ESP32 Universal IoT Dev Kit - Actuating

IoT Actuator Kit

Abstract

This is an ESP-32 based dev kit/IoT unit mainly designed for connecting actuators to IoT based systems. It is ideal for researchers, engineers, and hobbyists in IoT and embedded system development. It is versatile enough to be used in R&D and perfect for long-running applications and systems. This document explains the components available in the module and their utilization. The users are advised to go through this booklet before using a board.

ThingsInNet

Innovate. Connect. Empower

Contents

1: Introduction

This unit is an ESP32-based dev kit/IoT unit mainly designed to connect actuators to IoT-based systems. It has four integrated relay switches to control devices. Further, it has a range of I/O pins for connecting sensors or actuators according to user requirements. The OLED display, buzzer and four push buttons are available for standard input and output. The unit can be programmed and employed using Arduino IDE, MicroPython and other Espressif chips compatible platforms such as ESP IDF. The block diagram of the module is shown below. This module is ideal for researchers, engineers, and hobbyists in IoT and embedded system development. It is versatile enough to be used in R&D and perfect for long-running applications and systems.

2: Block Diagram

Figure 1 shows the overall system design and other options available in the IoT dev kit for actuating.

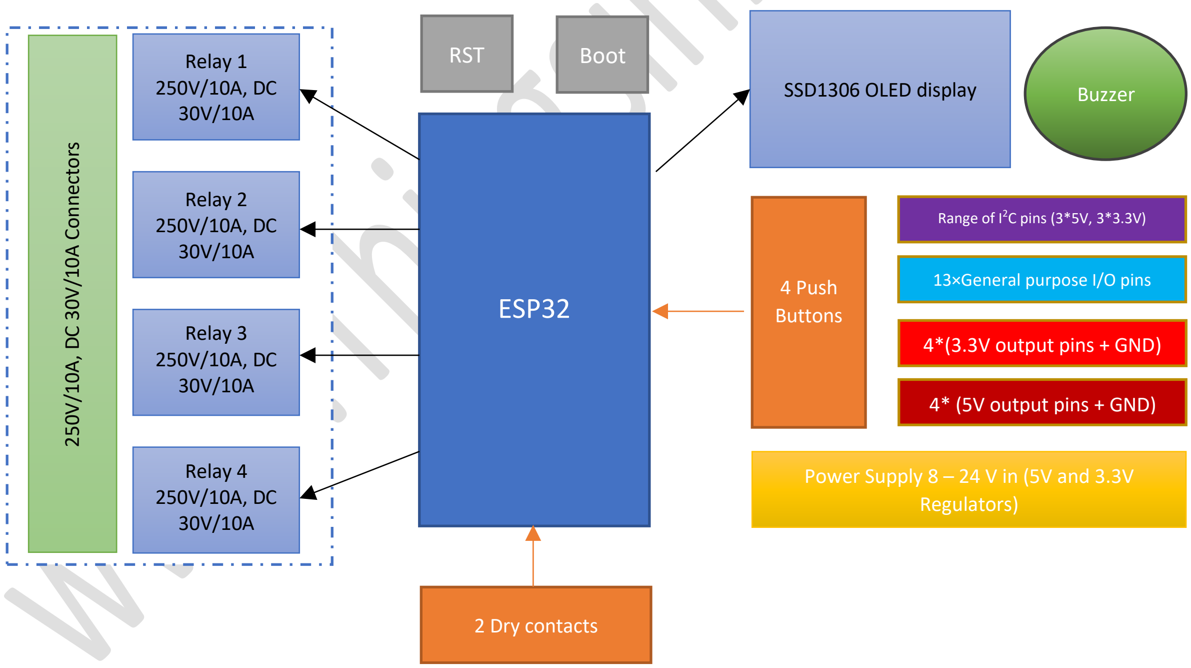

Figure 1: Block diagram of the module

3: IoT Module Components

This section explains the MCU, relays, expanded I/O pins and other modules available in the IoT dev kit.

