

Developed by Crazy Thursday

- Team Name: Crazy Thursday
- Team Members: Zheyu Li, Shun Yao, Zhuozi Xie
- Github Repository URL: <https://github.com/upenn-embedded/final-project-crazy-thursday.git>
- Github Pages Website URL: <https://ssysy.github.io/crazythursday.github.io/>
- Description of hardware: ATmega328PB, ESP32 S2 Feather, 6-DOF IMU, LCD, LED, Buzzer

Introduction

Swingtronics is a **motion sensing game controller** that can be held in one hand. By detecting the player's movement as an input, it is converted into a digital signal and transmitted wirelessly to a portable processor. All devices are **portable** and **do not require WiFi**. Just have batteries ready to play in **indoor/outdoor environments**!

Minimum of just one game controller and one portable processor to play the game (Multiplayer/multi-input device is also supported). Currently, a baseball simulation game has been developed, which requires 2 players with 1 processor. More game (tree-cutting, sword battle etc.) is on the way!

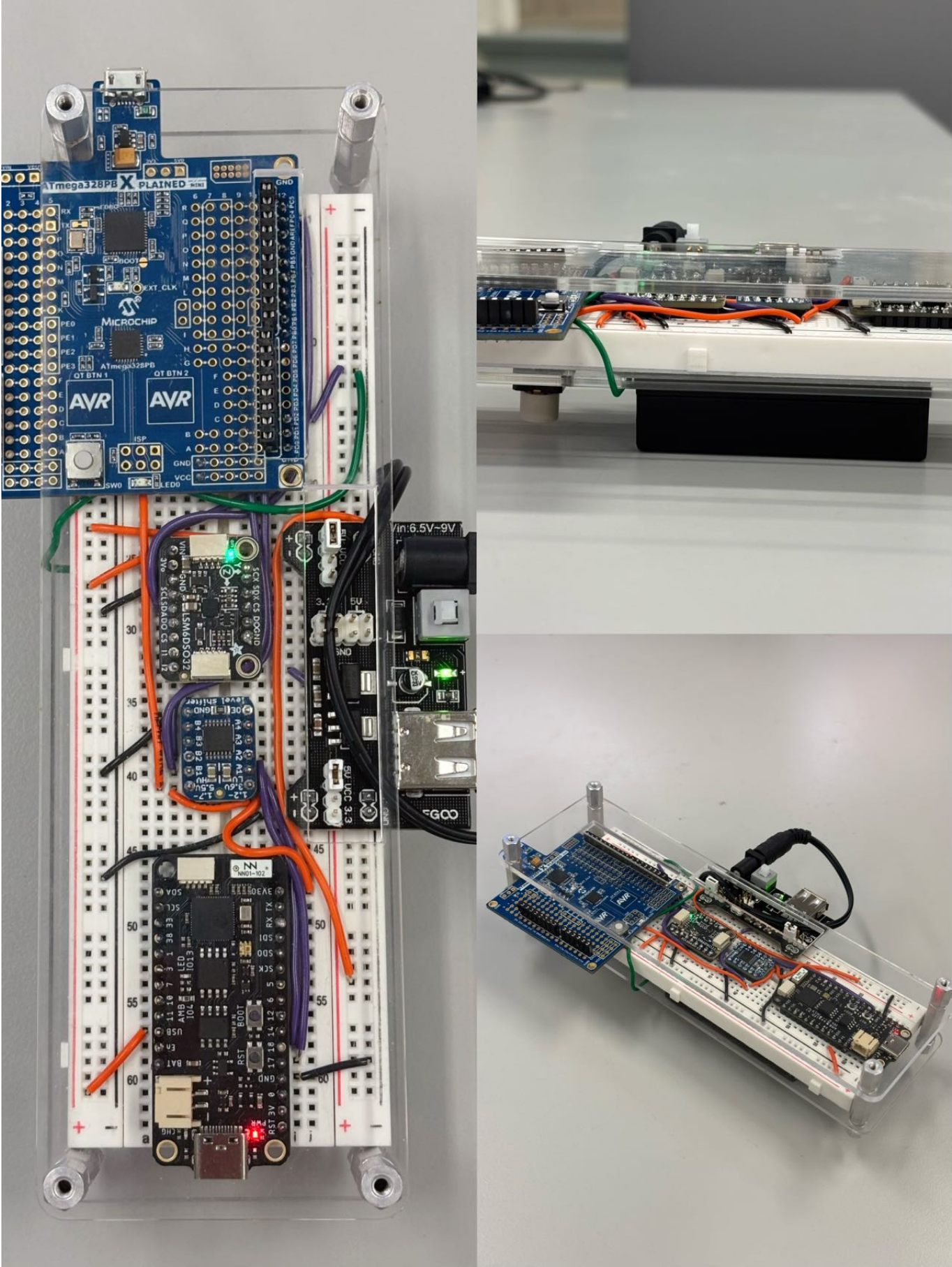
1. Video

[Click Here for Our Final Demo Video!](#)

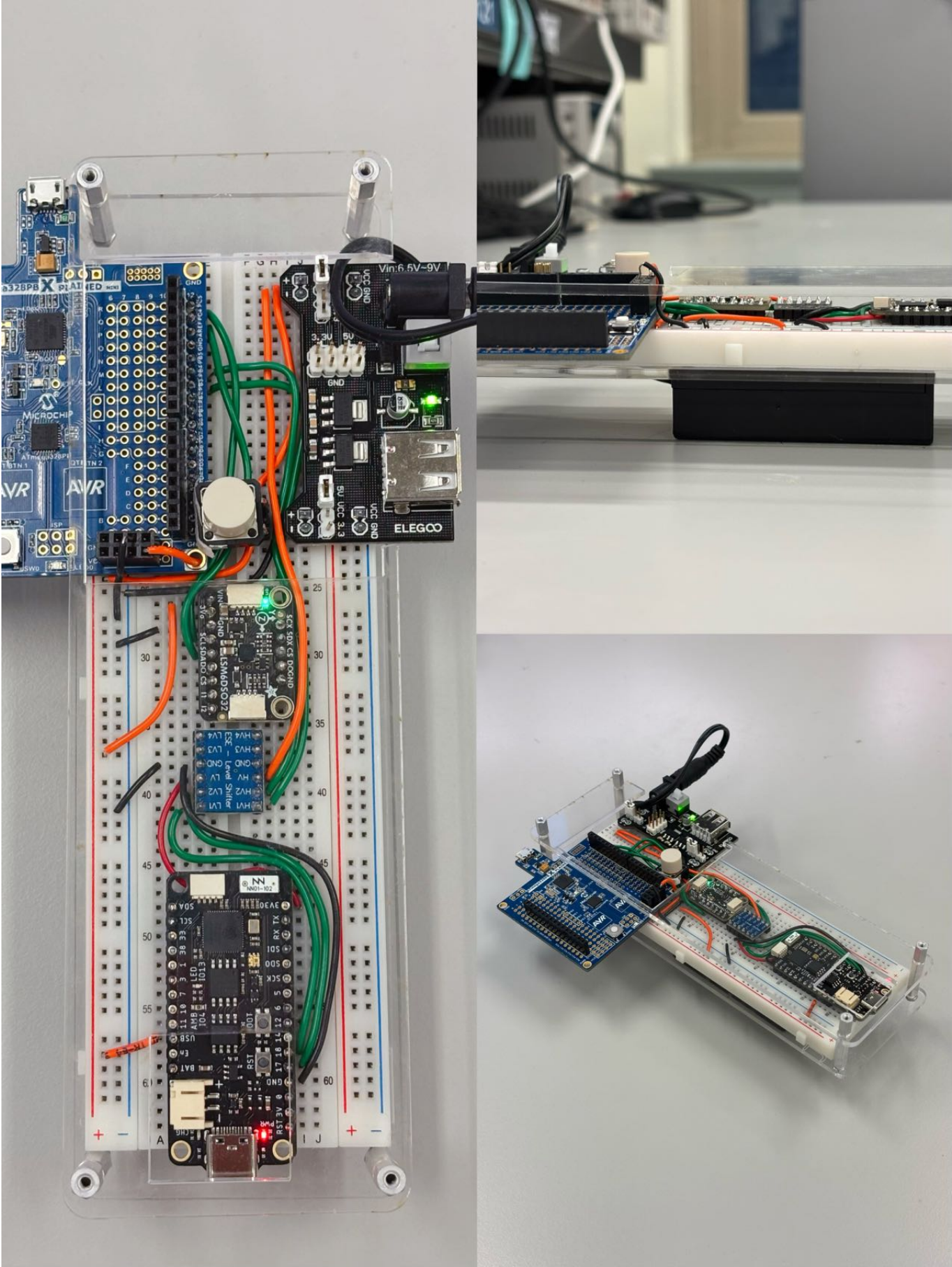
Here are two videos for our [Indoor Stability Test](#) and [Outdoor Stability Test](#).

