**Project: Scoreboard Controller System**

This repository provides the code and documentation for controlling a scoreboard and shot clock system using multiple microcontrollers, including Arduino Mega 2560, NodeMCU 8266, and ESP8266-based boards. The system allows for real-time control and synchronization of the scoreboard and shot clock via wireless communication.

**Table of Contents**

1. [Overview](#overview)
2. [Hardware Setup](#hardware-setup)
3. [Programming Method](#programming-method)
4. [Programming Steps](#programming-steps)
5. [Connections and Pinouts](#connections-and-pinouts)
6. [DIP Switch Settings](#dip-switch-settings)
7. [Troubleshooting](#troubleshooting)
8. [License](#license)

**Overview**

The project consists of multiple components that communicate with each other over Wi-Fi and serial communication:

* **Remote Controller Board:** Arduino Mega 2560 + ESP8266
* **Wi-Fi Module for Remote Controller:** NodeMCU 8266
* **Shotclock Board:** NodeMCU 8266 for controlling the shot clock

**Key Notes:**

* **USE LEVEL SHIFTER WHEN CONNECTING MEGA+WIFI TO NODEMCU8266.**
* The ESP8266 on the Mega+WiFi setup is **disabled** and not used due to performance issues.
* The system includes firmware for the different components:
  + **Scoreboard\_Controller\_FW:** Runs on the Arduino Mega 2560.
  + **Scoreboard\_Controller\_WiFi\_FW:** Runs on the NodeMCU 8266 for the Wi-Fi module.
  + **Shotclock\_FW:** Runs on the NodeMCU 8266 for the shot clock.

**Hardware Setup**

**Components:**

* **Arduino Mega 2560:** Acts as the main controller for the scoreboard.
* **NodeMCU 8266 (Wi-Fi Module):** Communicates with the Mega+WiFi board via UART.
* **NodeMCU 8266 (Shotclock Board):** Controls the shot clock functionality.
* **Level Shifter:** Converts voltage between Arduino Mega (5V) and NodeMCU 8266 (3.3V) for UART communication.

**Required Components:**

* Arduino Mega 2560
* NodeMCU 8266 (for Wi-Fi communication)
* NodeMCU 8266 (for shot clock control)
* Level shifter (3.3V to 5V for serial communication)

**Programming Method**

Follow these steps to flash the correct firmware onto each board:

1. **Clear the Mega+WiFi ESP8266 Code:** Flash the ESP8266 module with blank Arduino code.
2. **Flash the Mega+WiFi Controller Firmware:** Flash the Arduino Mega 2560 with the Scoreboard\_Controller\_FW firmware.
3. **Flash the Wi-Fi Module for Mega+WiFi:** Flash the NodeMCU 8266 for the Mega+WiFi board with the Scoreboard\_Controller\_WiFi\_FW.
4. **Flash the Shotclock Firmware:** Flash the NodeMCU 8266 shot clock with the Shotclock\_FW.

**Programming Steps**

**Step 1: Programming the Mega+WiFi ESP8266**

1. **Set DIP Switch to 00001110:** Configures the ESP8266 on the Mega board to programming mode.
2. **Flash the ESP8266:** Using Arduino IDE, upload blank Arduino code to the ESP8266 on the Mega to clear its memory.
3. **Unplug Power:** Disconnect the power from the Mega+WiFi board.

**Step 2: Flash the Arduino Mega 2560**

1. **Set DIP Switch to 00110000:** Configures the Mega board to load the scoreboard controller firmware.
2. **Upload the Firmware:** Using Arduino IDE, upload the Scoreboard\_Controller\_FW firmware to the Mega 2560.
3. **Unplug Power:** Disconnect the power from the Mega+WiFi board.

**Step 3: Flash the Wi-Fi Module (NodeMCU 8266 for Mega+WiFi)**

1. **Flash the Wi-Fi Firmware:** Using Arduino IDE, upload the Scoreboard\_Controller\_WiFi\_FW firmware to the NodeMCU 8266 Wi-Fi module.
2. **Unplug Power:** Disconnect the power after flashing the firmware.

**Step 4: Flash the Shotclock Board**

1. **Upload Shotclock Firmware:** Using Arduino IDE, upload the Shotclock\_FW firmware to the NodeMCU 8266 controlling the shot clock.

**Connections and Pinouts**

**Main Board (Mega+WiFi):**

* Set the DIP switch to 00110000 to configure the Mega for scoreboard control.
* Set serial connection switch to RXD0 and TXD0 to ensure no collision with RXD3 and TXD3

**Wi-Fi Module (NodeMCU 8266 for Mega+WiFi):**

* **RX/TX Communication:** The Mega's Serial3 (TX/RX – PIN 14 & 15) must be connected to the RX/TX pins on the NodeMCU 8266.
  + Use a level shifter for proper voltage conversion between the 5V logic of the Mega and the 3.3V logic of the NodeMCU.
  + **Connections:** RX > TX and TX > RX.

**Shotclock Board (NodeMCU 8266):**

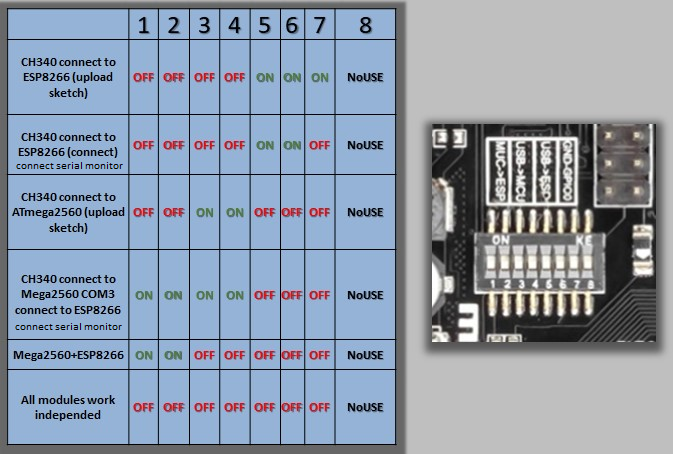
* The shot clock board is pre-configured with its firmware (Shotclock\_FW) and works independently.

**DIP Switch Settings**

Ensure the following DIP switch settings are correctly configured for each step:

1. **For Clearing Mega+WiFi ESP8266:**
   * Set DIP switch to 00001110.
2. **For Flashing the Mega+WiFi 2560 with the Scoreboard Controller Firmware:**
   * Set DIP switch to 00110000.
   * Set Serial switch to RX0/TX0

Refer to the DIP switch diagram for further details.



**Troubleshooting**

**Issue: Communication Errors**

* **Solution:** Ensure the serial communication lines between the Mega 2560 and the NodeMCU are connected properly, and that level shifters are used to protect the 3.3V NodeMCU from the 5V Mega.

**Issue: Firmware Not Uploading**

* **Solution:** Check the DIP switch settings and ensure that the correct board is selected in the Arduino IDE. Verify that the correct port is being used for uploading.

**Issue: No Wi-Fi Connection**

* **Solution:** Double-check the Wi-Fi credentials in the Scoreboard\_Controller\_WiFi\_FW. Ensure the NodeMCU 8266 is correctly configured.

**License**

This project is licensed under the MIT License. See the LICENSE file for details.