# 1. INTRODUCTION

## 1.1. Brief Overview of Tektork IoT Kit-V1.0


The Tektork IoT kit -V1.0 development board is a robust and versatile platform designed for learning, prototyping, and deploying Internet of Things (IoT) solutions. Our boards undergo rigorous testing before delivery. Tektork IoT board is manufactured using high-quality PCBs, components, assembly processes, and testing procedures to ensure reliability and performance. The Tektork IoT board represents a culmination of academic expertise and professional manufacturing capabilities. The innovative and user-friendly design concept was developed and tested at the School of Computing, Amrita Vishwa Vidyapeetham, Coimbatore. This forward thinking design bridges theoretical knowledge with practical application, creating an IoT platform suitable for both educational and commercial purposes.

At its core is the ESP32-WROOM-32 module, a dual-core microcontroller known for its powerful processing capabilities, integrated Wi-Fi and Bluetooth, and extensive peripheral support.

- The main processor is a dual-core 32-bit Xtensa LX6 CPU running up to 240 MHz, supported by 520 KB SRAM and 4 MB flash memory.

- Built-in Wi-Fi (802.11 b/g/n) and Bluetooth (v4.2, including BLE) allow seamless wireless communication, making the board ideal for IoT ecosystems.
- The board accommodates real-time operating systems like FreeRTOS and includes on-board interface chips (e.g., CH340 or CP2102) for easy USB connectivity and programming.
- A jumper-based system connects onboard components (LEDs, buttons, potentiometer, LDR) to dedicated General Purpose Input/Output (GPIO) pins, which can be easily reconfigured for different experiments and projects.
- Essential on-board controls such as reset and boot push buttons simplify programming and troubleshooting.

## 1.2. IDENTIFICATION OF BOARD VERSION AND IOT APPLICATION SCOPE

- Board Version: Tektork IoT -V1.0 typically refers to boards where the ESP32-WROOM-32 module is mounted, offering maximum flexibility for beginner and advanced projects.
- Pinout and Expansion: The board provides labeled pin headers for GPIO, UART, SPI, $I^2C$, CAN, and analog inputs, supporting direct connection to various sensors and modules. Clear silkscreen labels enable quick identification and ensure error-free wiring.
- Size and Layout: Physical dimensions are optimized for breadboard compatibility (e.g., 60 x 105 mm, or standard 30 or 36 pins), ensuring that students can rapidly prototype without soldering.
- Onboard Components for Immediate Learning:
  Two user LEDs (GPIO33 and GPIO4) for output exercises
  One push button (GPIO32) for digital input
  10K analog potentiometer (GPIO36) and LDR (GPIO39) for analog input activities.

## 1.3. IOT APPLICATION SCOPE

The Tektork IoT -V1.0 is the ideal development board for modern IoT training and R and D prototype development:

- Wireless Connectivity: Thanks to integrated Wi-Fi and Bluetooth, the board can directly connect to local networks or cloud platforms, collect sensor data, control devices remotely, and implement smart automation.
- Flexible Prototyping: The jumper-based design means students can quickly reconfigure pin assignments for new experiments without re-writing much code.
- Sensor and Actuator Integration: The board supports direct interfacing of sensors, actuators, and communication modules (SPI flash, CAN transceivers, gyroscope/accelerometer) for creating complete IoT solutions.

- Software Support: Open-source platforms like Arduino IDE and MicroPython firmware are fully supported, making programming and debugging approachable for third-year engineering students and beyond.
- Security Features: Secure boot, hardware encryption (AES, SHA, RSA, ECC), and safety features allow basic exploration of IoT device security for coursework.
- Real-World IoT Applications: This board is suitable for projects like smart homes, environmental monitoring, wearable devices, and industrial automation—providing students with hands-on experience that translates to industry and research.

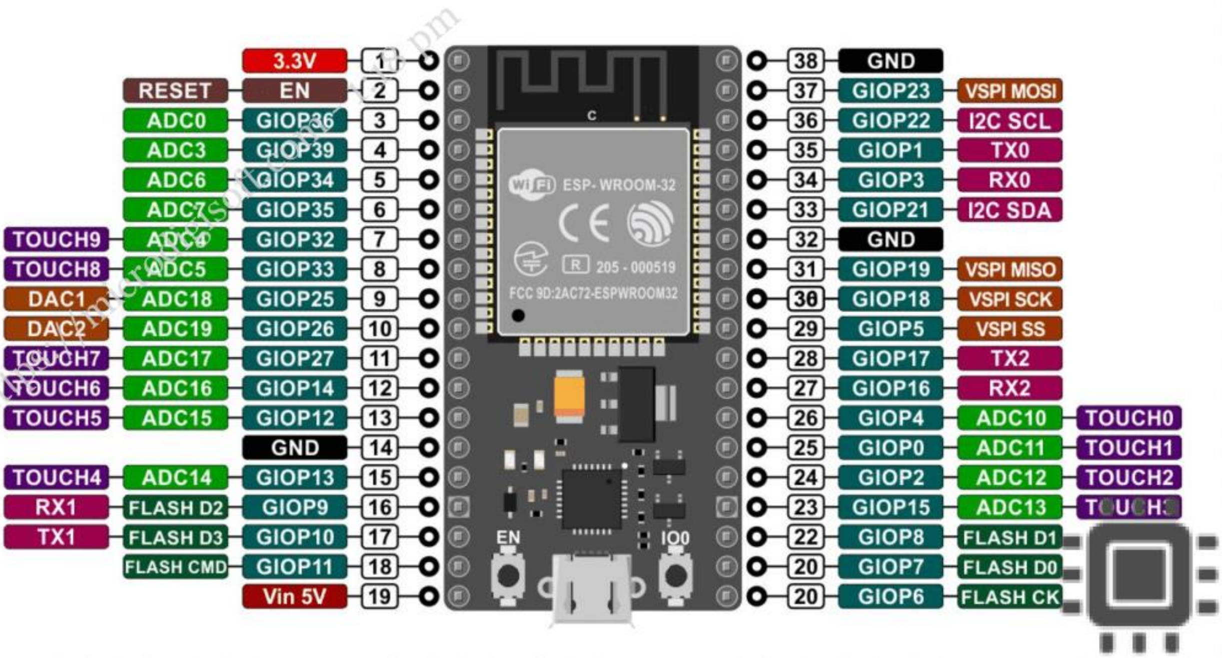
# 2. Tektork IoT V1.0 BOARD FEATURES AND INSTRUCTIONS

## 2.1. BOARD OVERVIEW AND VERSION IDENTIFICATION

### 2.1.1. Tektork IoT KIT V1.0 INTRODUCTION

The Tektork IoT- V1.0 development board is a powerful IoT development platform built around the ESP32-WROOM-32 Wi-Fi and Bluetooth module. This comprehensive development kit is specifically designed for training and prototyping IoT applications, making it an excellent choice for students and developers learning embedded systems programming.

