

Voice Activated Scorekeeper

Team 31 : Jason Hwang, Christopher Kalebich, Allan Zou

Overview

- Problem
- Solution
- Design
- Challenges
- Final Product
- Future Work

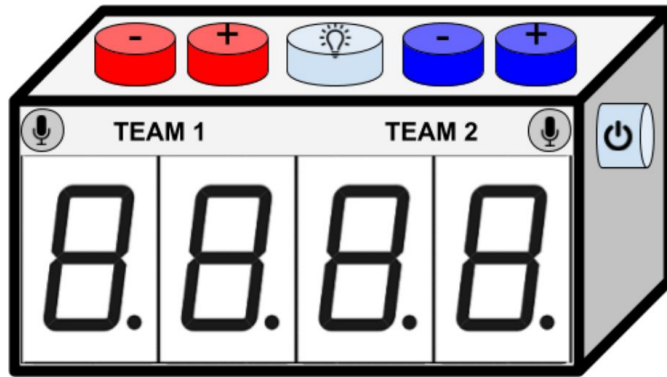
The Problem

Scorekeeping is a hassle and
disrupts the game.

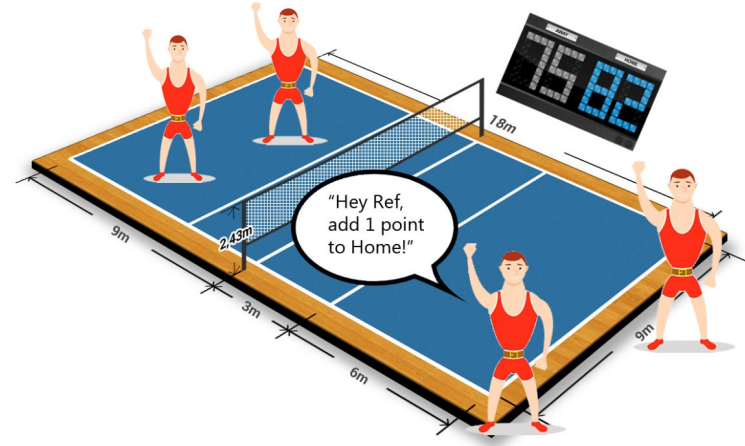
The Solution

A portable scoreboard controlled
using speech recognition.

