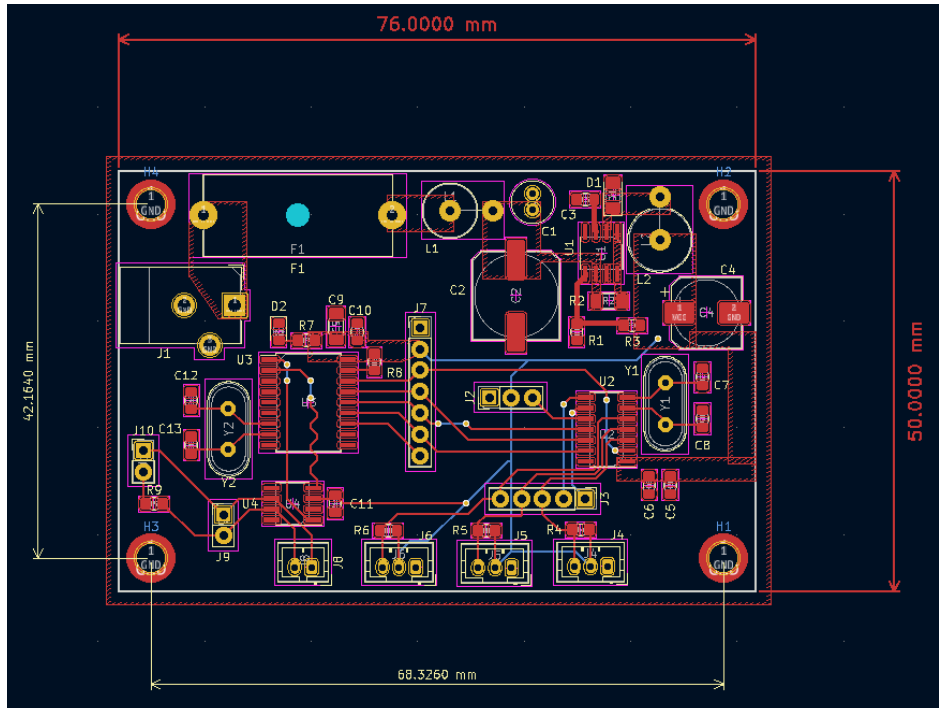


Figure 4.8.2: Schematic of the Final PCB Block

JST-PH connectors were chosen for the CAN and Sensor connectors as one of our group members has a collection of those connectors. A standard DC power jack was chosen for the 24V Power In to the system. A 20mm Fuse holder was added to add some protection for the circuit since this system is a prototype designed to be used on a Forklift. Finally, male pin headers were added throughout the PCB to serve as test points for easier troubleshooting with the lab equipment.

Interface	Properties
otsd_fnl_pcb_dcpwr	Inominal: 100mA Ipeak: 500mA Vmax: 24V Vmin: 15V
fnl_pcb_otsd_comm	Messages: The Chain detection result Protocol: CAN Vmax: 4.5V Vmin: 0.5V
chn_msrmnt_snsr_fnl_pcb_dsig	Logic-Level: Logic 0: Vmax = 1.5V Logic-Level: Logic 0: Vmin = 0V Logic-Level: Logic 1: Vmax = 5V Logic-Level: Logic 1: Vmin = 3.5V

Table 4.8.1: Interface Properties

4.8.3 - General Validation:

The physical dimensions of the PCB are 2in x 3in, making the overall footprint 6in², meeting the system requirement. The PCB also contains every electrical circuit except the chain measurement sensors making the system meet the “no more than 50% purchased” universal constraint.

4.8.4 - Interface Validation:

otsd_fnl_pcb_dcpwr

Interface Property	Why is this interface this value?	Why do you know that your design details for this block above meet or exceed each property?
Vmax: 27V	Forklift battery is a 24V battery and start-up voltage is taken into account to make the value about 27V and 17V for our original minimum 15V. (should expect a voltage spike of about 2-3V during start-up)	MC34063A-E datasheet [1], page 11: Expected input voltage is 24V. Max input voltage is 50V in the datasheet page 5. Start-up voltage is between 1.5-2.1V in the datasheet page 7.
Vmin: 17V	Start-up voltage makes the value 17V instead of 15V. Value is arbitrary because too low of a voltage will not provide enough current to supply the microcontroller, CAN controller, and sensors.	MC34063A-E datasheet [1], page 11: Expected input voltage is 24V. Start-up voltage is between 1.5-2.1V in the datasheet page 7.
Ipeak: 100mA	Based on the schematic in the datasheet.	MC34063A-E datasheet [1], page 5: Maximum driver collector current is 100mA. Switch current can take 1.5A max Inominal would be 150mA (1 magnitude smaller).
Inominal: 10mA	Nominal current typically operates at a magnitude lower than Ipeak for a buck converter. Value is arbitrary because the datasheet does not have Inominal for the driver collector current.	MC34063A-E datasheet [1], page 5: Maximum driver collector current is 100mA. Switch current can take 1.5A max Inominal would be 150mA (1 magnitude smaller).

