Production Model Design Report

F2019 – ECE 298

|  |  |  |  |
| --- | --- | --- | --- |
| Lab Section: | 5 | Group: | 16 |

# Team Members

|  |  |  |
| --- | --- | --- |
| # | Name | Role |
| 1 | Simon Cousineau | LCD display, keypad API, PCB assembly |
| 2 | Ben Finch | Limit switches and indicators, stepper motor API, PCB assembly |

# Design Overview

## Problem Statement

Manufacturing facilities have a demand for high-throughput, in order to fulfil large orders. To achieve this goal, they require efficient picker-placer machines which are capable of manufacturing electronic circuits with high speed, high accuracy and minimal downtime. This device should be able to articulate in two axis, within a bounded plane, while navigating through user input coordinates.

## Design Scope

In order to achieve this goal, our device utilizes a keypad to input coordinates, while providing prompts and feedback through the board’s LCD display. Once all coordinates are enterer, the device will actuate its stepper motors to navigate to the specified location, while limit switches enforce and indicate boundaries.

While designing the device, it was assumed that the payload carried would be relatively small, since PCB components are generally light. It was also assumed that the user would input a maximum of 5 coordinates. It was also assumed that the device would run on a low voltage source (5V).

## Project Design Requirements

Functional:

* Must have at least two functional axis of movement.
* Must initialize to center of axis range, for both axis. (start x = width / 2, start y = height / 2)
* Must support coordinates inputs that are positive and negative.
* Must actuate LED indicator upon reaching range limits.
* Must stop appropriate motor upon reaching range limits.
* Must allow user to input coordinate values via keypad.
* Must allow user to start movement sequence via keypad input.
* Must display current movement progress on LCD display (percentage to target).

Non-functional:

* Must be easy to use (input coordinates, start sequence, etc.).
* Must reach coordinates within reasonable time delay.
* Must travel in one smooth motion (no jitter, grinding, skipping).
* Must reach targets with reasonable accuracy and/or precision
* Must be energy-efficient.

Constraint:

* Must cost less than 100$ per unit.

