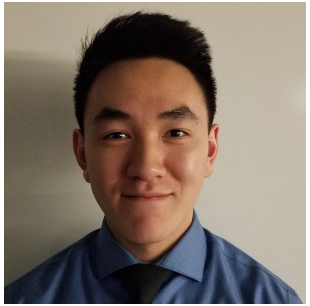
TrueTouch

Final Project Review

Team 19

Our Team



Anthony Chan: EE

Alex Dickopf: CSE

Cameron Kluza: CSE

Jonathan Yip: CSE

Problem Statement

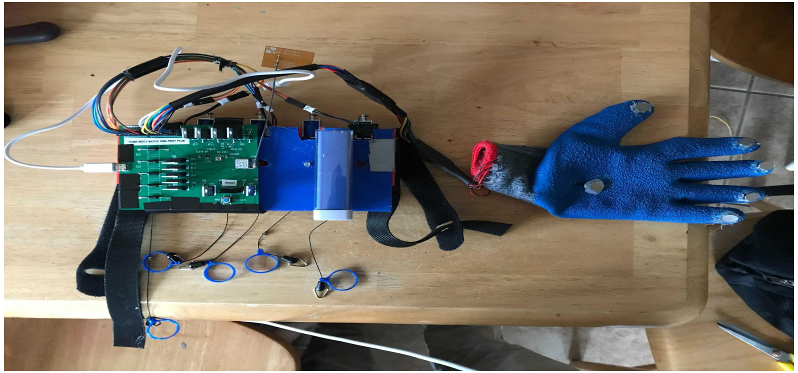
- VR lacks sensation of physical touch
- Not realistic if your fingers can go through an object
- Hand held controllers
  - Hand locked in a specific shape
  - Limited vibration areas
- Improvement of immersion

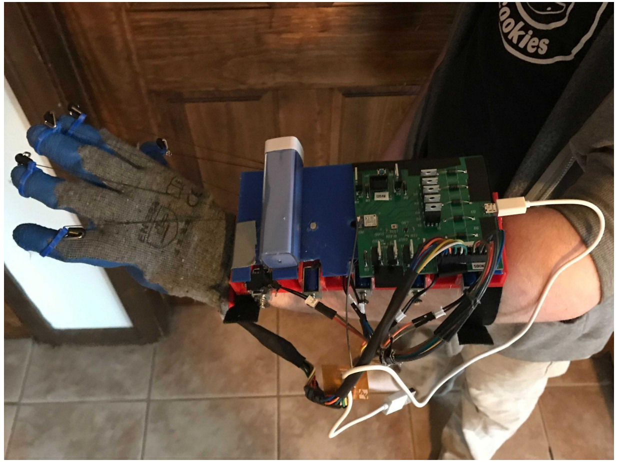
Haptic Feedback

- Feedback to simulate physical touch
- Different kinds of feedback
  - ...such as force, vibrotactile, ...
- Sense of impact and vibration
  - vibration motors (ERM, LRA, etc)
- Sense of shape
  - Exoskeletons

Proposed Solution: TrueTouch

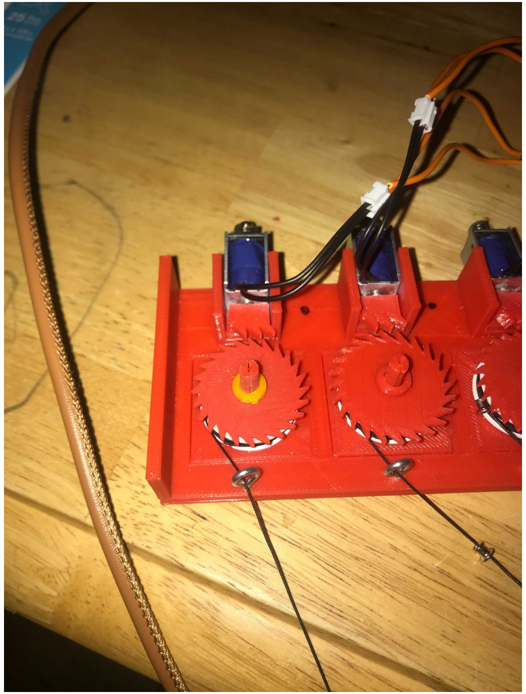
- Combining both: feeling of shape and impact
- Controller free
- Lightweight glove, extending onto the arm
- Integrate with hand tracking





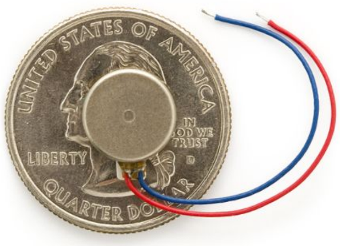
TrueTouch's "Braking System"