Table 4.8.2: Interface Property Validations for the Power Input Interface

Fnl_pcb_otsd_comm

Interface Property	Why is this interface this value?	Why do you know that your design details <u>for this block</u> above meet or exceed each property?
Vmax: 4.5V for CAN_H	In silent mode, with 5 V input, the can_h signal is from 2.5 V to 4.5 V(on oscilloscope), so the Vmax should be 4.5 V.	TJA1051 datasheet [2] page 8 shows CAN high pin's output voltage will be from 2.75 V to 4.5 , which means the maximum voltage output will be 4.5 V.
Vmin: 2.5V for CAN_H	In silent mode, with 5 V input, the can_h signal is from 2.5 V to 4.5 V(on oscilloscope), so the Vmin should be 2.5 V.	TJA1051 datasheet [2] page 8 shows CAN high pin's output voltage will be from 2.75 V to 4.5. In the actual 2.5 V situation, the receiver can receive the signal normally.
Vmax: 2.5V for CAN_L	In silent mode, with 5 V input, the can_l signal is from from 0.5 V to 2.5 V (on oscilloscope), so the Vmax should be 2.5 V.	TJA1051 datasheet [2] page 8 shows CAN low pin's output voltage will be from 0.5 V to 2.25 V. In the actual 2.5 V situation, the receiver can receive the signal normally.
Vmin: 0.5V for CAN_L	In silent mode, with 5 V input, the can_l signal is from from 0.5 V to 2.5 V (on oscilloscope), so the Vmin should be 0.5 V.	TJA1051 datasheet [2] page 8 shows CAN low pin's output voltage will be from 0.5 V to 2.25 V, which means the minimum voltage output will be 0.5 V.

Table 4.8.3: Interface Property Validations for the CAN Output Interface

Chn_msrmnt_snsr_fnl_pcb_dsig

Interface Property	Why is this interface this value?	Why do you know that your design details <u>for this block</u> above meet or exceed each property?
Logic 0: $V_{min} = 0V$.	According to the datasheet [2], $0.3 \times V_{DD}$ is the maximum voltage an I/O pin can handle to be considered a LOW logic, and $-0.2V$ is the minimum voltage. Since V_{DD} will be $V_{CC}(5V)$, these value calculates to be $1.5V$ and $-0.2V$.	As long as the sensor has enough time to change the output logic levels, i.e. $2ms$, then the logic levels will fall between the range.
Logic 0: $V_{max} = 1.5V$.	According to the datasheet [2], $0.3 \times V_{DD}$ is the maximum voltage an I/O pin can handle to be considered a LOW logic, and $-0.2V$ is the minimum voltage. Since V_{DD} will be $V_{CC}(5V)$, these value calculates to be $1.5V$ and $-0.2V$.	As long as the sensor has enough time to change the output logic levels, i.e. $2ms$, then the logic levels will fall between the range.
Logic 1: $V_{min} = 3.5V$	According to the datasheet [2], $V_{DD} + 0.2V$ is the maximum voltage an I/O pin can handle to be considered a HIGH logic and $0.7 \times V_{DD}$ is the minimum voltage. Since V_{DD} will be $V_{CC}(5V)$, these value calculates to be $5.2V$ and $3.5V$.	As long as the sensor has enough time to change the output logic levels, i.e. $2ms$, then the logic levels will fall within the range.
Logic 1: $V_{max} = 5V$	According to the datasheet [2], $V_{DD} + 0.2V$ is the maximum voltage an I/O pin can handle to be considered a HIGH logic and $0.7 \times V_{DD}$ is the minimum voltage. Since V_{DD} will be $V_{CC}(5V)$, these value calculates to be $5.2V$ and $3.5V$.	As long as the sensor has enough time to change the output logic levels, i.e. $2ms$, then the logic levels will fall within the range.

Table 4.8.4: Interface Property Validations for the Sensor Input Interface

4.8.5 - Verification Process:

The same verification processes are used in the Sensor/MCU, Power Filter, and CAN Blocks.

Otsd_fnl_pcb_dcpwr

Verifying Vmax, Vmin, Ipeak, and Inominal Pwr_fltr_pwr_cnvtr_dcpwr (input)

1. Set up prototype board with schematic provided from the [MC34063A-E datasheet](#).
Important Note (Must read): Use Figure 2 for reference because the original schematic in Figure 1 has pins backwards.
2. Plug connectors to power supply and DC load.
3. Connect supply to Vcc and GND.
4. Connect load to GND and Vout.
5. Set load to 500mA and set channel 1 on the power supply to 27V.
6. Inspect the power supply that 27V is being properly supplied to the circuit and there is no short circuit on the supply side (no current going in about 0A).
7. Inspect load that 500mA and 5V is being displayed on the load screen.
8. Shut off load and supply to let circuit components cool down.
9. Repeat steps 1.5-1.8 for Vmin and Inominal (set supply voltage and current to 17V/10mA set load to 10mA). Note, Inominal values must hold their value for 30 seconds.
10. Note: if connections are not long enough, use alligator clips as needed or other connector types.

Fnl_pcb_otsd_comm

Verifying Vmax, and Vmin cn_cntrlr_otsd_comm (output)

1. Solder two CAN controller PCBs, one MCU PCB, and prepare one Arduino UNO board.
2. Connect one CAN controller to an MCU via SPI lines, and connect another CAN to the UNO board via SPI lines.
3. Connect two CAN controllers together via CAN bus.
4. Set the UNO board as a receiver to receive signals, and show results on the serial monitor.
5. Set MCU as transmitter, and send messages to the receiver. The message should include the three properties mentioned above: ID, logic 0, and logic 1.
6. Repeat steps from 1.a to 1.e.
7. Connect CANH to the oscilloscope, and observe the high and low of CANH respectively. These two values correspond to Vmax and Vmin of CANH respectively
8. Connect CANL to the oscilloscope, and observe the high and low of CANL respectively. These two values correspond to Vmax and Vmin of CANL respectively

Chn_msrmnt_snsr_fnl_pcb_dsig

To verify the Sensor Properties, the PCB will be programmed to output the state of the sensors via the CAN interface. The sensors will then be moved over a chain to show that the sensor is able to detect the chain link and gaps between chain links. An oscilloscope will be connected to the male pin headers on the PCB to show the voltage level of the signal wire during each state of the sensors. If this value falls in between the voltage ranges for HIGH and LOW logic, then these properties have been met.