## System-Level Design (High-Level)



## Completed Prototype

**

Figure 1: Negative coordinate input on keypad

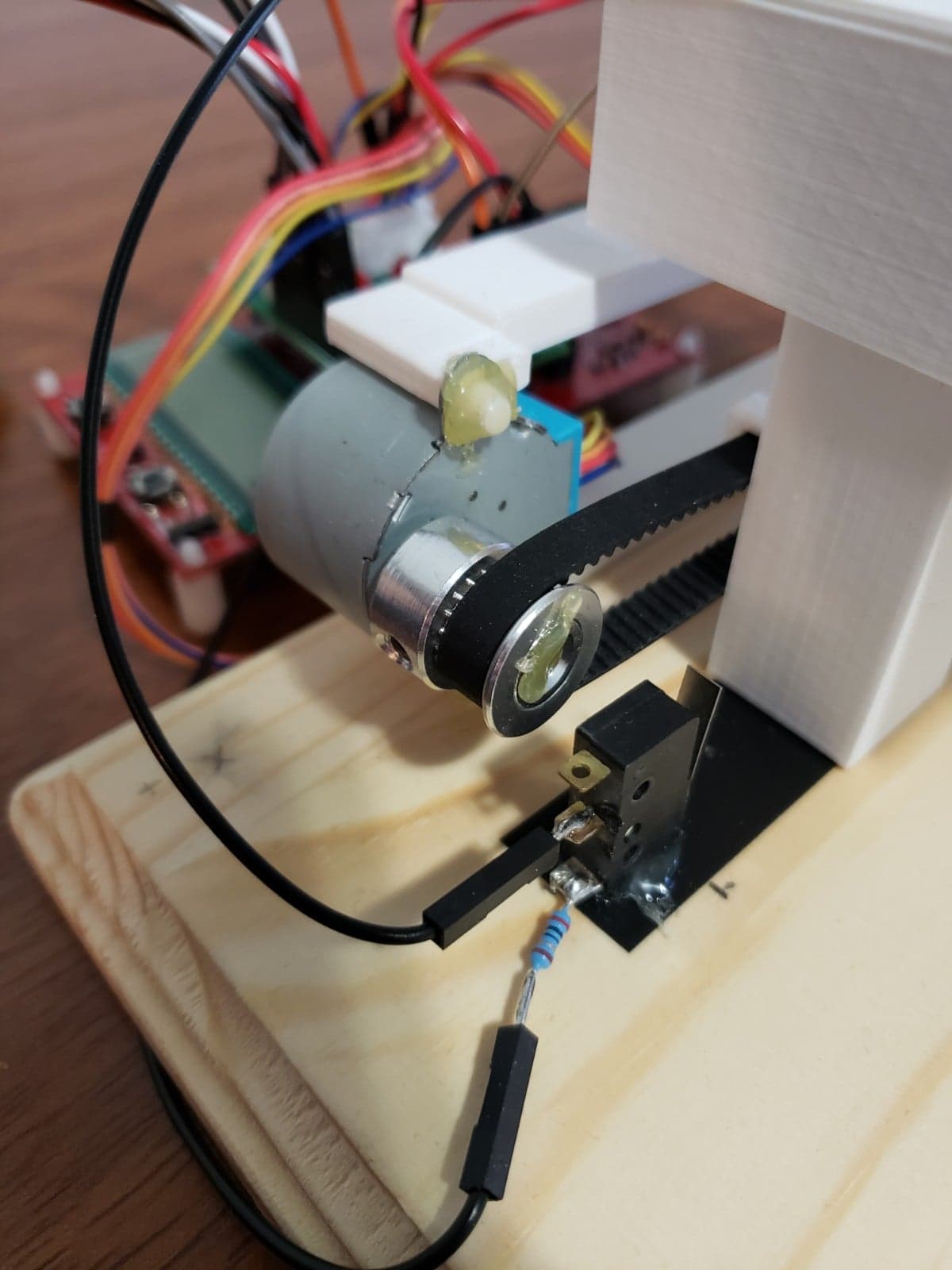


Figure 2: Limit switch push button and stepper motor setup

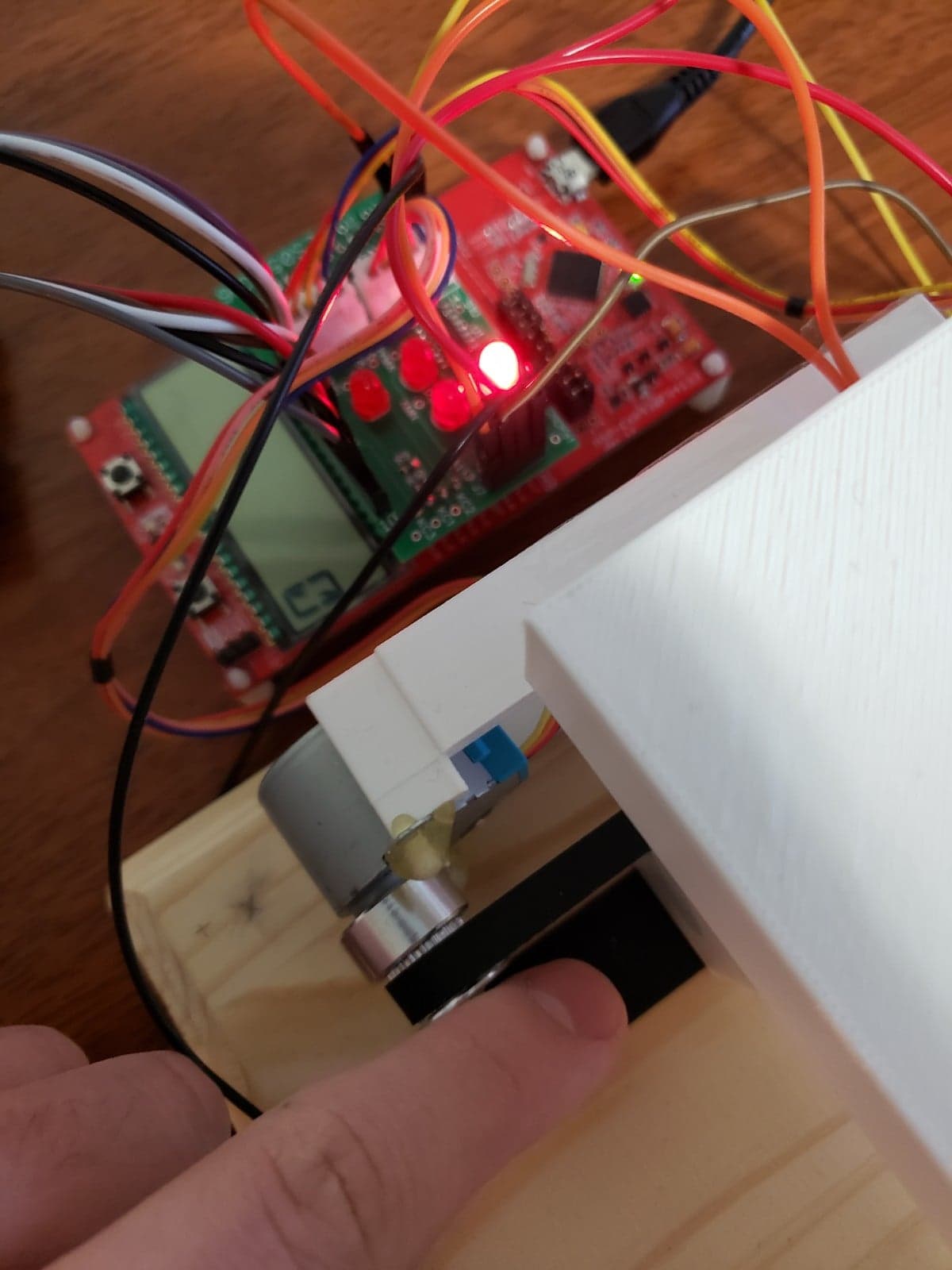


Figure 3: X-max limit switch activation indicator

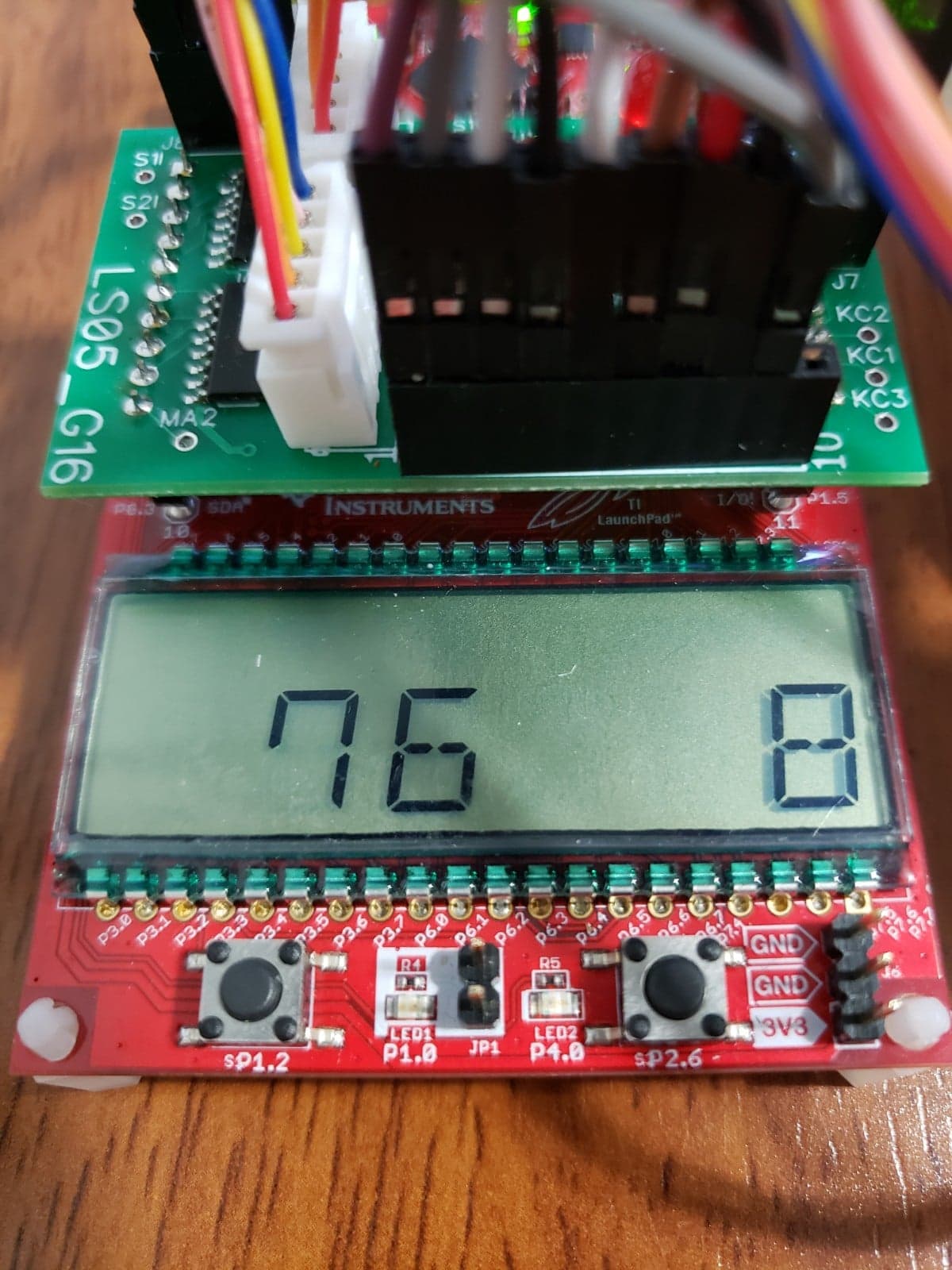


Figure 4: Stepper motor progress percentage display

## Preliminary Production Design Changes

* Include stronger stepper motors. This would allow us to move the platform faster and move larger payloads.
* Lighter limit switches, to ensure more consistent limit switch activation. The current model will occasionally fail to detect limit switch contact immediately.
* More robust mechanical rig. With stronger motors, this would allow the device to include a larger platform and ensure that the axis do not tip during movement.
* Fix PCB design: Reverse direction of motor header ports, remove LED resistor short circuits, increase LED resistor values to 10k ohms.

# Member 1 Production Details

Simon Cousineau – ID# 20717856

## Further Integration

Further integration comprises the next steps necessary to bring this prototype design into a state that is useful for real-world applications. This includes changes that will allow the product to work in unison with other technologies already in place (instead of as a standalone device like the prototype), as well as optimization of software and hardware designs.

This is relevant to our project as our prototype will require some changes before it is ready to be integrated into a manufacturing facility’s production line. The first major issue that must be addressed for the production model is that lack of automation capabilities. Currently the device requires manual user input of coordinates and requires human supervision to start the movement. The second issue that must be addressed in the production model is the device’s mechanical rig. The prototype design is used to demonstrate the device’s ability to navigate x and y coordinates by moving a platform. In the production model, the device should be able to move some assembly head in order to perform its duties.