**GPIO Pins of ESP32**




| GPIO Pin | Spl Notes | FXN 1 | FXN 2 |
|---|---|---|---|
| 0 | General | ADC 11 | TOUCH 1 |

| | | | |
|---|---|---|---|
| 1 | General | TX 0 | |
| 2 | Bootstrap | ADC 12 | TOUCH 2 |
| 3 | General | RX 0 | |
| 4 | Bootstrap | ADC 10 | TOUCH 0 |
| 5 | Bootstrap | VSPI SS | |
| 6 | Internal use | FLASH CK | |
| 7 | Internal use | FLASH D0 | |
| 8 | Internal use | FLASH D1 | |
| 9 | Internal use | FLASH D2 | RX 1 |
| 10 | Internal use | FLASH D3 | TX 1 |
| 11 | Internal use | FLASH CMD | |
| 12 | Bootstrap | ADC 15 | TOUCH 5 |
| 13 | General | ADC 14 | TOUH 14 |
| 14 | General | ADC 16 | TOUCH 6 |
| 15 | Bootstrap | ADC 13 | TOUCH 13 |

| GPIO Pin | Spl Notes | GPIO Pin | Spl Notes |
|---|---|---|---|
| 16 | General | RX 2 | |
| 17 | General | TX 2 | |
| 18 | General | VSPI SCK | |

| | | | |
|---|---|---|---|
| 19 | General | VSPI MISO | |
| 21 | General | I2C SDA | |
| 22 | General | I2C SCL | |
| 23 | General | VSPI MOSI | |
| 25 | General | ADC 18 | DAC 1 |
| 26 | General | ADC 19 | DAC 2 |
| 27 | General | ADC 17 | TOUCH 7 |
| 32 | General | ADC 04 | Touch 9 |
| 33 | General | ADC 05 | Touch 8 |
| 34 | General | ADC 06 | |
| 35 | General | ADC 07 | |
| 36 | General | ADC 00 | |
| 39 | General | ADC 03 | |

### 2.1.2. KEY FEATURES OF Tektork - IoT V1.0

The Tektork IoT V1.0 board incorporates the ESP32-WROOM-32 module as its core processor, featuring a dual-core 32-bit Xtensa LX6 microprocessor operating at frequencies up to 240 MHz. The board includes 520 KB of SRAM and 4 MB of flash memory, providing ample space for complex IoT applications.

**Core Specifications:**

- Processor: Dual-core 32-bit Xtensa LX6 CPU, up to 240 MHz
- Memory: 520 KB SRAM, 4 MB Flash memory
- Wireless: Wi-Fi 802.11 b/g/n (up to 150 Mbps), Bluetooth v4.2 BR/EDR and BLE
- Operating Voltage: 2.2V to 3.6V (with onboard 3.3V regulator)

- GPIO Pins: 34 programmable GPIO pins with multiple functions

### 2.1.3. SIGNIFICANCE OF VERSION V1.0 FOR TRAINING/PROTOTYPING

The Tektork IoT V1.0 is particularly well-suited for educational purposes and rapid prototyping due to several key advantages:

**Educational Benefits:**

- Comprehensive onboard components eliminate the need for external breadboarding for basic experiments
- Clear pin labeling and silk-screen markings make it easy for beginners to understand connections
- Jumper-based configuration system allows students to learn different GPIO assignments
- Built-in examples and sample programs provide hands-on learning opportunities

**Prototyping Advantages:**

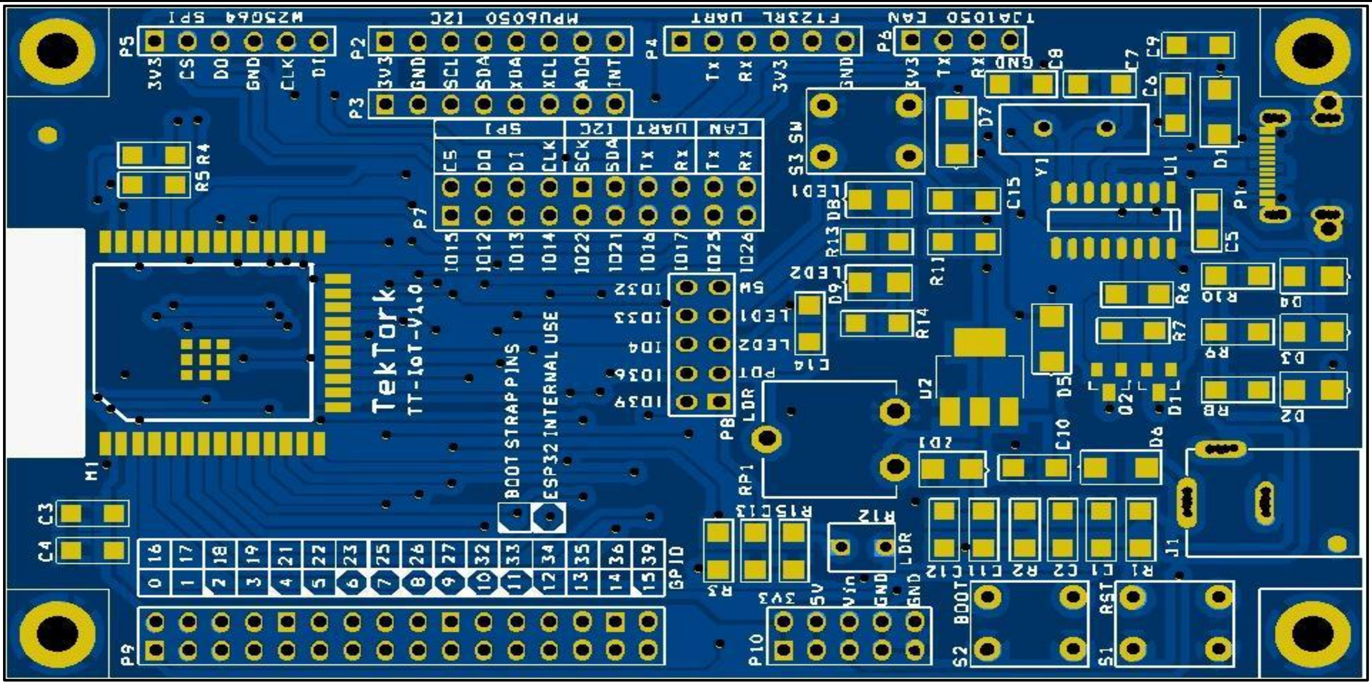
- Modular design with expansion connectors for UART, $I^2C$, SPI, and CAN protocols
- Integrated components (LEDs, buttons, sensors) for immediate testing
- USB-C connector for modern programming and power delivery

Robust PCB design (60mm × 105mm) suitable for laboratory environments

## 2.2. PCB LAYOUTS AND COMPONENT POSITIONS

### 2.2.1. PCB TOP SIDE LAYOUT WITH LABELS AND KEY COMPONENTS

The top side of the Tektork IoT V1.0 PCB features a well-organized layout with clearly labeled components and connection points. The design prioritizes accessibility and clarity for educational use.

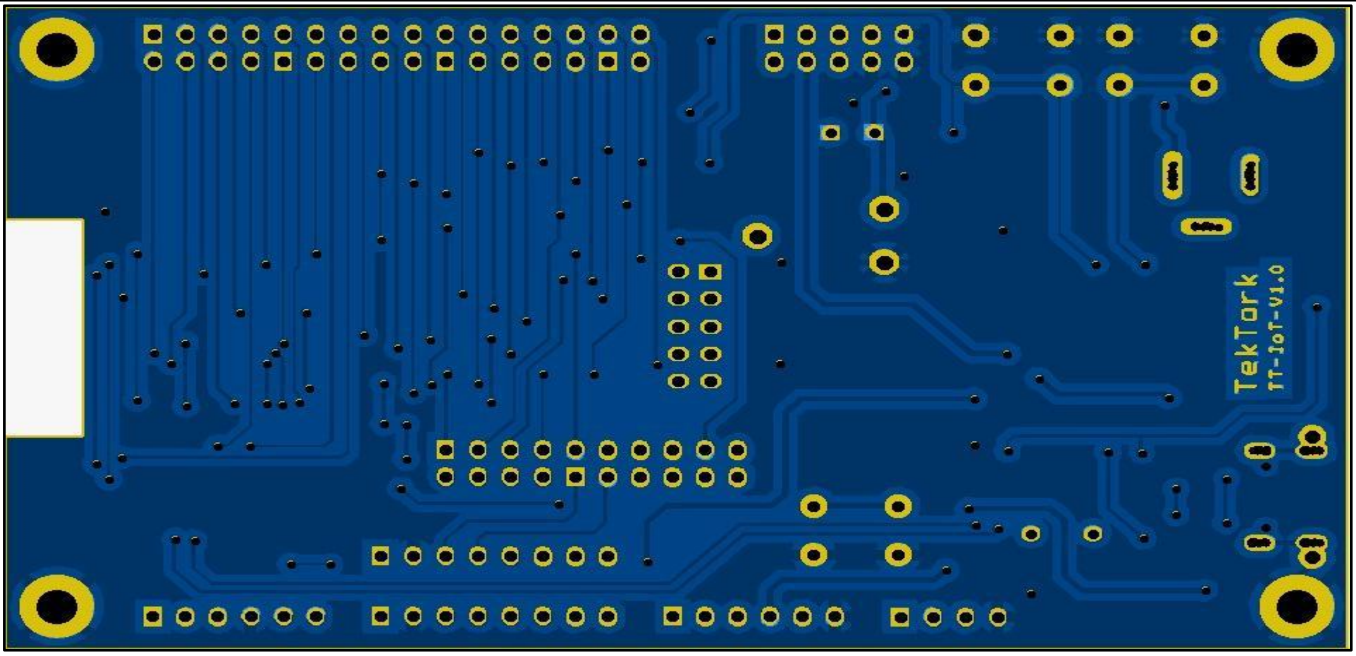



**Major Components on Top Side:**

- ESP32-WROOM-32 Module: Central processing unit located in the middle of the board
- USB-C Connector: Modern USB interface for programming and power
- CH340 USB-to-UART: Enables communication between PC & Board.
- User Components: Two LEDs, one push button, potentiometer, and LDR.
- GPIO Headers: Clearly labeled pins arranged in rows for easy access