Solution



Scorekeeper Concept



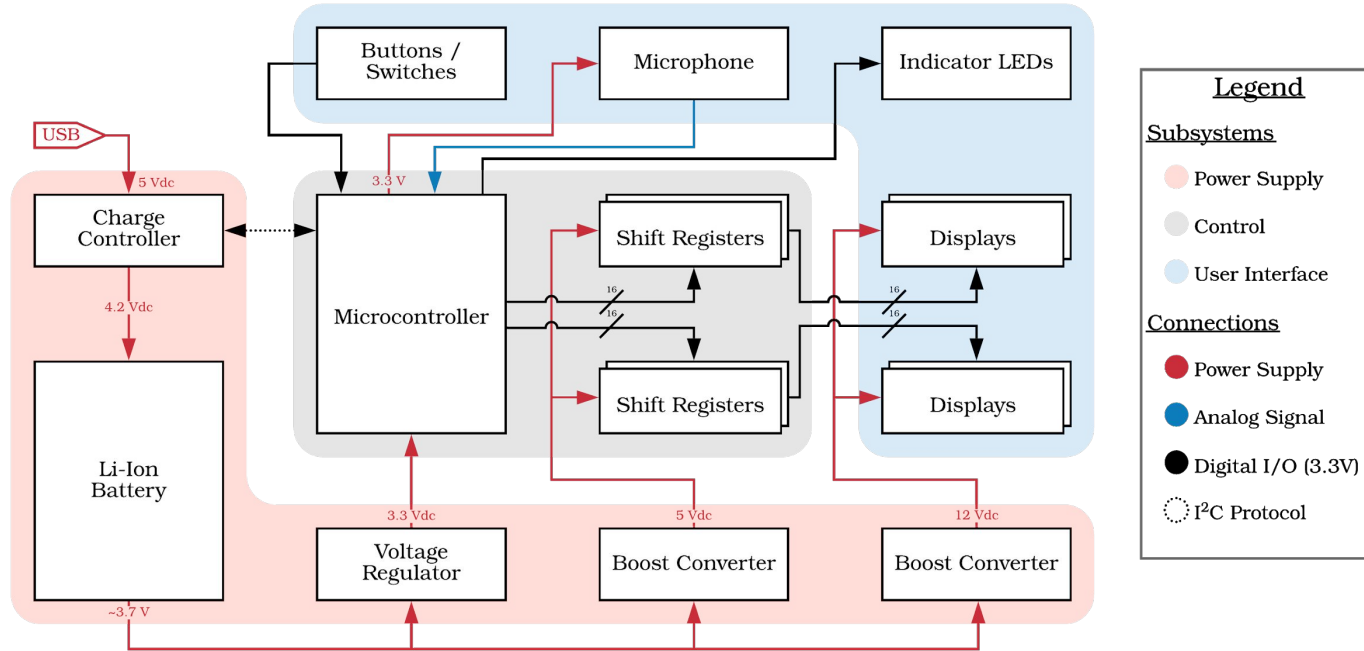
Interaction Concept

The Design

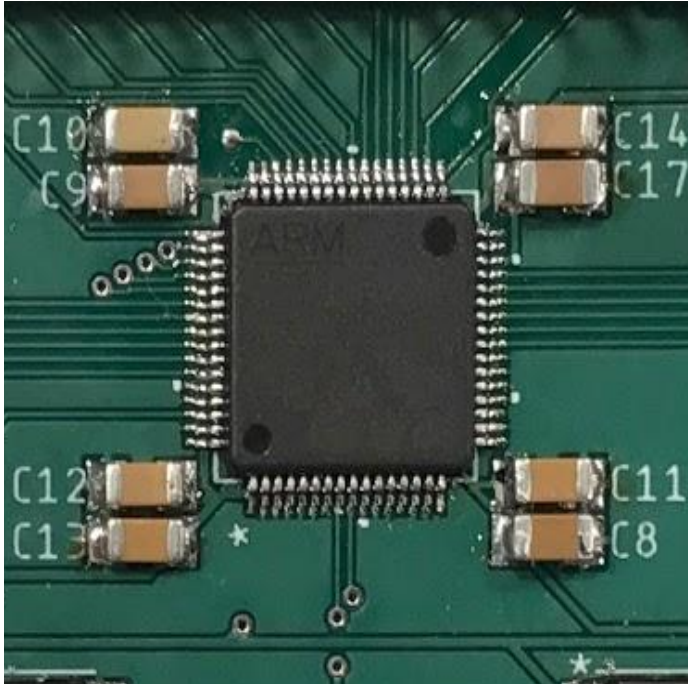
High Level Requirements

1. Change score using keywords with 85% accuracy
2. Pickup audio commands ≤ 10 ft from user with $75 \pm 5\%$ dB background noise
3. Rechargeable battery lasting ≥ 4 hrs