In order to address the issue of automation, the production model should implement a different input interface, such as UART or serial communication (I2C), while also maintaining the keypad interface for manual overrides. This interface would allow the input of target coordinates to be controlled from a distance and would allow automation capabilities. Furthermore, this would allow the devices to be controlled and monitored from a central location. The production model device will also need to move some extrusion, milling or pincer head, in order to accomplish its manufacturing duties. This change would require some stronger stepper motors, in order to move these heavy components, as well as a redesign of the hardware rig. Currently the rig is tailored for a platform, but it would need to be redesigned to properly support these manufacturing heads.

## Energy Efficiency

Energy efficiency is the process of accomplishing the same task, while using less energy. At its essence, energy efficiency is the optimization of energy utilisation, in order to reduce energy waste and make better use of energy resources.

Energy efficiency is especially interesting for large scale operations, such as manufacturing facilities, where small savings add up quickly, and energy optimization allows the company to increase its profit margin. In our case, the device is designed for these manufacturing facilities, which will most likely require multiple devices to fulfil their manufacturing needs. Therefore, optimizing the device’s energy usage is very attractive, as it will allow these companies to reduce energy expenses and reduce their environmental impact related to energy usage.

In order to improve the device’s energy efficiency, the production model should be designed to use the MSP430’s low power modes. In order to make use of the board’s low power modes, some changes to the software design are necessary. The device should make use of interrupts for coordinate input instead of tight polling. This allows the device to put the processor to sleep while it waits for inputs, at the expense of slightly slower input recognition, although this change would be unnoticeable to the user. Furthermore, the device should use timers for the stepper motor control, instead of using software delays. This also allows the processor to be put to sleep, while only keeping a clock active for the timer. Once these changes are made, the software design will be able to accommodate the board’s low power mode 3 (LPM3), which disables everything except interrupts and the timer’s clock, when they are unused [2]. As mentioned in [2], this low power mode allows the board to “go from 300uA down to less than 1uA”, which means the device will be very energy efficient.

# Member 2 Production Details

Ben Finch – ID# 20714219

## Design for Reliability (DfR)

To design for reliability, the engineer must consider methods of minimizing frequent failure, covering and correcting failure points, recovering from failure, and analyzing new designs to ensure reliability.  This is a major focus in systems engineering where multiple components are integrated and can fail. As a system gets more complex, there are more failure points and symptoms for each particular failure, so it is very important to design to mitigate these errors, as well as to quickly diagnose unavoidable errors to minimize downtime.

The pick and place machine is one such system that is suited for reliability analyses.  It is an electromechanical system, such that it has two domains of failure. The mechanical platform can grind and wear down, the axis can bend, and the tension pulley can snap.  The electrical system can burn out, and stepper motors can be overworked and stall. As its purpose is to increase the efficiency and throughput of electronic devices in an industrial setting, it is important that downtime is minimal, and accuracy is maintained for the duration of product life.

To consider accuracy, the production engineer must consider failure points for inaccuracy.  This is subtler than detecting complete system failure. Functional inaccuracy occurs at the mechanical level.  Should the pulley system slip, or the axis block misalign with the limit switch rather than hit it, the number of steps tracked for an axis will misrepresent the actual position of the axis, causing a constant offset of parts picked and placed.  This can be fixed with a system reboot and mechanical reinforcement to tighten the pulley and prevent slippage. To detect an error early, the system must analyze the environment data it obtains during startup calibration, where it evaluates its limits and detects its workspace.  If the calibration data were to be sent to a centralized production server, statistical analysis can find anomalies in particular calibration runs and discover when a failure can occur. Put simply, the production engineer should be keeping track of calibration data to detect errors in accuracy.  To prevent slippage later in runtime, the machine should avoid running towards the limits of the workspace, such that it doesn’t miss hitting the limit switch and skew position data. The production engineer should ensure the input coordinates are bounded and does not hit limits where possible.

Should any mechanical parts break, they can be quickly and easily replaced.  The system has been designed such that there are few parts and they are simple to put together.  The production engineer should keep spare platform parts ready for fast replacement to minimize downtime.  Should any mechanical part break, this should be considered a total system failure as it halts correct functioning.  This is more obvious to detect, as the platform may not move or articulate correctly in both axis. On the electrical side, failure can be detected input side, via noise on the keypad (or certain keys becoming undetectable), or output side, via stepper motor failure.  Both sides can be inspected visually by observing input and platform motion but can also be verified by electrical test points on the PCB. The production engineer should continually observe the system and if an irregularity occurs, test points should be evaluated to isolate the error.  If the keypad test points are noisy, the pulldown resistance on that column has failed. If a particular test point is pulled to zero, this is a multiplexor fail. If all test points are pulled to zero, then the pulldown is shorted. If the steppers are acting irregularly, there are test points on the input and output of the driver to isolate if the error is in GPIO or in the stepper.  If a limit fail is suspected, the production engineer can visually observe the LED indicating a limit hit. If this does not occur, a test point for that limit can determine if the limit switch has failed or if the LED has burnt out. All of these failures can be fixed by repairing the PCB or replacing it entirely. The production engineer should keep replacement PCBs available for quick system repairs.  Electrical failures can only be fixed and not avoided.