### 2.2.2. PCB BOTTOM SIDE LAYOUT OVERVIEW

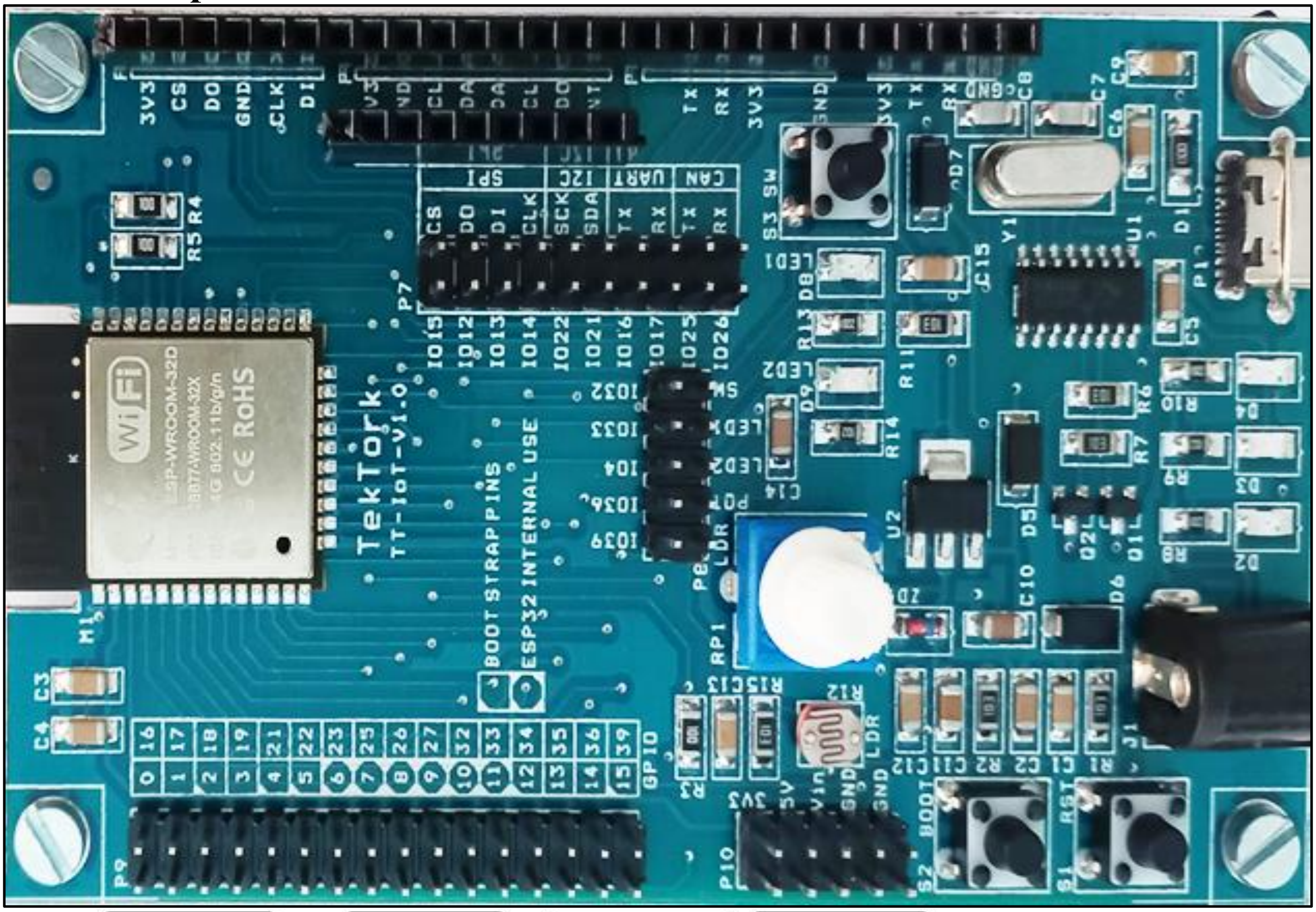
The bottom side of the PCB contains primarily the ground plane and supporting circuitry. The design maintains signal integrity while providing adequate power distribution across the board.

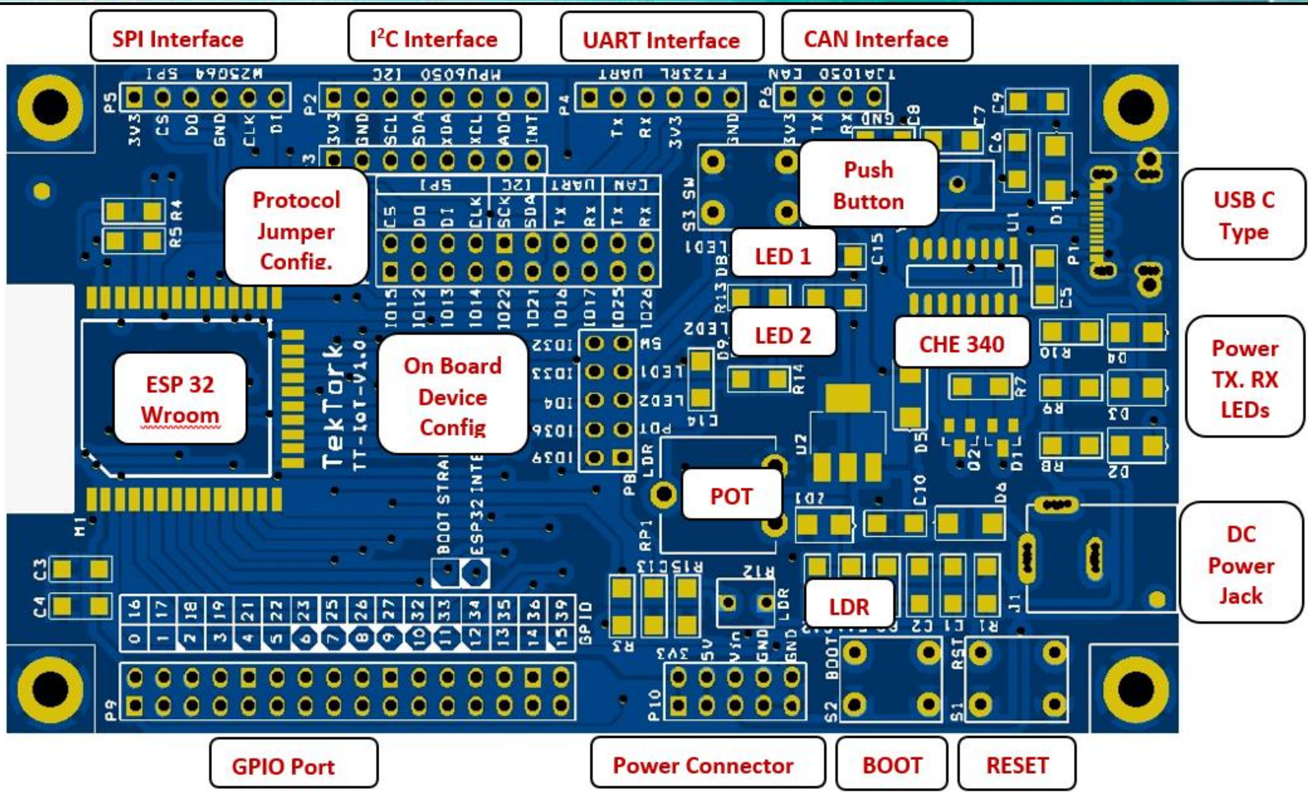



### 2.2.3. PLAIN PCB ORIENTATION AND FIXED COMPONENT POSITIONING

The board is designed with a horizontal orientation, with the USB-C connector typically positioned on one end for easy cable access. Component placement follows standard PCB design practices with critical components positioned to minimize interference and maximize accessibility.