2. Images

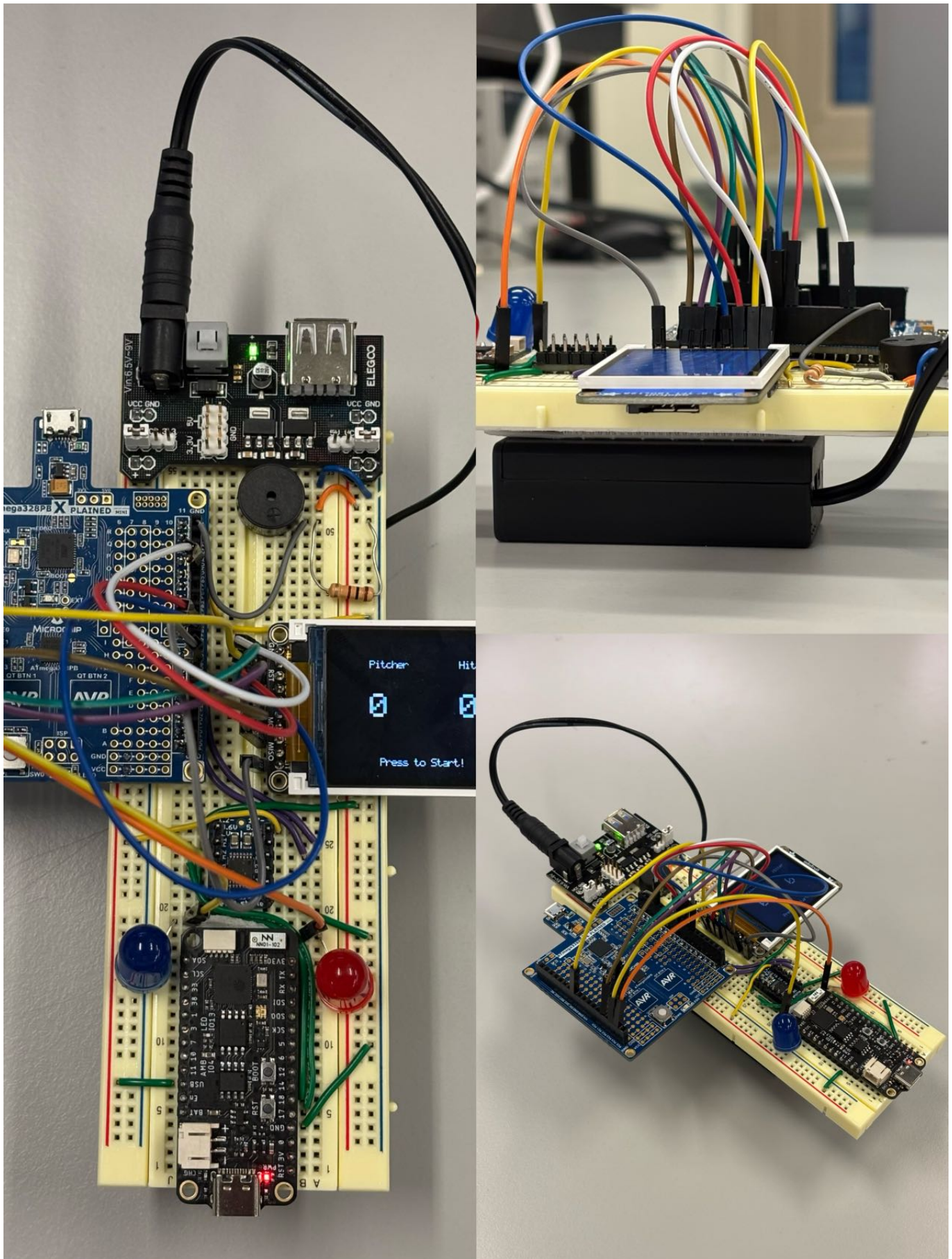
Game Controller 1 (Input Device/Pitcher)



Game Controller 2 (Input Device/Hit Bat)



Portable Central Processor (Output Device/Score Board)



3. Results

- General Purpose Devices: 2 motion-sensing game controller and 1 central processor, all devices are portable and no WiFi requirement.