## Supply Chain Management

Supply chain management ensures the supply of production materials as well as the steady throughput of the production process to maintain an ideal stock of finished product to match consumer demand.  This is an interesting problem for engineering operations as the consumer demand must be predicted early to ensure the correct amount of product material is received and processed. Too many completed products becomes an overhead in storage, and too few completed products becomes a loss in demand.  This has virtually no impact in software products, but electrical and mechanical systems have to balance this very carefully.

The picker placer product, as an electromechanical system, comes with the challenge of ordering enough electrical components and plastic material to be assembled and produced in house to send out to customers quickly and without the above-mentioned overheads.  PCBs and electrical components must be ordered and soldered, while plastic filament must be ordered to print and process the mechanical platform parts. Other mechanical parts, such as the axels and pulleys, also need to be ordered for assembly. As this is an industrial product, which requires installation in a timely manner and integration into other company operations, the product must be made quickly and sent out as available.

Fortunately for the production engineer, the system is small, so product can be pre-emptively made and stored with minimal space footprint.  This allows for some margin to overproduce in the event consumer demand is underestimated. One challenge the production engineer must face is keeping separate inventories of multiple mechanical parts and PCBs.  Not all need to be assembled as some of the parts may go out as replacement pieces, such that the consumer doesn’t have to buy a whole new system and waste the broken product. It is important that the production engineer statistically analyzes the failure rate of particular components and produces those components proportionally to consumer demand.  In the event of overproducing passed the storage margin, the production engineer should intentionally underproduce and use the stored margin as an offset to bring back storage stability. This should be done when consumer demand is not expected to grow so as to smoothly correct the production margin.

# References

|  |  |
| --- | --- |
| [1] | "IEEE Style," 2019. [Online]. Available: https://pitt.libguides.com/citationhelp/ieee. |
| [2] | Argenox, "MSP430 Interrupts and Low Power," Argenox, [Online]. Available: https://www.argenox.com/library/msp430/msp430-interrupts-and-low-power-chapter-7/. [Accessed 30 11 2019]. |

# Appendix – Detailed Design

Figure 5: Feasibility model design

*Notes: The input resistance to the LED is now 10 ohms. Added 10k pulldown resistors to the keypad columns. Keypad rows are now multiplexed.*



Figure 6: Software flowchart

*Link to code repository:* [*https://github.com/blazingbbq/xyplatform*](https://github.com/blazingbbq/xyplatform)

Table 1: Design changes from feasibility model

|  |  |  |
| --- | --- | --- |
| # | Change | Reason/Notes |
| 1 | No longer planning to use UART to send commands. Will only be using it in testing context. | Doesn’t offer any additional value over other main components of the project. |
| 2 | We were originally planning to use 2 separate boards, in order to have access to enough GPIO pins. Will only be using one board. | We now realize that we have enough pins on a single board. |
| 3 | No longer planning to use interrupts for button presses (mainly, limit switches). | We use sufficiently small steps, that tight polling the button statuses should be responsive enough for our needs. Moreover, this simplifies our code. |
| 4 | LEDs will not be connected to their own GPIO pin. Instead, they will be wired in-line with the limit switches. | Reduces usage of GPIO pins, allowing us to consolidate the design to a single board. Also, these LEDs should only be on when the button is pressed, therefore there is no need to control them independently from the limit switch push buttons. |
| 5 | Changed the input resistance to the limit LEDs from 300Ω to 200Ω. | Changed this value so that we use less resistors (because at 300Ω, we’d need to use three 100Ω resistors), whereas we only use two 100 Ω resistors now. This resistance still allows us to limit the input current within the nominal range acceptable to drive the LED. |

Table 2: Important notes from feasibility model

|  |  |
| --- | --- |
| # | Note |
| 1 | As per the suggestion of the TA conducting our Feasibility Model demo, we investigated using hall effect sensors as our limit switches. We found that these were functionally equivalent to the push buttons (but with inverted output), however they also required their output to be amplified to 3.3v, in order to properly read their statuses. This adds unnecessary complexity to our design. Additionally, upon investigating similar products that exist already, we found that they commonly use physical limit switches, similar to our design. |
| 2 | We should always remember to initialize the required pins before attempting to use them. It is best to initialize them with all the other initializers (at the top of the file), since we disable interrupts in this section. |
| 3 | We sample the keypad buttons multiple times, in order to reduce possible noise on these inputs. |
| 4 | We purposefully omit debouncing the limit switch push buttons because, in the event of noise, the steppers will only skip one movement due to this noise and continue normally afterwards. |