Block Diagram



Control Subsystem



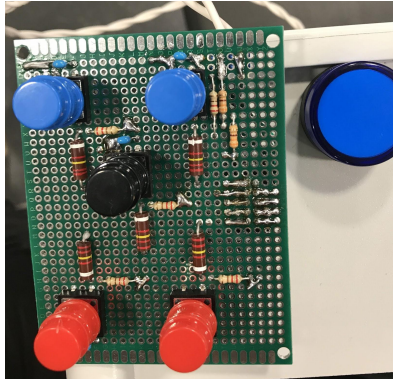
STM32F4 Microcontroller

Clock Frequency
84 MHz

Memory
256 kB

RAM
64 kB

User Interface Subsystem

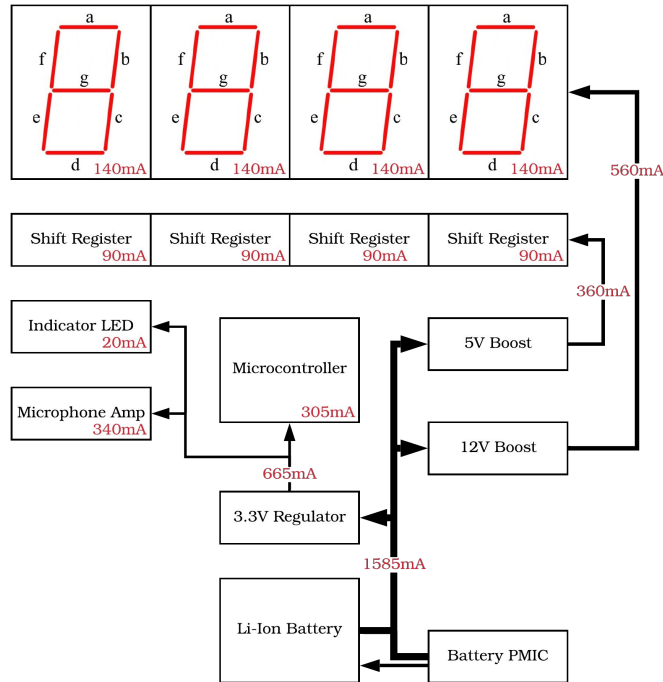


Main Components

- Microphone
- 7 Segment Displays
- Indicator LEDs
- Buttons/Switches

Button Board & 7 Seg. Display

Power Subsystem

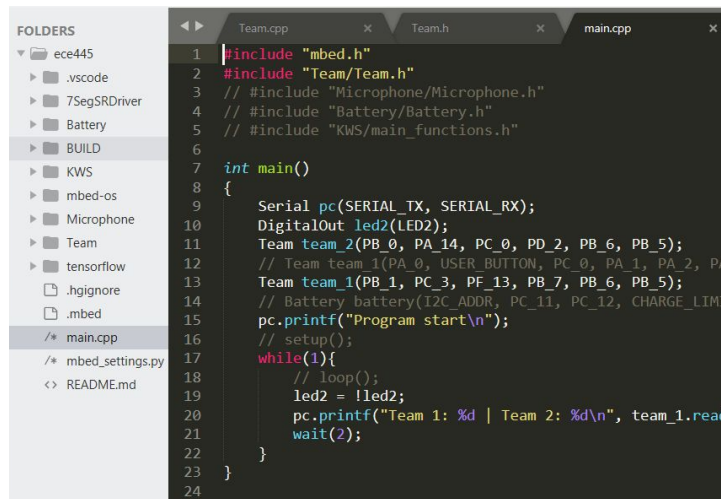


(Figure of Current Accounting)

Main Components

- 4.2V Charge Controller
- 3.7V - 6600mAh Li-ion Battery
- 3.3V Voltage Regulator
- 5.0V Boost Converter
- 12.0V Boost Converter

