UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

Simplifying Part Access

Team #26

Aidan Yaklin, Matheu Fletcher, Tejas Aditya

December 5th, 2023

Team Introductions

Aidan Yaklin

- Senior in CE
- Focus in chip design
- Acted as PCB lead

Matheu Fletcher

- Senior in CE
- Focus in security and system administration
- Acted as Mechanical & CAD lead

Tejas Aditya

- Senior in CE
- Focus in cloud computing
- Acted as code lead

Agenda

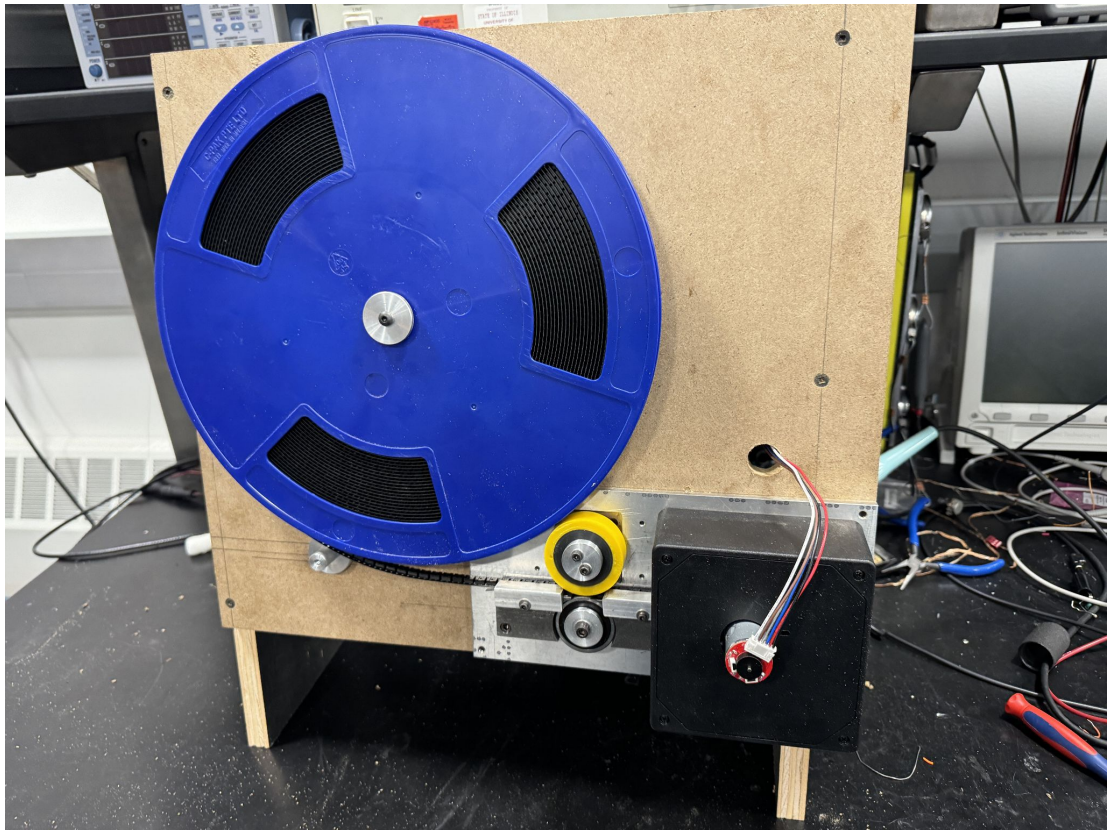
- ❏ Introduction
- ❏ Core components
- ❏ PCB design
- ❏ Software design
- ❏ Mechanical design
- ❏ Changes
- ❏ Demonstration
- ❏ Challenges
- ❏ Conclusion