Table 3: Hardware signal test plan

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Signal (TP\*) | Test Point name | Property | Required Software Mode | Min | Nominal | Max |
| X Axis MIN | X\_MIN\_TEST | Voltage | Limit Switch test mode | 0 V | 3.3 V | 5 V |
| X Axis MAX | X\_MAX\_TEST | Voltage | Limit Switch test mode | 0 V | 3.3 V | 5 V |
| Y Axis MIN | Y\_MIN\_TEST | Voltage | Limit Switch test mode | 0 V | 3.3 V | 5 V |
| Y Axis MAX | Y\_MAX\_TEST | Voltage | Limit Switch test mode | 0 V | 3.3 V | 5 V |
| Keypad COL 1 |  | Voltage | Keypad Column 1 test mode |  | 3.3 V |  |
| Keypad COL 2 |  | Voltage | Keypad Column 2 test mode |  | 3.3 V |  |
| Keypad COL 3 |  | Voltage | Keypad Column 3 test mode |  | 3.3 V |  |
| Keypad ROW 1 | ROW\_1\_TEST | Voltage | Keypad Column [1|2|3] test mode |  | 3.3 V |  |
| Keypad ROW 2 | ROW\_2\_TEST | Voltage | Keypad Column [1|2|3] test mode |  | 3.3 V |  |
| Keypad ROW 3 | ROW\_3\_TEST | Voltage | Keypad Column [1|2|3] test mode |  | 3.3 V |  |
| Keypad ROW 4 | ROW\_4\_TEST | Voltage | Keypad Column [1|2|3] test mode |  | 3.3 V |  |
| Stepper 1 IN 1 | STEPPER\_1\_OUT\_TEST | Voltage | Stepper test mode | 0 V |  | 5V |
| Stepper 1 IN 2 |  | Voltage | Stepper test mode | 0 V |  | 5V |
| Stepper 1 IN 3 |  | Voltage | Stepper test mode | 0 V |  | 5V |
| Stepper 1 IN 4 |  | Voltage | Stepper test mode | 0 V |  | 5V |
| Stepper 2 IN 1 | STEPPER\_2\_OUT\_TEST | Voltage | Stepper test mode | 0 V |  | 5V |
| Stepper 2 IN 2 |  | Voltage | Stepper test mode | 0 V |  | 5V |
| Stepper 2 IN 3 |  | Voltage | Stepper test mode | 0 V |  | 5V |
| Stepper 2 IN 4 |  | Voltage | Stepper test mode | 0 V |  | 5V |

Table 4: Hardware signal connectivity

|  |  |  |  |
| --- | --- | --- | --- |
| Signal | MSP430FR4133 Pin | LaunchPad J1/J2 Pin | Prototype Connection |
| Stepper 1 IN 1 | P8.1 | J1 pin 2 | IC1 1B |
| Stepper 1 IN 2 | P1.1 | J1 pin 3 | IC1 2B |
| Stepper 1 IN 3 | P1.0 | J1 pin 4 | IC1 3B |
| Stepper 1 IN 4 | P2.7 | J1 pin 5 | IC1 4B |
| Stepper 2 IN 1 | P8.0 | J1 pin 6 | IC2 1B |
| Stepper 2 IN 2 | P5.1 | J1 pin 7 | IC2 2B |
| Stepper 2 IN 3 | P2.5 | J1 pin 8 | IC2 3B |
| Stepper 2 IN 4 | P8.2 | J1 pin 9 | IC2 4B |
| Keypad COL 2 | P8.3 | J1 pin 10 | J4 pin 1 |
|  |  |  |  |
| Keypad ROW 2 | P1.5 | J2 pin 1 | J5 pin 2 |
| Keypad ROW 3 | P1.4 | J2 pin 2 | J5 pin 1 |
| Keypad COL 3 | P1.3 | J2 pin 3 | J4 pin 5 |
| Keypad ROW 4 | P5.3 | J2 pin 4 | J4 pin 4 |
| Y Axis MIN | P5.2 | J2 pin 5 | J1 (3.3 V) |
| Y Axis MAX | P5.0 | J2 pin 7 | J1 (3.3 V) |
| X Axis MIN | P1.6 | J2 pin 8 | J1 (3.3 V) |
| X Axis MAX | P1.7 | J2 pin 9 | J1 (3.3 V) |
|  |  |  |  |
| Keypad COL 1 | P2.6 | J3 pin 1 | J4 pin 3 |
| Keypad ROW 1 | P4.0 | J3 pin 2 | J4 pin 2 |