Software Subsystem



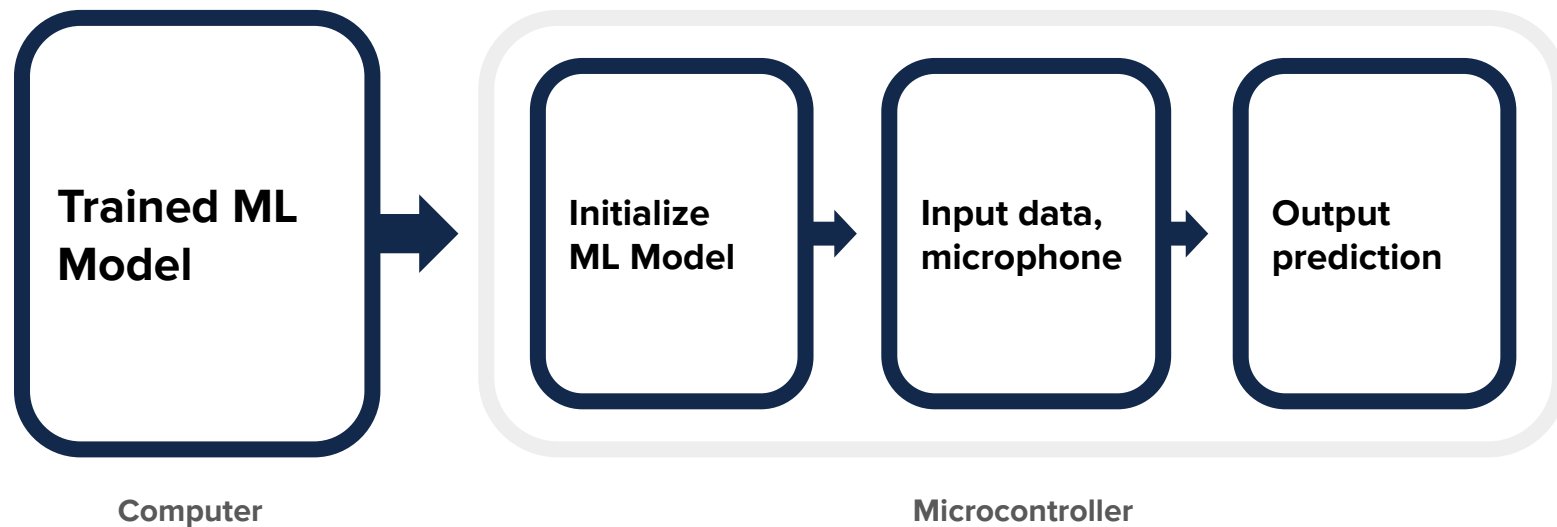
The screenshot shows a code editor with a project structure on the left and C++ code in the main window. The project structure includes folders like .vscode, 7SegSRDriver, Battery, BUILD, KWS, mbed-os, Microphone, Team, tensorflow, .hgignore, .mbed, main.cpp, mbed_settings.py, and README.md. The main window displays the code for Team.cpp, Team.h, and main.cpp. The code includes headers for mbed.h, Team/Team.h, Microphone/Microphone.h, Battery/Battery.h, and KWS/main_functions.h. The main function initializes a serial port, a LED, and two Team objects, then enters a loop that prints team scores and waits for 2 seconds.

```
1 #include "mbed.h"
2 #include "Team/Team.h"
3 // #include "Microphone/Microphone.h"
4 // #include "Battery/Battery.h"
5 // #include "KWS/main_functions.h"
6
7 int main()
8 {
9     Serial pc(SERIAL_TX, SERIAL_RX);
10    DigitalOut led2(LED2);
11    Team team_2(PB_0, PA_14, PC_0, PD_2, PB_6, PB_5);
12    // Team team_1(PA_0, USER_BUTTON, PC_0, PA_1, PA_2, PA_3);
13    Team team_1(PB_1, PC_3, PF_13, PB_7, PB_6, PB_5);
14    // Battery battery(I2C_ADDR, PC_11, PC_12, CHARGE_LIMIT);
15    pc.printf("Program start\n");
16    // setup();
17    while(1){
18        // loop();
19        led2 = !led2;
20        pc.printf("Team 1: %d | Team 2: %d\n", team_1.read(), team_2.read());
21        wait(2);
22    }
23 }
24
```

Main Components

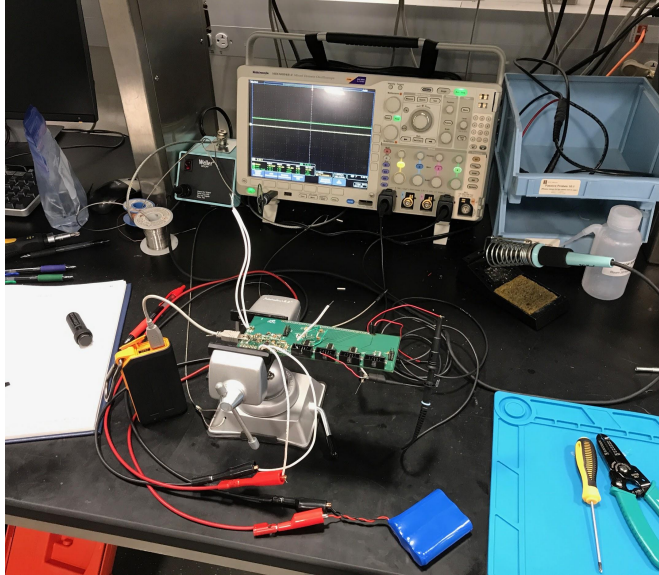
- Tensorflow Lite Keyword Spotting
- Keyword/score state control
- Software/Hardware integration