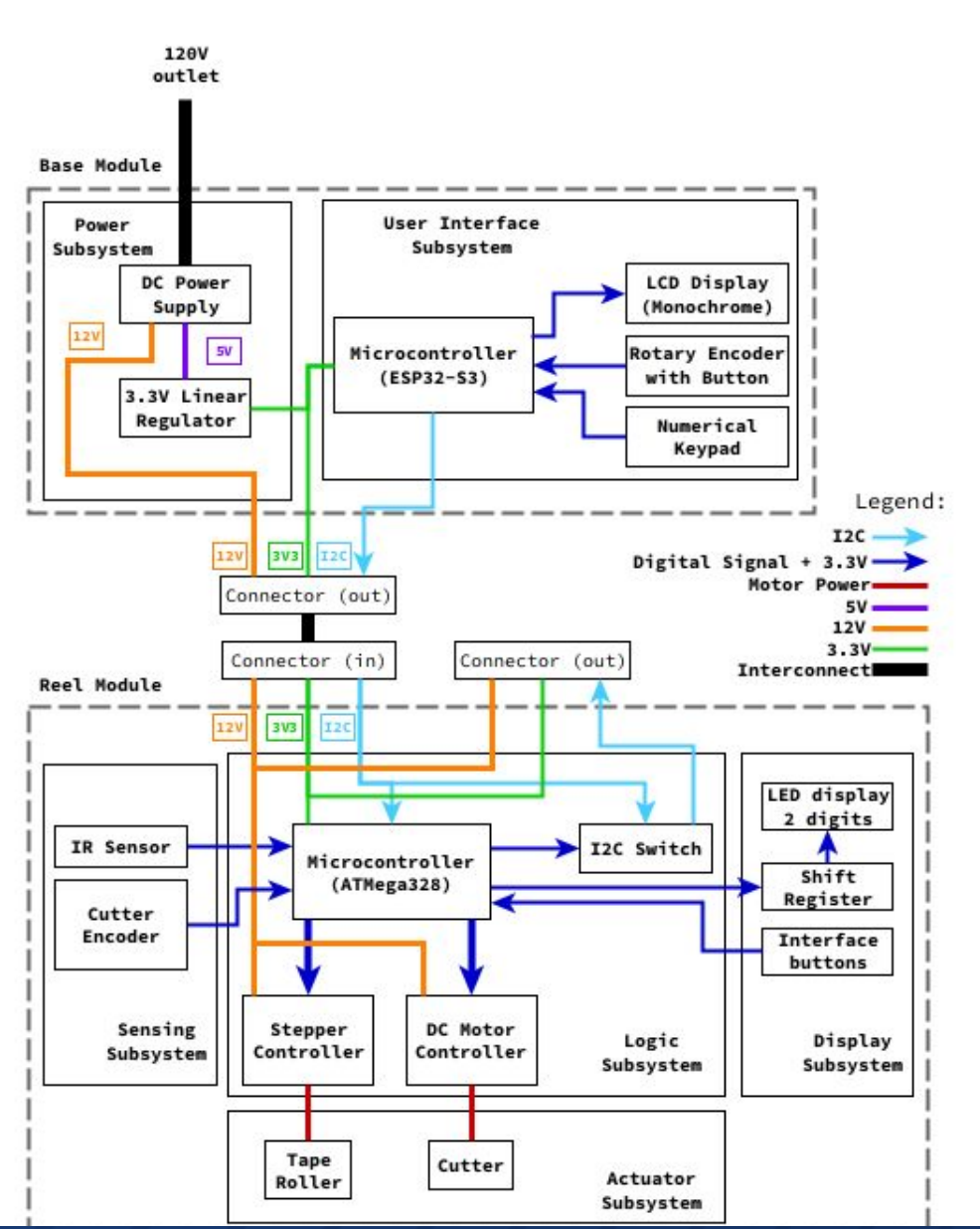
Introduction:

Create an automated solution for dispensing tape and reel SMD components

High Level Objectives:

- ❑ Accept 7” and 13” diameter reels
- ❑ Reels up to 25mm wide, and 4mm deep
- ❑ Precise feed of reel, within +/- 1mm
- ❑ Yield rate of 90%
- ❑ Support up to three modules simultaneously





Block Diagram:

Subsystems consisting of:

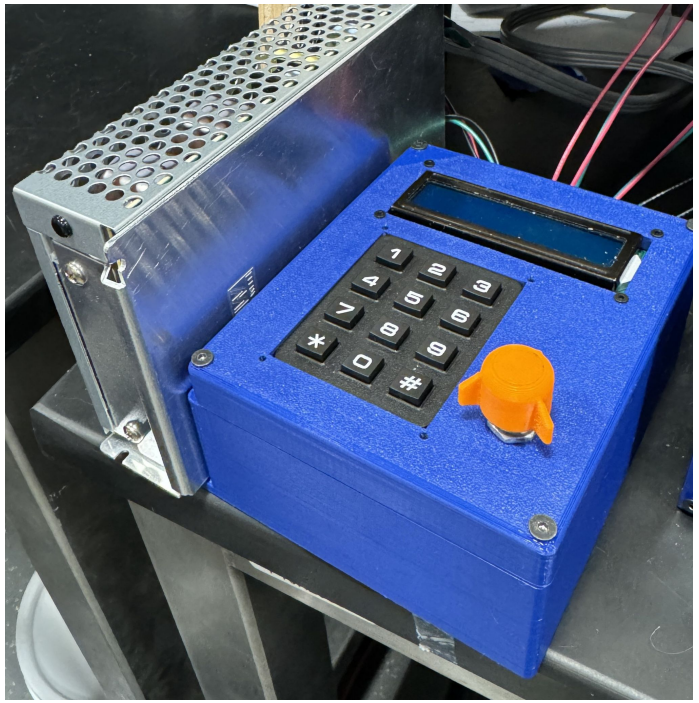
- ❑ Base Module
  - ❑ Power
  - ❑ User Interface
- ❑ Reel Module
  - ❑ Sensing
  - ❑ Logic
  - ❑ Display
  - ❑ Actuator

Requirements & Verifications

Base Module & UI

Able to verify:

