ESP32



The predecessor of ESP32, the ESP8266 has a built-in processor. However due to multitasking involved in updating the Wi-Fi stack, most of the applications use a separate micro-controller for data processing, interfacing sensors and digital Input Output. With the ESP32 you may not want to use an additional micro-controller. **ESP32 has Xtensa® Dual-Core 32-bit LX6 microprocessors, which runs up to 600 DMIPS.** The ESP32 will run on breakout boards and modules from **160 MHz up to 240MHz**. That is very good speed for anything that requires a microcontroller with connectivity options.

The two cores are named **Protocol CPU (PRO\_CPU)** and **Application CPU (APP\_CPU)**. That basically means the **PRO\_CPU** processor handles the Wi-Fi, Bluetooth and other internal peripherals like SPI, I2C, ADC etc. The **APP\_CPU** is left out for the application code. This differentiation is done in the Espressif Internet Development Framework (ESP-IDF). ESP-IDF is the official software development framework for the chip. Arduino and other implementations for the development will be based on ESP-IDF.

ESP-IDF uses freeRTOS for switching between the processors and data exchange between them. We have done numerous tutorials on freeRTOS and with all the bare-metal programming tutorials for ESP32 we will try and cover this aspect in detail. Although the feature set is great at the price at which the chip is being sold, the complexity is enormous. For the chip to get widely adopted, it will require huge efforts from Espressif as well as the community.

**PIN SPECIFICATIONS**

|  |  |  |
| --- | --- | --- |
| **Pin Category** | **Pin Name** | **Details** |
| Power | **Micro-USB, 3.3V, 5V,** **GND** | **Micro-USB:** ESP32 can be powered through USB port  **5V:** Regulated 5V can be supplied to this pin which is we be again regulated to 3.3V by on board regulator, to power the board.  **3.3V:** Regulated 3.3V can be supplied to this pin to power the board.  **GND:** Ground pins. |
| Enable | **En** | The pin and the button resets the microcontroller. |
| Analog Pins | **ADC1\_0 to ADC1\_5 and ADC2\_0 to ADC2\_9** | Used to measure analog voltage in the range of 0-3.3V.  12-bit 18 Channel ADC |
| DAC pins | **DAC1 and DAC2** | Used for Digital to analog Conversion |
| Input/Output Pins | **GPIO0 to GPIO39** | Totally 39 GPIO pins, can be used as input or output pins. 0V (low) and 3.3V (high). But pins 34 to 39 can be used as input only |
| Capacitive Touch pins | **T0 to T9** | These 10 pins can be used a touch pins normally used for capacitive pads |
| RTC GPIO pins | **RTCIO0 to RTCIO17** | These 18 GPIO pins can be used to wake up the ESP32 from deep sleep mode. |
| Serial | Rx, **Tx** | Used to receive and transmit TTL serial data. |
| External Interrupts | All GPIO | Any GPIO can be use to trigger an interrupt. |
| PWM | All GPIO | 16 independent channel is available for PWM any GPIO can be made to work as PWM though software |
| VSPI | GPIO23 (MOSI), GPIO19(MISO), GPIO18(CLK) and GPIO5 (CS) | Used for SPI-1 communication. |
| HSPI | GPIO13 (MOSI), GPIO12(MISO), GPIO14(CLK) and GPIO15 (CS) | Used for SPI-2 communication. |
| IIC | GPIO21(SDA), GPIO22(SCL) | Used for I2C communication. |
| AREF | **AREF** | To provide reference voltage for input voltage. |

### ****ESP32 Technical Specifications****

|  |  |
| --- | --- |
| Microprocessor | Ten silica Xtensa LX6 |
| Maximum Operating Frequency | 240MHz |
| Operating Voltage | 3.3V |
| Analog Input Pins | 12-bit, 18 Channel |
| DAC Pins | 8-bit, 2 Channel |
| Digital I/O Pins | 39 (of which 34 is normal GPIO pin) |
| DC Current on I/O Pins | 40 mA |
| DC Current on 3.3V Pin | 50 mA |
| SRAM | 520 KB |
| Communication | SPI(4), I2C(2), I2S(2), CAN, UART(3) |
| Wi-Fi | 802.11 b/g/n |
| Bluetooth | V4.2 – Supports BLE and Classic Bluetooth |