4.8.6 - Reference and File Links

- [1] STMicroelectronics, "DC-DC converter control circuits," DocID5257, Nov. 20, 2007. [Online]; ST life augmented: Worldwide, 2020, URL: <https://www.st.com/content/ccc/resource/technical/document/datasheet/03/f9/c4/3d/7f/eb/4c/5e/CD00001232.pdf/files/CD00001232.pdf/jcr:content/translations/en.CD00001232.pdf> (May 14, 2023).
- [2] NXP Semiconductors, "TJA1051 High-speed CAN transceiver," Data Sheet number, Nov. 28, 2017. [Online]; NXP Semiconductors: USA, 2017, URL: <https://www.digikey.com/htmldatasheets/production/616039/0/0/1/tja1051.html> (May 14, 2023).

4.8.7 - Revision Table

Revision Number	Revision Date	Description	Revision Author
1	3/14	Initial Revision	Kai
2	5/14	Updated the interface properties and validations to match the finalized properties. Updated References and File Links. Updated Verification Process	Kai

Section 5 - System Verification Evidence

5.1 - Universal Constraints:

5.1.1 - The system may not include a breadboard:

Our design meets this constraint because our final design currently is on a PCB with no breadboards used. Since all of the non-sensor circuits reside on a single PCB and the Sensor array connects directly to the PCB. No breadboards are needed in the system.

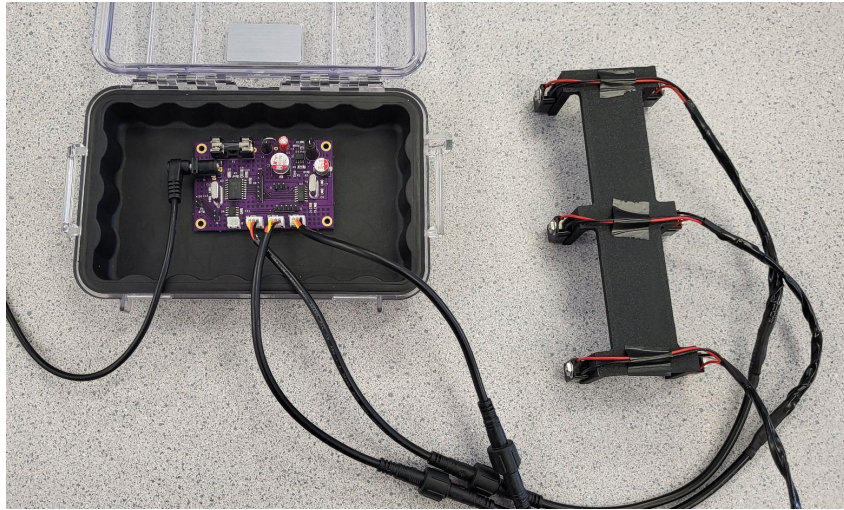


Figure 5.1.1.1 - Evidence showing that the electronics of the system does not use a breadboard

5.1.2 - The final system must contain a student designed PCB:

Our design meets this constraint because one of our blocks is called Final PCB and all relevant documentation is in [Section 4.8](#).

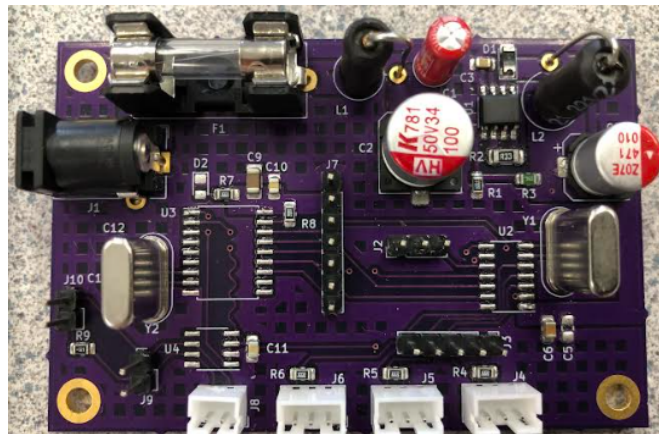


Figure 5.1.2.1 - Evidence showing that our system contains a student designed PCB

5.1.3 - All connections to PCBs must use connectors:

Our design meets this constraint because our PCB has no wires attached by soldering. Refer to [Section 4.8](#) for evidence.

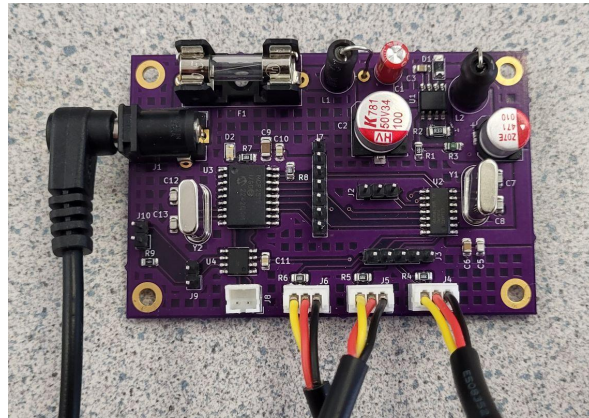


Figure 5.1.3.1 - Evidence showing that all connections to the PCB involve connectors of some kind

Since all non-sensor circuits reside on a single PCB, which has dedicated connectors for power input, CAN output, and the sensor array. There are no connections that are soldered directly to the PCB. Even in the scenario where one of the circuits does not work on the PCB, Pin Headers were added to ensure that we could attach our testing modules to the PCB without needing to solder any wires to the PCB.