- ❑ Behaves as expected
- ❑ Had problems clearing errors

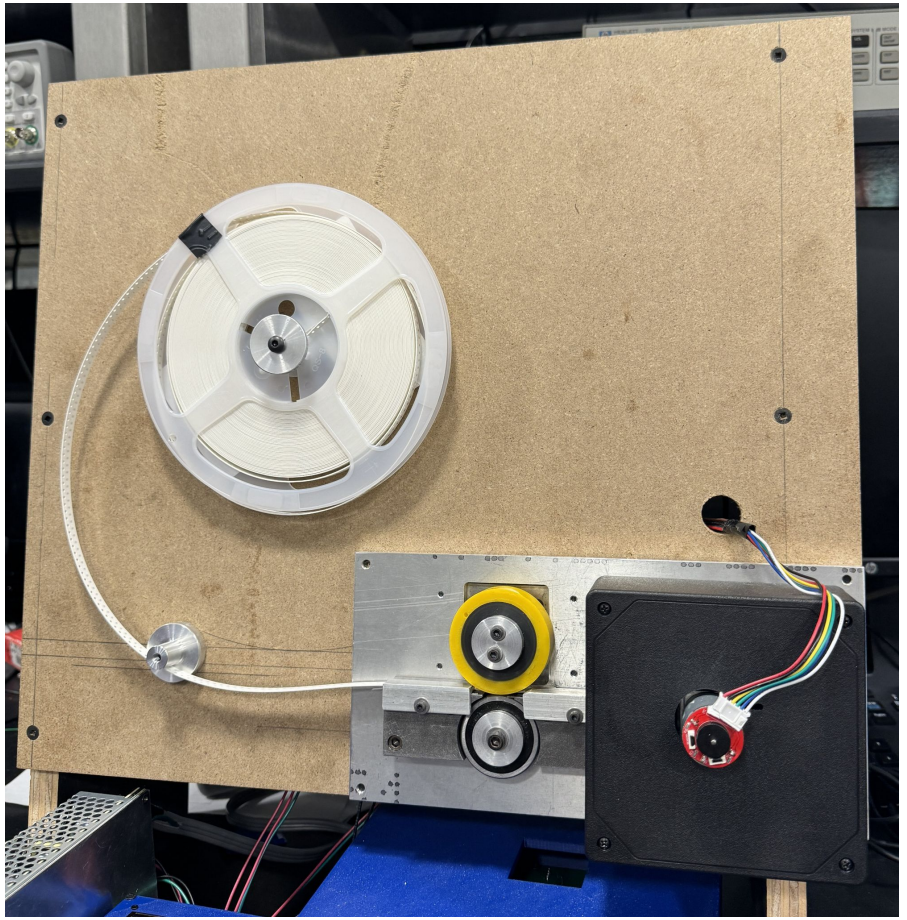


Requirements	Verification
• The ESP32 can enumerate the attached reel modules on the I2C bus • The ESP32 can communicate with the attached reel modules on the I2C bus • The ESP32 can detect communication errors and fail safely	• Connect a reel module or microcontroller with reel module simulation software to the I2C bus • Verify that the ESP32 enumerates the connected module, and reads spacing • Disconnect the module and verify that an error is reported
• The LCD shows: • #cuts and #repeats • reel position in the daisy chain • If an error status is present, the component information should be replaced with an error message • If no modules are attached, the display should show a “No Reels” message	• Verify that the LCD displays “No Reels” • Disconnect power, add a microcontroller with a reel module simulation program to the I2C bus, and reconnect power • Verify that the LCD shows appropriate job information
• The ESP32 can read the rotary encoder and keypad	• Can identify turns of encoder • Can detect encoder button press • Can detect numpad buttons

Reel Module

Able to verify:

- ❑ Reel module attaches to base
- ❑ Reel module can drive stepper at correct voltage
- ❑ Saves spacing to EEPROM