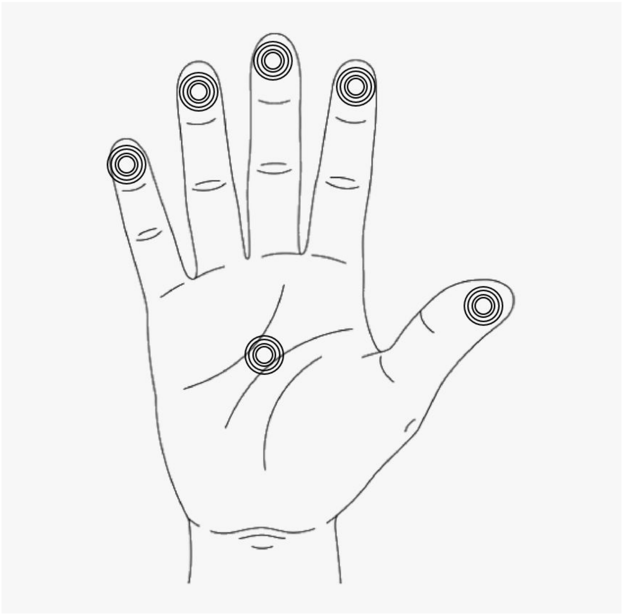
- System is mounted on the top of the forearm
- Retractable string attached to the back of each fingertip
- Length of the string attached to the back of the fingertips vary as you bend your finger
- Limiting the length of a string along the back of your finger will limit how far you can bend
- Using five ratchet and pawls to lock string length (inspired by Wireality)



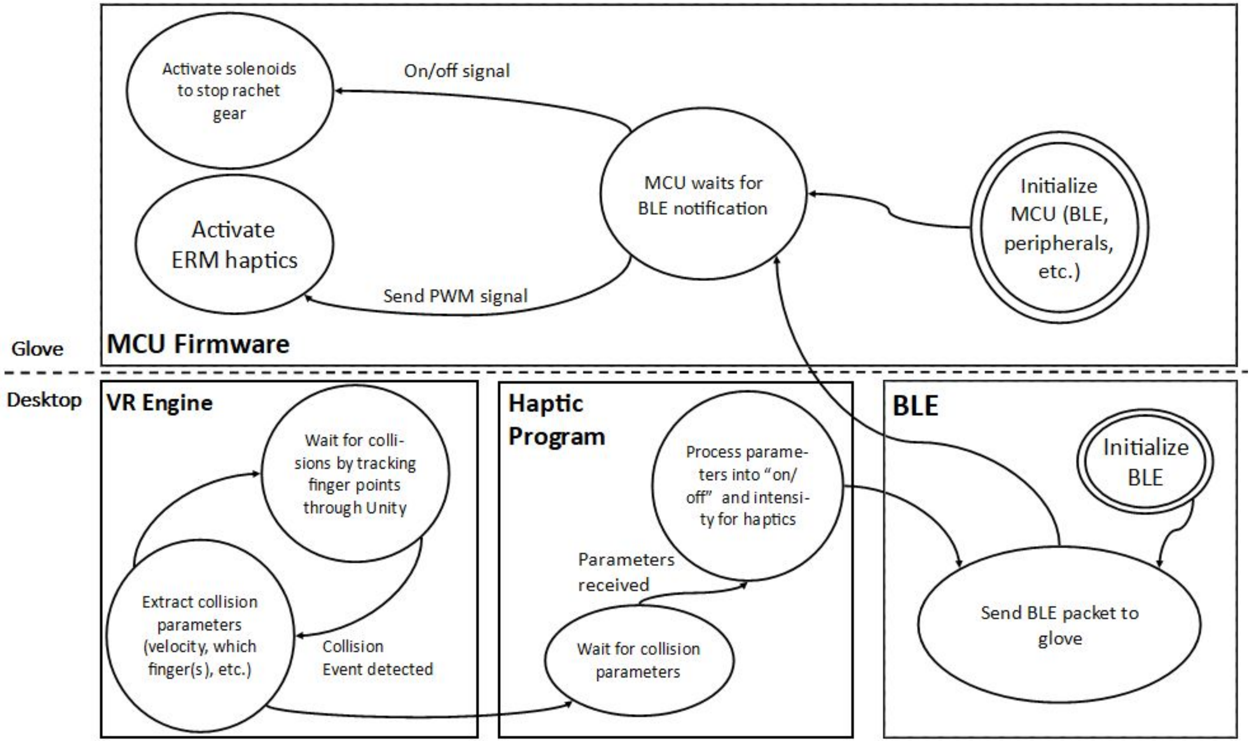
TrueTouch's Vibrotactile System

- Small Eccentric Rotating Mass (ERM) motors will provide vibrations, simulating impact
- ERM motors will be placed on palm and each finger to provide sensation of impact against hand

