CAN Bus Development with ESP32-WROOM32 Development Board
Copperhill Technologies – https://copperhilltech.com

This post will demonstrate how to add a CAN Bus port to the ESP32-WROOM32 development board, i.e., regarding hardware and software. As a matter of fact, we already offer a hardware utilizing the ESP32 processor and an onboard CAN Bus transceiver as shown in the image below:

The ESP32 WiFi, Bluetooth Classic, BLE, CAN Bus Module comes with an onboard ESP32 WROOM-32 WiFi, Bluetooth Classic, BLE Module, and a CAN Bus port with a transceiver. Also, onboard is an RGB LED, IO pins on a 0.1" pad.

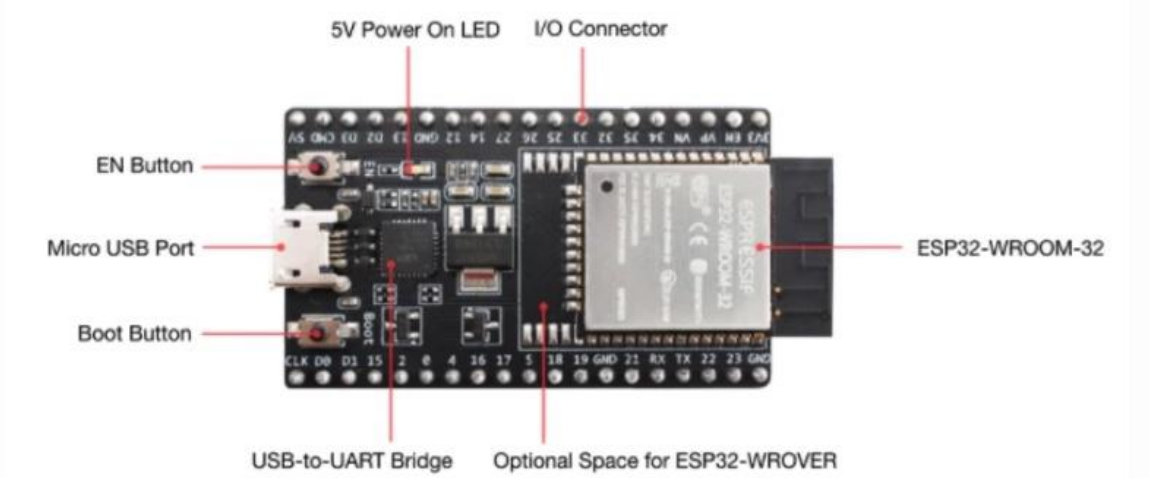
The programming is accomplished through the popular Arduino IDE connected to the USB-to-Serial converter with USB-C connector, automatic bootloader and reset.

However, if you don't need the supported 12 VDC power connection and extended temperature range (specifically suited for automotive and industrial applications) but are satisfied with the power supplied per USB port, you may prefer the lower-priced application as described here.

For this project we are using:

- ESP32-WROOM-32D Bluetooth, BLE, And WIFI Development Board with PCB Antenna
- CAN Bus Mini Breakout Board

The ESP32 is a series of low-cost, low-power system-on-chip microcontrollers with integrated Wi-Fi and dual-mode Bluetooth. The ESP32 series employs a Tensilica Xtensa LX6 microprocessor in both dual-core and single-core variations and includes built-in antenna switches, RF balun, power amplifier, low-noise receive amplifier, filters, and power management modules. It is a successor to the ESP8266 SoC.




*ESP32-DevKitC V4 with ESP32-WROOM-32 module soldered*

| Key Component | Description |
|---|---|
| ESP32-WROOM-32 | A module with ESP32 at its core. |
| EN | Reset button. |
| Boot | Download button. Holding down **Boot** and then pressing **EN** initiates Firmware Download mode for downloading firmware through the serial port. |
| USB-to-UART Bridge | Single USB-UART bridge chip provides transfer rates of up to 3 Mbps. |
| Micro USB Port | USB interface. Power supply for the board as well as the communication interface between a computer and the ESP32 module. |
| 5V Power On LED | Turns on when the USB or an external 5V power supply is connected to the board. |
| I/O | Most of the pins on the ESP module are broken out to the pin headers on the board. You can program ESP32 to enable multiple functions such as PWM, ADC, DAC, I2C, I2S, SPI, etc. |

The ESP32 DevKit Board contains the ESP-WROOM-32 as the main module and some additional hardware to easily program ESP32 and connect with the GPIO Pins.

### *ESP32 Features*

- Single or Dual-Core 32-bit LX6 Microprocessor with clock frequency up to 240 MHz.
- 520 KB of SRAM, 448 KB of ROM and 16 KB of RTC SRAM.
- Supports 802.11 b/g/n Wi-Fi connectivity with speeds up to 150 Mbps.
- Support for both Classic Bluetooth v4.2 and BLE specifications.
- 34 Programmable GPIOs.
- Up to 18 channels of 12-bit SAR ADC and 2 channels of 8-bit DAC
- Serial Connectivity include 4 x SPI, 2 x I2C, 2 x I2S, 3 x UART.
- Ethernet MAC for physical LAN Communication (requires external PHY).
- 1 Host controller for SD/SDIO/MMC and 1 Slave controller for SDIO/SPI.
- Motor PWM and up to 16-channels of LED PWM.
- Secure Boot and Flash Encryption.
- Cryptographic Hardware Acceleration for AES, Hash (SHA-2), RSA, ECC and RNG.
-

### *ESP32 Board Components*

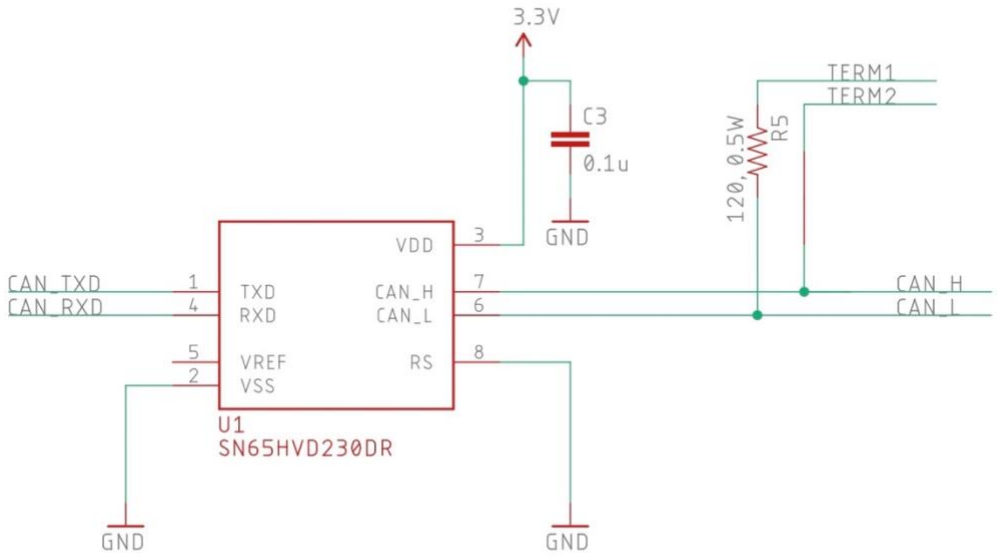
- ESP-WROOM-32 Module
- Two rows of IO Pins (with 15 pins on each side)
- CP2012 USB – UART Bridge IC
- Micro–USB Connector (for power and programming)
- AMS1117 3.3V Regulator IC
- Enable Button (for Reset)
- Boot Button (for flashing)
- Power LED (Red)
- Some passive components

## CAN Bus Connection

The ESP32 integrates a CAN Bus controller compatible with the NXP SJA1000. Thus, it is CAN 2.0B specification compliant.

As with the SJA1000, the ESP32 CAN Bus controller provides only the data link layer and the physical layer signaling sublayer. Therefore, an external transceiver module is required, which converts the CAN-RX and CAN-TX signals of the ESP32 into CAN_H and CAN_L bus signals. The transceiver, such as the MCP2551 or SN65HVD23X, provides compatibility with ISO 11898-2.

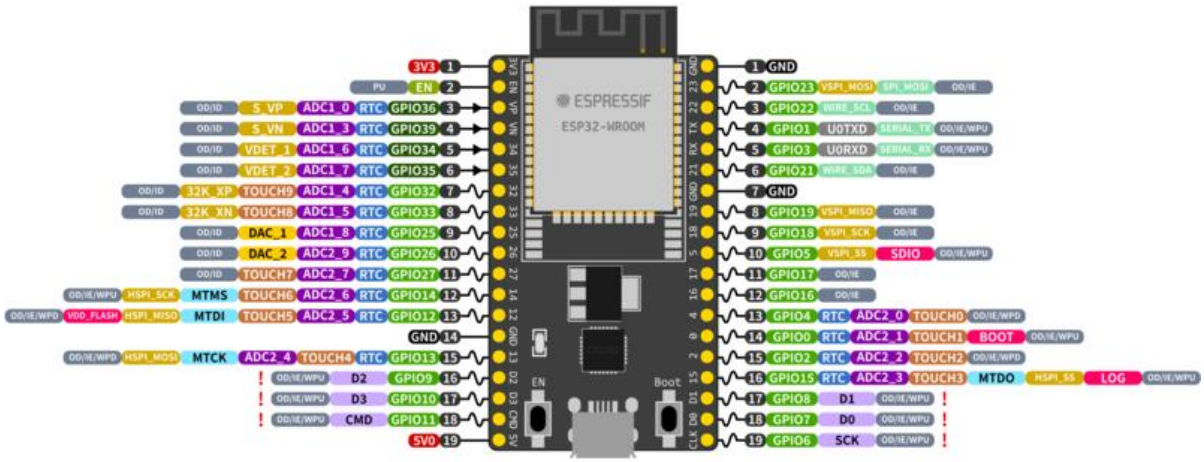
***Note:***
*The SJA1000 does not support CAN-FD and is not CAN-FD tolerant.*




CAN_TXD = ESP32 – IO25
CAN_RXD = ESP32 – IO26

ESP32-DevKitC

ESPRESSIF




**ESP32 Specs**

32-bit Xtensa® dual-core @240MHz
Wi-Fi IEEE 802.11 b/g/n 2.4GHz
BLuetooth 4.2 BR/EDR and BLE
520 KB SRAM (16 KB for cache)
448 KB ROM
34 GPIOs, 4x SPI, 3x UART, 2x I2C,
2x I2S, RMT, LED PWM, 1 host SD/eMMC/SDIO,
1 slave SDIO/SPI, TWAI®, 12-bit ADC, Ethernet

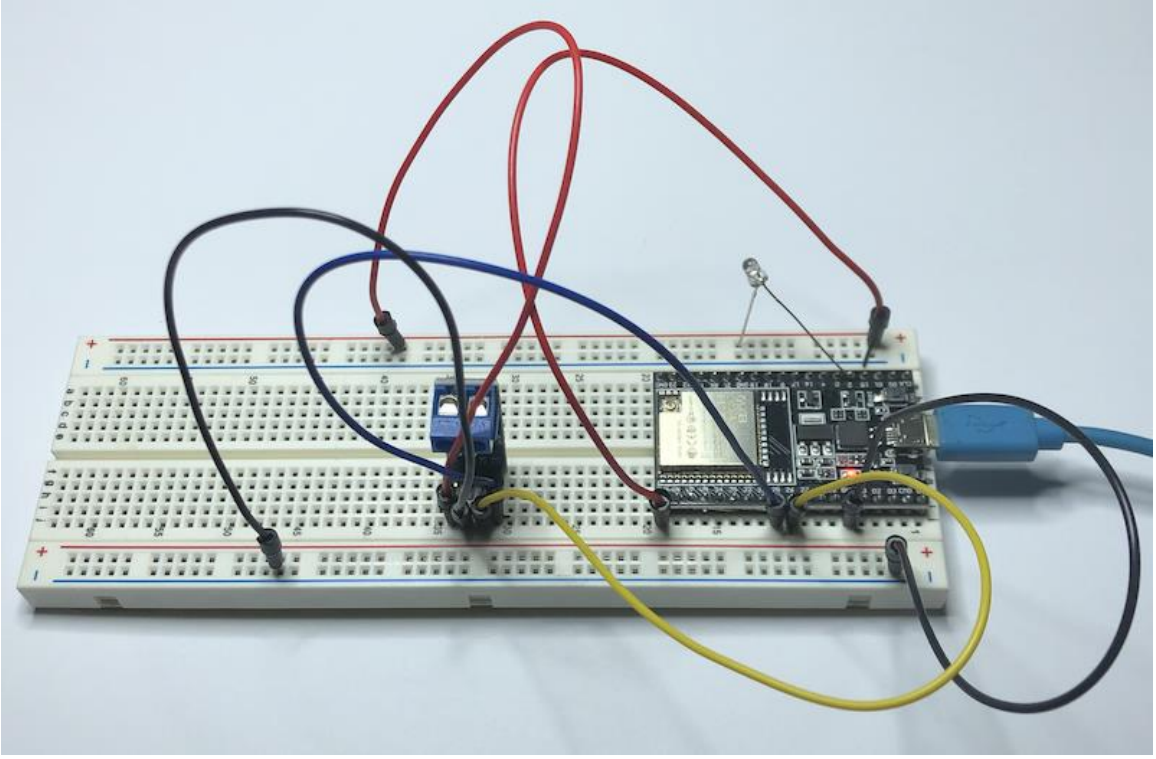
PWM Capable Pin
GPIOX GPIO Input Only
GPIOX GPIO Input and Output
DAC_X Digital-to-Analog Converter
JTAG/USB JTAG for Debugging and USB
FLASH External Flash Memory (SPI)
ADCX_CH Analog-to-Digital Converter
TOUCHX Touch Sensor Input Channel
OTHER Other Related Functions
SERIAL Serial for Debug/Programming
ARDUINO Arduino Related Functions
STRAP Strapping Pin Functions

RTC RTC Power Domain (VDD3P3_RTC)
GND Ground
PWD Power Rails (3V3 and 5V)
! Pin Shared with the Flash Memory
*Can't be used as regular GPIO*

GPIO STATE
**WPU**: *Weak Pull-up (Internal)*
**WPD**: *Weak Pull-down (Internal)*
**PU**: *Pull-up (External)*
**IE**: *Input Enable (After Reset)*
**ID**: *Input Disabled (After Reset)*
**OE**: *Output Enable (After Reset)*
**OD**: *Output Disabled (After Reset)*

For better resolution, download the PDF file.

Our test setup looks like this:



In addition to the CAN port, we added an LED to indicate CAN Bus data traffic.

## CAN Bus Programming

For the CAN Bus programming we used the resources as shown in the following. We didn't deem it necessary to repeat information in this post that is already sufficiently provided.

The CAN Bus driver source code plus code samples can be found at:

- sandeepmistry/arduino-CAN: An Arduino library for sending and receiving data using CAN bus. (github.com)

For instructions on how to program the ESP32 using the Arduino IDE, see:

- How to Program ESP32 with Arduino IDE? (electronicshub.org)

Further Development Resources

- ESP32-DevKitC V4 Getting Started Guide...
- ESP32-WROOM-32D & ESP32-WROOM-32U Data Sheet (PDF)...
- Getting Started with ESP 32 | Introduction to ESP32...
- ESP32 vs ESP8266 - Which One To Choose?
- How to Program ESP32 with Arduino IDE?
- Interfacing I2C LCD with ESP32 | ESP32 I2C LCD Tutorial...
- A Beginner's Tutorial on ESP32 Bluetooth | Learn…
- A Complete Beginner's Tutorial on How to Create…
- In-depth tutorial on ESP32 Servo Control | Web…


**Internet of Things Projects with ESP32: Build exciting and powerful IoT projects**

The ESP32, a low-cost MCU with integrated Wi-Fi and BLE capabilities, comes with a variety of modules and development boards for building IoT applications efficiently. Wi-Fi and BLE are standard network stacks for Internet-of-Things applications providing cost-effective solutions for your business and project requirements.

This book is a fundamental guide for developing ESP32 programs and starts by explaining GPIO (General Purpose I/O) programming with sensor devices. The reader gets up to speed with ESP32 development through several IoT projects such as weather stations, sensor loggers, smart homes, Wi-Fi cams, and Wi-Fi wardriving. The reader learns how to use ESP32 boards to facilitate interactions between mobile applications and cloud servers, such as AWS. By the end of this book, you'll have learned how to control a range of IoT projects using the ESP32 chip.

More Information...