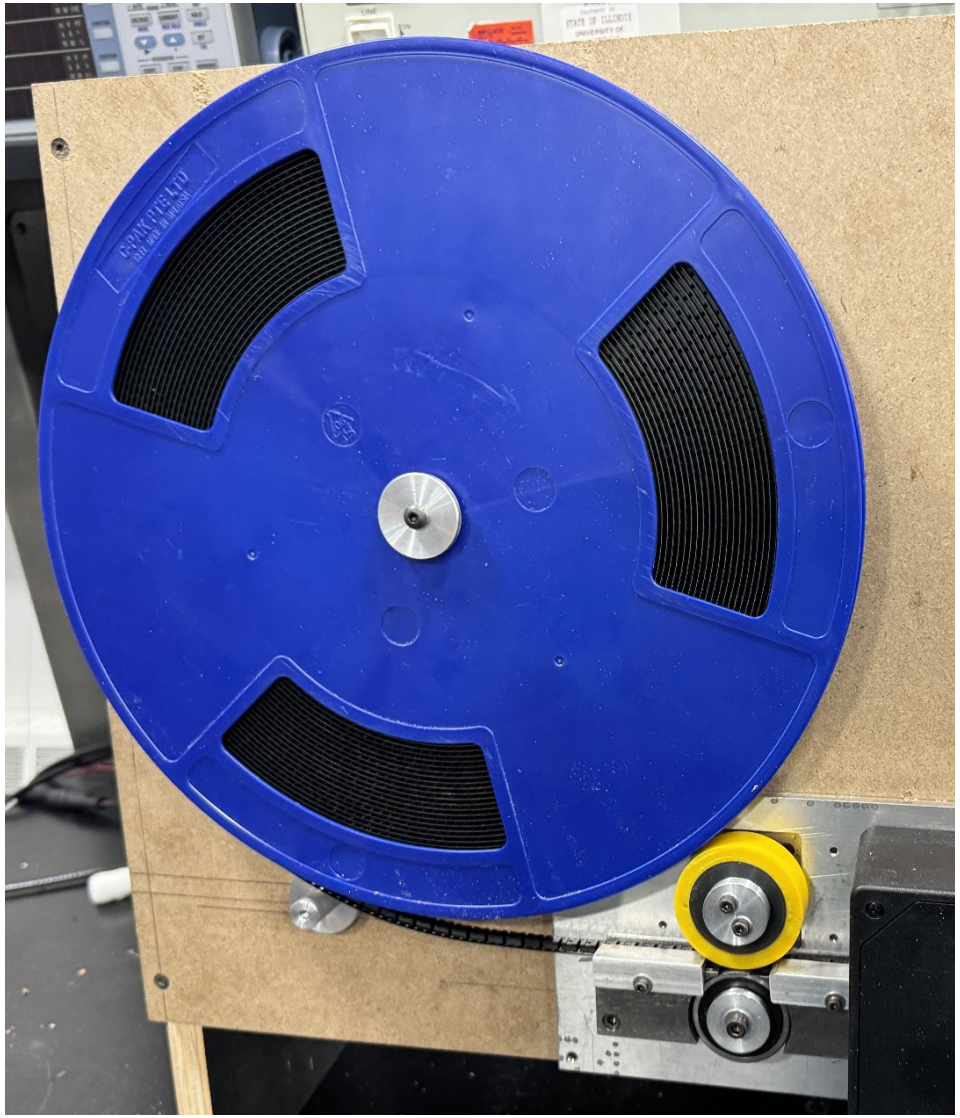


Requirement	Verification
• Can communicate with the base module over I2C at 400 kHz	• Confirm device is enumerated over the I2C bus and should be shown on the base module display
• Confirm pulses to the stepper to accurately dispense the correct number of components	• Use the on-board interface to command the module to dispense 100 components, then verify
• Can save reel configuration and calibration to the chip when powered off	• Set reel spacing • Cycle power to the reel module • Use the on-board interface to verify that the spacing is the same

Feed System verified:

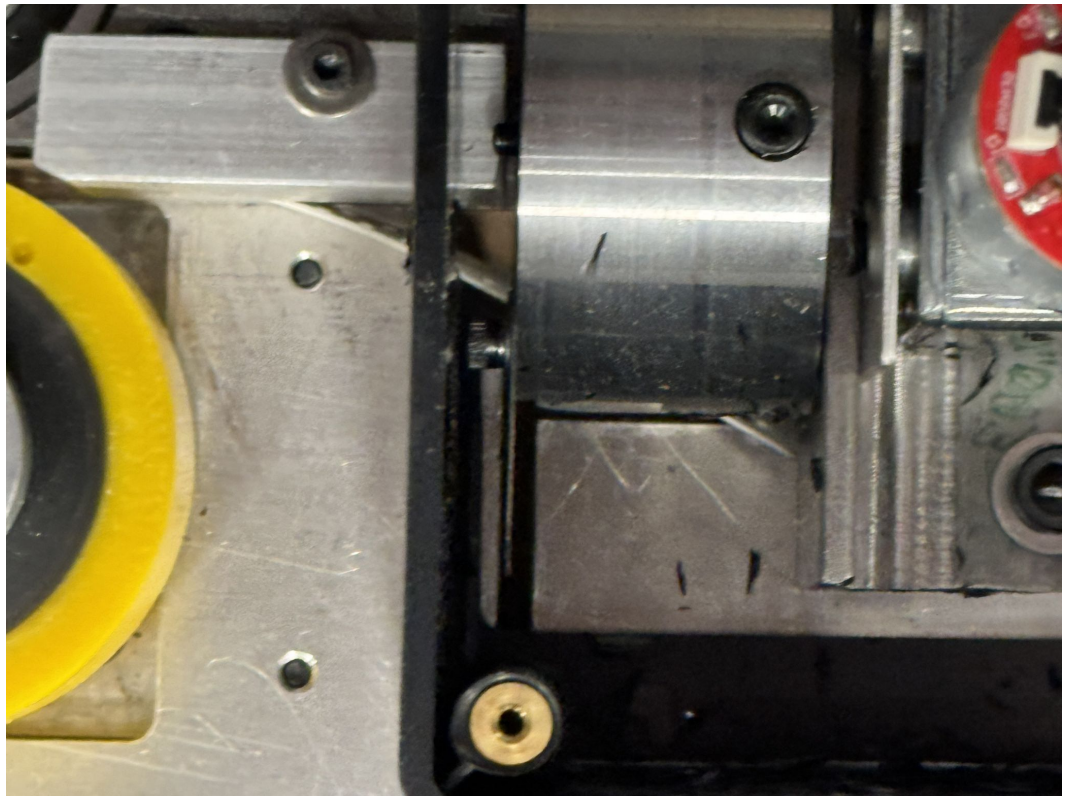
- ❑ Accuracy per step at ~0.43mm
- ❑ ¼ of that with microstepping
- ❑ Met stepper torque requirement at minimum speed allowed (runs at higher)

Requirement	Verification
• Maximum feed rate of at least 5 mm/s	• Command the module to cut 100 components • Measure the distance moved and time to calculate the feed rate
• Tape feed positional accuracy of at most +/- 0.5 mm	• Cut 10 individual components • Measure the width of each cut component and verify that it is within tolerance
• Sufficient torque to pull tape from the reel, especially at low speeds	• Command the module to cut 1 component • Verify that 1 component is cut from the tape, the stepper motor does not stick, and the tape does not slip



Cutter System verified:

- ❑ Cut speed actually 3 seconds (could be improved in software)
- ❑ Kill switch - halfway