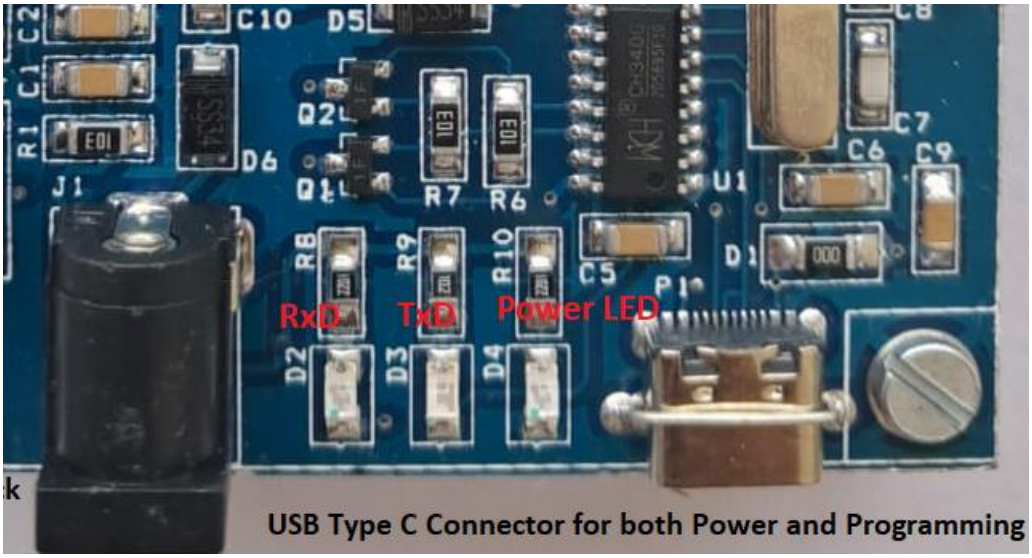
**Fixed Component Locations:**






- Power management circuitry near the USB connector
- Crystal oscillator close to the ESP32 module for stable timing
- Bypass capacitors strategically placed for power filtering
- User interface components positioned for easy access during experiments

## 2.3. ON-BOARD COMPONENTS DESCRIPTION




**On Board LEDs: LED D2 -> RxD, LED D3 –> TxD, LED D4 –> Board Power .**

### 2.3.1. USER LEDS CONNECTED TO GPIO33, GPIO4

The ESP32 V1.0 board includes two user-controllable LEDs that serve as visual indicators for program execution and debugging.

**LED Specifications:**

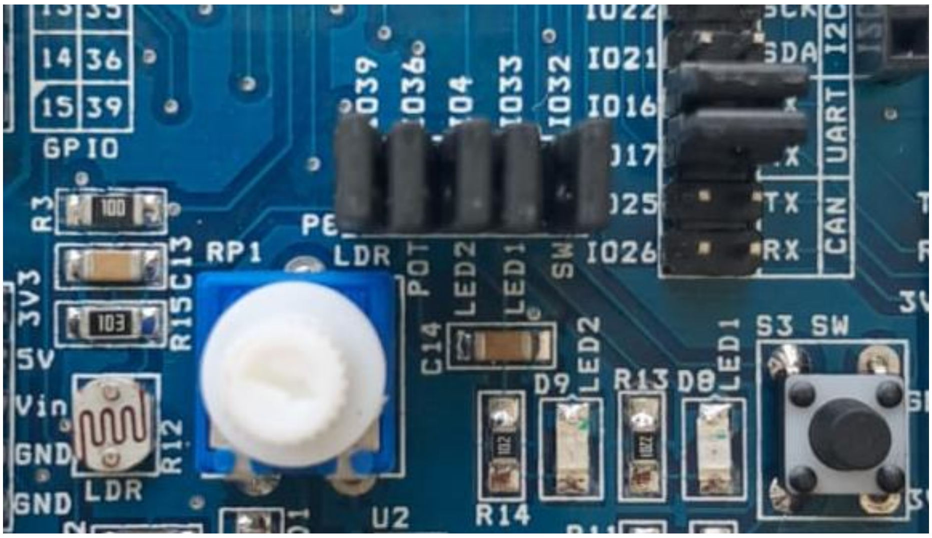
- LED1: Connected to GPIO33 by default
- LED2: Connected to GPIO4 by default
- Purpose: Visual feedback for program execution, status indication, and debugging
- Control: Can be programmed to blink, fade, or respond to sensor inputs
- Reconfiguration: Connection can be changed using jumpers to any available GPIO pin

### 2.3.2. USER PUSH BUTTON CONNECTED TO GPIO32

A single push button provides user input capability for interactive programs and system control.

**Button Specifications:**

- Connection: GPIO32 (default)
- Type: Momentary push button with internal pull-up resistor
- Function: User input for program control, mode switching, or triggering events
- Programming: Can be configured for interrupt-driven or polling-based input reading.




**On Board User Button (32), LED1 (33), LED2 (04), POT (36), LDR (39)**

### 2.3.3. ANALOG INPUTS: POTENTIOMETER (GPIO36), LDR (GPIO39)

The board includes two analog input devices for learning analog signal processing and sensor interfacing.

**Potentiometer (POT):**

Connection: GPIO36 (ADC channel)

Type: 10kΩ variable resistor

Function: Provides variable analog voltage (0-3.3V) for testing analog input reading

Applications: Volume control simulation, threshold setting, analog signal generation

**Light Dependent Resistor (LDR):**

Connection: GPIO39 (ADC channel)

Function: Measures ambient light intensity

Output: Resistance varies with light level, converted to voltage for ADC reading

Applications: Automatic lighting control, light sensing projects, environmental monitoring

### 2.4. JUMPER CONFIGURATION SETTINGS

#### 2.4.1. DEFAULT JUMPER SETTINGS EXPLAINED

The ESP32 V1.0 board uses a sophisticated jumper system that allows flexible GPIO pin assignment for onboard components. This system enables users to customize pin assignments based on project requirements.

**Default Pin Assignments:**

- GPIO33: LED1 (User LED 1)
- GPIO4: LED2 (User LED 2)
- GPIO32: User push button (S3)
- GPIO36: Potentiometer (POT)
- GPIO39: Light Dependent Resistor (LDR)

| User Peripherals | GPIO Pin |
|---|---|
| Switch SW (S3) | 32 |
| LED 1 (D7) | 33 |
| LED 2 (D8) | 4 |
| POT (RP1) | 36 |
| LDR (R12) | 39 |

By placing the individual respective jumpers, on board User Button (SW) , USER LED1 and 2, POT and LDR will be connected to the particular GPIO mentioned in the above table.

By removing the Jumper, they can be connected to any available GPIO pins as per design.

For example, suppose, if you want to connect USER LED 2 to GPIO 5, you have to remove the respective jumper connecting USER LED2 to GPIO4 and a jumper wire is used to connect one end of the USER LED 2 to GPIO 5 as illustrated in the following image.

**PLs Note that BOOT Button at GPIO 0 can also be used as user button during program running time.**

**So, users can have two buttons, two LEDs and two Analog inputs POT and LDR in the board itself.**

### 2.4.2. EFFECTS ON BOARD IO PIN MAPPINGS AND COMPONENT FUNCTIONS

The jumper system directly affects how onboard components connect to ESP32 GPIO pins, providing flexibility for different project configurations.