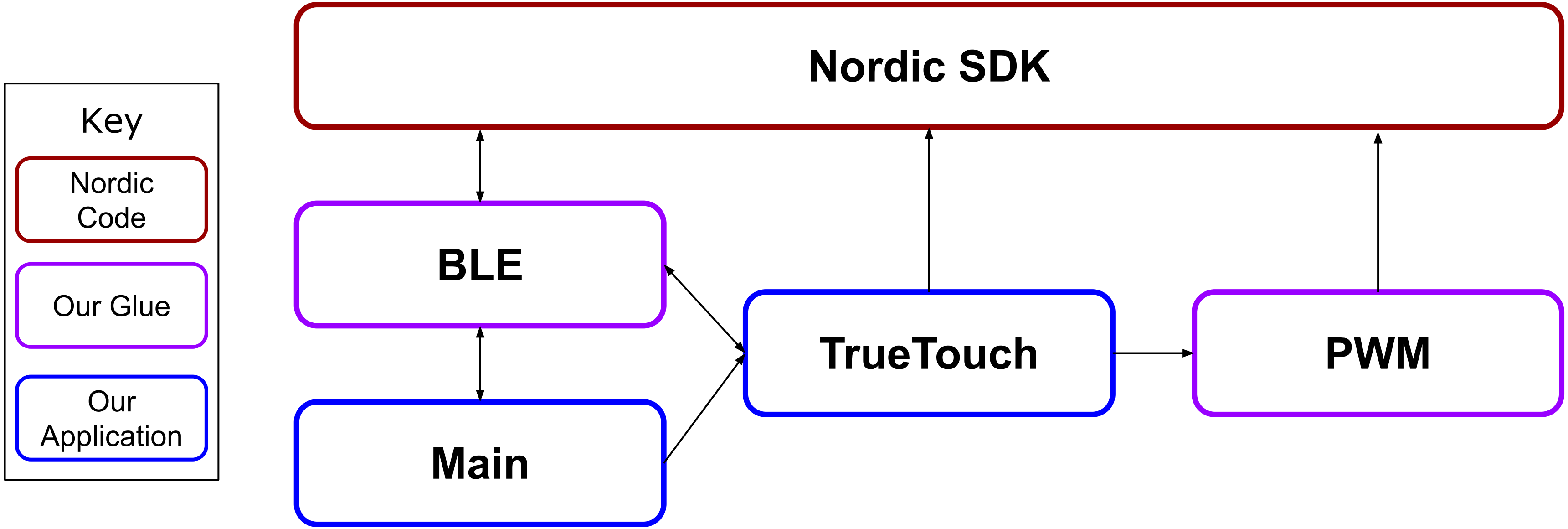




Software Diagram



System Software - Glove Firmware



System Software - Glove Firmware

- Glove firmware is built using the Nordic SDK
  - Nordic SDK includes binary for the Nordic BLE stack, libraries to interact with it, and useful middleware
- Uses the Nordic UART Service for underlying communication
  - BLE service provided in Nordic SDK
- Implements custom TrueTouch protocol on top of this
  - Sends commands + parameters formatted in packed structs

Glove Firmware - Main

```
ble::init(ble_event_callback);

/* CTOR registers BLE callback and configures solenoid/ERM pins */
TrueTouch truetouch {};

// Start execution.
ble::advertise();

// Enter main loop.
for (;;)
{
    idle_state_handle();
    truetouch.service();
}
```

Glove Firmware - TrueTouch Service

```
/* First byte is command, peek that. */
const Command command = static_cast<Command>(_buffer[0]);

/* Perform command-specific processing (break if all data bytes aren't received yet) */
switch (command) {
    case Command::SOLENOID_WRITE: {
        handle_solenoid_write();
    } break;

    case Command::SOLENOID_PULSE: {
        handle_solenoid_pulse();
    } break;

    case Command::ERM_SET: {
        handle_erm_set();
    } break;
}
```