Requirements	Verification
• Motor can cut through the the tape reliably within the span of a couple seconds	• Send cut operation, confirm it cuts accurately, not jamming or stalling • Cuts take maximum of 2 seconds per cut
• Confirm that the blade safety mechanism is working	• Ensure user can not fit finger in hole • Ensure that the cutter motor stops entirely after pressing the kill switch

IR System:

Did not get implemented in the end. As such, verifications here were not met

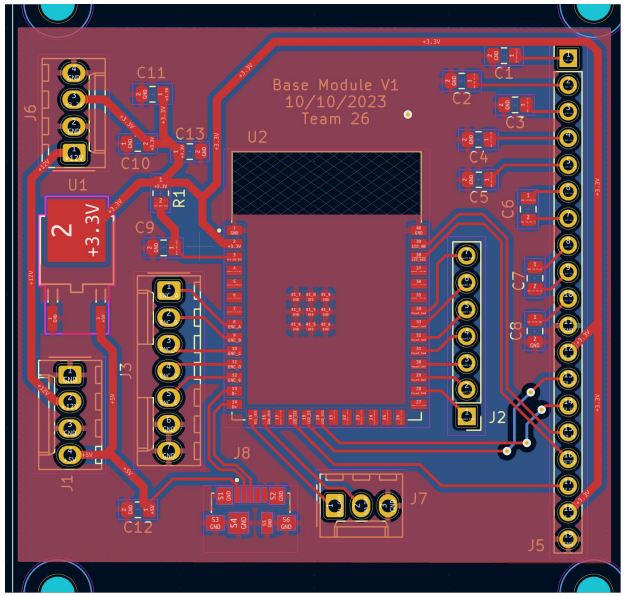
Requirements	Verification
• The IR sensor can distinguish between the tape and a sprocket within the tolerance of ±1%	• Ensure that nothing is in between the IR sensor module and then record the boolean value from the ATMega328's • Pass tape through it and record the value when the tape is blocking the IR path. Ensure that it is different from the initial value • Move the tape so a sprocket is now in between the transmitter and receiver. Confirm that the value is the same as the default value

Assembly

PCB work

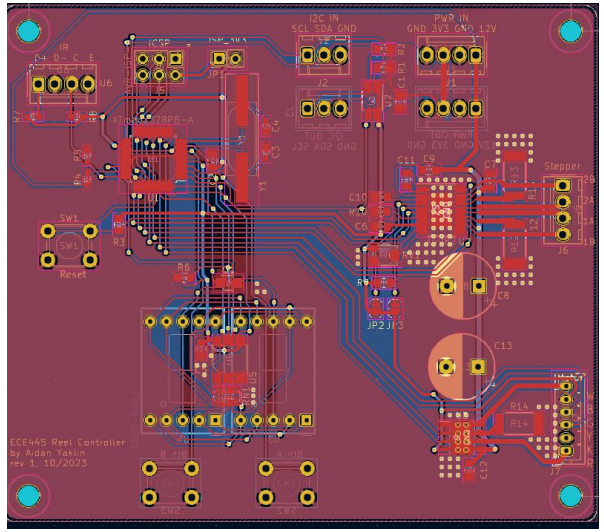
Base Module

- ESP32-S3 Chosen
  - Availability
  - Micro-USB connection
  - Wifi capabilities if upgrades made
- Design with molex connectors in mind
- Optimized component layout



Reel Module

- ATmega328PB
  - Availability
  - Pin count
  - High reliability
- Motor Drivers
  - DRV8231 H-bridge motor driver for the cutter,
  - A4982 stepper controller
- I2C switch for daisy-chaining
- LED display for debugging



Code

Base Module Code

- Based on the Arduino Core for ESP32-S3 library
- Polling driven design
  - Keyboard and encoder input at desired time
  - Error and run availability polling

Reel Module Code

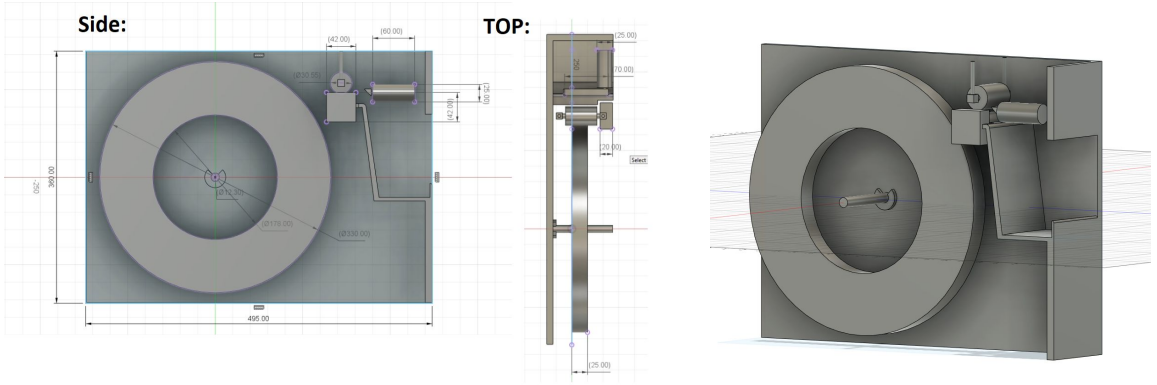
- Bare-metal C code (no Arduino library)
- Interrupt-driven design
  - Timers for regular tasks
  - Pin-change interrupts
  - I2C command interrupts
- Main loop implemented as a state machine