**Configuration Benefits:**

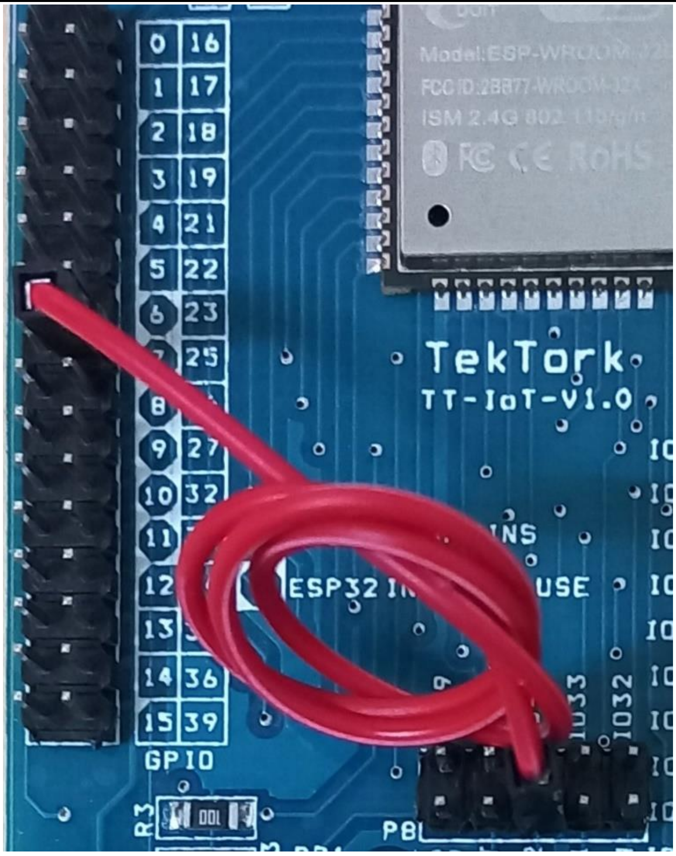
- Pin Flexibility: Any component can be reassigned to different GPIO pins
- Conflict Resolution: Allows resolution of pin conflicts when using multiple peripherals
- Learning Enhancement: Students can experiment with different pin assignments
- Project Customization: Enables optimization for specific project requirements

### 2.4.3. EXAMPLE OF JUMPER RECONFIGURATION FOR CUSTOM USE

**Re-configuring LED2 from GPIO4 to GPIO5:**

- Remove Default Jumper: Disconnect the jumper connecting LED2 to GPIO4
- Free GPIO4: GPIO4 becomes available for other uses (e.g., PWM output)
- New Connection: Use a jumper wire to connect LED2 to GPIO5
- Update Code: Modify the program to control LED2 on GPIO5 instead of GPIO4
- Verify Operation: Test the new configuration to ensure proper operation

This flexibility is particularly valuable when specific GPIO pins are needed for specialized functions like PWM, ADC, or communication protocols.




## 2.5. POWER SUPPLY AND USB INTERFACE

### 2.5.1. USB TYPE C CONNECTOR AND EXTERNAL POWER JACK DETAILS

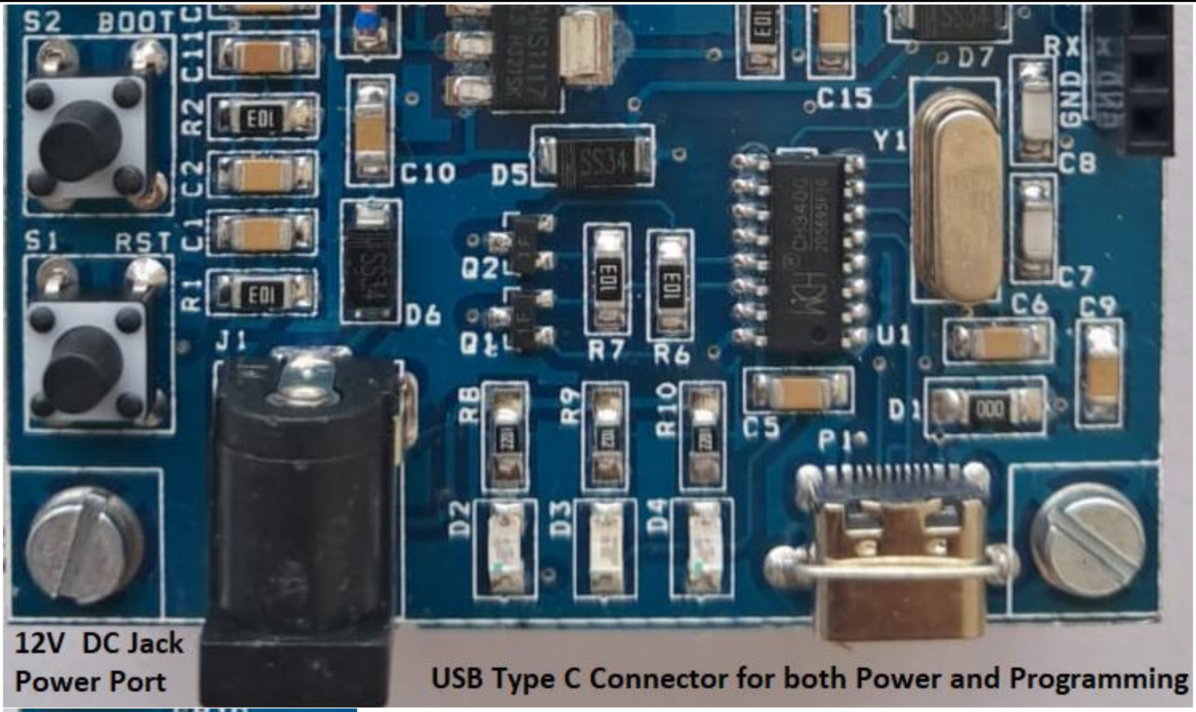
The ESP32 V1.0 board provides multiple power input options to accommodate different usage scenarios.

**USB-C Connector Features:**

- Modern Interface: USB-C provides reliable power and data connectivity
- Power Delivery: Supplies 5V from computer USB port or USB power adapter
- Data Transfer: Enables programming and serial communication with computer
- Current Capacity: Standard USB ports provide up to 500mA, sufficient for basic operations

**External Power Jack:**

- Voltage Range: Accepts 7V to 12V DC input
- Current Rating: Higher current capacity for power-hungry applications
- Connector Type: Standard DC barrel jack for easy connection to external power supplies
- Use Cases: Laboratory power supplies, battery packs, or dedicated power adapters

- 


Users can take 3.3V , 5V and Ground from the Connector P10 for other peripheral devices. User can give external 5 V to the Board through VIN pin. (Note: Take care while powering the board other than USB port. VIN- Max 5V and the board can be also powered through DC Jack Socket with a maximum of 12V DC source)

### 2.5.2. ON-BOARD RESET AND BOOT PUSH BUTTONS

Two essential control buttons are provided for system management and programming.

**Reset Button:**

- Function: Restarts the ESP32 and re-runs the uploaded program
- Usage: Press to restart the system after programming or during debugging
- Circuit: Connected to the ESP32's reset pin (EN)
- Behavior: System performs complete restart including initialization routines

**Boot Button:**

- Function: Places ESP32 into bootloader mode for programming
- Programming Mode: Hold during upload to enable firmware programming
- Alternative Use: Can be configured as additional user input during runtime

- Pin Connection: Connected to GPIO0 (bootstrap pin)


**Switch S1 is Reset (RST) and Switch S2 is Boot Button (also user Button GPIO 0 after uploading Code)**

### 2.5.3. CH340 USB TO UART CONVERTER AND ITS ROLE

The CH340 chip serves as a critical interface between the computer's USB port and the ESP32's serial communication pins.