5.1.4 - All power supplies in the system must be at least 65% efficient:

Our design theoretically meets this constraint because our power supply is the power converter block and is rated to be 80% efficient with 25V/125mA input and 5V/500mA output. Reference is under [4.1.6 - References and File Links](#). Video evidence is provided on this [link](#).

To verify the efficiency of our power supply, we will remove all of the non-power related ICs/Circuits from the PCB to isolate the power supply circuit. We will then connect the input to a variable power supply set to 24V and an I_{peak} of 250mA and a 5V output to an electronic load. We will then test a variety of output currents and record the input current draw and the output voltage. We will start with 200mA which is our approximate $I_{nominal}$ based on the specifications for the other circuits and end at 500mA, which is our I_{peak} for our power circuit.

V_in (V)	I_in (mA)	V_out (V)	I_out (mA)	Efficiency
24	58	4.75	200	68.25%
24	84	4.66	300	69.35%
24	112	4.53	400	67.41%
24	140	4.41	500	65.63%

Table 5.1.4.1 - Table of Power Supply Efficiency

As the table shows, the Power supply efficiency is at least 65% across all output current draws meaning that we have met this universal requirement.

5.1.5 - The system may be no more than 50% built from purchased ‘modules’:

Our design meets this constraint as the only part that purchased ‘modules’ is the Enclosure, which we modified, the waterproof connectors that connect the sensor array to the enclosure/PCB, and the individual break-beam sensors that attach to our custom bracket. Everything else resides on a custom, student designed PCB. This includes the [power converter](#), [power filter](#), [microcontroller](#), and [CAN controller](#) circuits. By the measure of anything modified is considered built instead of pre-purchased, then our system is not built using any purchased ‘modules’.

Block	Power Converter	Power Filter	Enclosure	Code	CAN Controller	Chain Sensor	Final PCB
Built v. Bought	Built	Built	Built (Modified)	Built	Built	Built (Bracket)	Built

Table 5.1.5.1 - Table showing what blocks have been built and what blocks have been bought

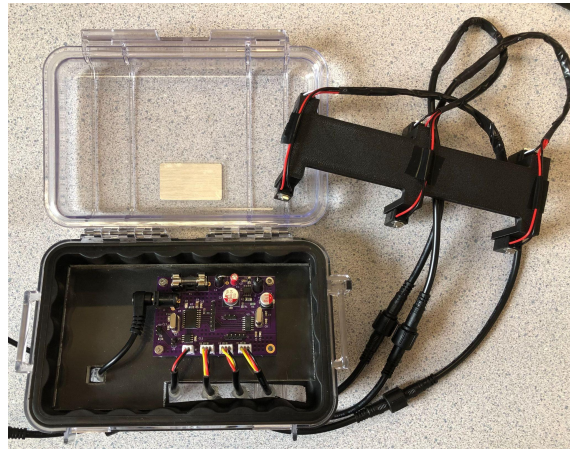


Figure 5.1.5.2 - Evidence showing that all ‘blocks’ have either been made or modified

5.2 - System Requirements:

5.2.1 - PCB:

5.2.1.1 - Project Partner Requirement:

Cost less than \$10 or less at production quantity of 15,000/yr

5.2.1.2 - Engineering Requirement:

The system will have a single PCB that is no larger than 6 in² and contains all non-sensor circuits.

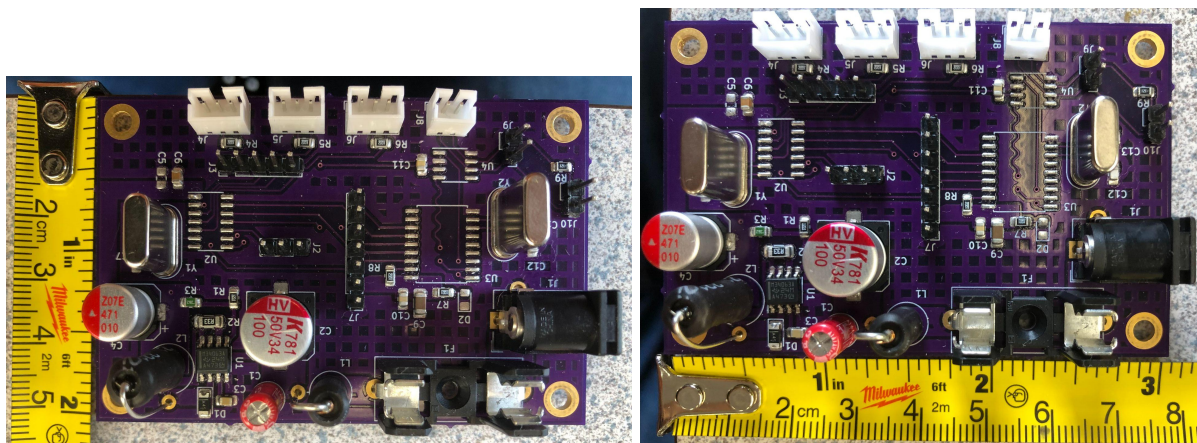
5.2.1.3 - Verification Process:

This requirement can be verified by measuring the dimensions of the PCB as well as inspecting that all the non-sensor circuits reside on the PCB. The dimensions can be measured in the design files themselves, or by measuring the physical PCB using a ruler or calipers. An inspection of the schematic and/or physical board shows that all of the non-sensor circuits, i.e. the Power Filter, Power Converter, Microcontroller, and CAN Controller circuits all reside on the PCB.