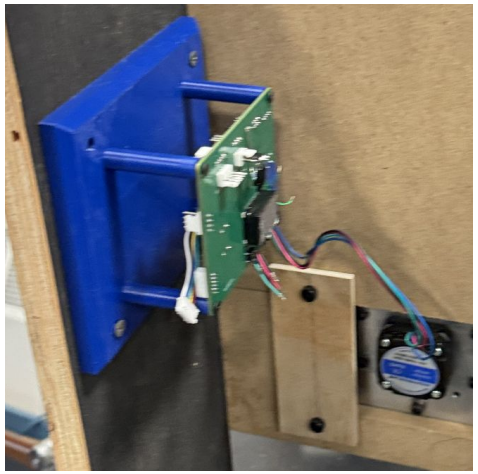
```
uint8_t lsb = spacing & 0x00FF;
uint8_t msb = (spacing & 0xFF00) >> 8;
Wire.beginTransmission(attached_reels[addr_idx].addr);
Wire.write(SPACING_REGISTER_LB_CMD); // lower byte is 
Wire.write(lsb);
Wire.write(SPACING_REGISTER_UB_CMD);
Wire.write(msb);
return Wire.endTransmission();
```

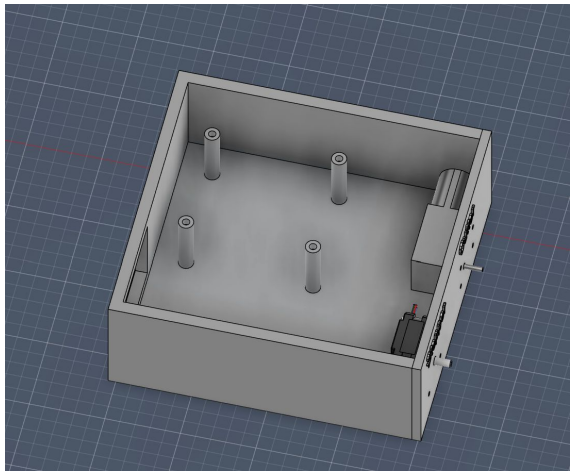
```
//  - COM3A = 0
//  - COM3B = 0
//  - WGM3 = 0100 ([1:0] = 00)
TCCR3A = 0;
//  - ICNC3 = 0
//  - ICES3 = 0
//  - WGM3 = 0100 ([3:2] = 01)
//  - CS1 = 000 (no clock)
TCCR3B = (1 << WGM32);
```

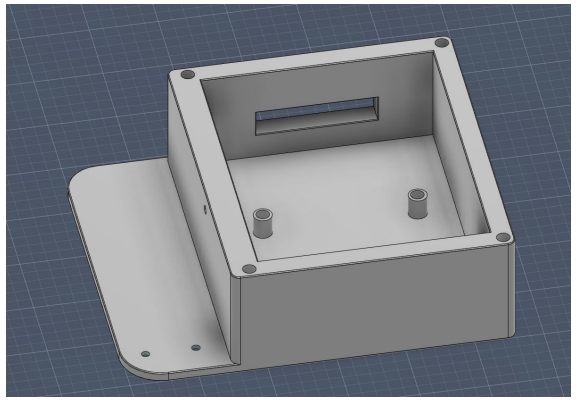
General Design:

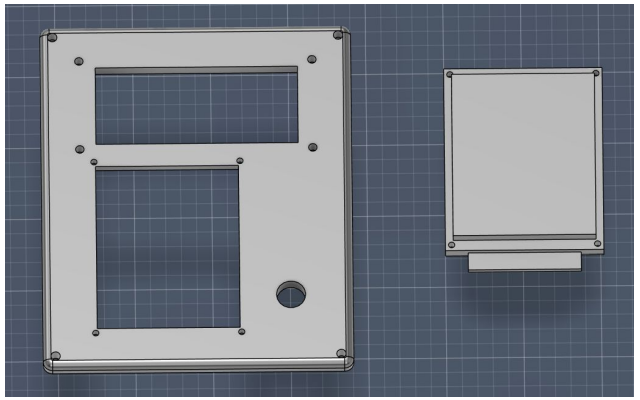
- CAD work consisted of:
  - Initial motor mounts and feed design
  - Mounting standoff for reel module
  - Box for reel module
  - Knob for rotary encoder
  - Box for base module
  - Power supply cover