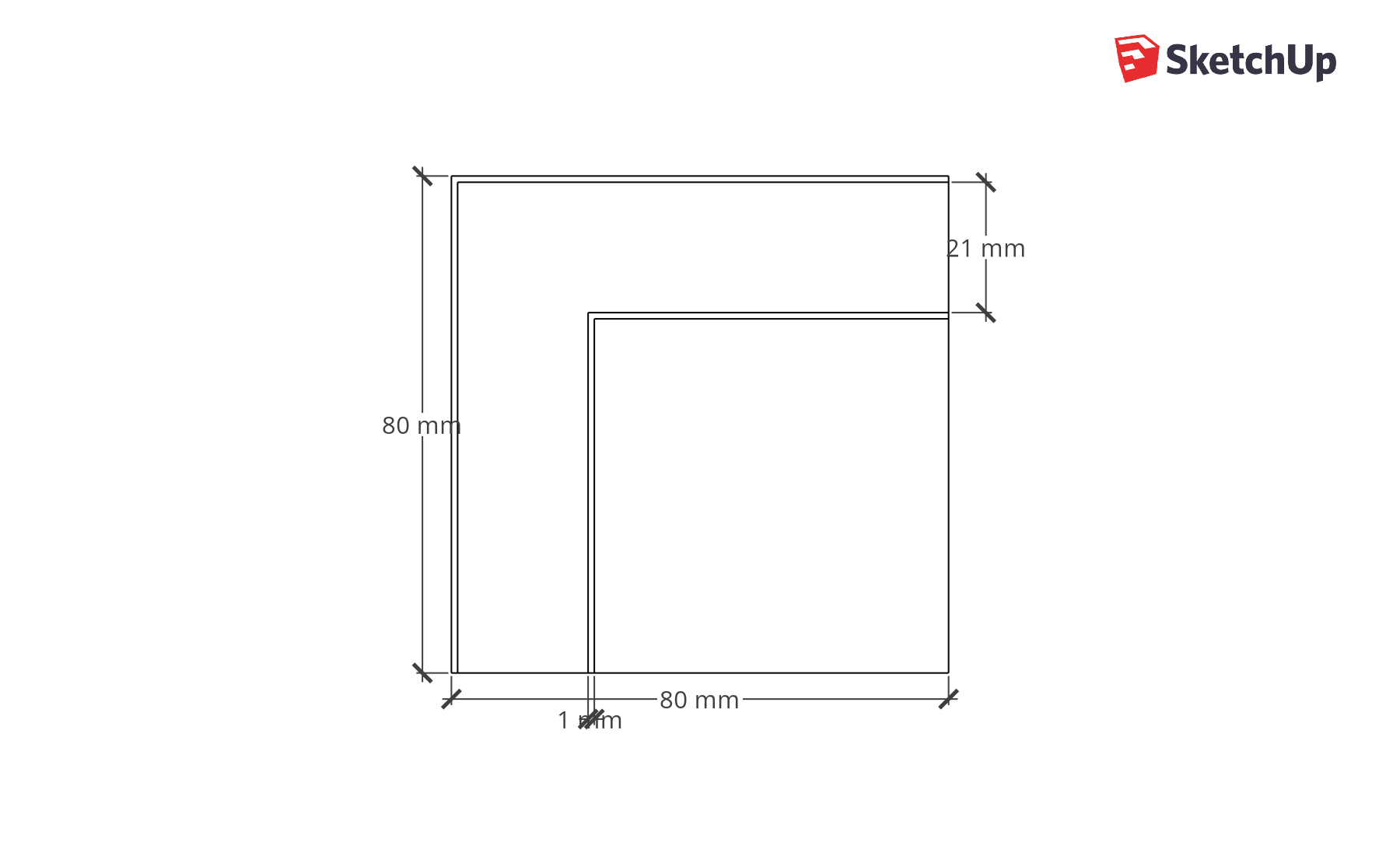


Figure 7: Mechanical drawing of prototype platform

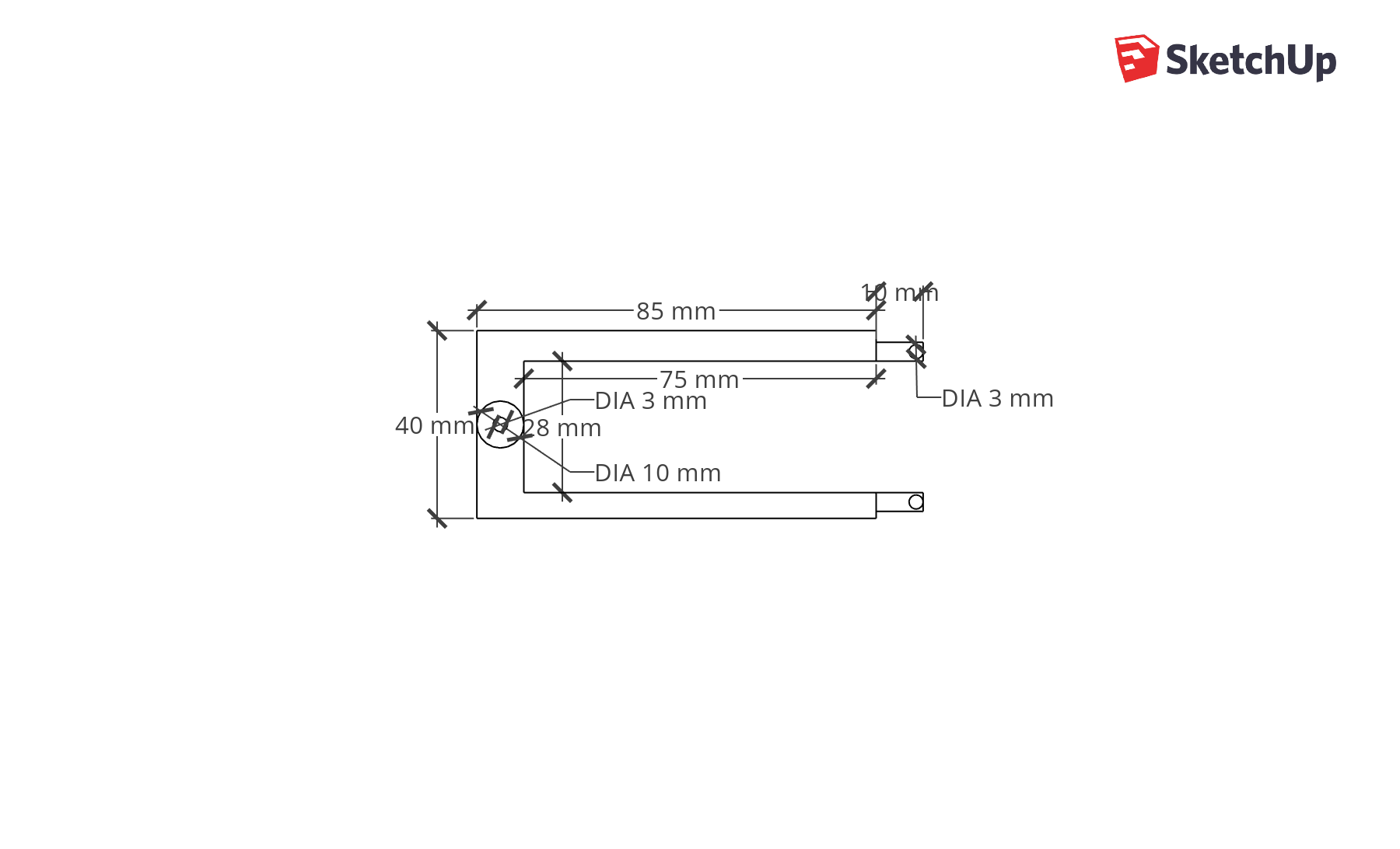


