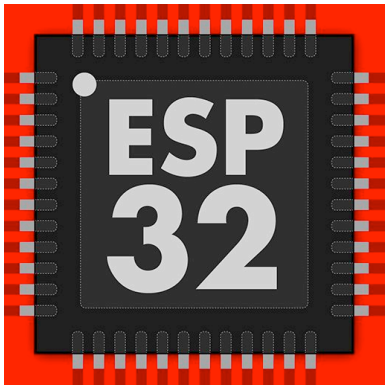
RIoT Strikes Back – Medan

Unleash the power of Espressif ESP32 Dual Core CPU

Universitas Pelita Harapan - Medan

2 December 2017



$whoami

Albert Suwandhi, ST, MTI.

Lecturer – STMIK IBBI

Bachelor Degree in Electrical Engineering

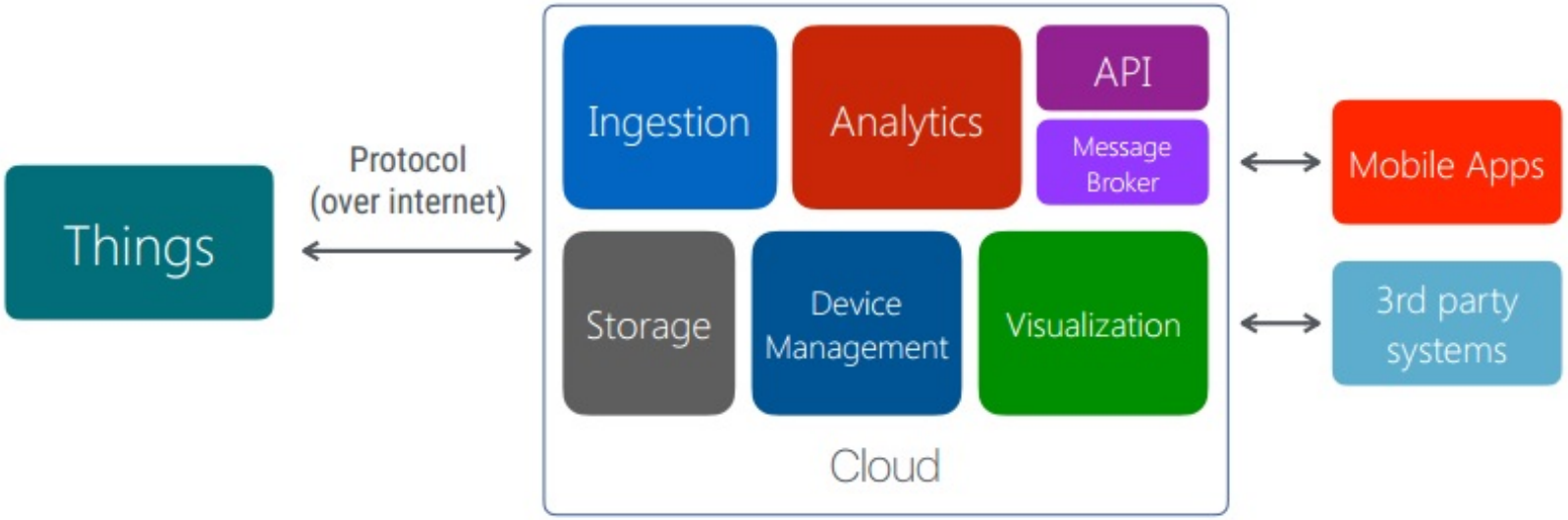
Master Degree in Information Technology

https://id.linkedin.com/in/albertsuwandhi

albert.suwandhi@gmail.com

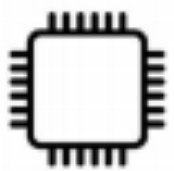
087868577265

IoT Typical Architecture

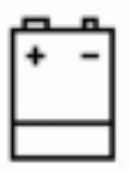


What are inside the Things

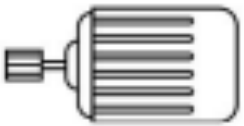
Sensors



MCU/MPU



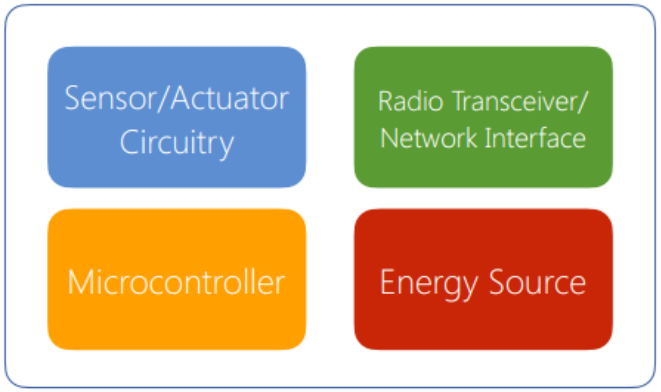
Energy Source



Actuators

Network Interface

Firmware



Today's Focus

ESPRESSIF

Espressif ESP8266

- SoC/MCU that integrates TCP/IP Stack and WiFi to enable it to connect to the network and communicate with other devices
- Made by Espressif – China
- Have many modules, eq : ESP-01, ESP-02, ESP-12F, ESP-WROOM2
- It was sold as a Serial-WiFi adapter for other microcontrollers, however it is more powerful as the ESP8266 integrates a 32 bits microcontroller.
- Feature :

  Microcontroller: Tensilica L106 (32 bits)

  Clock Frequency: 80 MHz

  WiFi: 802.11 b/g/n

  Interfaces: SPI, I2C, I2S, UART, PWM

  GPIO: 17 pins (12 mA max current)