Glove Firmware - TrueTouch Pulse Timer

```
/* Turn off the current pin. */
auto pin = finger_to_solenoid_pin(_current_pulse_bit);
nrf_gpio_pin_clear(pin);

/* Stop if no more pins to pulse */
if (!_pulse_pin_bitset) {
    _pulse_dur_ms = 0;
    _current_pulse_bit = NO_ACTIVE_BIT;
    return;
}

/* Start the next pin. */
_current_pulse_bit = util::get_highest_bit(_pulse_pin_bitset);
util::clear_bit(_pulse_pin_bitset, _current_pulse_bit);
pin = finger_to_solenoid_pin(_current_pulse_bit);

nrf_gpio_pin_set(pin);
APP_ERROR_CHECK(app_timer_start(_timer, APP_TIMER_TICKS(_pulse_dur_ms), this));
```

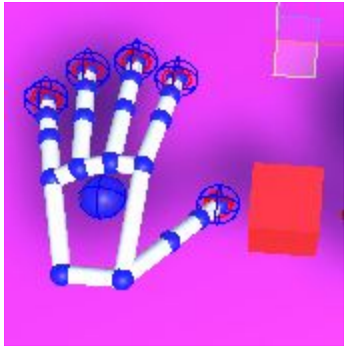
Leap Motion Controller

- Used for hand tracking
  - Replaced HTC Hand Tracking SDK
- Uses IR light to track the hand
- Has tracking points on the tip and middle joint of each finger and on the palm

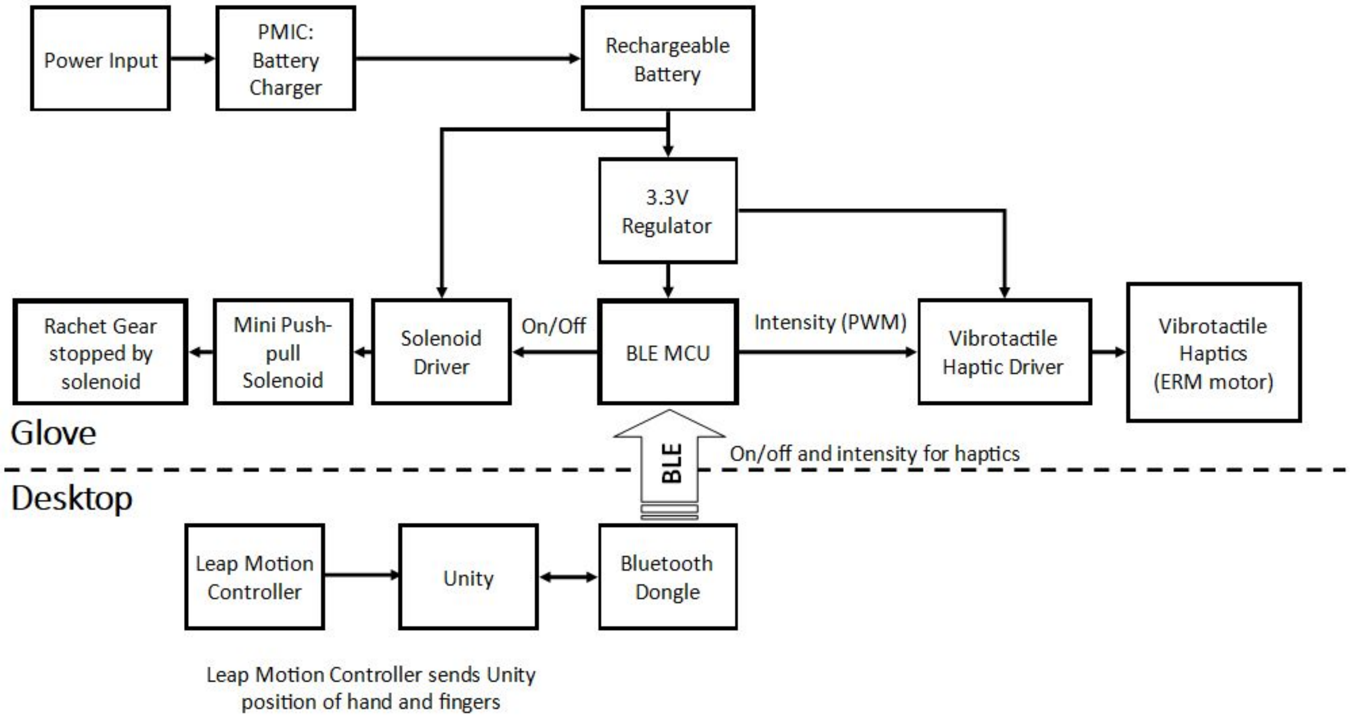
Desktop Unity

- Connected MCU to Unity through BLE
- Simple objects in the scene that the user can interact with
- Detect when collisions happen between finger and object
- Each finger’s solenoid and ERM motor can be independently controlled