- Available Game: A baseball simulation game, which requires 2 game controller (one as pitcher and another as hit bat) and 1 processor (perform as scoreboard), allowing 2 players to enjoy. The game can be played at low latency in both indoor and outdoor environments with a player distance within 10m.

4. Software Requirements Specification (SRS) Validation

- SRS 01 – Two microcontroller needs to be able to communicate with each other without external WiFi connection. Communication delay should be less than 10ms. In our project, ESP32 device communicates directly with another ESP32 by specifying its MAC address.

[Proof of Work Video](#)

- SRS 02 – Mutiple MCU devices required to be able to identify others, allowing one Master device able to communicate with mutiple slave divesces. This project use ESP32 MAC addresses to identify specific peers, allowing a single device to receive data from multiple others.

[Proof of Work Video](#)

- SRS 03 - The Master MCU device needs to be able to generate a appropriate animation or score board on an LCD or equivlent display based on the signal send from slave devices.

[Proof of Work Video](#)

- SRS 04 - With the acceleration on X and Y axis of IMU, one calculation algorithm that can calculate the trajectory and timing slot between the Pither and HitBall needs to be programmed.

ESP_NOW_Broadcast_Master.ino esp_now.h cJSON

```

185         cmd |= (1<<7);
186         if (boardsStruct[0].vel > 1.2){
187             cmd |= (1<<1); //double score
188         }
189     }
190 }
191
192 // Calculate and print the time interval between the two boards (if both are received)
193 if (boardsStruct[0].Time != 0 && boardsStruct[1].Time != 0) {
194
195     cmd |= (1<<7);
196     Serial.println();
197     float T = (GAME_DISTANCE/boardsStruct[0].vel);
198     Serial.printf("T_calc: %.3f \n", T);
199     float timeInterval = boardsStruct[1].Time - boardsStruct[0].Time;
200     if(timeInterval<0){
201         timeInterval += 60;
202     }
203     Serial.printf("T_actual: %.3f \n", timeInterval);
204
205     // Catch detection & Score
206     if ((float)T - timeInterval > 1.0) {

```

Output Serial Monitor x

Message (Enter to send message to 'Adafruit Feather ESP32-S2' on '/dev/cu.usbmodem2101')

Ball thrown!

Velocity: 0.747
Acceleration: 0
Ball released Time: 17.191

Velocity: 0.372
Acceleration: 1
Hit Bat Swiped Time: 18.598

T_calc: 2.008
T_actual: 1.407
Hitter win!
Passing command:10000011

- SRS 05 - The timing and predicted trajectory are used to determine which player win points, also the gaming logic will be programmed within one project.

[Proof of Work Video](#)

5. Hardware Requirements Specification (HRS) Validation

- HRS 01 – Microcontroller with enough communication ports to establish at least three communication channels. This project is based on ATmega328PB microcontroller coded with baremetal C.

[Please See File:"GameController_ATmega.X" for Proof](#)

- HRS 02 - Microcontroller with WiFi communication ability or equivalent. Here wireless communication parts will be handled by ESP32 S2 Feather through ESP-NOW protocol.

[Proof of Work Video](#)