3.1: Microcontroller Unit (MCU)

This board is based on the ESP-WROOM-32D-N8 microcontroller, which is integrated with Wi-Fi and Bluetooth capabilities. Hence, it is one of the ideal microcontrollers for IoT applications. Further, ESP-32 is a robust, reliable, and low-power device that could be employed in both R&D and industrial applications.

3.2: Relays

The IoT dev kit has four integrated relays for driving devices and four push buttons for Input. Each relay module is 250V/10A, DC 30V/10A. The MCU and relays are entirely opto isolated to protect the MCU pins from back emf. More information about the placement of relays and pushbuttons available on the board is shown below.

SN#	Device	I/O Pins in MCU		Comment
		Label	Pin	
1	Relay 1	RLY1	GPIO25	Active Low
2	Relay 2	RLY2	GPIO14	Active Low
3	Relay 3	RLY3	GPIO16	Active Low
4	Relay 4	RLY4	GPIO17	Active Low
5	Pushbutton 1	S1	GPIO34	Active High
6	Pushbutton 2	S2	GPIO35	Active High
7	Pushbutton 3	S3	GPIO39	Active High
8	Pushbutton 4	S4	GPIO36	Active High

Table 1: Relays and pushbuttons

3.3: USB-to-UART Bridge

CP2102 USB chip from Silicon Labs has been employed in this board to enable auto switching between ESP-32 boot modes.

3.4: OLED Display

The OLED display from JMD Electronics is available on board. It is 0.96 inches in dimensions, 128×64 resolution and white. It supports I2C communication and drives by SSD1306 driver IC.

3.5: Input Pushbuttons

There are six pushbuttons available on the dev kit, one for Rest, one for ESP-32 Boot and another four for standard Input. More information (I/O pins) about the Input pushbuttons is available in Table 1.

3.6: Buzzer

The buzzer available on the board can be employed to Output warnings. It is connected to GPIO26 of the ESP-32.

3.7: LEDs

Two LED indicators are available on the board. LD1 is the power indicator and LD2 can be used for user defined indications such as status of the Wi-Fi connectivity. LD2 connected to GPIO2 of the ESP-32.

3.8: Expanded I/O Pins and Specifications

The board has a series of I/O pin headers for connecting any extra sensors or actuators according to the user requirements. The specifications of the available I/O and power supply pin headers are shown in Table 2.

<table>
<tr><th>Pin Label</th><th>Description</th><th colspan="2">Corresponding ESP32 Pin</th></tr>
<tr><td rowspan="2">I2C_5V</td><td rowspan="2">3×5V I^2C pins. Users can connect 5V I^2C modules using these headers.</td><td>SDA</td><td>GPIO21</td></tr>
<tr><td>SCL</td><td>GPIO22</td></tr>
<tr><td rowspan="2">I2C_3V3</td><td rowspan="2">3×3.3V I^2C pins. Users can connect 3.3V I^2C modules using these headers.</td><td>SDA</td><td>GPIO21</td></tr>
<tr><td>SCL</td><td>GPIO22</td></tr>
<tr><td>FL1</td><td>This port has inbuilt debouncing filter and can be utilized as a dry contact such as connecting a floating switch.</td><td colspan="2">GPIO32</td></tr>
<tr><td>FL2</td><td>This port has inbuilt debouncing filter and can be utilized as a dry contact such as connecting a floating switch.</td><td colspan="2">GPIO33</td></tr>
<tr><td>GPIOs</td><td>ESP32 GPIO pins for Input and Output</td><td colspan="2">Tx0, Rx0, 23,19,18,5,4, 0,13,12,27,2,15,GND</td></tr>
<tr><td>3V3_OUT</td><td>4×3.3V/200mA voltage output pins</td><td colspan="2"></td></tr>
<tr><td>5V_OUT</td><td>4×5V/600mA voltage output pins</td><td colspan="2"></td></tr>
</table>

Table 2: Expanded I/O Pins and Specifications

3.9: USB

There is a USB Type-C port available on board. It can be used to program the ESP-32 MCU and can be used to power up the module.