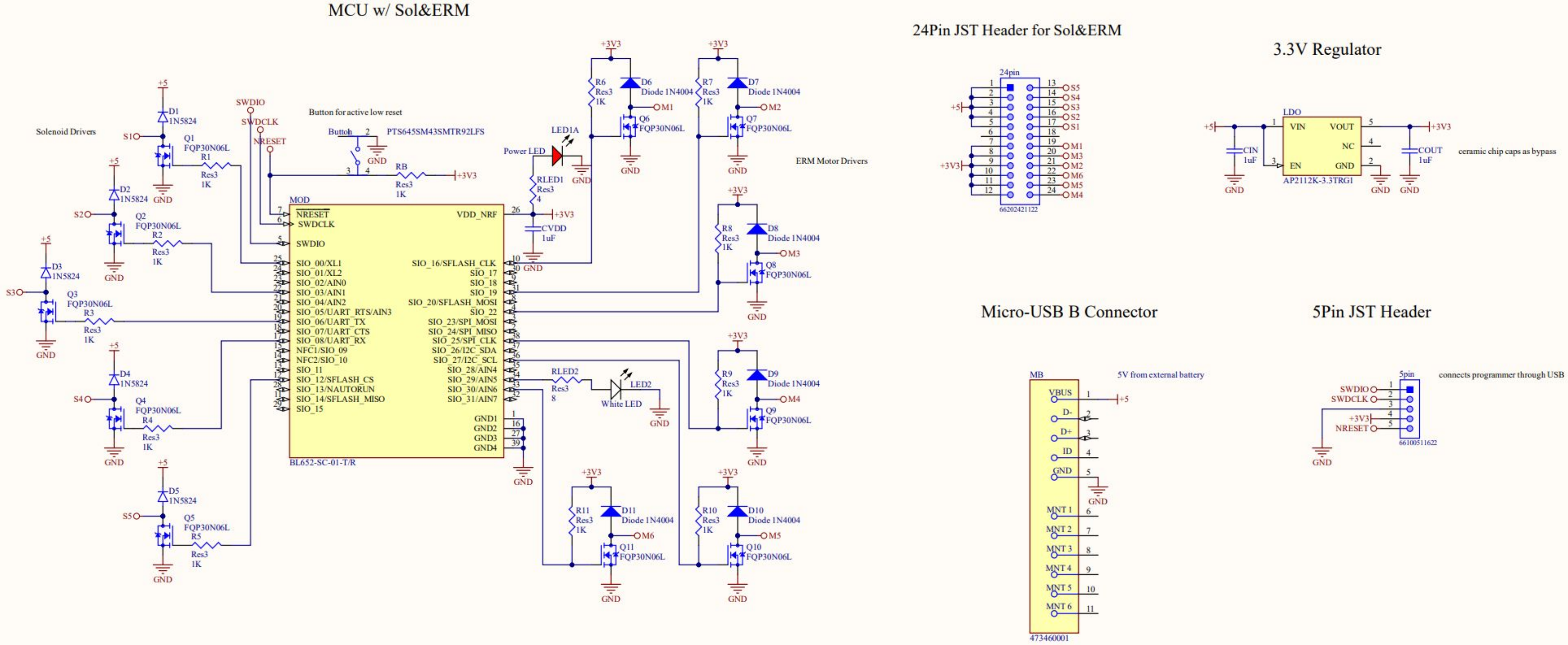
```
public void Thumb()
{
        tooch.UpdateFinger(TrueTouchBLE.Finger.THUMB, TrueTouchBLE.UpdateType.PULSE_SOLENOID);
        print("ThumbS");
}

public void ThumbERMOn()
{
        tooch.UpdateFinger(TrueTouchBLE.Finger.THUMB, TrueTouchBLE.UpdateType.SET_ERM, 255);
        print("ThumbEOn");
}
```



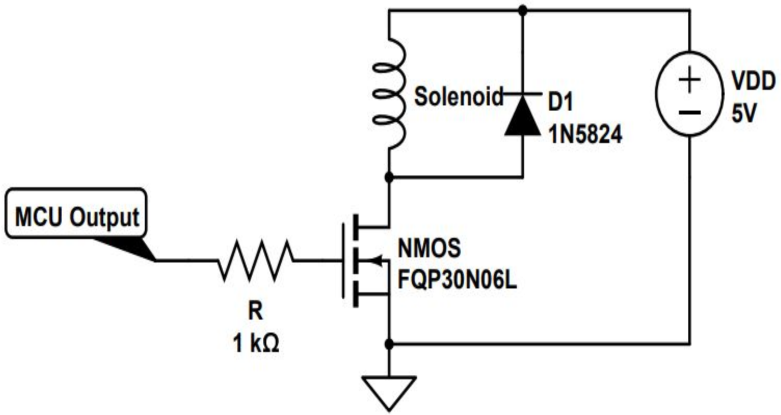
Hardware Diagram

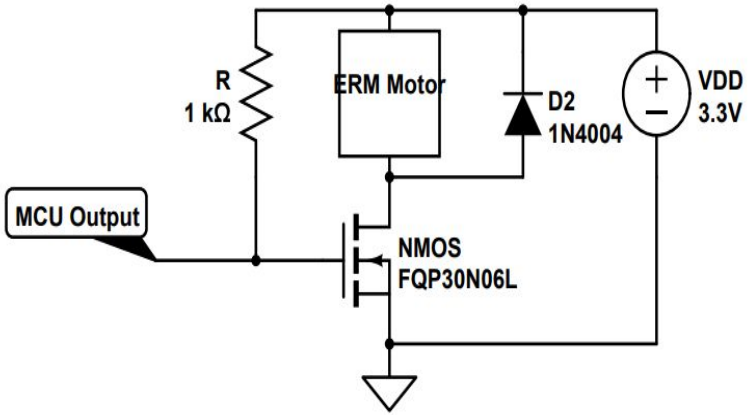


Custom PCB - Schematic

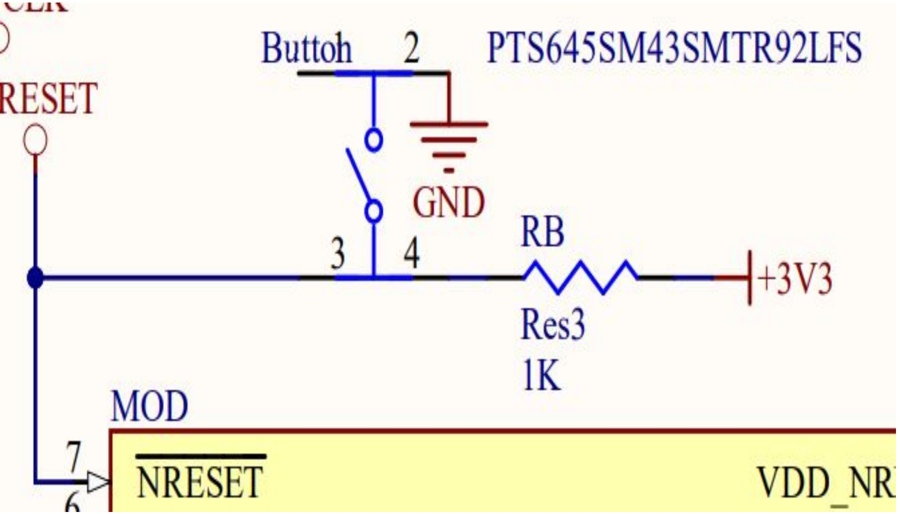


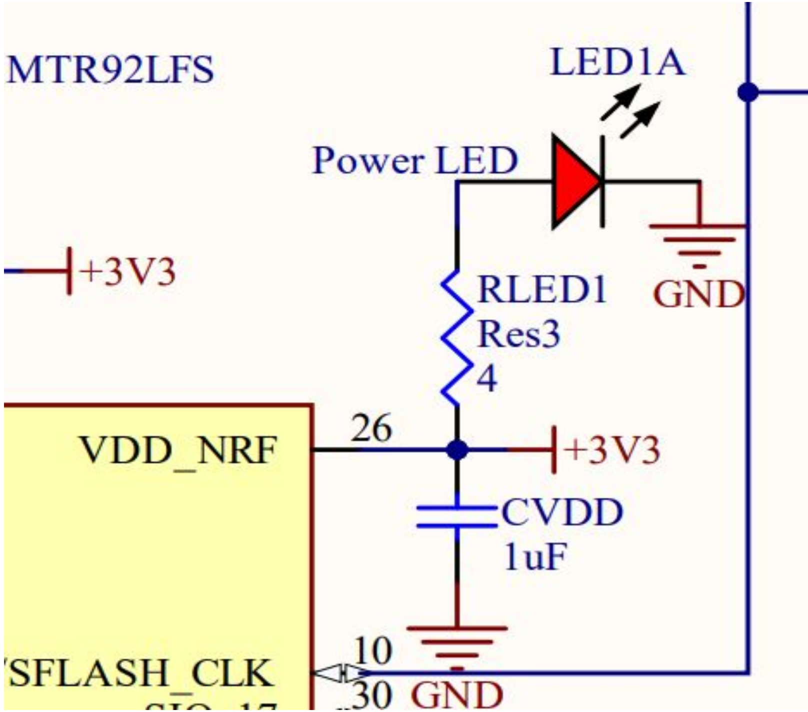
Custom PCB - ERM and Solenoids Schematic