- HRS 03 – A 6-DOF orientation IMU shall be used for movement detection. The sensor shall measure acceleration and angular velocity on the Pitcher Glove and the Hit Bat, with a detection range of at least -4g to 4g and a resolution of 2^{16} steps.

```

COM5 - PuTTY
AX: 244, AY: 1511, AZ: 8433 | GX: -230, GY: -1220, GZ: 5734 | Temp: 26
AX: -209, AY: 968, AZ: 8317 | GX: -4868, GY: -37, GZ: 15175 | Temp: 26
AX: 192, AY: 293, AZ: 8454 | GX: -12, GY: 653, GZ: 14423 | Temp: 26
AX: 523, AY: 1183, AZ: 9203 | GX: -168, GY: -4280, GZ: -32764 | Temp: 26
AX: 356, AY: 1080, AZ: 8677 | GX: 14047, GY: 2387, GZ: 11149 | Temp: 26
AX: -183, AY: 251, AZ: 8484 | GX: -3921, GY: 437, GZ: 27997 | Temp: 26
AX: -1197, AY: 1836, AZ: 8488 | GX: 13166, GY: -7531, GZ: -32764 | Temp: 26
AX: 1039, AY: 1592, AZ: 8499 | GX: -13374, GY: 392, GZ: 21547 | Temp: 26
AX: -318, AY: 745, AZ: 7955 | GX: 17350, GY: 1242, GZ: -28532 | Temp: 26
AX: 1272, AY: 2047, AZ: 8879 | GX: -7964, GY: -1717, GZ: 15572 | Temp: 26
AX: -169, AY: -196, AZ: 7944 | GX: 4829, GY: 1061, GZ: -3004 | Temp: 26
AX: 181, AY: 1081, AZ: 8297 | GX: 14981, GY: 1029, GZ: 760 | Temp: 26
AX: 809, AY: 2079, AZ: 7874 | GX: 3655, GY: -3672, GZ: 1256 | Temp: 26
AX: 694, AY: 1500, AZ: 8094 | GX: -5579, GY: 2007, GZ: -8574 | Temp: 26
AX: 2673, AY: -1731, AZ: 7729 | GX: -142, GY: -11818, GZ: 5758 | Temp: 26
AX: 3748, AY: 1511, AZ: 7360 | GX: 462, GY: 1814, GZ: 2981 | Temp: 26
AX: 3291, AY: 764, AZ: 7100 | GX: 2143, GY: 10589, GZ: 5743 | Temp: 26
AX: 910, AY: 1206, AZ: 7914 | GX: 8360, GY: 12385, GZ: -8258 | Temp: 26
AX: 375, AY: 1388, AZ: 10177 | GX: -9805, GY: 1709, GZ: -1790 | Temp: 26
AX: -92, AY: 622, AZ: 3644 | GX: -1925, GY: -12871, GZ: 4228 | Temp: 26
AX: -1794, AY: -1099, AZ: 10882 | GX: -10088, GY: 11343, GZ: 1169 | Temp: 26
AX: 989, AY: -903, AZ: -2726 | GX: -2219, GY: -6808, GZ: -2100 | Temp: 26
AX: -1904, AY: 10680, AZ: 24542 | GX: 23874, GY: -9412, GZ: 8903 | Temp: 26
AX: 3334, AY: -953, AZ: 16908 | GX: 7901, GY: 9812, GZ: 8324 | Temp: 26
AX: 912, AY: -2948, AZ: -6036 | GX: -25153, GY: 2089, GZ: 350 | Temp: 26
AX: -1816, AY: -3582, AZ: -7417 | GX: -14700, GY: -3922, GZ: -7719 | Temp: 26
AX: -2112, AY: -2790, AZ: 1481 | GX: -2710, GY: -10818, GZ: -8723 | Temp: 26
AX: -126, AY: 2543, AZ: 8382 | GX: -6176, GY: -10096, GZ: -1218 | Temp: 26
AX: 3553, AY: 1806, AZ: 13681 | GX: 1600, GY: -6960, GZ: 2936 | Temp: 26
AX: 2615, AY: -148, AZ: 9375 | GX: -2968, GY: 4842, GZ: -5551 | Temp: 26
AX: -1365, AY: 938, AZ: 7326 | GX: 3893, GY: -5640, GZ: -4988 | Temp: 26
AX: -1665, AY: 5218, AZ: 8089 | GX: 4116, GY: -13078, GZ: -7999 | Temp: 26
AX: -3369, AY: 2816, AZ: 11418 | GX: -12819, GY: -25559, GZ: -22356 | Temp: 26
AX: 5913, AY: 2074, AZ: 7326 | GX: 2095, GY: 1995, GZ: -12598 | Temp: 26
AX: 3736, AY: 4968, AZ: 8367 | GX: 15913, GY: -16550, GZ: -2175 | Temp: 26
AX: 2770, AY: 5327, AZ: 4736 | GX: 13497, GY: -15386, GZ: 12836 | Temp: 26
AX: 12034, AY: 4993, AZ: 8319 | GX: -15508, GY: -13427, GZ: -13531 | Temp: 26
AX: 11090, AY: 1661, AZ: 4647 | GX: 18378, GY: 12869, GZ: 32764 | Temp: 26
AX: 7488, AY: 615, AZ: 3775 | GX: 5724, GY: -526, GZ: 25474 | Temp: 26
AX: 3760, AY: 2219, AZ: 6715 | GX: 15852, GY: 25982, GZ: -10944 | Temp: 26
AX: 1700, AY: 7288, AZ: 2718 | GX: 28817, GY: 30660, GZ: -10903 | Temp: 26
AX: -2831, AY: 5796, AZ: -1984 | GX: 27742, GY: 32764, GZ: -22110 | Temp: 26
AX: -5551, AY: 7742, AZ: -4988 | GX: 102, GY: 423, GZ: -19674 | Temp: 26
AX: -2946, AY: 7178, AZ: -2926 | GX: -870, GY: -1592, GZ: 1529 | Temp: 26
AX: -3665, AY: 6011, AZ: -2941 | GX: 3292, GY: 3206, GZ: -1941 | Temp: 26
AX: -3839, AY: 6653, AZ: -2988 | GX: 62, GY: 312, GZ: -1045 | Temp: 26
AX: -3674, AY: 6574, AZ: -3045 | GX: 51, GY: 9, GZ: -248 | Temp: 26
AX: -3711, AY: 6504, AZ: -3030 | GX: 88, GY: 19, GZ: 149 | Temp: 26
AX: -3737, AY: 6497, AZ: -3034 | GX: 78, GY: 26, GZ: 2 | Temp: 26
AX: -3699, AY: 6540, AZ: -3070 | GX: 66, GY: 39, GZ: 23 | Temp: 26
AX: -3732, AY: 6511, AZ: -3036 | GX: 85, GY: 13, GZ: 30 | Temp: 26
AX: -3709, AY: 6529, AZ: -3053 | GX: 279, GY: 57, GZ: 237 | Temp: 26
AX: -3739, AY: 6513, AZ: -3037 | GX: 87, GY: 56, GZ: 57 | Temp: 26
AX: -3721, AY: 6527, AZ: -2835 | GX: 53, GY: 15, GZ: 3 | Temp: 26
AX: -3722, AY: 6508, AZ: -3053 | GX: 65, GY: 25, GZ: -25 | Temp: 26
AX: -3705, AY: 6532, AZ: -3319 | GX: 58, GY: 14, GZ: 6 | Temp: 26
AX: -3736, AY: 6491, AZ: -2838 | GX: 48, GY: 4, GZ: 23 | Temp: 26
AX: -3724, AY: 6543, AZ: -3073 | GX: 69, GY: 1, GZ: 23 | Temp: 26
  