5.2.1.4 - Testing Evidence:

According to the PCB file, the PCB has the dimensions of 76mm x 50mm which is 3600mm². In inches these dimensions are approximately 2.99 in x 1.97 in (5.89 in²), which meets this system requirement.

According to the Schematic File, this PCB contains the following circuits/blocks: Power Filter, Power Converter, Microcontroller, CAN Controller.



Images of these files can be found in [Section 4.8.2](#) of this document.

The KiCad Schematic and PCB Files can be found [here](#).

5.2.2 - Power Input:

5.2.2.1 - Project Partner Requirement:

Outputs an analog (5v) or digital signal (CAN or other serial protocol signal)

5.2.2.2 - Engineering Requirement:

The system will operate on 24VDC with up to 3Vpp of noise and 500mA peak of current.

5.2.2.3 - Verification Process:

This requirement can be verified by using the DC power supply initially set to 24 ± 3 VDC and current limit set to 500mA. This power supply will then be connected to the input power of the system. To verify the "3Vpp of noise," we will raise and lower the supply voltage by 3V (i.e power the system off of 27V and 21V respectively). During this we will observe the current being drawn by the power supply and the output of the system (the CAN Communication). If the current draw stays under 500mA and the system continues to communicate via the CAN network, then our system can handle 3Vpp of noise at 24V.

5.2.2.4 - Testing Evidence:

Our system draws less than 500mA of current at all voltages tested and the system continues to measure the chain and output that measurement via the CAN Network. Additionally, our system is designed with a 500mA Fuse, so that if the current does exceed 500mA the fuse would blow, protecting the rest of the system.

A video of this can be found here: [DC Voltage Checkoff Video](#)

5.2.3 - Project Partner Checkoff:

5.2.3.1 - Project Partner Requirement:

Hyster-Yale engineer will checkoff design

5.2.3.2 - Engineering Requirement:

The system will be approved by a Hyster-Yale electrical engineer (project partner).

5.2.3.3 - Verification Process:

We will send an email to our project partner to show him our design and run through the functionality of our design and show our testbench, code, and design with enclosure. More importantly, we will show how each aspect of design and testbench works and theoretical output, code to program both the testbench and the sensors, and the PCB/schematic files.

5.2.3.4 - Testing Evidence:

"Hey All,

I saw your email about your engineering checkoff. Everything looks great. The code is nice, neat, and well commented. The PCB is well laid out and built/soldered very well.

I am excited to see that you are getting some preliminary results from your test set up and am excited to see how well this can be improved!

I was asked to mention that the budget change from \$10 to \$15 was indeed approved by me.

I approve the checkoff for your engineering requirements.

Talk to you all soon and have a great week.

Matt"

The source image can be found in our Google Drive [here](#).

5.2.4 - Robust Enclosure:

5.2.4.1 - Project Partner Requirement:

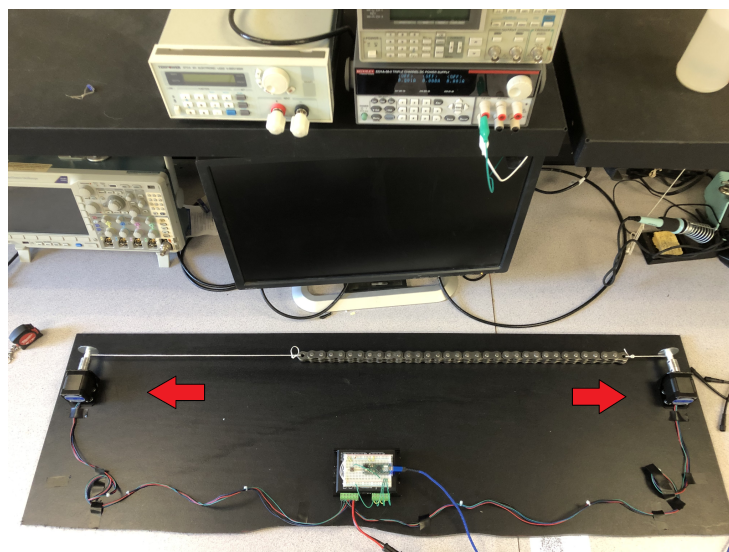
Be robust enough to survive abrasive (high chain wear) applications.

5.2.4.2 - Engineering Requirement:

The system will continue to operate at designed accuracy after 250 chain rotations.

5.2.4.3 - Verification Process:

To prove our enclosure is robust enough, we will run our system on our test bench for at least 250 chain 'rotations' and take a video to prove it happened and succeeded. Based on the construction of our test bench, a 'rotation' is one periodic back and forth movement of the chain on a flat surface.



Our testbench uses a pair of stepper motors controlled by an Arduino Micro and a TB6600 Stepper Motor Driver. The Arduino is programmed to run the motors CCW first and then CW. The testbench is coded to move 4000 Steps in one direction and then run the same number of steps in the opposite direction. This is what one 'rotation' is on our test rig. This translates to the chain moving a distance equivalent to 64 cm per cycle at a speed of about 6.4 cm/s. Each Cycle will take 10 seconds to complete, meaning that it will take approximately 42 minutes to complete. We will run the system for 45-50 minutes and if the measurement output of the sensor is the same as it was at the start of the test, then we have passed this requirement.

The verification method is inspecting the video that the design continues to provide measurement data after 250 periods of chain passage.