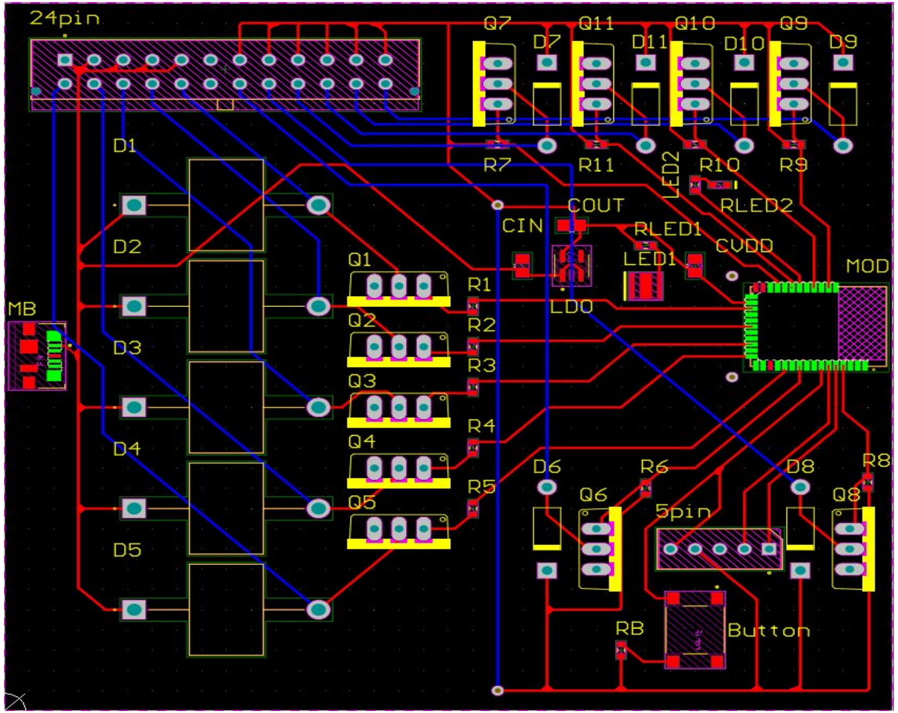
Custom PCB - MCU Schematic

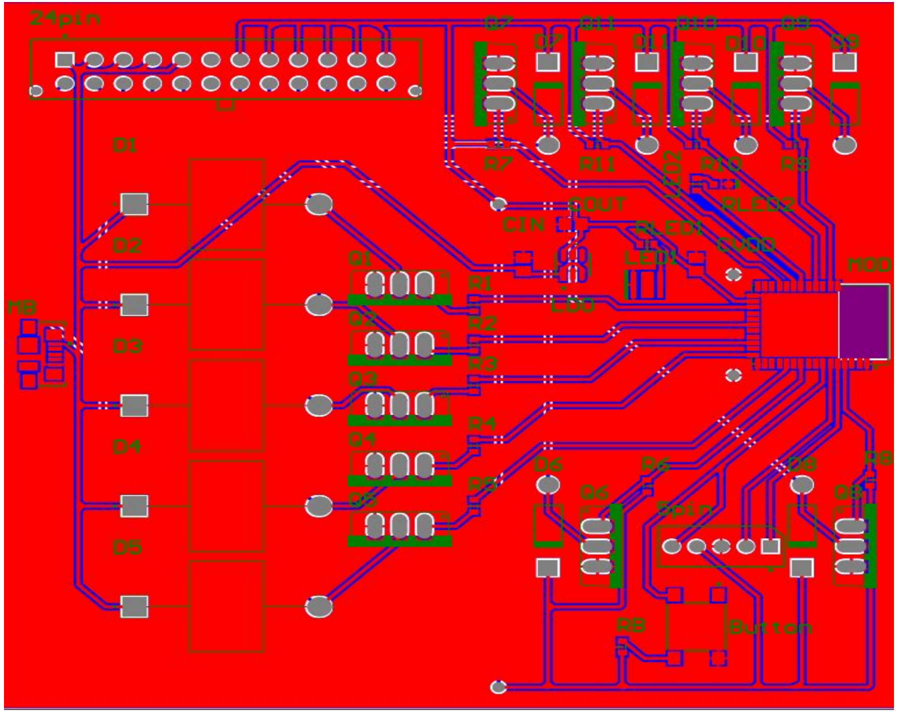




UMassAmherst

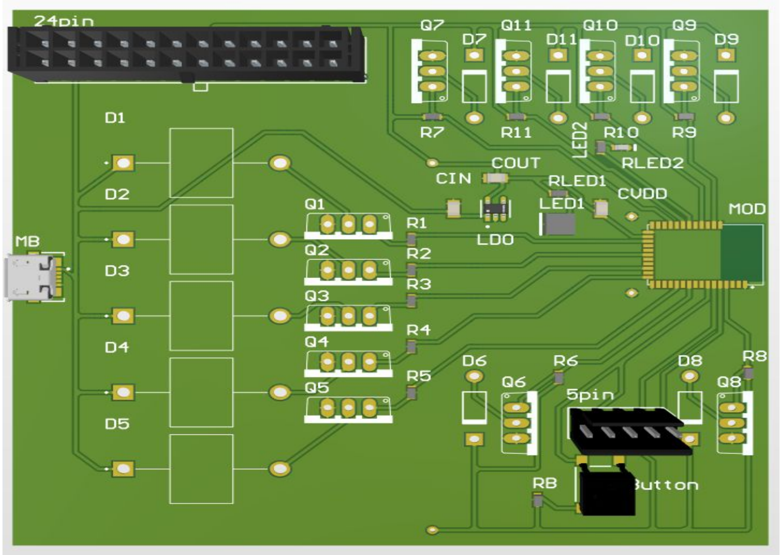
Custom PCB - Layout





Custom PCB - 3D

- LxWxH
  3.6 x 3.5 x 0.9 in
  (9.14 x 8.9 x 2.3 cm)



Demo Time!

Verifying Requirements

- Wireless system
  - No testing needed
- Single glove with system weighs no more than 2lbs
  - We weighed our system with a scale and it was 0.8125 lbs
- Braking system able to withstand 20N of pulling force from finger-bending strength
  - We attached a 20N force at the end of one of the ratchet and pawl systems to verify it wouldn’t break

Verifying Requirements - Battery Life

Rechargeable battery of minimum 1 hour of life

- The following power consumptions have been measured:

Parameter	Current
BLE Module Active Mode (Connected)	**5.4mA**
Power LED ON State	**2.2mA**
3.3V Voltage Regulator	**80uA**
NMOS OFF State (x11)	**10uA** * 11 = **110uA**
NMOS ON State (x11)	**250uA** * 11 = **2.75mA**
ERM Motors ON State (x6)	**60mA** * 6 = **360mA**
Solenoid Pulse (10ms) (x5)	**1A** * 5 = **5A**

Current	7.79mA

Verifying Requirements - Battery Life

- Rechargeable battery capacity - 2,200mAh
- Assuming worst case scenario
  - Scenario: The system is turned on with all 6 ERM motors vibrating for an hour