Code Repository



The Challenges

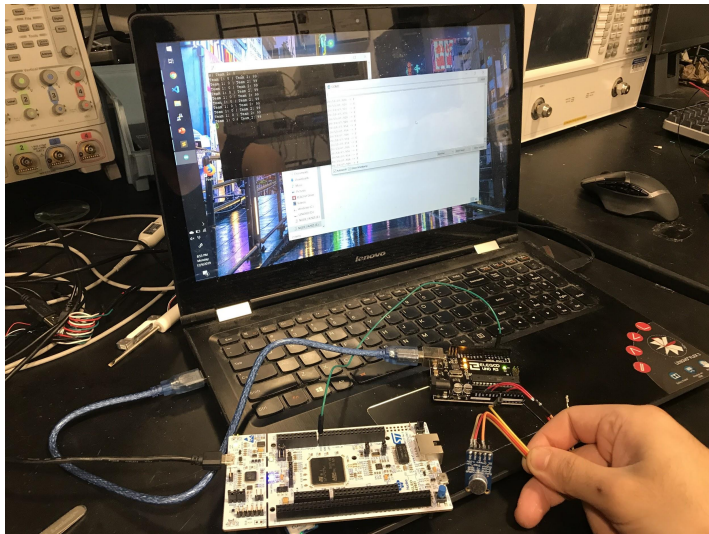
PCB Design and Assembly



Verification and Debugging

- 3.3V regulators
- Datasheet oversights
- Pacing and design flow

Keyword Spotting



KWS Subsystem

- Tensorflow incorrect predictions
- uSpeech library low accuracy

Phoneme	Literal
e	The 'e' sound.
h	The '/sh/' sound. It can also be raised by 'ch', 'j' and 'z'
v	The 'v' sound, occasionally triggered by 'z' (may need to be fixed)
f	The 'f' sound.
s	The 's' sound.
o	'a', 'o', 'i', 'r', 'l', 'm', 'n' and 'u' sounds.
' '	Too Quiet for anything

μSpeech

System Integration

```
class Team {  
public:  
    Team(  
        PinName inc_pin,  
        PinName dec_pin,  
        PinName voice_pin,  
        PinName srData,  
        PinName srClock,  
        PinName srLatch  
    );  
  
    void increment_score();  
    void decrement_score();  
    void reset_score();  
    int read_score();  
  
private:  
    InterruptIn _inc_button;  
    InterruptIn _dec_button;  
    volatile int _score;  
    // SSegSRDriver _display_tens;  
    // SSegSRDriver _display_ones;  
    InterruptIn _voice;  
    SSegSRDriver _display;  
  
    void update_display();  
    void split_score_digits(int score, uint8_t* digits);  
};
```

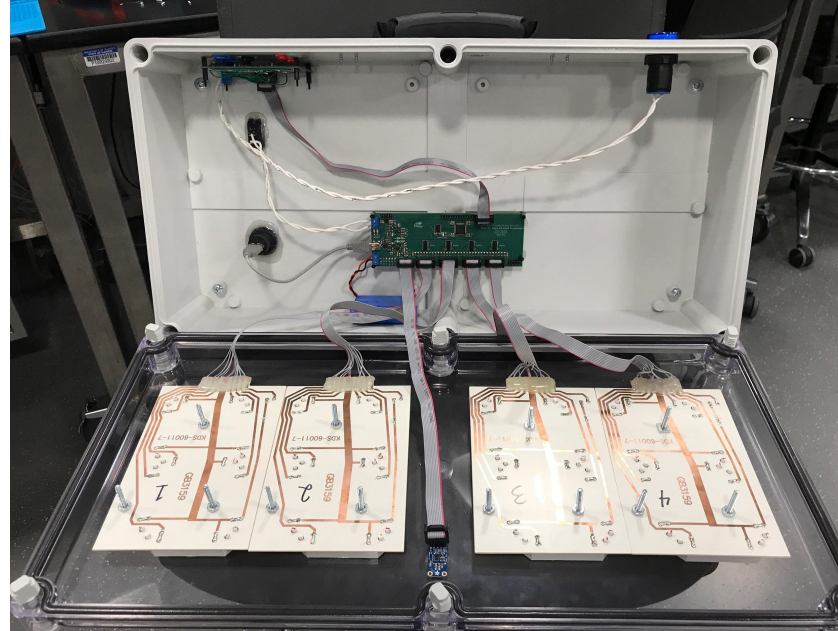
- Tracking and displaying scores
- Importing tensorflow

Scorekeeping Program

The Final Product

The Final Product

- Battery Charging
- Voltage Regulation



- KWS Speech Recognition
- Team scorekeeping
- Integration between various subsystems

Future Work

Improvement Ideas



Alternative KWS System



**Signal Processing on
Microphone**



Additional Features (Timer, etc)

The Team

The Team



Jason Hwang

Computer Engineering
Software Subsystem



Christopher Kalebich

Electrical Engineering
Hardware Subsystem



Allan Zou

Computer Engineering
Hardware-Software Subsystems

Thank you!