Universitat de Lleida

Enginyeria Informàtica

Embedded and Ubiquitous Systems

Prof. Fernando Guirado Fernández

e-Game Board

Meeple Showdown

Technical Documentation

Jordi García Ventura

Christian López García

January 10, 2025

Contents

1 Component wiring schematics

1.1 Meeple

It consists on 2 LEDs, a green one and a yellow one, a hall sensor, a microcontroller (ESP-01) and a battery, as in the **Figure** 1. Each component has the following function:

1. **ESP-01**: Microcontroller that controls the LEDs, reads the Hall Sensor and reads/writes feedback with MQTT.
2. **Battery**: Power source for the ESP-01
3. **Hall Sensor**: Detects when the meeple is being detected on board, detects meeple movement and death.
4. **Green LED**: Has two modes, when blinking it indicates that it's the meeple turn for moving, when solid it indicates the hall sensor is detecting a magnetic field (the meeple is detecting the board).
5. **Yellow LED**: Indicates if the player has the bullet on the shooting stage.

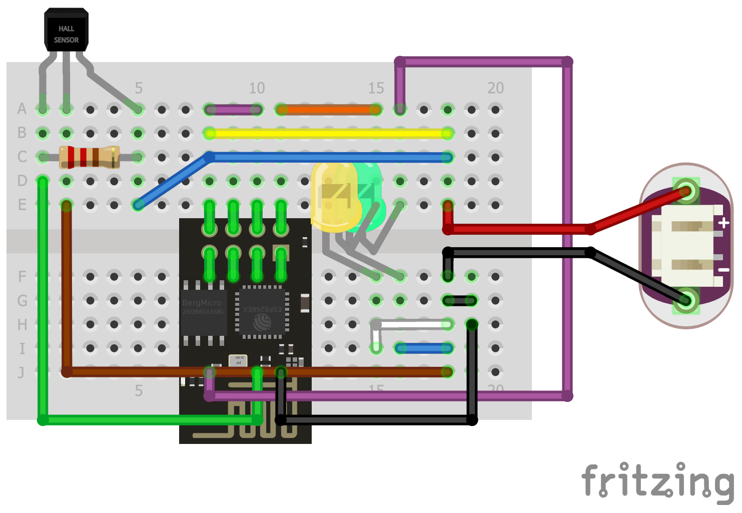


Figure 1: Meeple components schematic

1.2 Operation base

It consists on a red LED, a button, a buzzer, a microcontroller (ESP-32) and a LCD screen connected through a I2C module, as shown in the **Figure** 2. Each component has the following function:

1. **ESP-32**: Microcontroller that controls the LCD screen, reads the button, reads/writes feedback with MQTT (through WiFi) and controls the buzzer.

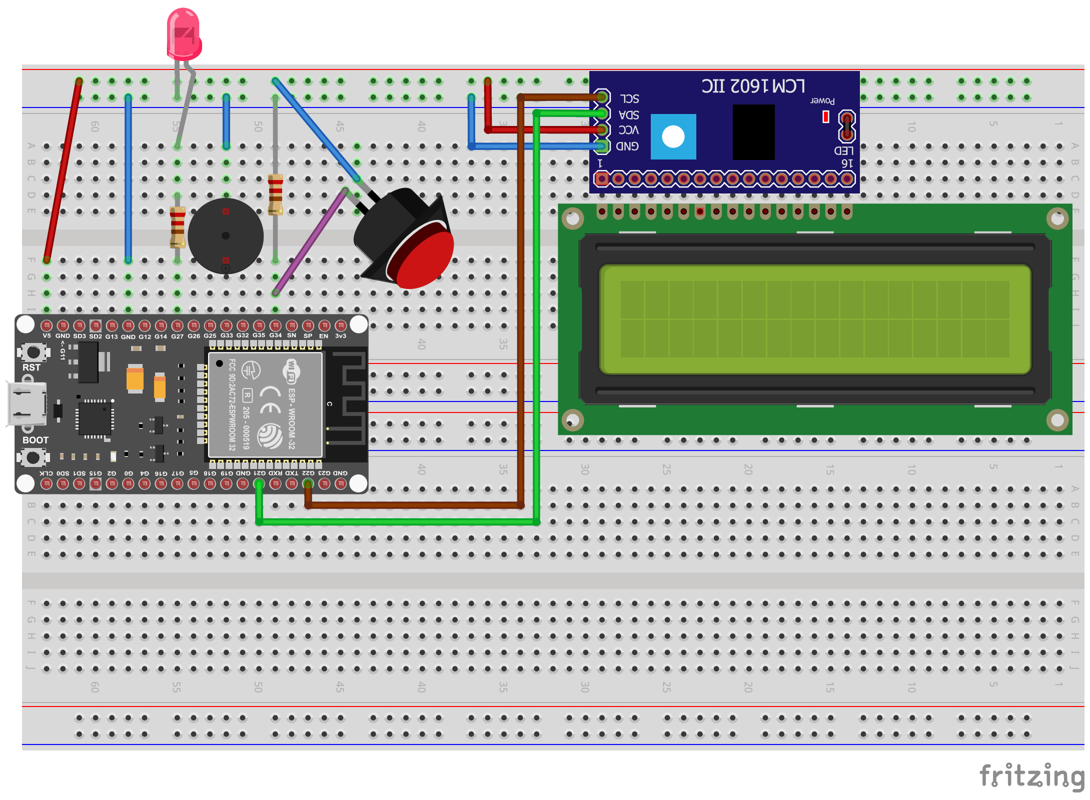


Figure 2: Operation base components schematic

2. **LCD screen**: Displays the current game stage and the player's turn.
3. **Button**: Used to shoot the bullet.
4. **Buzzer**: Indicates the end of the game.
5. **Red LED**: Indicates whether the player has the bullet.
6. **I2C module**: Module that connects the ESP-32 with the LCD screen through I2C protocol.
7. **Resistors**: Both resistors are 220 Ohm and are used to limit the current.

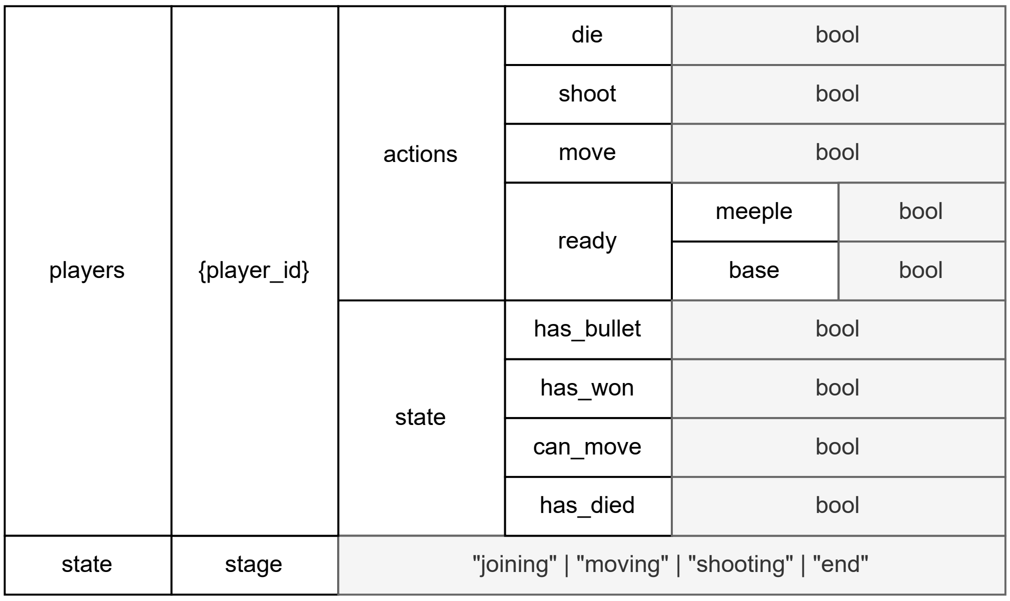
The board can be powered through a Micro-USB cable from a computer USB port.

2 MQTT topics

We can see the used MQTT topics in **Figure** 3, defined specifically in **Section** 2.1, 2.2.

2.1 Player topics

With the base topic **players/<player_id>/**, where *player_id* is the player's predefined unique identifier (e.g. in our case *duo_jc*), we have the following separated topics, where the *actions* subtopics are used to send feedback from the players and the *state* subtopics are used to receive



players	{player_id}	actions	die	bool	
			shoot	bool	
			move	bool	
			ready	meeple	bool
				base	bool
		state	has_bullet	bool	
			has_won	bool	
			can_move	bool	
			has_died	bool	
state	stage	"joining" | "moving" | "shooting" | "end"			

Figure 3: MQTT topics resume

feedback from the game controller:

Player actions topics:

- **actions/die**: *bool.* Indicates the player's death.
- **actions/move**: *bool.* Indicates the player's movement.
- **actions/shoot**: *bool.* Indicates the player's shooting decision.
- **actions/ready/meeple**: *bool.* Indicates the player's meeple is ready.
- **actions/ready/base**: *bool.* Indicates the player's base is ready.

Player state topics:

- **state/has_bullet**: *bool.* Indicates if the player has the bullet.
- **state/has_won**: *bool.* Indicates if the player is the last alive.
- **state/has_died**: *bool.* Indicates if the player has died.
- **state/can_move**: *bool.* Indicates if it's the player's turn to move.

2.2 State topics

With the base topic **state/**, we have the following topics:

- **stage**: *"joining"* | *"moving"* | *"shooting"* | *"end"*. Indicates the current game stage for all the players.