  - What is the maximum number of solenoid activations/pulses allowed?

7.79mA + 360mA + 1.5mA = 369.29mA , 2,200mAh - 369.29mAh = 1,830.71mAh

- This scenario draws about 369.29mA, leaving just under 6 hours of battery life total or about 1830.71mAh remaining after an hour of runtime
- Solenoids pulse for 16ms @ 1A each activation

1Hr / 16ms = 225,000 Activations , 1,830.71mA / 1Ah = 1.83071

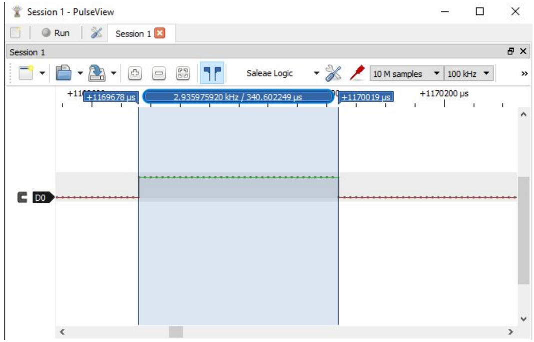
225,000 Activations * 1.83071 = 411,909.75 Activations

- 1Ah or 1,000mAh will be consumed after 225,000 solenoid activations
- The remaining 1,830.71mAh calculated earlier will leave enough battery life for about **411,909.75** solenoid activations in that same hour

Verifying Requirements - 100ms Response Time

Braking system and vibration activates within 100ms of seeing the object being touched in virtual world

- Time for BLE communication determined by measuring round-trip time and halving it
- Time for MCU to process determined by measuring GPIO with logic analyzer
  - On average 0.34 ms spent processing
- Time to extend solenoid calculated based on force applied and distance to travel
  - Anticipate <10ms to extend
- Overall average: ~87.5ms



BLE RTT Statistics	
Average	155.4 ms
Std. Dev.	18.8 ms
Min	101.5 ms
Max	185.4 ms

Thank you!

Questions?