# Magic Mice

Team C3: Bradley Zhou, Jade Wang, Jenny Han

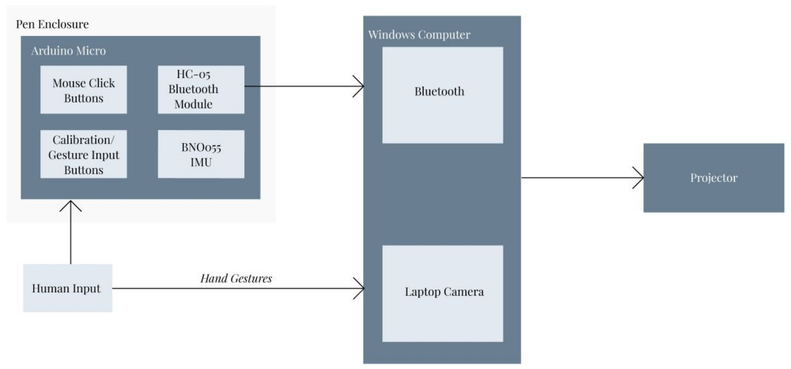
Electrical & Computer ENGINEERING

## Product Pitch

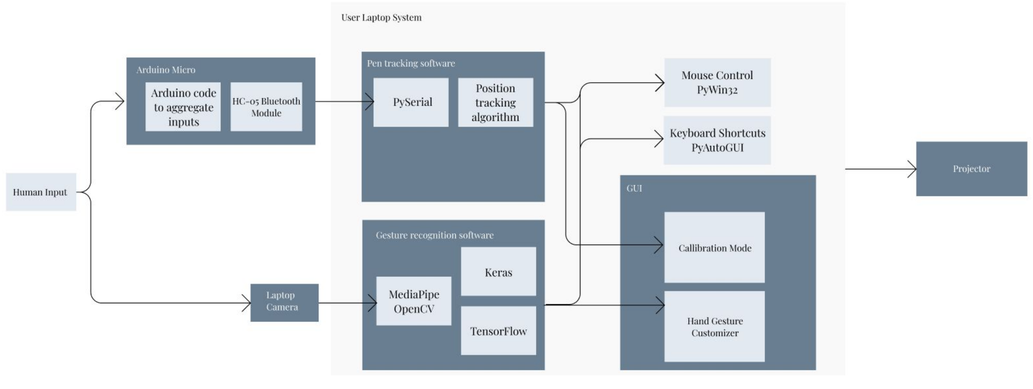
Our goal is to make collaboration in teams more accessible. Magic Mice is a projector computer attachment that displays a Windows computer screen on an indoor wall from 5 to 10 ft away. There are two types of input, a custom pen device and the computer webcam video feed. The pen controls mouse movement and camera detects customizable hand gestures as well as hand tracking. The combination of these two inputs simulates the functionality of a smartboard and more, by not only allowing the user to mimic mouse movement including, click, drag, and hover, but also keyboard commands that can be custom mapped to the user's hand gestures. Magic Mice allows users to interact with the projected wall via pen with a delay of 20 ms, and via hand gesture with a delay of less than 50 ms. Magic Mice is an affordable collaboration tool for team workers in educational, professional, entertainment settings, etc., as it allows users to draw on sketch websites, navigate browsers and file systems, and more.

## System Architecture

### Hardware Diagram




### Software Diagram




## System Description

*Pen*

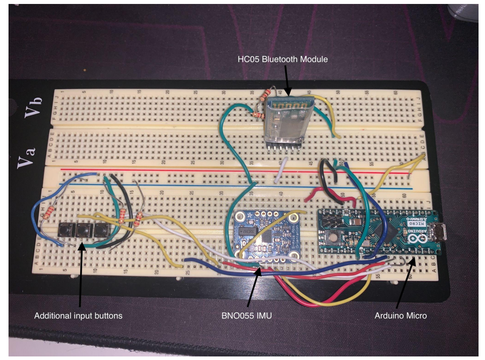
The Arduino Micro serves as the main microcontroller board of the pen that connects all of the Arduino sensors and components together. It was chosen for its small and manageable size and laptop mouse compatibility. The Arduino Micro is based on the ATMega32U4 board with built-in USB communication, allowing the Micro to appear to a connected computer as a mouse. The additional components we used in conjunction with the microcontroller are the BNO055 IMU as well as the HC-05 bluetooth module. The BNO055 was chosen for its excellent orientation/gyroscope data as well as its 100Hz polling rate. The HC-05 was chosen due to its size and ease to integrate with the rest of our components.

*Gesture Recognition*

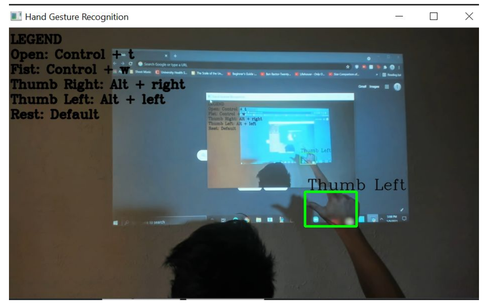
The gesture recognition software is used to map the user's webcam to specific macros. We use mediapipe, Keras, and Tensorflow to identify the hand's location and subsequently detect hand gestures. We use PyAutoGUI to map these hand gestures to user defined macros and shortcuts. Users can save new gestures with custom macros and the model is able to learn the new gesture and recognize it going forward. Users can also delete gestures and update the model to remove the gesture from the known movements. This gives the user the ability to customize their experience with Magic Mice.

*System*

The system will take the inputs from the gesture recognition (hand location and gesture) and pen software (click type and coordinates) and calculate the corresponding action. For gesture recognition, the system will need to map the hand gesture to a pre-saved Windows user macro or an error. For the pen output, our Python program will read in the IMU data using the PySerial library combined with the hand location, already known from the OpenCV hand gesture system. The system also recognizes clicks (both left and right). After this, the system will update the projector to update the display that the user sees. Orientation is calculated using a calibration button on the pen and the hand position location is calibrated using the camera.




*Wiring of Pen Components*




*Gesture Detection*

## System Evaluation

For our pen, we isolated the X,Y, and Z axis and drew a straight line to ensure that the mouse was moving correctly. Also, we isolated those axis for rotation as well to ensure that tip of the pen was positioned correctly. For the buttons, we tested single clicks, hold for drag, and hovering mouse movement as well. We found that the tip of the pen was around a 2 inch radius from the actual mouse, and the clicks were around 20ms long with a 50 Hz polling rate. For the hand expressions, we tested our gesture recognition algorithm by verifying our camera input maps to the correct hand expression. We found that the system was correct ~17 times out of 20 when recognizing new gestures. As for our projector, we wanted to work on distance ranges from 5 - 15 ft . 5 ft wouldn't allow for a big enough screen projection and 15 ft is slightly larger than the size of an average room. However, we found that 15 ft made it challenging to see the screen resolution clearly, so we had capped it at 10 ft.

**Metrics and Validation**

| Segment | Metric | Performance |
|---|---|---|
| Pen | ~1 inches when used with a projector<br>~50 ms button clicks response<br>~100 Hz polling rate | ~2 inches when used with a projector<br>~20 ms button clicks response<br>~50 Hz polling rate |
| Gesture Recognition | ~95% accuracy<br>Add/remove new gestures | ~85% accuracy<br>Saves up to 10 different gestures |
| System/Projector | Calibration for distances of 5-15 ft | Calibration for 5-10 ft |