5.2.4.4 - Testing Evidence:

A video of this can be found here: [Robust Enclosure Checkoff Video](#)

5.2.5 - Sensor Accuracy:

5.2.5.1 - Project Partner Requirement:

Measure chain "stretch" (the cause of "stretch" is actually wear) within +/- 5% of actual

5.2.5.2 - Engineering Requirement:

The system will measure chain "stretch" within +/- 5% of the actual value

5.2.5.3 - Verification Process:

We will first measure the length of a new and used chain by hand and calculate the percentage difference between the two chains using the new chain as the base value. This value will be the actual "stretch" between the two chains. We will then use our sensor to measure the "Length" of a new chain and the "Length" of a used chain. We will calculate a percentage difference ("stretch") between those two chains. This value will be the measured "stretch" between the two chains. If the value measured by the system falls within +/- 5% of the hand-measured value then we have met this requirement.

5.2.5.4 - Testing Evidence:

We were not able to meet this requirement by the system checkoff date.

5.2.6 - Signal Output:

5.2.6.1 - Project Partner Requirement:

Outputs an analog (5V) or digital signal (CAN or other serial protocol signal)

5.2.6.2 - Engineering Requirement:

The system will communicate with the Hyster Forklift's electronics using a 5V digital serial signal

5.2.6.3 - Verification Process:

To prove the system is able to output CAN protocol signals, we will connect our board to the receiver composed of CAN module and UNO board and see whether UNO is able to receive correct signals and show them on the serial monitor. Since our board is already programmed, after connecting the board to the receiver, give the board a 5V DC voltage through Vin pin and then check the serial monitor. In addition, we will connect the CAN signal output from the PCB to the oscilloscope to check whether the signal conforms to the description of the datasheet. If the monitor shows correct information, as well as the digital signal meets the expectation, then we can say our system meets the requirement.

5.2.6.4 - Testing Evidence:

Our system is able to send a message to an Uno with a CAN Module connected to it, and the signal shown on an oscillator also conforms to the description of the datasheet. This message can change depending on the state of the sensors, proving that the message is being sent from our system to the Arduino.

A video of this can be found here: [Signal Out Checkoff Video](#)

5.2.7 - System Cost:

5.2.7.1 - Project Partner Requirement:

Cost less than \$10 or less at production quantity of 15,000/yr

5.2.7.2 - Engineering Requirement:

The system, not including the enclosure, will cost less than \$15 at a production quantity of 15,000/yr

Note: This requirement changed to \$15 instead of \$10 from the project partner requirement and has been approved by our project partner. ([see Evidence Folder](#))

I was asked to mention that the budget change from \$10 to \$15 was indeed approved by me.

I approve the checkoff for your engineering requirements.

5.2.7.3 - Verification Process:

A Bill of Materials will be created containing the components for the electronics board and the sensors using the bulk pricing for all of the components. If the total price of the components is under \$15, then we have passed this requirement.

5.2.7.4 - Testing Evidence:

A BOM was created to track the total price of the electronics, which is linked at the end of this section. Octopart was used to find out the bulk price for each component.

The electronics can be split into two categories: The PCB and the Sensors. The total price of the PCB and its components at bulk price comes to \$11.00. The sensors are simply IR Breakbeam sensors, which only need two components for each sensor, an IR LED and an IR Receiver. The total cost for three sensors is \$1.67 when purchased in bulk. This means the total price of the electronics is \$12.67 which is below the \$15 budget that we have.

The total price is counted in this [Electronics BOM](#).

5.2.8 - Water Resistance:

5.2.8.1 - Project Partner Requirement:

Be robust enough to survive abrasive (high chain wear) applications and rain

5.2.8.2 - Engineering Requirement:

The system shall function normally after applying a 12oz bottle of water from above.

5.2.8.3 - Verification Process:

The first step of the verification process will be to demonstrate our system working before applying water. Next, the board will be unplugged from the power supply and the PCB will be removed and replaced with dry paper towels. We will pour an entire >12oz bottle of water on top of the enclosure. After applying the bottle of water, we will demonstrate that no water got into the enclosure by displaying the dry paper towels. If no water got on the paper towels and there is no visible water inside the enclosure, then we have demonstrated that the enclosure is water resistant.

5.2.8.4 - Testing Evidence:

A video of this can be found here: [Water Resistance Test Video](#).

5.3 - References and File Links:

Guide to Chain Stretch Measurement: <https://www.youtube.com/watch?v=LAkxU9RCLjA>

5.4 - Revision Table:

Revision Number	Revision Date	Description	Revision Author
1	3/15	Initial Revision	Kai, Corbin,

			Adam, Tianhao
2	4/26	Updated verification processes	Corbin, Adam, Tianhao
3	5/4	Review, Comments, and Update of Section 5 Some edits to section 5.1, but most of the edits are within Sections 5.2 (specifically 5.2.2, 5.2.3, 5.2.4, 5.2.5, 5.2.7, 5.2.8)	Kai
4	5/7	Edits to Verification sections	Corbin, Adam, Kai

Section 6 - Project Closing

6.1 - Future Recommendations

6.1.1. - Technical recommendations:

During our time developing this project, we had to order several parts, big and small. A good approach to ordering parts in a helpful way is to avoid placing an abundance of orders; collect a list of parts everyone needs and order all at once when possible. This cuts down on packaging waste and wait times, as well as having particular members of the team much further ahead than others due to component arrival time differences [1].