Figure 8: Mechanical drawing of axis mount

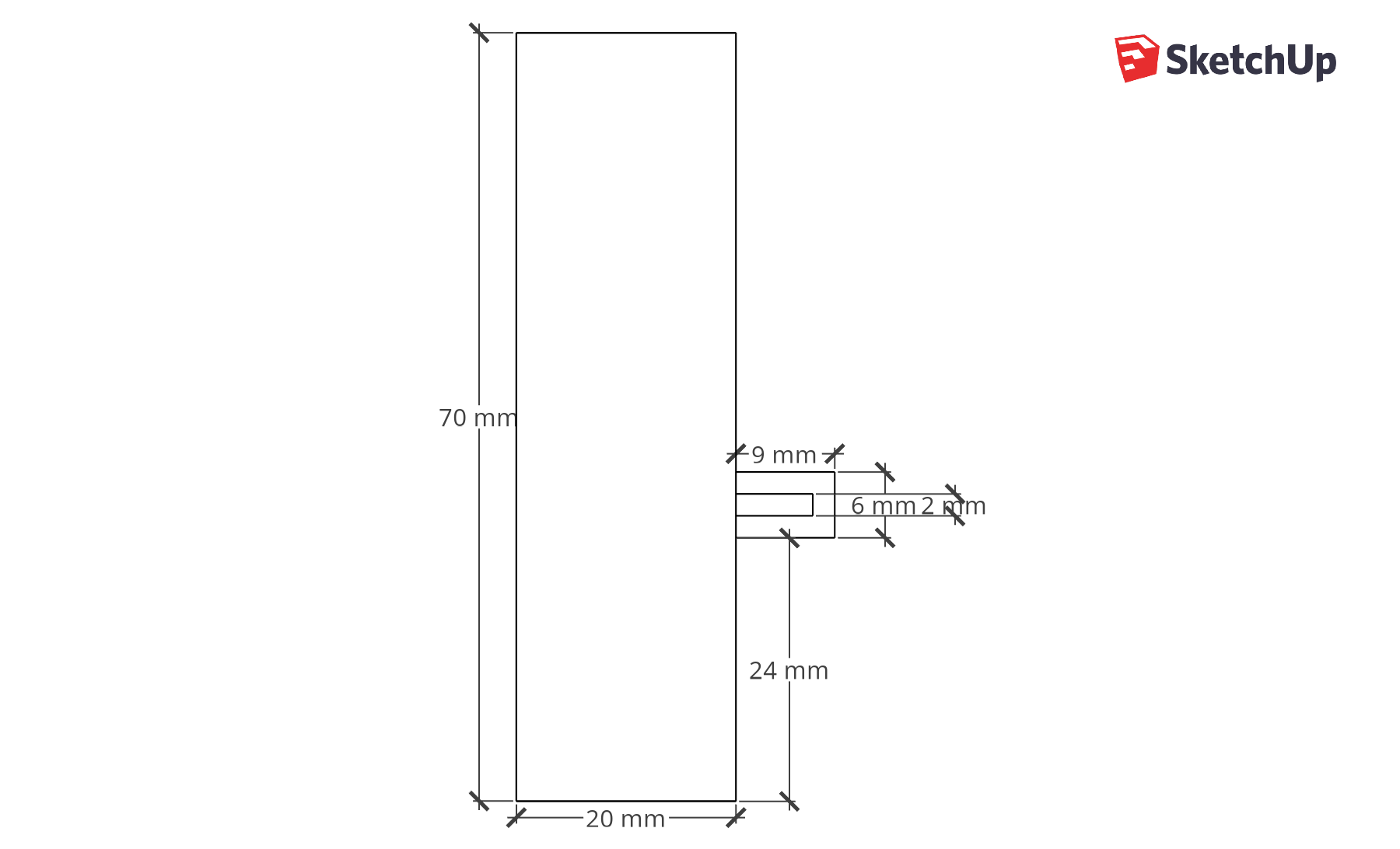


Figure 9: Mechanical drawing of axis beam