```

- HRS 04 – An LCD used for user interface. The display shall be able to communicate with the microcontroller via UART, I2C or equivalent communication protocol.

[Proof of Work Video](#)

- HRS 05 - A power supply that can provide rated power for the used microcontrollers, in this case: 3.3V for ESP32 S2 Feather and 5V for ATmega328PB. (Since we are using ATmega and ESP32 that have different logic voltage levels, an additional level shifter is needed for communication between the two)

[Please See File:"datasheet" for Proof](#)

- HRS 06 - One push button (momentary switch) shall be used at the Pitcher Glove to detect whether the 'ball' has been thrown out or not.

[Please See File:"datasheet" for Proof](#)

- HRS 07 - Actuators: LED, Buzzer, etc. could be added to any socre board devices as player interfaces. This project used two LEDs and one buzzer on socre board.

[Proof of Work Video](#)

6. Conclusion

What we learned

- Based on chip's datasheet to write bare metal C
- Application of Timer, interrupt, GPIO on AVR (ATmega328PB)
- Learn how to write drivers for I2C, UART, SPI on AVR chips
- ESP-NOW configuration on ESP32 using Arduino

Encountered Issues

- High Complexity of calculating velocity and angle (Kalman filtering and quaternions can be used to improve detection accuracy)
- Short Battery life (Final demo: Playable for only 10-15 mins per battery cycle)
- Conflicts caused by multiplexing of communication protocol pin (UART1 and SPI0, we therefore change to SPI1)
- Integration of different parts could be difficult (3 devices, 4 different kind of communication protocols)
- Game logics can be complicated and cause some bugs.

Next steps

- More Reliable, Longer Lasting, Rechargeable Power Supply Solutions
- Better visualization (Larger LCD etc.)
- More Robust Hardware Connection (PCB, More fitted cases)
- More Available Games