  ADC: 10 bits

  Operating voltage: 3.3 V

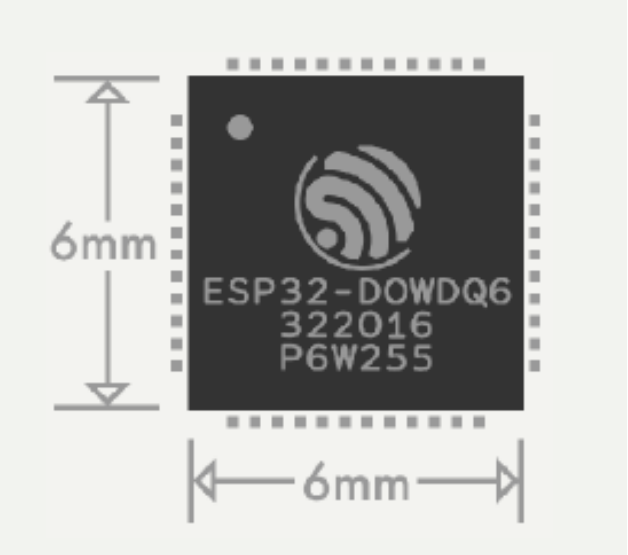
ESP8266 Development Framework

Sming

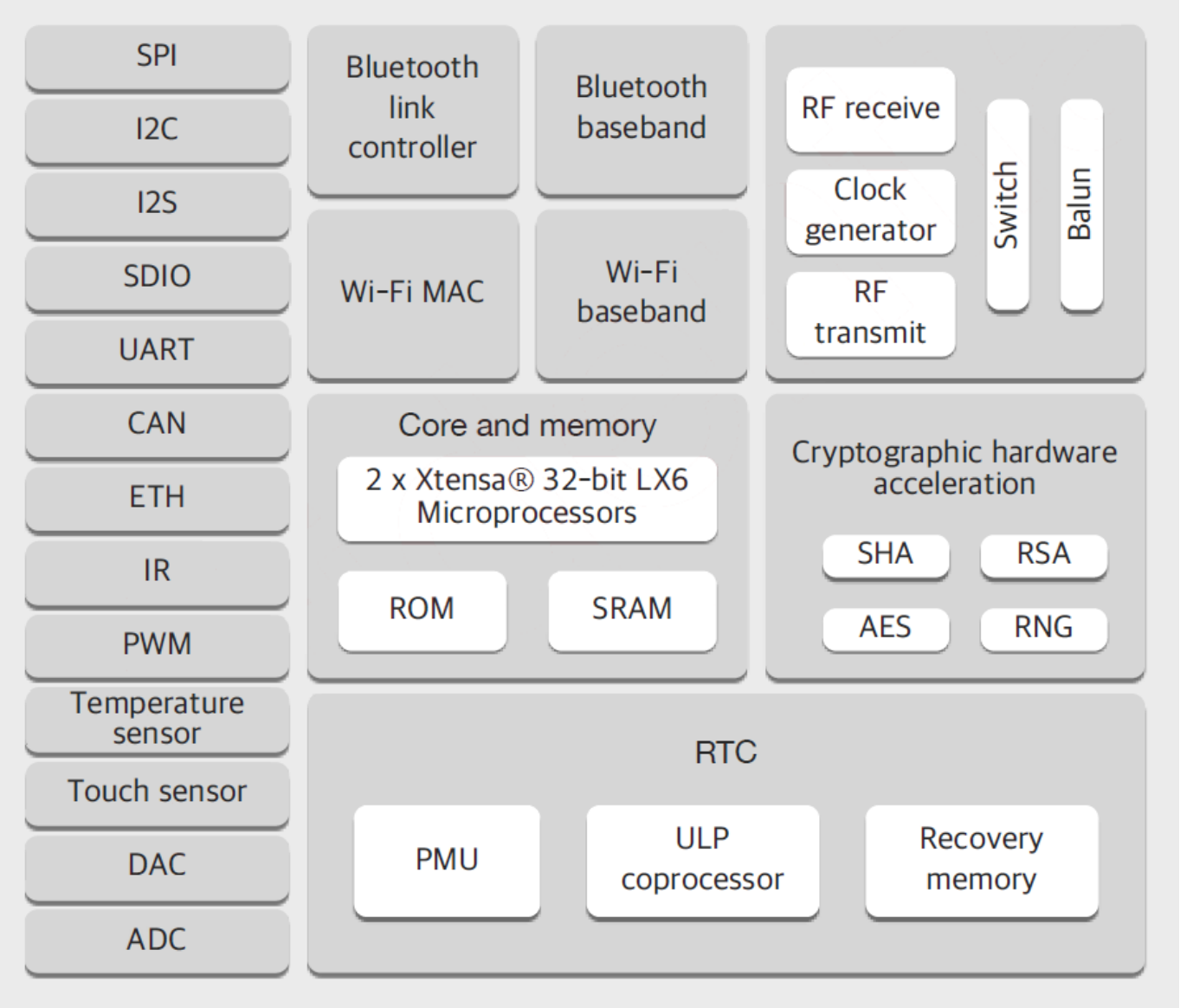
Espruino

Espressif ESP32

- CPU: Tensilica Xtensa 32-bit dual core, up to 240 MHz, 600 DMIPS.
- Operating voltage: 3.3V
- Memory: 448 KB ROM, 520 KB SRAM 16 KB SRAM in RTC, 1 Kbit of eFuse. External Flash: 512 KB to (4 x 16) MiB
- WiFi (802.11): b/g/n/e/i
- Bluetooth: v4.2 BR/EDR & BLE
- Peripherals: GPIOs, PWM, ADC, DAC, I2S, UART, SPI, I2C, CAN, RMII, Cap Touch



ESP32 Block Diagram

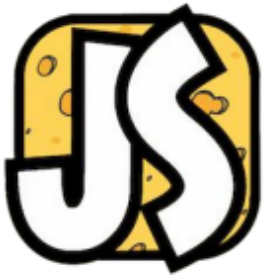


ESP32 Development Framework

ESP-IDF

Espruino

ARDUINO



ESP-IDF

- ESP IoT Development Framework (ESP-IDF) is the official development framework for ESP32.
- Built on Assembly, C and C++
- ESP-IDF is build on top of FreeRTOS.
- Advantages : More control over the low level APIs and the configurations
- Disadvantages : It is harder to getting started with
- More info : https://github.com/espressif/esp-idf

Arduino ESP32

- Arduino framework support for Espressif ESP32 platform
- Advantages : Easier to get started with, low level stuffs are hidden to developer, huge community and libraries.
- Trade Off : Lose control of the configurations & low level stuffs
- Can be embedded as ESP-IDF components
- More info : https://github.com/espressif/arduino-esp32

MicroPython

- A software implementation of the Python 3 programming language written in C, that is optimised to run on a microcontroller.
- Initially written by Damien George for PyBoard
- Easy to Learn High Level Language
- More info : https://micropython.org/

Mongoose OS

- An Operating System for connected products
- Language supported : C/C++ and JavaScript
- Relatively new in the market
- More info : https://mongoose-os.com/