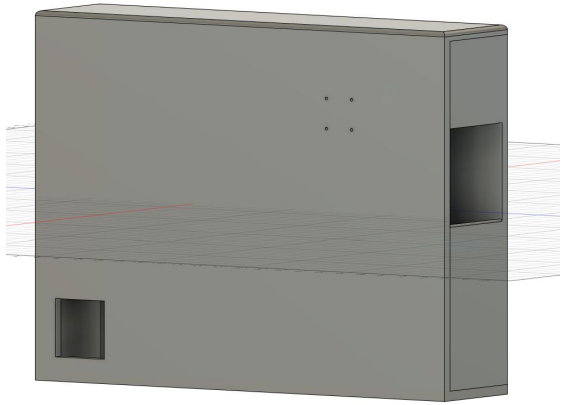


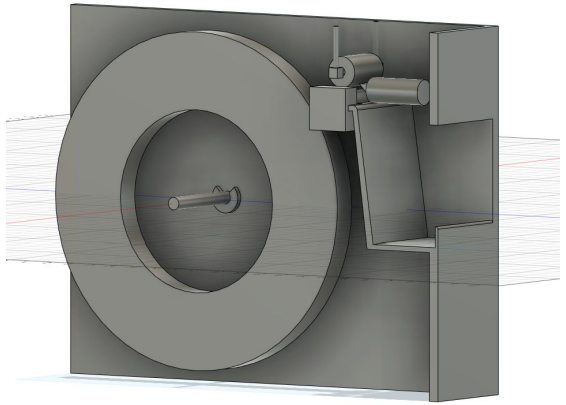


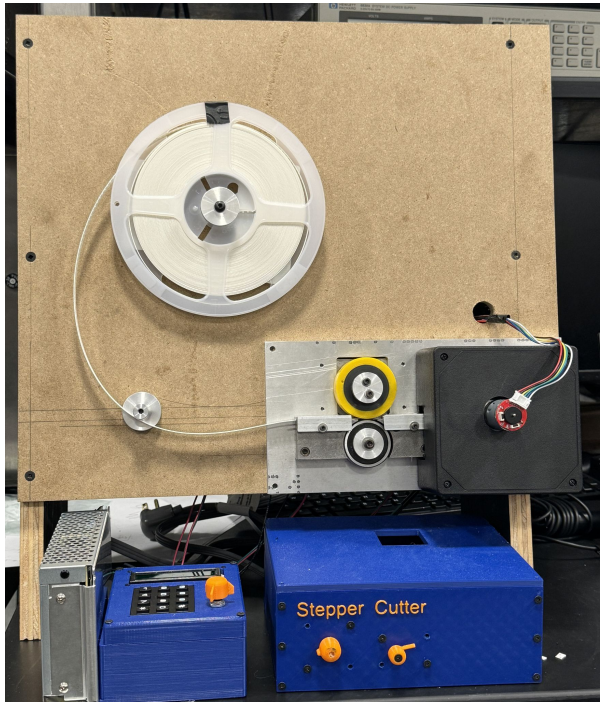
Design Changes

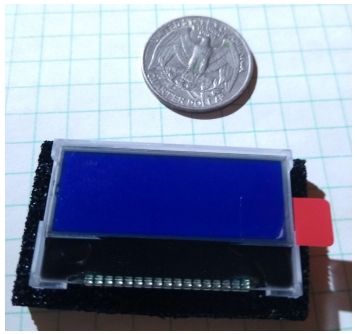
From concept to physical

- ❑ Multiple -> Single mechanical module
- ❑ Reduced max reel dimensions
- ❑ Quantity of components per cut limited
- ❑ Changed screen component used
- ❑ Flipped CAD design
- ❑ Changed cutter motor
- ❑ Roller dimensions
- ❑ Definitive last Reel module





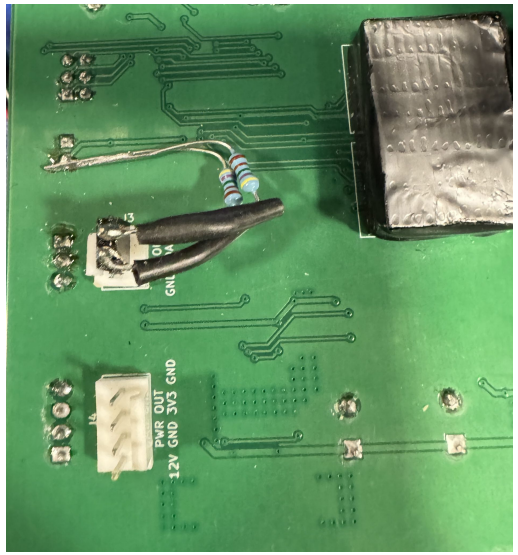


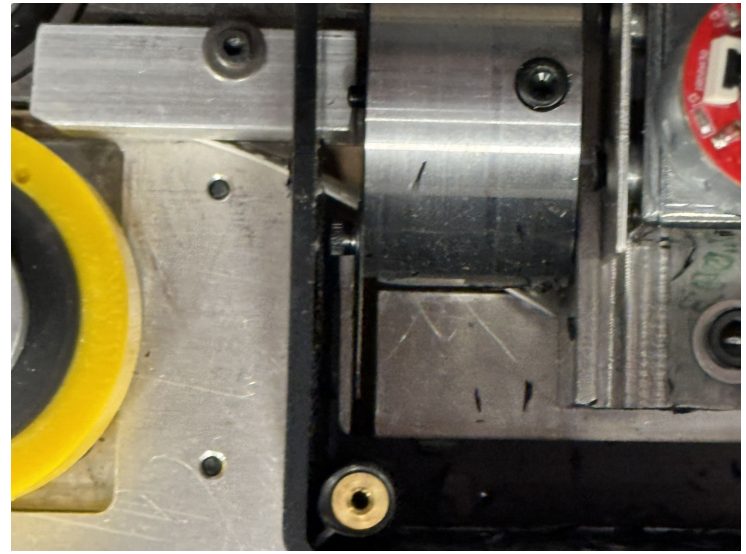


Numerous mistakes

We had a number of occurrences and problems that required debugging and redoing work in order to get the project in a testable state. These included:

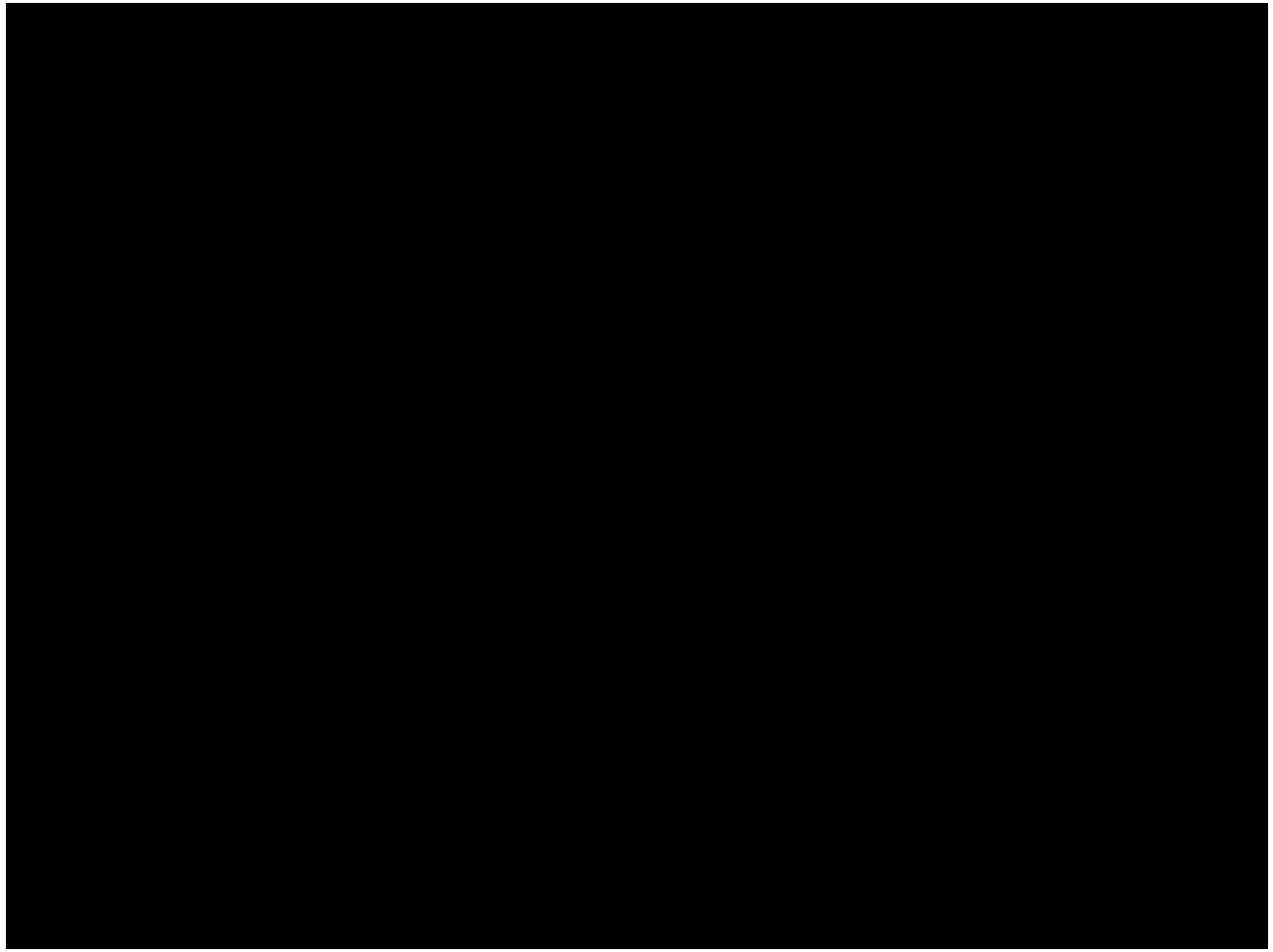
- ❑ Frying a pin of the Atmega chip
- ❑ Loose pins on the screen
- ❑ Lack of test points
- ❑ Missing program-enable button for ESP32
- ❑ Cutter motor jamming
- ❑ Fix floating I2C with jumper





Final touches

Demonstration:



Showcasing the operation of the final product

Conclusions

Works, but needs calibration

- ❑ Feasible
- ❑ Works
- ❑ Initial setup is tricky
- ❑ Assuring exact measurements

We learned:

- ❑ Implementation of PCB design
- ❑ I2C
- ❑ General design and test of a prototype

If done again, would get parts earlier to better test subsystems early

Focus on accuracy, and scaling

To continue this project, the primary goals would be to:

- ❑ Focus on calibrating the feed distance
- ❑ Improve cut speed (software patch and improved motor)
- ❑ Make cut pieces fall out cleaner
- ❑ Make design slimmer (via board design and motor choices)
- ❑ Keep smaller tapes from trying to curl up
- ❑ Better aligned cutting blade
- ❑ Being able to run the dispensing operation on multiple reels simultaneously

Ethics Concerns

- Cutter
  - Box mostly prevents harm
    - may have gap where tape feeds
  - Killswitch
    - Prefer hardware overcurrent software
- Seek honest criticism (IEEE 7.8. I.5)
- Treat others fairly (IEEE 7.8 II)

Thank You

Any Questions?

The Grainger College of Engineering

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN