DESIGN DOCUMENT FOR **SMART IRRIGATION SYSTEM USING ESP32**

**Version**: 1.0

**Date**: 06-05-2024

**Author**: TEAM\_A

**Document Control**

**Document Information​**

|  |  |
| --- | --- |
| **©** | **Information** |
| Document Owner | TEAM\_A |
| Issue Date | 06-05-24 |
| File Name | DESIGN DOCUMENT FOR SMART IRRIGATION SYSTEM USING ESP32 |

**Document History​**

|  |  |  |
| --- | --- | --- |
| **Version** | **Issue Date** | **Changes** |
| v1.0 | 06-05-2024 | Initial Version |
| v1.0 | 07-05-2024 | Specification and feature added |
| v1.0 | 08-05-2024 | API functions added |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Document Review History​**

|  |  |  |  |
| --- | --- | --- | --- |
| **Role** | **Name** | **Signature©** | **Date** |
| Drafted by | Team A |  |  |
| Reviewed by |  |  |  |
| Approved by |  |  |  |

**TABLE OF CONTENTS**

[**1. Introduction 4**](#_tku9nelzwvy3)

[**2. System Overview 4**](#_ss1becsh52il)

[**3. System Architecture 4**](#_7wkg22jfl3vb)

**4. System Hardware Design 4**

[**4. Software Block Diagram**](#_kcsx1yc7sz8n) **4**

4.1 ESP32 5

4.2 SPI Interface 5

4.3 I2C Interface 6

4.4 ADC Interface 7

4.5 Motor Driver 7

4.6 Power Supply 8

4.7 Circuit Diagram 8

[**5. Software Requirement**](#_ogi5jich8qwq) **9**

5.1 Flow Diagram 9

[5.2 ESP32 Microcontroller:](#_y77ya4bmwx9a) 9

[5.3 SPI](#_1zhll7go3bdk) 9

5.4 I2C 10

5.5 Application Layer 10

[**7. Conclusion**](#_qk654j7ebwdz) **11**

[**8. References 1**](#_3rol4ut1o83c)1

# 

# **1. Introduction**

This document outlines the software and hardware design considerations for integrating. This includes integration of Bluetooth, Serial Peripheral Interface (SPI), Inter-Integrated Circuit (I2C), and a moisture sensor with an ESP32 microcontroller for smart agriculture. This project aims to demonstrate the seamless integration of these technologies to create a robust and flexible sensing solution.

# **2. System Overview**

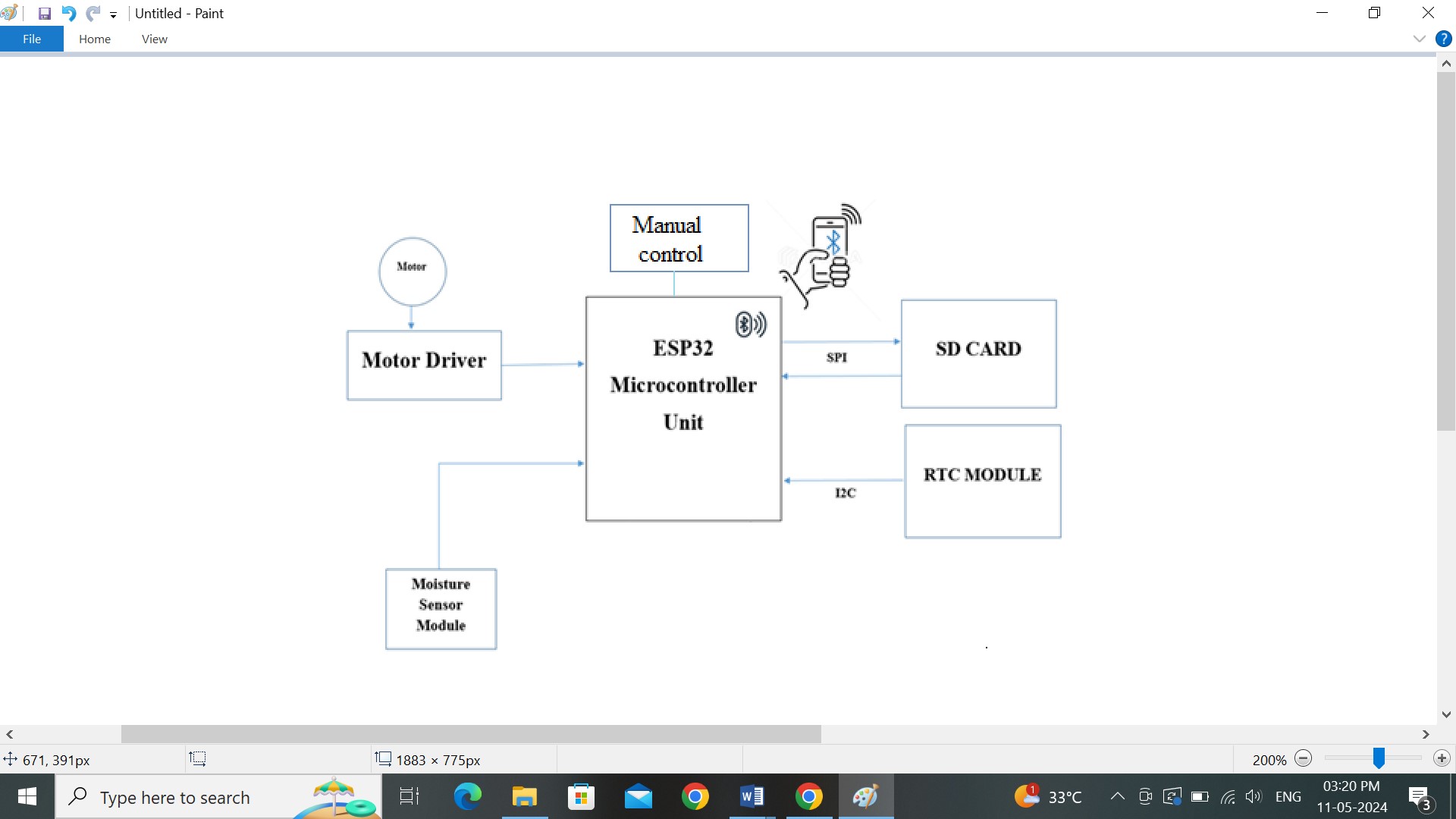
The system comprises an ESP32 microcontroller with soil moisture, I2C and SPI peripherals. The primary objective is to establish communication between the microcontroller and moisture sensor for controlling the motor operation, I2C for reading at timings for motor on and off, SPI external SD card for data storage and retrieval purposes.

# **3. System Architecture**

The system architecture involves the following components:

* ESP32 Microcontroller (e.g., ESP32 Development board).
* SPI (SD card).
* I2C Peripheral Interface.
* Moisture sensor.
* Motor Module.

**4. System Hardware Design**



## 

**4.1 ESP32**

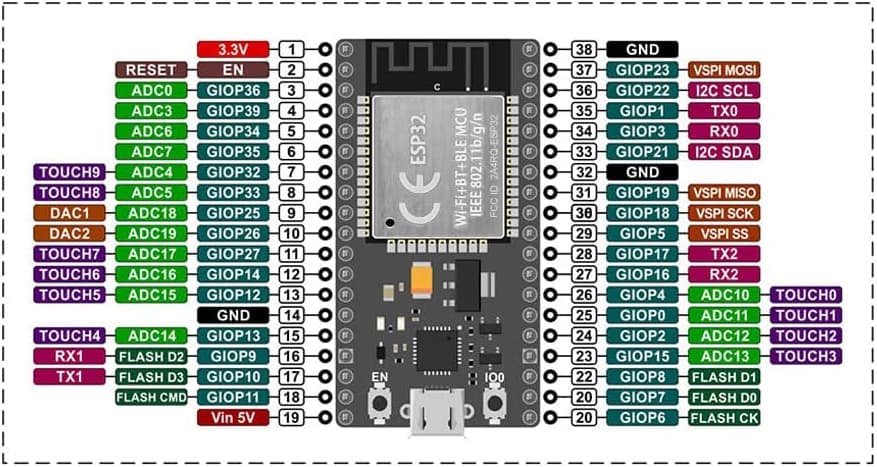
**4.1.1 FEATURES:**

* It features a dual-core Xtensa LX6 CPU running at up to 240MHz. It integrates Wi-Fi, Bluetooth, and a wide range of peripheral interfaces.
* It interfaces with connecting sensors, actuators, displays, and other external devices include SPI, I2C, UART, GPIO, ADC more.
* It is supported by the Arduino IDE, ESP-IDF (Espressif IoT Development Framework), and other development environments.
* It is open source community support provides documentation, libraries, tutorials, and support forums.

**4.1.2 SPECIFICATIONS:**

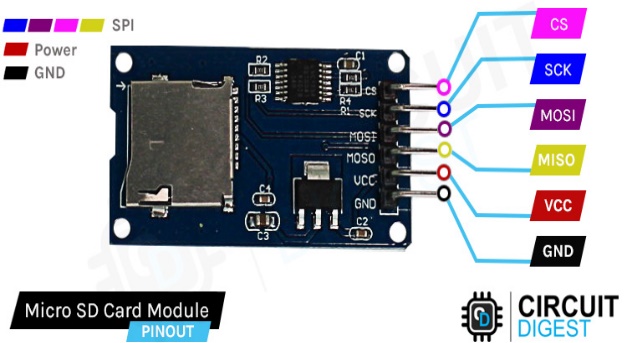
* Microcontroller: Dual-core Xtensa LX6 CPU running at up to 240MHz
* Wireless connectivity: Wi-Fi(802.11 b/g/n) and Bluetooth (BLE) connectivity
* Flash memory: range from 4MB to 16MB
* RAM: range from 520KB to 4MB
* Power Supply: 3.3v to 3.6v

**4.1.3 PIN Description:**



## **4.2 SPI INTERFACE**

The SPI interface involves the physical connections between the ESP32 microcontroller and the SPI SD card. Key considerations include:

* Assigning appropriate GPIO pins on the microcontroller for SPI communication (e.g., MOSI, MISO, SCK, CS).

**4.2.1 FEATURES:**

* It have SD card slot which is designed to accommodate standard-size SD cards, as well as smaller variants like microSD cards with the use of an adapter.
* It have voltage regulator to provide a stable power supply to the SD card. This helps ensure reliable operation, especially when powered from a variable power source.
* It have CS pin is used to select the SD card when communicating over SPI.

**4.2.2 SPECIFICATIONS:**

* Capacity: 8 Gigabytes.
* Interface: SPI(Serial Peripheral Interface)
* Speed class: 2 megabytes per second(MB/s)
* File system: FAT32 file system
* Operating voltage: 2.7 to 3.6 volts.

|  |  |  |  |
| --- | --- | --- | --- |
| **SL.NO** | **Port** | **Connected to** |  |
| 1 | 14 | HSP1\_CLK | CLK |
| 2 | 13 | HSPI1\_MOSI | DO |
| 3 | 12 | HSPI1\_MISO | DI |
| 4 | 15 | CS | CS |

## 

## **4.3 I2C INTERFACE**

The Inter-Integrated Circuit (I2C) communication protocol is often used for interfacing Real-Time Clocks (RTCs). The RTC is an essential component in these systems, as it provides accurate timekeeping and scheduling capabilities for irrigation schedules and data logging.



**4.3.1 FEATURES:**

* To keep track of time, including hours, minutes, seconds, and often dates.
* To maintain timekeeping functionality even when the main power source is disconnected.
* Suitable for battery-operated devices consume less power.

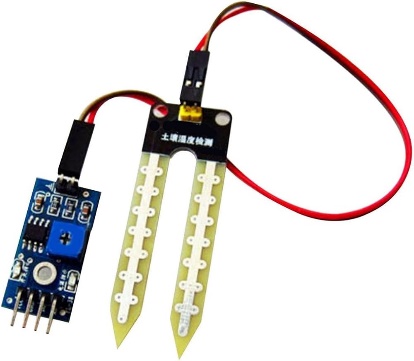
**4.3.2 SPECIFICATIONS:**

* Package Type: 8-pin DIP (Dual Inline Package)
* Supply voltage: voltage range of 4.5V to 5.5V DC
* Battery Backup: 3v lithium coin cell battery
* Crystal Oscillator: 32.768kHz crystal oscillator circuit
* Interface: I2C interface (2-wire communication for data transfer with SDA & SCL.

|  |  |  |
| --- | --- | --- |
| **SL.NO** | **RTC Pins** | MCU Pins |
| 1 | SCL | D21 |
| 2 | SDA | D22 |
| 3 | Vcc | Vcc |
| 4 | Gnd | gnd |

## **4.4 ADC INTERFACE**

The ADC convert sensor measured data to digital data. The sensor measures the soil moisture content in the root zone before a scheduled irrigation event and bypasses the cycle if the soil moisture is above the threshold.



**4.4.1 FEATURES:**

* Even the smallest change can be detect
* Quick response time
* Sensing method: uses resistive, capacitive and electromagnetic techniques

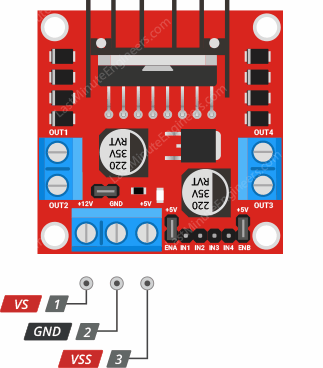
**4.4.2 SPECIFICATIONS:**

* Operating voltage: 3.2vots.
* Pins: 4pins ( A0, D0, Vcc, Gnd);

|  |  |  |
| --- | --- | --- |
| **SL.NO** | **Sensor pins** | **MCU Pins** |
| 1 | A0 | D4 |
| 2 | VCC | VCC |
| 3 | GND | GND |

## **4.5 Motor Module**

## The motor is connected with ESP32 which can works both automatically and manually by auto adjusting the switch as per required aspects within the moisture level motor is off state when it reduces motor turn on automatically and water the soil.



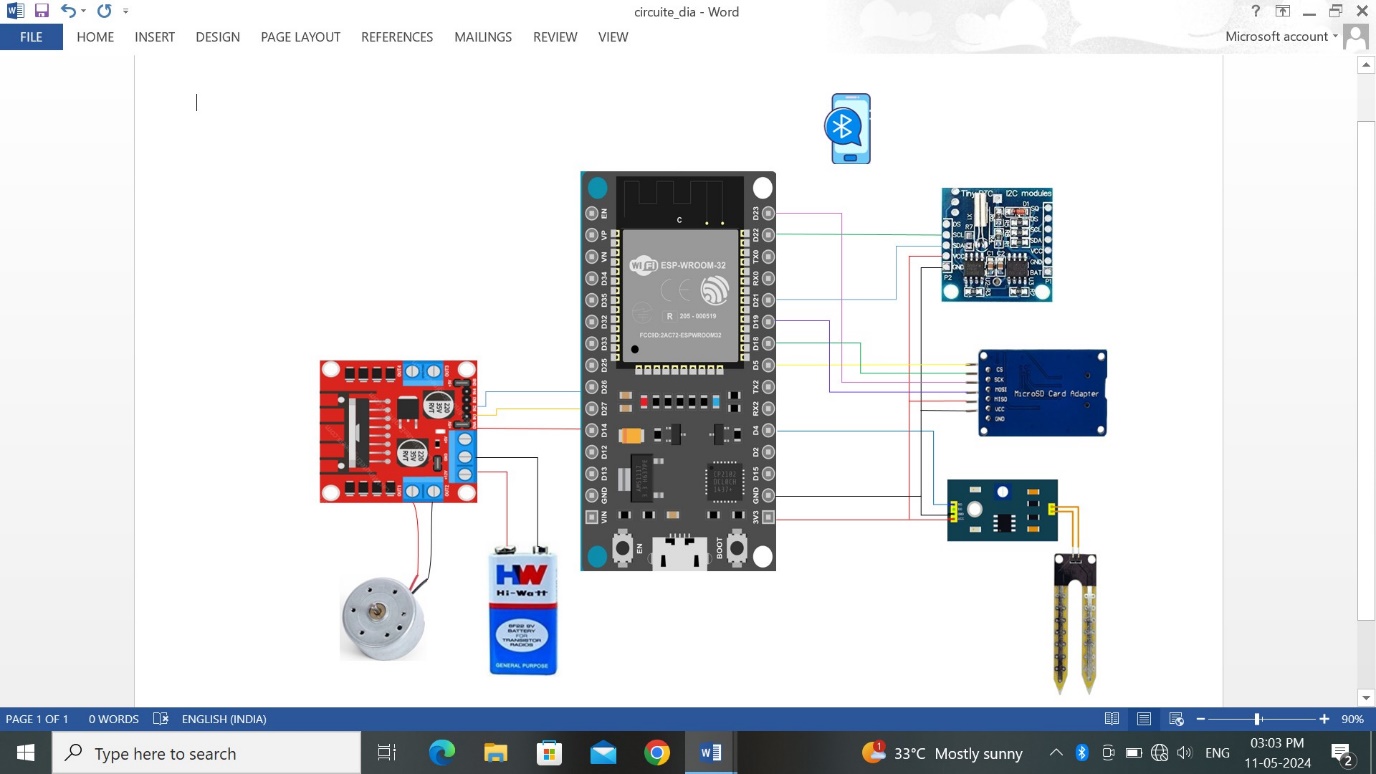
**4.5.1 SPECIFICATIONS:**

* Driver Voltage: 5 - 35V
* Heatsink for better performance
* IN1 & IN2 input pins Used to control the Motor A.
* ENA Enables PWM signal for Motor A.

|  |  |  |
| --- | --- | --- |
| **SL.NO** | **L298N** | **MCU** |
| 1 | IN1 | 26 |
| 2 | IN2 | 27 |
| 3 | ENA | 14 |
| 4 | GND | GND |
| 5 | VCC | VCC |

## **4.6 Power Supply**

Provide adequate power supply connections to both the ESP32 microcontroller and the SPI. Adhere to the voltage and current requirements specified in the datasheets to prevent voltage fluctuations and ensure stable operation of the components

**4.7 CIRCUIT DIAGRAM**

# **5. Software Requirement**

The software block diagram depicts the interactions and dependencies between various software components involved in SPI and I2C, including SPI, I2C drivers, memory management, and application layers.

# **5.1 Flow Diagram:**

# C:\Users\FAZULULLA\AppData\Local\Packages\Microsoft.Windows.Photos_8wekyb3d8bbwe\TempState\ShareServiceTempFolder\Screenshot (4).jpeg **5.1.1 Flow Description:**

## 

## **5.2 ESP32 Microcontroller:**

* **Functionality**:
  + Provides the foundation for the system's operation by serving as the central processing unit.
  + Manages hardware peripherals and executes software instructions.
  + Interfaces with higher-level software components such as the SPI, I2C drivers and memory management module.

## **5.3 SPI:**

* **API Functions:**

SD.open(); // This function is used to opening the file in SD Card.

SD\_card\_write(); // To write data into file in SD card.

SD\_card\_read\_function(); // Read function to read data from SD card.

String integer\_to\_string(); // converting integer values into string

SD.begin(chipSelect); // selecting SD card for communication

* **Functionality:**
  + Establishes communication between the ESP32 microcontroller and the SPI collecting the data throw sensor.
  + Configures SPI communication parameters such as clock frequency and data format.
* Manages data transmission and reception between the microcontroller and SPI SD card.

## **5.4 I2C:**

* **API Functions:**

Wire.begin(); // starting I2C communication

Wire.requestFrom(); // requesting bytes of data from RTC

Wire.beginTransmission(); // start condition

Wire.endTransmission(); // stop condition

setDS1307Time(); // setting time to RTC

readRTCTime(); // reading time from RTC.

* **Functionality:**
* Establishes communication between the ESP32 microcontroller and RTC module collecting the proper time.
* The I2C reads the time collected and store the real time clock pulses.
* By this MCU know the timings of motor on or off state and further it is send SPI.

## **5.5 Application Layer:**

* **API Functions:**

Serial.begin(115200); // baud rate setting to esp32.

file.readStringUntil('\n'); // reading from file.

digitalWrite(); // writing to required pins.

Serial.println(); // printing message on serial terminal.

SerialBT.readString(); // reading string from Bluetooth terminal.

analogRead(); // reading data from sensor.

void startMotor(); // starting Motor

void stopMotor(); // Stopping Motor.

* **Functionality:**
* Updating the soil irrigation system with additionally combined with the SPI and I2C can give the detailed information of running of motor which is useful for reference.
* Adding a two switches increases a chances to run the motor as per requirement whether it should be operate automatically or else manually both can be attained.

# **6. Conclusion**

This document provides guidelines for both software and hardware design aspects of integrating the ESP32 with SPI, I2C & Bluetooth module which works for more efficient in checking water level in soil and store the readings using SPI to SD card, which are observed and noted by I2C using RTC. By following the outlined architecture and design considerations, developers can ensure efficient and reliable communication between the microcontroller and the other peripherals, facilitating seamless integration into embedded systems applications.

# **7. References**

* ESP32 Microcontroller Datasheets and Reference Manual.
* SPI Memory Chip Datasheet.
* I2C Datasheet.
* Arduino IDE Programming reference.