Another technical recommendation is to consider the other scenarios where a forklift chain can fail. Although the focus of this project is to measure chain elongation, you should account for other factors such as protruding or loose pins, plate cracking, misalignment due to worn pins, and broken links because these cases can cause harm worse than chain elongation such as complete breaks and can result in injury of the forklift operator and potential injury for anybody around the forklift [2]. We recommend adding additional sensors to detect protruding or loose pins, plate cracking, and misalignment. An example of including additional sensors would be applying computer vision to detect these other potential faults in a chain.

A good approach to technical checkoffs is to rehearse what you are showing to the TA/professor *multiple times* before your checkoff slot. This will allow you to get into the checkoff and get out in a very timely manner and ensure your parameters are actually met. Additionally, take a video of your project working! Electronics in development get stage fright so often and bugs and other problems that never existed before will suddenly present themselves only because the TA/professor is standing there grading you. Having a video will allow you to prove that your block does, indeed, work [3].

Finally, unless your project is strictly software, having a good toolkit is essential. Ensure you, as a group, have soldering equipment, a few good powered tools like a drill or voltmeter, some basic tools like allen wrenches or alligator clips, electrical wires, and finally screwdrivers - these are an extremely important tool and in a class of their own when it comes to usefulness. Additionally, there are several good arduino kits that include an abundance of materials that will prove useful for this project and for any hobby projects in the future. Considering you got this far in engineering, this equipment will be useful for the rest of your life, so don't be afraid to make investments if you are able to. We ran into issues a few times when only one member of our team had a toolkit with a particular type of tool (screwdrivers), so when he took his toolbox home we were unable to make modifications requiring screwdrivers. Avoid this by having redundancy [4].

6.1.2. - Global impact recommendations

One of our global recommendations is to consider the amount of electronic waste that is being generated from producing prototypes to the final design. The reason for this is because of the fact that 18% of electronic waste is accounted for which means that the majority of waste is leaking toxins into the environment [5]. Some ways to reduce the amount of electronic waste is to research pre existing components and build them instead of creating a custom one that may include trial and error which will result in electronic waste. Also, use simulation software such as KiCad, and Spice to run tests in software to determine any hardware and software bugs with a design before it is sent to manufacturing.

Another global recommendation is to consider the materials that require mining solutions for our electrical components, which consequently require mining. Mining is an incredibly polluting and degrading activity. Native peoples are disproportionately affected by these activities, especially those groups living on or near the land directly affected. For example, mining that takes place in northern Canada affects aboriginal communities who live in those regions. Colonialism has affected aboriginal communities and values because of treaties that displaced aboriginal peoples to acquire metal deposits and other mineral resources. Treaties have also deliberately cut aboriginal peoples from decision making to encourage industry to be created in the north. Additionally, industry originates from the south of Canada which embodies western culture, values, and communication values. Another aspect is that mining communities have workers who have long hours and sometimes, both parents of a family might be working which would require grandparents or guardians to assume childcare. This style is different compared to the aboriginal communities where hunting and gathering for food is common and reliance on family members. Also, the introduction of westerners from the south diminish the use of aboriginal languages. An example is from 1989 to 1999, the aboriginal languages spoken in the northern regions of Canada dropped about 10 percent. Some positive aspects of mining are the cultural values it promotes such as independence, freedom, and pride which could restore traditional lifestyles. Another aspect is increasing disposable income to improve quality of life [6].

6.1.3. - Teamwork recommendations

An extremely helpful practice when working with the team is meeting with them multiple times a week; at least two times a week but aim for three. We recommend meeting this often because there will be times during the academic year when you will need time to meet and work on documentation and design deadlines as often as possible. Also, meetings often can account for time lost with group members who cannot attend all of the group meetings during the week [7].

Another practice we recommend is to meet with your project partner once a week to update them on overall progress and to ask technical questions that your professor/TA's may not have time to answer during certain weeks out of the year. The opposite is also true in case your project partner is not available to communicate your progress/questions and may have to

rely on the professor and TA's to keep your project on the right track to not waste time. Although it may seem like one year is more than enough time to complete this project, it is not if your group is inexperienced and new to certain concepts such as power electronics [8].

6.2 - Project Artifact Summary with Links

[Final PCB Schematic and PCB Layout](#)

[Group 15 Gantt Chart](#)

[Sensor Bracket V1 \(Single Sensor Pair\)](#)

[Sensor Bracket V2 \(All 3 sensor Pairs\)](#)

[Device Bill of Materials and Total Cost price](#)

6.3 - Presentation Materials

Student Showcase Site:

<https://eecs.engineering.oregonstate.edu/capstone/ece/student/index.php>

Engineering Requirements

The System will...

- **PCB** - Have a single PCB that is no larger than 6 inches squared and contains all non-sensor circuits.
- **Power Input** - Operate on 24VDC with up to 3Vpp of noise and 500mA peak of current.
- **Project Partner Checkoff** - Be approved by a Hyster-Yale electrical engineer (project partner).
- **Robust Enclosure** - Continue to operate at designed accuracy after 250 chain rotations.
- **Sensor Accuracy** - Measure chain "stretch" within +/- 5% of the actual value.
- **Signal Output** - Communicate with the Hyster Forklift's electronics using a 5V digital serial signal.
- **System Cost** - Cost less than \$15 at a production quantity of 15,000/yr, including the enclosure.
- **Water Resistance** - Function normally after applying a 12oz bottle of water from above.



