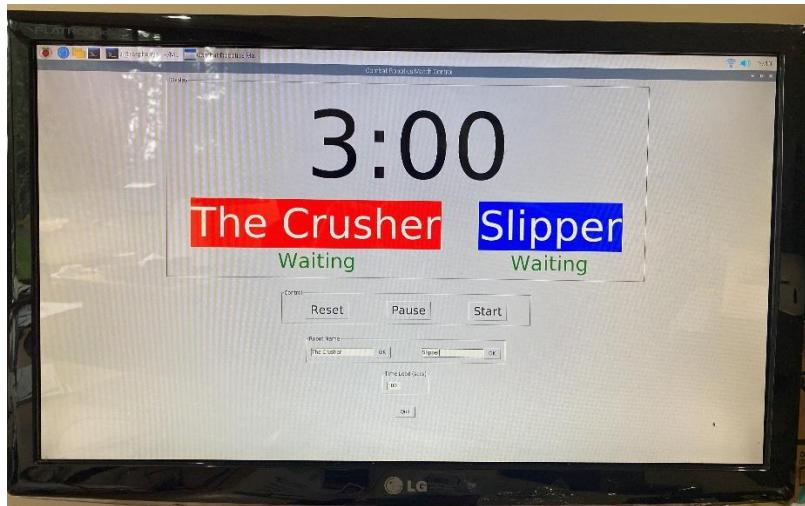


Match Manager

Overview

Match Manager provides a system to manage combat robot fights. It consists of an Administrator Console and a Display to allow the competitors and audience to see the status of the match. It provides Tap-In and Tap-Out buttons for the competitors and an optional trap door management system.

Administrator Console



Display Manager

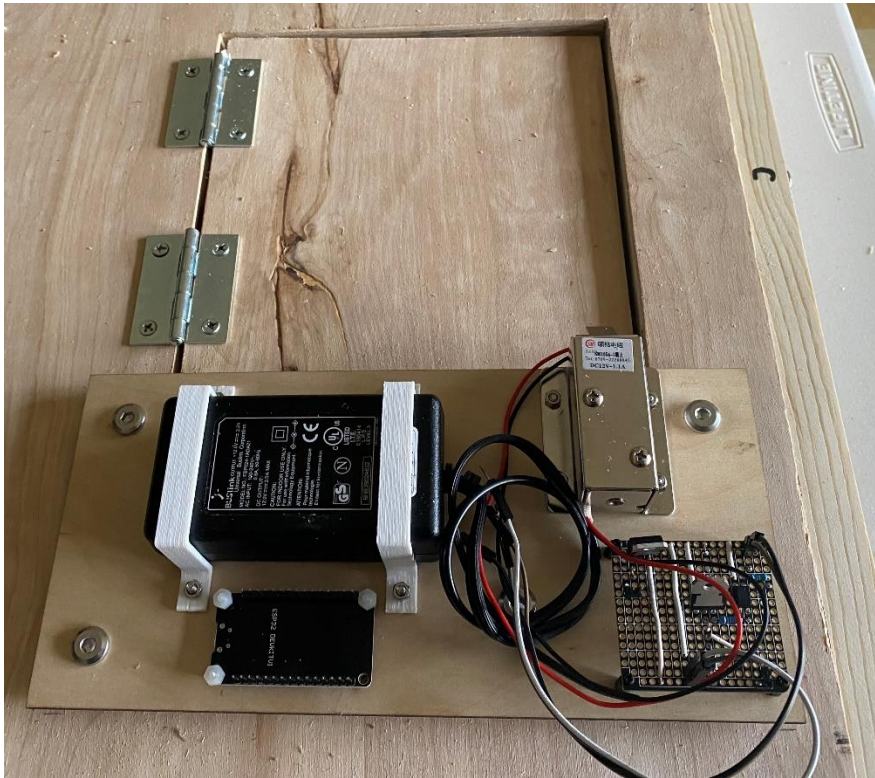


Match Manager

Tap-In/Out Buttons

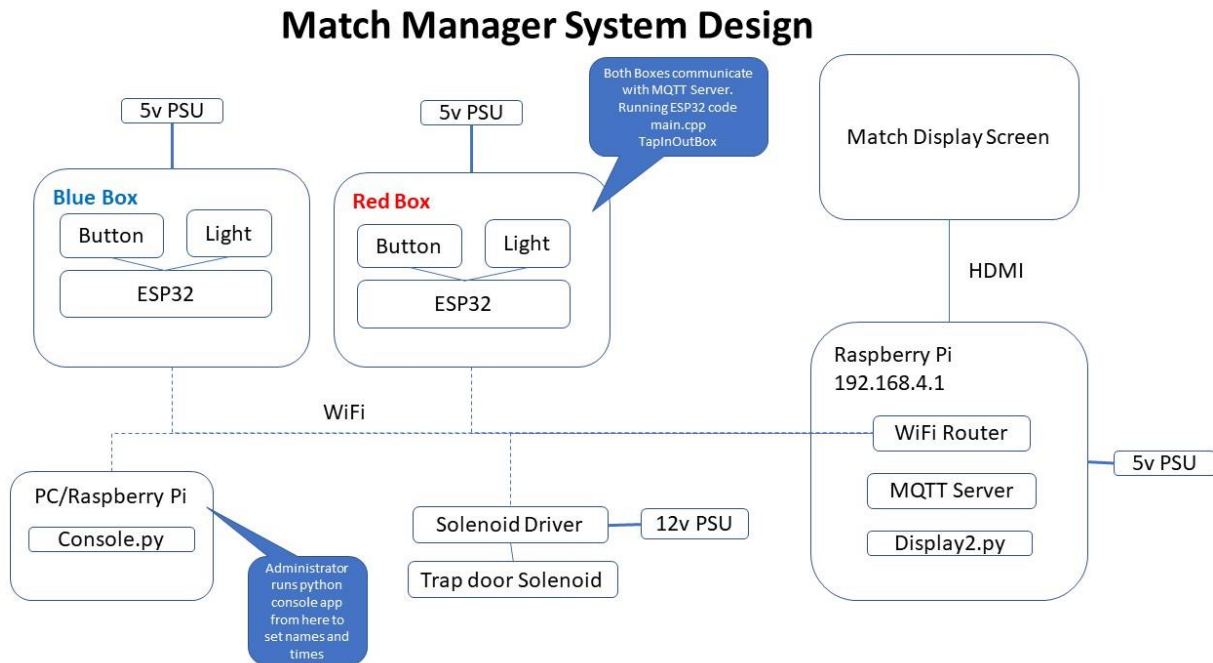


Trap Door Solenoid



Match Manager

Architecture



The heart of the system is the Display Manager running on a Raspberry Pi which is connected to the competition display but this system is also where the competition state machine is stored and it is also the source of control for Tap-In/Out lights and the for the solenoid that operates the trap door. The Display Manager Raspberry Pi also acts as a wireless access point to allow communication between all system components.

The Display Manager is configured to operate as a wireless access point that will route traffic to and from the eth0 port to the wireless network. Note that Match Manager will work even if the eth0 is not connected to the internet. This wireless access point provides DHCP addresses to components in the range 192.168.4.1 to 192.168.4.40. Communication from the Administrator Console, Tap-In/Out Buttons and Trap Door is by MQTT. The Display Manager runs the Mosquitto MQTT server to which all components publish/subscribe. The Display Manager software is written in Python and uses tkinter to manage the display elements

The Administrator Console is written in Python and can be run on either a PC or a Raspberry Pi. It uses tkinter to manage the display elements.

The Tap-In/Out buttons run on ESP32 Dev boards written in CPP.

Operation

The Display Manager is the first component to be running as it provides the wireless hosting and MQTT server for all other components. It is recommended that a keyboard and mouse be connected to this Raspberry Pi to launch the Display Manager software. (Shouldn't be necessary in later versions)

The Tap-In/Out buttons, Administrator Console and Trap Door should be powered next. They will connect using WiFi to the MQTT server where they will publish their status and subscribe to messages.

Match Manager

On the Administrator Console, the administrator will enter the names of each combat robot which will immediately update the Display Manager screen. By default, the match duration is set to 3 minutes but this can be modified prior to the start of the match from the Administrator Console. The Display Manager will show that the system is “Waiting” for the competitors to Tap-In.

When a competitor presses the Tap-In button, an MQTT message is published and the Display Manager updates the competitor’s status to “Ready”. The Display Manager publishes a request that the Tap-In LED be lit. When both competitors have Tapped In (and both LEDs are lit), the system is primed for the fight.

On the Administrator Console, the administrator starts the fight by pressing the “Start” button. The system will start counting down from 3:00 minutes reflected on both the Display Manager screen and the Administrator Console.

When the countdown reaches 20 seconds from the end of the match (currently fixed at 20 secs but to be made variable in later versions), the Trap Door system releases the latch and the trap door opens.

When the countdown reaches zero, the match status is displayed as “Finished”

If a competitor presses their Tap-In/Out button during the fight, this is interpreted as a Tap-Out request. The countdown immediately stops and the competitor status is updated to show they have Tapped-Out.

Design

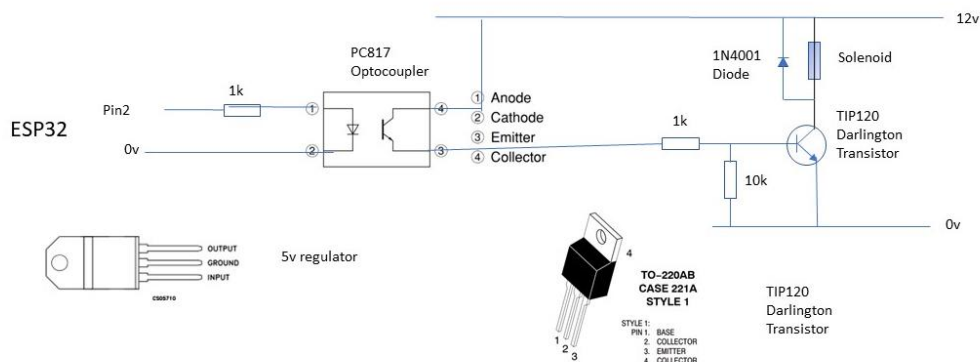
The following **Python libraries** are required for Administrator Console and the Display Manager:

- Maho-mqtt for MQTT messaging
- Tkinter for drawing user interface

The following **ESP32 C++ libraries** are required for the Tap-In/Out buttons and the Solenoid

- WiFi.h, SPI.h for WiFi connectivity
- PubSubClient.h for MQTT messaging

The **Trap Door Solenoid** is driven from the ESP32 with the following circuit



The **Tap-In/Out** button runs on 5v with the microswitch connected to pin 23. The LED is connected to pin 2

Match Manager

Access Point Configuration

See separate document

Requirements

Design

Components

- Display / Server

- Console

- Tap In/Out buttons

- Trapdoor solenoid

Hardware

- Raspberry Pi

- ESP32

- Solenoid

- Solenoid Driver Board & PSU

Software

- Display / MQTT Server

 - Mosquitto Server

 - Python

 - Python Libraries

 - MQTT

Match Manager

Console

Tap In/Out buttons

Theory of Operation

Wireless connectivity

Use of MQTT

Display Visualization