**CH340 Functionality:**

Protocol Conversion: Converts USB signals to UART (serial) for ESP32 communication

Programming Interface: Enables code upload from Arduino IDE to ESP32

Serial Monitor: Provides bidirectional communication for debugging and data exchange

Driver Requirements: Requires CH340 driver installation on host computer

**Driver Installation Process:**

Download: Obtain CH340 driver from manufacturer website

Installation: Run installer appropriate for operating system (Windows/Mac/Linux)

Verification: Check Device Manager (Windows) or system information for proper recognition

Port Selection: Select correct COM port in Arduino IDE for programming

### 2.5.4. POWER PINS (5V, 3.3V, GND) LOCATIONS

The board provides multiple power access points for connecting external components.

**Power Pin Specifications:**

| Pin Type | Voltage | Current Capacity | Usage |
|---|---|---|---|
| 5V | 5.0V | 500mA(USB),Higher(External) | Power external 5V components |
| 3.3V | 3.3V | Up to 600mA | Power 3.3V sensors and modules |
| GND | 0V | N/A | Common ground reference |

**Power Distribution:**

Multiple GND Pins: Several ground pins distributed across the board for easy access

Regulated 3.3V: Onboard LDO regulator provides stable 3.3V from 5V input

Power Isolation: Proper power filtering to minimize noise and interference

## 2.6. COMMUNICATION PORTS AND EXPANSION CONNECTORS

### 2.6.1. DETAILS ON UART, SPI, I²C, AND CAN PORTS

The ESP32 V1.0 board provides dedicated connectors for major communication protocols, enabling easy interfacing with external devices and sensors.

**UART (Universal Asynchronous Receiver-Transmitter):**

Default Pins: GPIO16 (TX), GPIO17 (RX)

Purpose: Serial communication with other microcontrollers, sensors, or computers

Features: Asynchronous, full-duplex communication without clock signal

Baud Rates: Supports standard baud rates from 9600 to 256000 bps

Applications: GPS modules, Bluetooth modules, sensor communication

**SPI (Serial Peripheral Interface):**

Pin Assignment: GPIO15 (CS), GPIO12 (MISO), GPIO13 (MOSI), GPIO14 (SCK)

Purpose: High-speed synchronous communication with peripherals

Features: Master-slave architecture with dedicated clock line

Speed: Supports high-speed data transfer for demanding applications

Applications: SD cards, displays, memory devices, sensors requiring fast data transfer

**I²C (Inter-Integrated Circuit):**

Pin Assignment: GPIO22 (SCL), GPIO21 (SDA)

Purpose: Multi-device communication bus for sensors and peripherals
Features: Two-wire interface supporting multiple devices on single bus
Addressing: 7-bit or 10-bit device addressing
Speed: Standard mode (100 kHz) and Fast mode (400 kHz)
Applications: Temperature sensors, accelerometers, EEPROMs, RTCs

**CAN (Controller Area Network):**
Pin Assignment: GPIO26 (TX), GPIO27 (RX)
Purpose: Robust communication for automotive and industrial applications
Features: Error detection, automatic retransmission, priority-based arbitration
Speed: Up to 1 Mbps for high-speed applications
Applications: Automotive systems, industrial automation, distributed control systems

### 2.6.2. POWER AND GROUND CONNECTOR LAYOUT

The board provides organized power distribution headers for external components.
**Power Header Organization:**
5V Rail: Multiple 5V pins for powering external modules requiring higher voltage
3.3V Rail: Regulated 3.3V supply for sensors and low-power devices
Ground Rails: Multiple ground connections distributed across expansion areas
Current Ratings: Each power rail designed to handle typical sensor and module requirements
Pin Mapping Tables for Communication Protocols
Communication Protocol Pin Mapping:

| Protocol | Signal | GPIO Pin | Function | Connector |
|---|---|---|---|---|
| UART | TX | GPIO16 | Transmit Data | P4 |
| UART | RX | GPIO17 | Receive Data | P4 |
| SPI | CS | GPIO15 | Chip Select | P5 |
| SPI | MISO | GPIO12 | Master In Slave Out | P5 |
| SPI | MOSI | GPIO13 | Master Out Slave In | P5 |
| SPI | SCK | GPIO14 | Serial Clock | P5 |
| 12C | SCL | GPIO22 | Serial Clock Line | P3 |
| 12C | SDA | GPIO21 | Serial Data Line | P3 |
| CAN | TX | GPIO26 | CAN Transmit | P4 |
| CAN | RX | GPIO27 | CAN Receive | P4 |

## 2.7. PRECAUTIONS

### 2.7.1. USE OF DATA-CAPABLE USB CABLE FOR STABLE CONNECTION

**Cable Selection Importance:**

- Data Lines Required: Programming requires USB cables with all four conductors (power and data)
- Charging-Only Limitation: Some USB cables only include power lines and cannot transfer data
- Quality Matters: Poor quality cables can cause intermittent connection problems during programming
- Length Considerations: Shorter cables provide more reliable communication and less voltage drop

**Cable Testing:**

- Connection Test: Verify board appears in Device Manager or system information when connected
- Programming Test: Attempt simple sketch upload to confirm data capability
- Serial Monitor: Check if serial communication works properly with connected cable

### 2.7.2. DRIVER INSTALLATION PRECEDENCE BEFORE CONNECTING BOARD

**Installation Sequence:**

- Driver First: Always install CH340 drivers before connecting the ESP32 board
- Avoid Recognition Issues: Connecting before driver installation may cause Windows to install incorrect drivers
- Clean Installation: If problems occur, uninstall existing drivers before installing correct ones
- System Restart: Restart computer after driver installation for proper recognition

**Prevention of Issues**:

- Proper Sequence: Download → Install Drivers → Restart → Connect Board
- Device Manager Check: Verify proper driver installation before first use
- Multiple Computers: Install drivers on each computer that will be used for programming

### 2.7.3. SAFE POWER SUPPLY PRACTICES TO AVOID CONFLICTS AND HARDWARE DAMAGE

**Power Supply Guidelines:**

Single Source: Use either USB or external power, never both simultaneously

Voltage Limits: Respect 3.3V operating voltage of ESP32 chip

Current Capacity: Ensure power supply can provide adequate current for application
External Power Range: Use 7-12V for external DC input, avoid exceeding specifications
**Damage Prevention:**
- Reverse Polarity: Double-check polarity when using external power supplies
- Overvoltage Protection: Use regulated power supplies to prevent voltage spikes
- Current Limiting: Implement current limiting for experimental circuits
- Ground Loops: Maintain proper grounding practices to avoid ground loops

### 2.7.4. RECOMMENDATIONS ON BREADBOARDING AND PIN USAGE

**GPIO Pin Considerations**:
- Input-Only Pins: GPIO34-39 are input-only and cannot be used as outputs
- Strapping Pins: Be careful with GPIO0, GPIO2, GPIO12, and GPIO15 as they affect boot behavior
- Internal Flash Pins: Avoid using GPIO6-11 as they connect to internal SPI flash
- Safe Pins: GPIO4, GPIO5, GPIO18-19, GPIO21-23, GPIO25-27, GPIO32-33 are generally safe for most applications

**Bread-boarding Best Practices:**
- Stable Connections: Ensure reliable connections to prevent intermittent operation
- Power Distribution: Use breadboard power rails for clean power distribution
- Wire Management: Keep wires organized and as short as practical
- Component Placement: Place components logically to minimize crossing wires

**Pin Assignment Strategy:**
- Communication Protocols: Reserve specific pins for $I^2C$, SPI, UART as needed
- Analog Inputs: Use ADC-capable pins for sensor reading (GPIO32-39)
- PWM Outputs: Most GPIO pins support PWM for motor control and LED dimming
- Interrupt Capability: Any GPIO pin can be configured for interrupt input

### 2.7.5. TROUBLESHOOTING TIPS FOR COMMON SETUP ISSUES

**Programming Failure Solutions:**
- "Failed to Connect" Error: Hold Boot button during upload process
- Port Not Found: Check driver installation and cable connection
- Upload Timeout: Try different USB ports or cables
- Board Not Recognized: Reinstall CH340 drivers and restart computer

**Power-Related Issues:**
- Brown-out Detector: Inadequate power can trigger brown-out reset
- USB Power Insufficient: Use external power for high-current applications

- Voltage Fluctuations: Add decoupling capacitors near power-hungry components
- Ground Issues: Ensure solid ground connections throughout the circuit

**Code Execution Problems:**

- Infinite Reset Loop: Check for conflicting pin usage or hardware shorts
- Erratic Behavior: Verify power supply stability and adequate current capacity
- GPIO Conflicts: Review pin assignments for conflicts with strapping pins
- Serial Monitor Issues: Check baud rate settings and driver installation

**Best Practices Summary:**

- Systematic Approach: Follow installation and setup procedures in correct sequence
- Quality Components: Use reliable cables, power supplies, and breadboarding materials
- Documentation: Keep track of pin assignments and circuit modifications
- Testing: Test each component and connection systematically during development
- Safety First: Always power down before making hardware connections or modifications

# 3. ESP WROOM32 MODULE TECHNICAL SPECS AND ROLE ON ESP32-V1.0 BOARD

## 3.1. MODULE OVERVIEW

The ESP WROOM32 is a compact, industry-standard module integrating Espressif's ESP32-series system-on-chip, providing rich wireless and processing capabilities in a small footprint.

- Processor Architecture: Dual-core 32-bit Xtensa® LX6 CPU, up to 240 MHz.
- Wireless Standards:
  – Wi-Fi 802.11 b/g/n (up to 150 Mbps)
  – Bluetooth v4.2 BR/EDR & BLE (up to 2 Mbps)
- Memory:
  – 520 KB internal SRAM
  – 4 MB external SPI flash
- Peripherals: ADC, DAC, SPI, I²C, UART, PWM, I²S, CAN, capacitive touch, SD-card interface, Ethernet MAC.

## 3.2. MODULE PINOUT AND SIGNAL FUNCTIONS

The ESP WROOM32 exposes 38 pins (49 pads) for power, ground, I/O, and interfaces. Key signal functions include:

| Pin Group | Typical Signals |
|---|---|

| | |
|---|---|
| Power & Ground | 3V3, VDD_SDIO, GND |
| Digital I/O | GPIO0-GPIO39(input/output, PWM, touch) |
| ADC Channels | GPIO32-GPIO39(12-bit ADC) |
| SPI Bus | GPIO6-GPIO11(flash), GPIO12-GPIO15, GPIO18-GPIO23 |
| $I^2C$ Bus | GPIO21(SDA), GPIO22(SCL) |
| UART | GPIO1(TX0),GPIO3(RX0),GPIO16-GPIO17 |
| CAN Bus | GPIO26(TX), GPIO27(RX) |
| Bluetooth | Internal antenna via antenna pad |

Each pin is multiplexed via the GPIO matrix, allowing flexible assignment to peripherals while respecting boot strapping pins (GPIO0, GPIO2, GPIO15) and input-only pins (GPIO34–GPIO39).

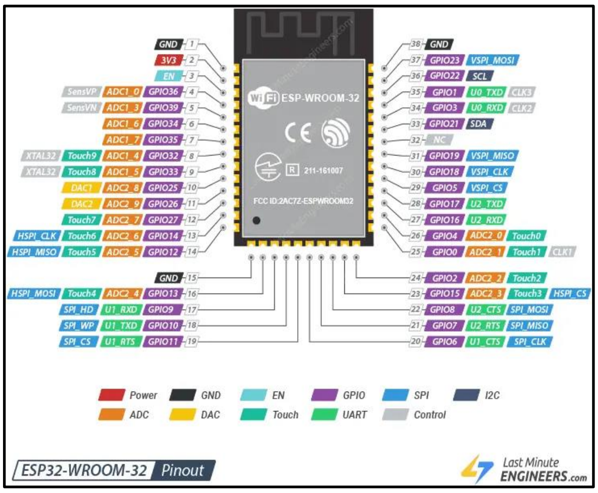
## 3.3. ROLE OF MODULE ON ESP32-V1.0 BOARD

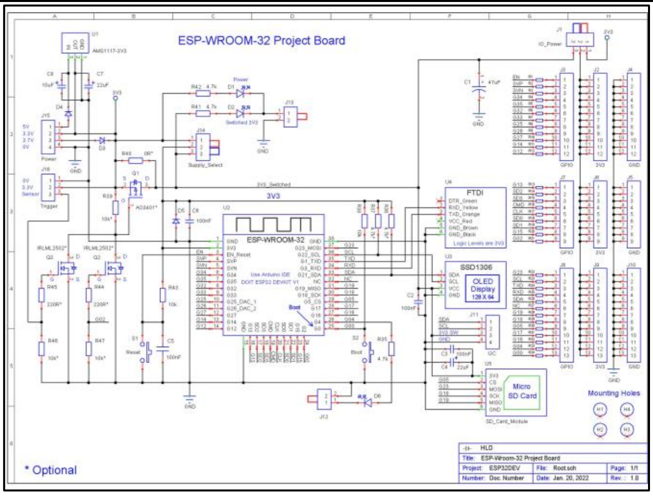
On the ESP32-V1.0 development board, the ESP WROOM32 module serves as the central processing & communication hub:

- Processing Core: Executes user firmware, RTOS tasks, and network stacks (LwIP, FreeRTOS).
- Wireless Connectivity: Provides Wi-Fi and BLE interfaces for IoT applications.
- GPIO Interface: Routes digital, analog, and peripheral signals through board-mounted headers and jumper-configurable connectors.
- Modularity: The module plugs into castellated pads, enabling replacement with alternative ESP32-series modules (e.g., ESP32-S2, ESP32-C3) without redesigning the PCB.
- Peripheral Integration: Directly interfaces with onboard LEDs, push buttons, sensors (POT, LDR), CH340 converter, and expansion connectors for UART, SPI, $I^2C$, and CAN devices.

This modular integration streamlines board design and allows reuse of the same base PCB for multiple ESP32-module variants, facilitating rapid prototyping and production scalability.

## 3.4. EXTERNAL DATASHEETS AND TECHNICAL MANUALS






For in-depth electrical and mechanical details, refer to Espressif's official resources:

- ESP32-WROOM-32 Datasheet (v3.6): https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf
- ESP32 Technical Reference Manual (v5.5): https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf

These documents provide complete pin assignments, electrical characteristics, RF performance data, mechanical dimensions, and manufacturing guidelines. When cross-referencing with the ESP32-V1.0 board documentation, verify that any custom pull-up/down resistors or signal routing on the PCB align with module requirements to ensure correct boot mode behavior and signal integrity.

# 4. CONCLUSION

## 4.1. SUMMARY OF ESP32-V1.0 BOARD LAYOUT, JUMPER SETTINGS, POWER, AND COMMUNICATION PORTS

The ESP32-V1.0 development board represents a comprehensive educational and prototyping platform that integrates sophisticated hardware features with student-friendly design principles. The board's well-organized layout provides immediate access to essential components while maintaining the flexibility needed for advanced projects.

**Board Layout Excellence:**
The PCB design prioritizes both functionality and educational value, featuring clearly labeled GPIO headers, strategically positioned onboard components, and logical component grouping. The horizontal orientation and standardized connector placement facilitate easy breadboarding and external device connections, making it ideal for laboratory environments and student experimentation.

**Flexible Jumper Configuration System:**
The innovative jumper-based configuration system distinguishes this board from typical ESP32 development platforms. Students can easily reconfigure default pin assignments for LEDs (GPIO33, GPIO4), push button (GPIO32), potentiometer (GPIO36), and LDR sensor (GPIO39). This flexibility enables experimentation with different GPIO assignments while teaching fundamental concepts about pin multiplexing and hardware resource management.
**Comprehensive Power Management:**
The dual power input options (USB-C and external DC jack) accommodate various project scenarios from classroom demonstrations to standalone battery-powered applications. The onboard voltage regulation provides stable 3.3V and 5V rails with adequate current capacity for typical educational projects and peripheral interfacing.
**Professional Communication Interfaces:**
Dedicated connectors for UART, $I^2C$, SPI, and CAN protocols provide hands-on experience with industry-standard communication methods. The organized header layout with clear pin labeling reduces connection errors while teaching students about different communication protocols and their applications in IoT systems.

## 4.2. EMPHASIS ON THE CRITICAL ROLE OF ESP WROOM32 MODULE FOR IOT DEVELOPMENT

The ESP32-WROOM-32 module serves as the technological foundation that enables the board's advanced IoT capabilities. Understanding this module's role is crucial for students developing expertise in modern embedded systems and IoT applications.
**Processing Power and Architecture:**
The dual-core 32-bit Xtensa LX6 processor operating at up to 240 MHz provides computational resources comparable to many single-board computers. This processing capability enables students to implement complex algorithms, real-time data processing, and multitasking applications that mirror industrial IoT systems.

**Wireless Connectivity Foundation:**
The integrated Wi-Fi 802.11 b/g/n and Bluetooth v4.2 BLE capabilities transform simple microcontroller projects into connected IoT applications. Students can progress from basic embedded programming to cloud-connected systems, learning essential concepts about wireless communication, network protocols, and remote device management.

**Memory Architecture Understanding:**
The combination of 520 KB SRAM and 4 MB flash memory provides practical experience with memory management in resource-constrained environments.

Students learn to optimize code size, manage dynamic memory allocation, and implement efficient data storage strategies essential for professional embedded development.

**Peripheral Integration Capabilities:**
The module's GPIO matrix system and built-in peripheral controllers (ADC, DAC, PWM, UART, SPI, $I^2C$) demonstrate modern microcontroller architecture. Students gain experience with hardware abstraction layers, peripheral initialization, and interrupt-driven programming essential for embedded systems development.

## 4.3. IMPORTANCE OF UNDERSTANDING BOTH BOARD AND MODULE TO FULLY UTILIZE THE PLATFORM

Effective utilization of the ESP32-V1.0 platform requires comprehensive understanding of both the development board features and the underlying ESP32-WROOM-32 module capabilities. This dual knowledge approach ensures students develop both practical skills and theoretical understanding.

**Hardware-Software Integration:**
Understanding the board's physical layout and jumper system enables students to make informed decisions about pin assignments and peripheral connections. Combined with knowledge of the module's capabilities, students can design projects that efficiently utilize available resources while avoiding common pitfalls like boot pin conflicts or ADC channel limitations.

**Scalability Considerations:**
Knowledge of both development board convenience features and module-level specifications prepares students for transitioning from prototypes to production systems. Students understand when to use development boards for rapid prototyping and when custom PCB designs with direct module integration become necessary for commercial applications.

**Troubleshooting and Optimization:**
Comprehensive platform knowledge enables effective debugging and performance optimization. Students can distinguish between board-level issues (power supply, jumper configuration) and module-level problems (GPIO conflicts, memory limitations), leading to more efficient problem-solving approaches.

**Professional Development Skills:**
Understanding the relationship between development boards and production modules mirrors industry practices where engineers progress from evaluation boards to custom implementations. This knowledge prepares students for professional embedded systems development where component selection and system architecture decisions significantly impact project success.

## 4.4. FOUNDATION LAID FOR FURTHER PROGRAMMING AND IOT PROJECT DEVELOPMENT

The ESP32-V1.0 board establishes a solid foundation for progressive learning in embedded systems and IoT development, enabling students to advance from basic concepts to industry-relevant applications.

**Programming Skill Development:**

The board supports multiple programming environments and languages, from Arduino IDE's beginner-friendly C++ environment to advanced ESP-IDF development. Students can progress naturally from simple LED blinking exercises to complex IoT applications involving wireless communication, sensor data processing, and cloud integration.

**Project Scalability Path:**

Starting with onboard components (LEDs, buttons, sensors), students can gradually expand to external peripherals, communication modules, and complete IoT systems. The board's expansion connectors and jumper system facilitate this progression without requiring immediate breadboarding expertise.

**Industry-Relevant Technology Exposure:**

Experience with ESP32 architecture, wireless protocols, and communication interfaces directly translates to professional IoT development. Students gain familiarity with technologies widely used in commercial IoT products, smart home systems, industrial automation, and edge computing applications.

**Research and Innovation Opportunities:**

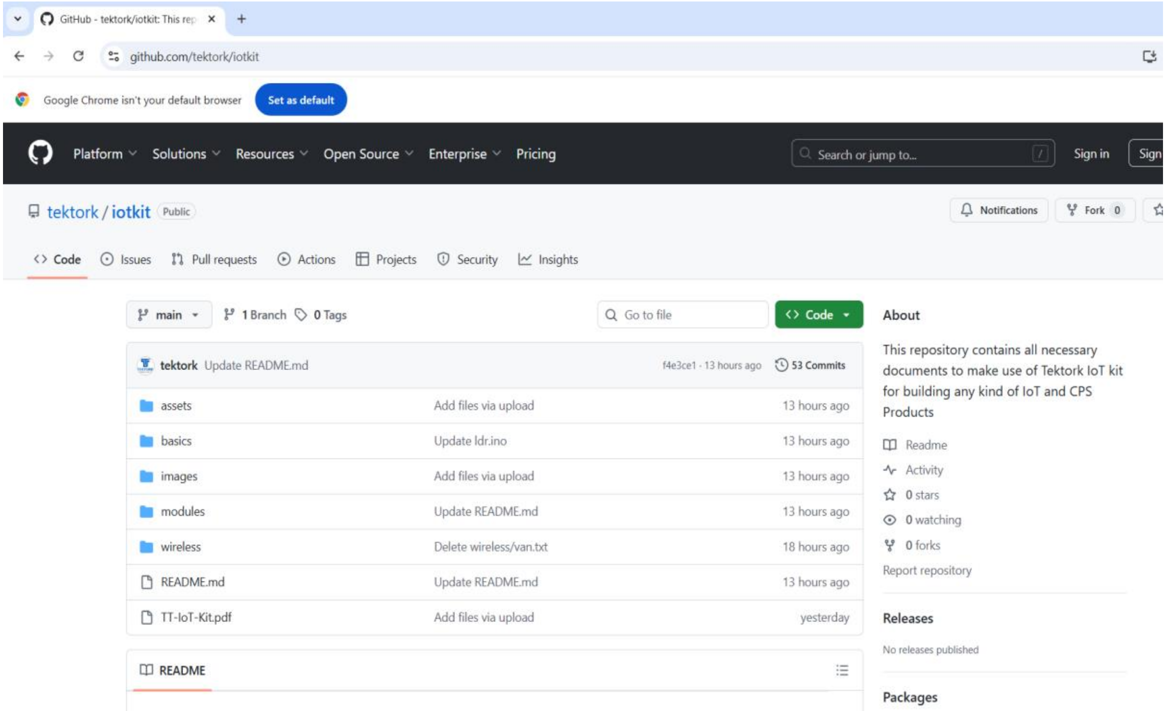
The platform's capabilities enable advanced projects in areas like machine learning at the edge, secure IoT implementations, and distributed sensor networks. Students can explore cutting-edge topics while building practical experience with real hardware systems.

**Educational Continuity:**

The comprehensive documentation, sample programs, and community support create an ecosystem that supports continued learning beyond formal coursework. Students can access progressive tutorials, contribute to open-source projects, and engage with the broader ESP32 development community.

The Tektork IoT -V1.0 development board successfully bridges the gap between academic learning and professional development, providing students with both the theoretical knowledge and practical skills necessary for success in embedded systems and IoT engineering careers. Its thoughtful design, comprehensive feature set, and educational focus make it an ideal platform for building the next generation of embedded systems engineers and IoT developers.

**Sample Programs.**




https://github.com/tektork/iotkit
LED Blink at https://github.com/tektork/iotkit/tree/main/basics/led_blink
Button and LED at https://github.com/tektork/iotkit/tree/main/basics/button
Analog Input (POT) and Serial Output at https://github.com/tektork/iotkit/tree/main/basics/potentiometer
Light Sensor LDR and Serial Output at https://github.com/tektork/iotkit/tree/main/basics/ldr