Fig. 4. Test bench used to determine the accuracy and durability of the system



Forklift Chain "Stretch" Monitor

Non-Contact Sensor to Determine Chain Wear for Heavy Duty Applications

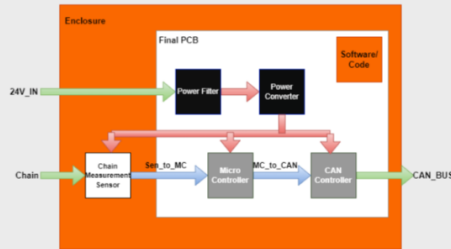


Fig. 1. Block diagram of the system

Power Filter/Converter

- Power supply with input from 24V battery
- Operates between 21-27VDC and up to but not including 500mA of current
- Uses a step-down converter (buck converter) to step down voltage and filter (LC) noise from the battery
- Efficiency rated between 66-80%. Efficiencies vary based on current
- Can replace fuse for circuit protection from current spikes from the battery

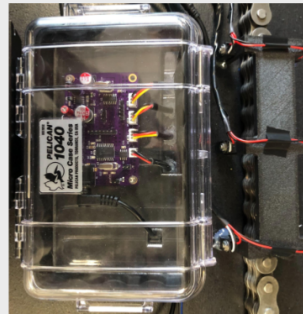


Fig. 3. Sensors bracket and PCB enclosure

CAN Controller

- The Controller Area Network (CAN) is one way of facilitating communication between electronic components
- CAN is a standard communication system in vehicles and allows for different sensors and components to communicate on the same line/bus in a priority-driven manner
- Many modern vehicles, and importantly Hyster Yale forklifts, use CAN controls
- Information from more crucial components is transmitted before other, less important devices all on the same network

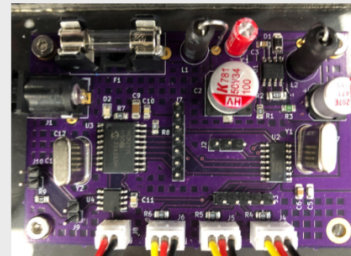


Fig. 2. Assembled PCB

Chain Measurement Sensor

- Three IR Break beam Sensors are used to measure Chain Elongation
- One sensor is used to measure speed, which the others are used to measure the elongation
- The timing difference between the sensors coupled with the speed gives us the elongation
- A 3D Bracket was designed for the sensors to keep the distance between them both constant and known for the measurement calculations

Enclosure

- Our enclosure consists of a pelican case with an acrylic plate for the custom PCB and holes drilled and then sealed to allow for waterproof cables to be added to the system. These cables connect to the Power, CAN, and sensors.



Project Summary



Fig. 5. Hyster Forklift

Our project is intended to determine whether it is feasible to create a reliable chain wear detection device that is economical. Hyster-Yale currently uses chain wear detection devices and our goal was to create a proof of concept that Hyster-Yale can obtain and improve upon.

Team Members



Fig. 6. All four members

(From left to right)

Tianhao Lin:
Microcontroller/CAN Controller Designer
lintia@oregonstate.edu

Adam Grzelewski:
Power Converter/Filter Designer
grzelewa@oregonstate.edu

Corbin Krecklow:
Programmer and Enclosure Designer
kreckloc@oregonstate.edu

Kai Roy:
Sensor and PCB Design
kaikurisakaroy@gmail.com

6.4 - References and File Links

- [1] Olivier. Danielle, "4 Ways to Buy Electronic Components and the Pros and Cons of Each" *escatec.com*, Jul. 13, 2017. [Online]. Available: <https://www.escatec.com/blog/4-ways-to-buy-electronic-components-and-the-pros-and-cons-of-each>. [Accessed May. 14, 2023].
- [2] Hodson. Jim, "How to Inspect Forklift Chains and Know When to Replace Them" *tmhnc.com*, Mar. 14, 2017. [Online]. Available: <https://www.tmhnc.com/blog/how-to-inspect-forklift-chains-and-know-if-forklift-chain-is-worn-out>. [Accessed Nov. 4, 2022].
- [3] Reilly. Rob, "How to Do a Great Hardware Demo" *thenewstack.io*, Aug. 12, 2014. [Online]. Available: <https://thenewstack.io/how-to-do-a-great-hardware-demo/>. [Accessed May. 14, 2023].
- [4] Mod. Shreepanjali, "10 Hardware Tools That Every Electrical Engineer Should Have In Toolbox" *engineersgarage.com*, May. 14, 2023. [Online]. Available: <https://www.engineersgarage.com/10-hardware-tools-that-every-electrical-engineer-should-have-in-toolbox/>. [Accessed May. 14, 2023].
- [5] Murthy. Venkatesha and Ramakrishna. Seeram, "A Review on Global E-Waste Management: Urban Mining towards a Sustainable Future and Circular Economy" *mdpi.com*, Jan. 7, 2022. [Online]. Available: <https://www.mdpi.com/2071-1050/14/2/647/htm>. [Accessed Oct. 31, 2022].
- [6] Gibson, Ginger, and Jason Klinck. "Canada's resilient north: the impact of mining on aboriginal communities." *Pimatisiwin* 3.1 (2005): 116-139. Available: <https://caid.ca/JAICH2005v3n1p115.pdf> [Accessed Dec. 2, 2022].
- [7] Hills. Charlotte, "5 Benefits of Team Meetings" *linkedin.com*, Nov. 19, 2020. [Online]. Available: <https://www.linkedin.com/pulse/5-benefits-team-meetings-charlotte-hills-acma-cgma>. [Accessed May. 14, 2023].
- [8] Academic Success Center. "Using Office Hours Effectively" *Oregon State University* [Online]. Available: <https://success.oregonstate.edu/office-hours-effectively>. [Accessed May. 14, 2023].