3.10: Power Supply

The input voltage to the board is 8 – 24 V. A standard power supply with 12V 1A can be used to power up the module.

3.11: Hard Reset

The GPIO13 pin has been wired to the RST pin of the MCU. Therefore, active HIGH GPIO13 could reset the ESP-32 microcontroller. To enable the feature, you may keep the JP1 closed. However, once you have enabled the hard resetting feature, the GPIO 13 cannot be used for other purposes.

3.12: Resettable Fuse

This board is equipped with a resettable fuse for over-current protection. It could protect the power output pins from short circuits.

3.13: Power Mode Selector Switch

This switch helps to switch the power supply between USB Type-C (USB Mode) and External power source (EXT Mode).

3.14: Jumpers (JP1 – JP4)

This section describes four different jumper settings available on the board.

JP1: You may close the jumper to enable hard reset feature in the board. For more information 3.11.

JP2-JP3: You may close JP2 and JP3 jumpers to powerup the relay modules directly from the on-board power supply.

JP4: To enable communication between the USB to UART bridge and the ESP32 module, JP4-1 and JP4-2 must be closed. This is necessary when uploading a program to the ESP32 through a computer. However, suppose you plan to use Tx0 and Rx0 on the GPIO pins. In that case, you can open the JP4 jumpers to avoid any conflicts between the two modes of communications.

4: Printed Circuit Board (PCB) Design

The dual-layer PCB has been printed using cutting-edge technologies and assembled using high-quality electronic components. Four relays are available on the board for driving devices, and a set of I/O pins for connecting other sensors and actuators according to the user preference.

4.1: Image

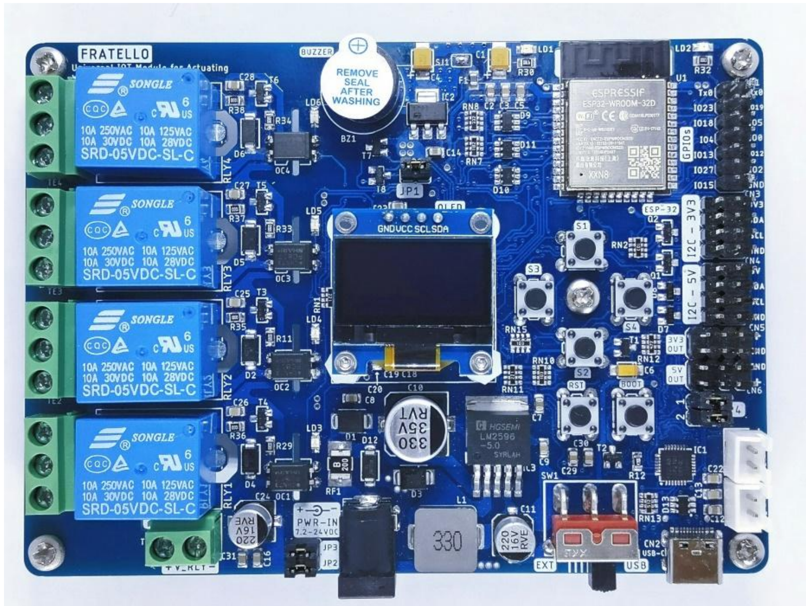

Figure 2: Image of the module

4.2: Layout

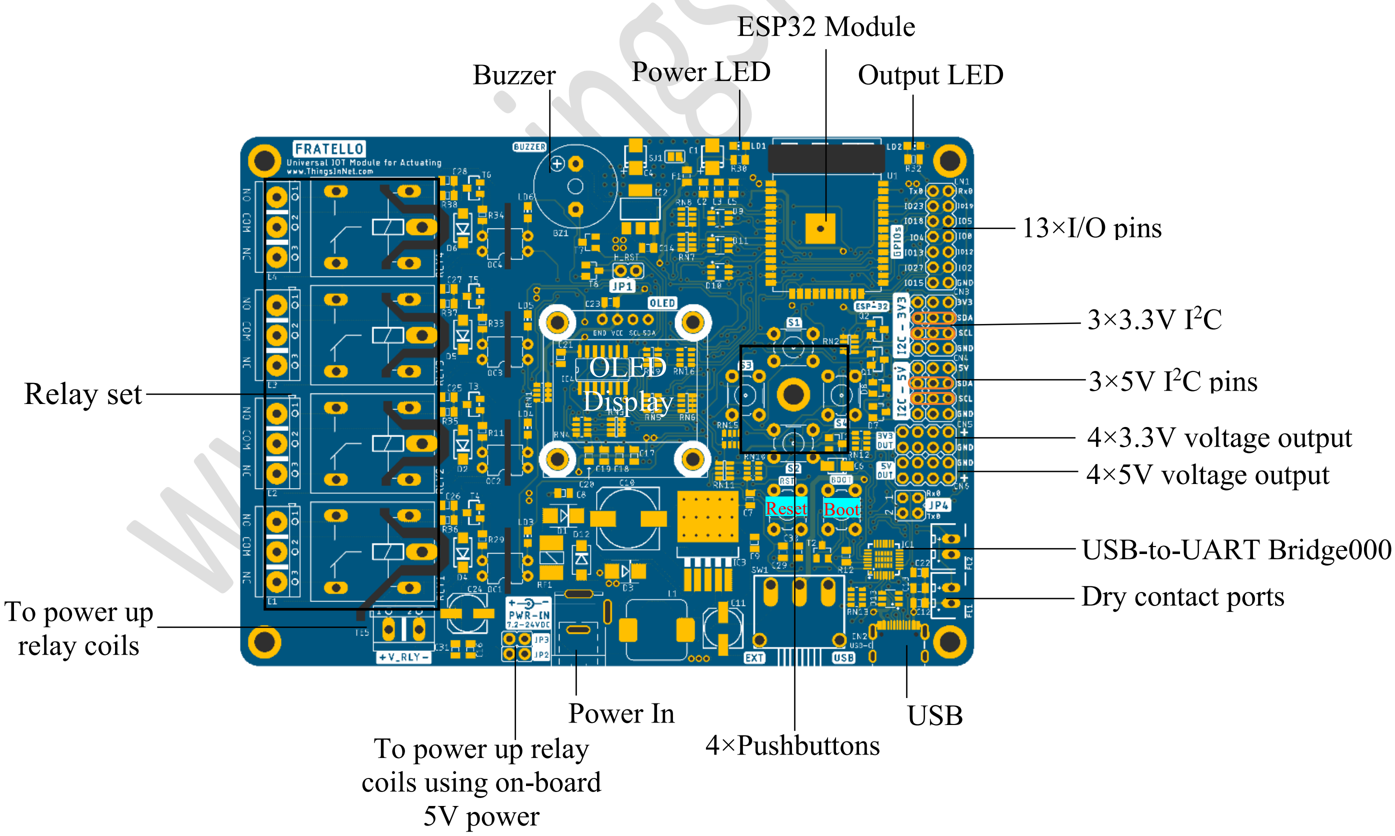

Figure 3: Layout of the module

4.3: Summary of the ESP-32 pin utilization and voltage output pins

SN#	Type	Component/Pin	Label	ESP-32 Pin
1	Relays	Relay 1	RLY1	GPIO25
2		Relay 2	RLY2	GPIO14
3		Relay 3	RLY3	GPIO16
4		Relay 4	RLY4	GPIO17
5	Pushbuttons	Pushbutton 1	S1	GPIO34
6		Pushbutton 2	S2	GPIO35
7		Pushbutton 3	S3	GPIO39
8		Pushbutton 4	S4	GPIO36
9	Dry contact ports	Dry contact 1	FL1	GPIO32
10		Dry contact 2	FL2	GPIO33
11	Onboard output	OLED Display	OLED	SDA GPIO21 SCL GPIO22
12		Buzzer	BUZZER	GPIO26
13		LED 2	LD2	GPIO2
14	Expanded I/O Pins	3×5V I^2C pins	I2C_5V	SDA GPIO21 SCL GPIO22
15		3×3.3V I^2C pins	I2C_3V3	SDA GPIO21 SCL GPIO22
16		ESP32 GPIO pins for Input and Output	GPIOs	Tx0, Rx0, 23,19, 18,5,4,0,13,12,27, 2,15, GND
17	Voltage output	4×3.3V/200mA	3V3_OUT	
18		4×5V/300 - 600mA	5V_OUT	

Table 3: Summary of the ESP-32 pin utilization and voltage output pins

4.4: Schematic diagram

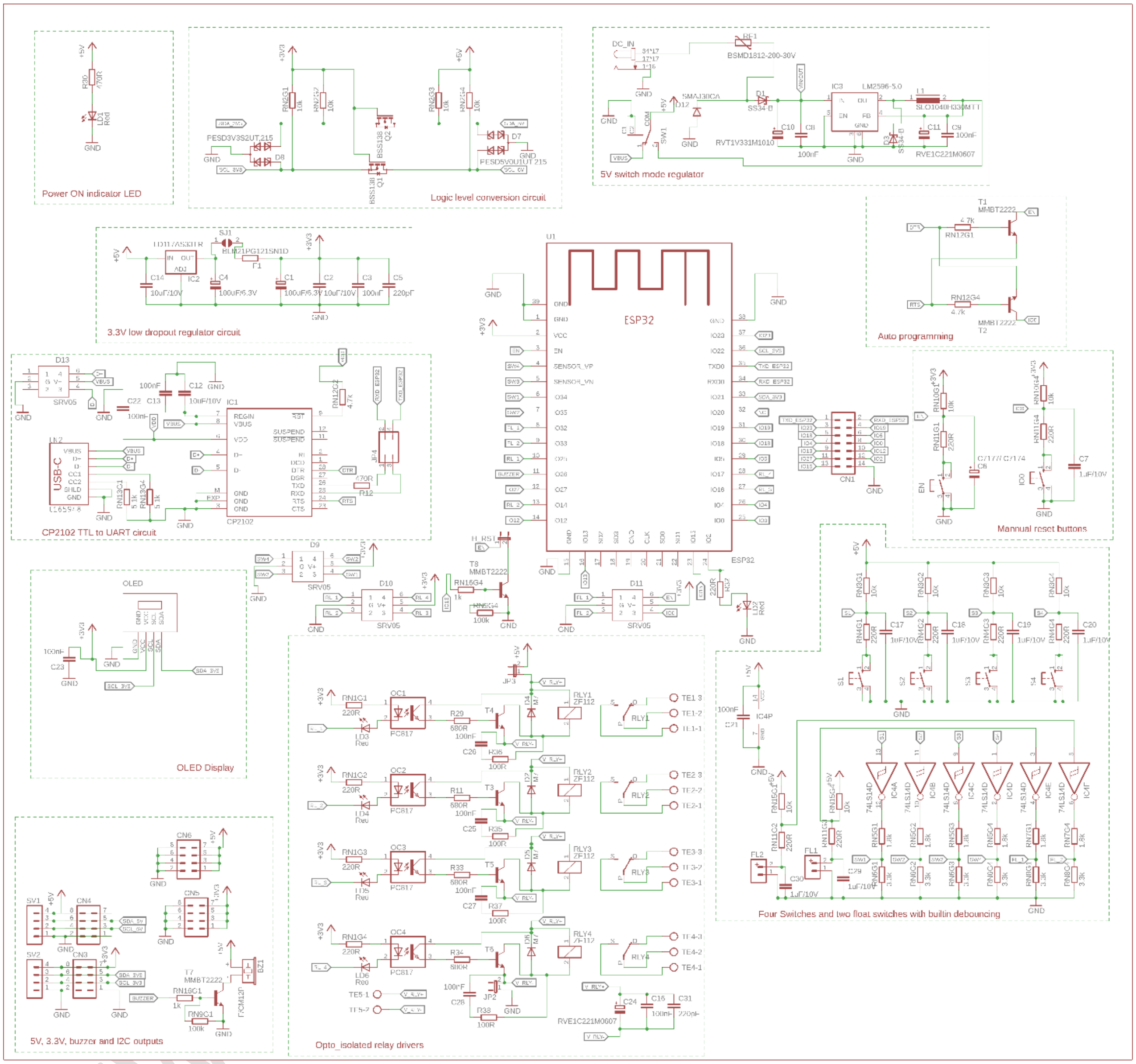

Figure 4: Schematic diagram of the module

5: Product Specifications

Name	Universal IoT Dev Kit for Actuating
MPN	GS3 ESP-32 IoT Unit for Actuating
Marketing status	Mass production
Temp (°C)	–40 ~ 85
Flash (MB)	4 MB
PSRAM (KB)	520 KB
Size (mm)	122×88
Antenna	PCB
Interfaces	I/O, I^2C, Dry contact, USB Type-C

Table 4: Product specifications

6: Applications of the Board

This board can be employed in a wide range of applications, from R&D to industrial applications where IoT and embedded systems are in place. Even though a set of application domains has been listed below, the module can also be utilized in many other fields.

1. Smart farming
2. Smart cities
3. Weather stations
4. Environmental monitoring systems
5. Autonomous robots
6. Security systems
7. Research
8. Teaching and learning

7: Special Notes

a. Even though the power drains through the USB is sufficient enough to function properly, it is always advisable to use the external power supply for both programming and normal operational modes.

b. The voltage and current specifications and GPIO properties of the I/O pins available in this module follow the original ESP-32 specifications.

c. If there is any requirement to have Analog to Digital conversion with the Dev Kits, we strongly recommend using ADS1115 ADC module which can easily connect with I2C bus.

d. Any sensor or component can be upgraded or removed as per client requests. If you want any customized versions of the IoT units e-mail contact@thingsinnet.com

8: Setup the board and tryout some samples

The contents, settings and programs in the third-party links below have been tested on the Dev Kit by the developer team (20/12/2022).

Step 1: Download and install the **USB drivers** here.
CP2102 USB-to-UART Bridge from Silicon Labs:
https://www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers?tab=downloads

Step 2: Installing the ESP32 Board in Arduino IDE (Windows, Mac OS X, Linux)
https://randomnerdtutorials.com/installing-the-esp32-board-in-arduino-ide-windows-instructions/

Step 3: In the Arduino IDE select the **ESP32 Dev Module**

Step 4: You may visit ThingsInNetBlog for examples from the ThingsInNet team

You may try out any simple examples available on the web and in the Arduino IDE. The development team will periodically update this section.

9: Copyright Notice

The content of this document, contents in the associated links and the dev kit layout and other options can be changed without prior notifications.

No warranty is provided for either hardware or software components.

The PCB design and the Schematic diagram of the dev kit module presented in this document has been submitted for copyright by ThingsInNet. All rights reserved. The PCB design and the Schematic diagram cannot be copied or duplicated as a whole or in part by any means without express prior agreement in writing or unless specifically noted in the documentation.

All the protocols, technologies, trademarks, and other third-party resources including URLs mentioned in this document are properties of their respective owners and are hereby acknowledged.