ESP32 Module

and
more…

ESP32 Boards (1)

Huzzah

Hornbill

ARS01119B

AnalogLamb ESP32

Node32S

FireBeetle

D-duino-32

and many more…

ESP32 Boards (2)

Espressif DevKit

ESP32 Things

ESP320

ESP32 N1

FiPy

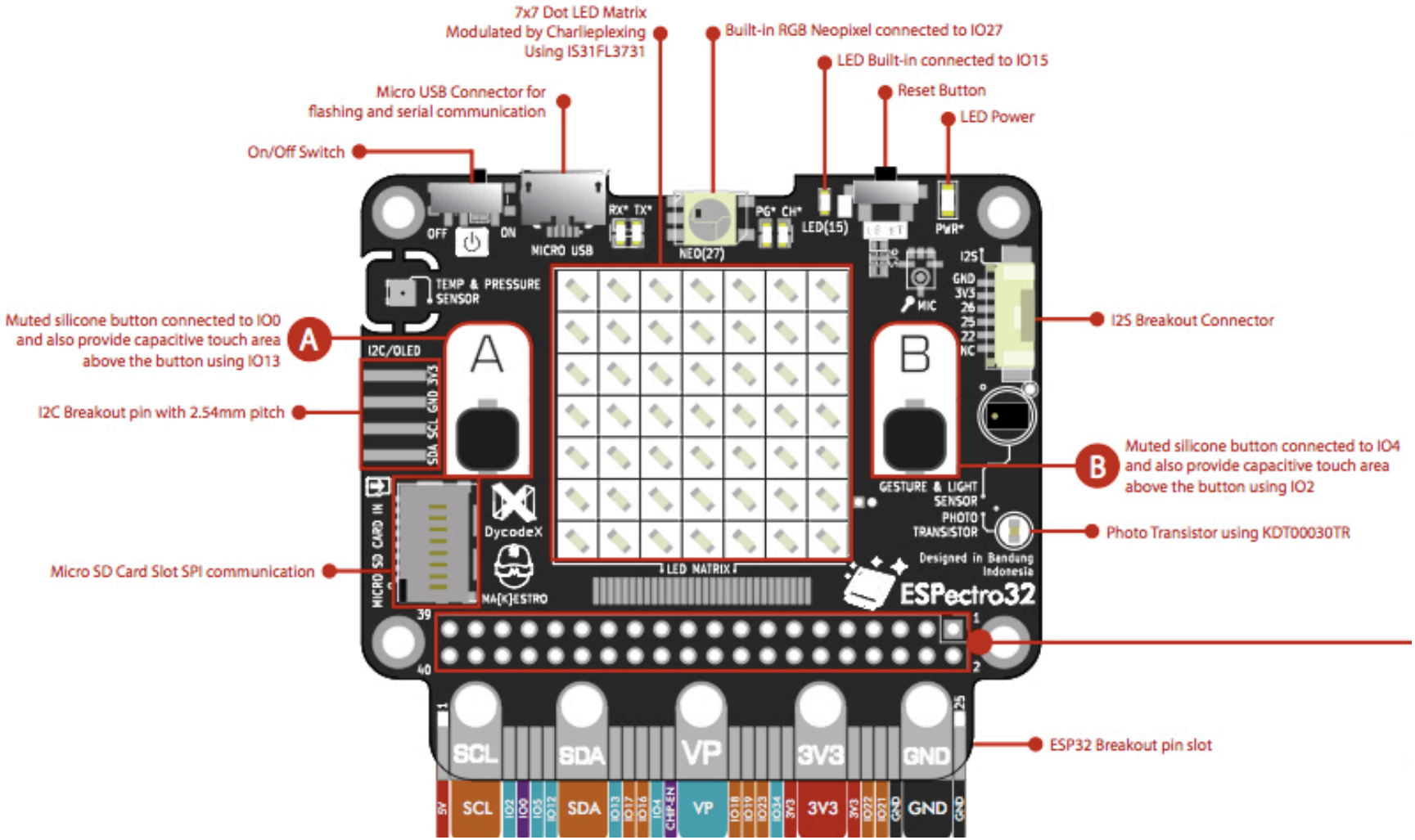
Nano32

Widora Air

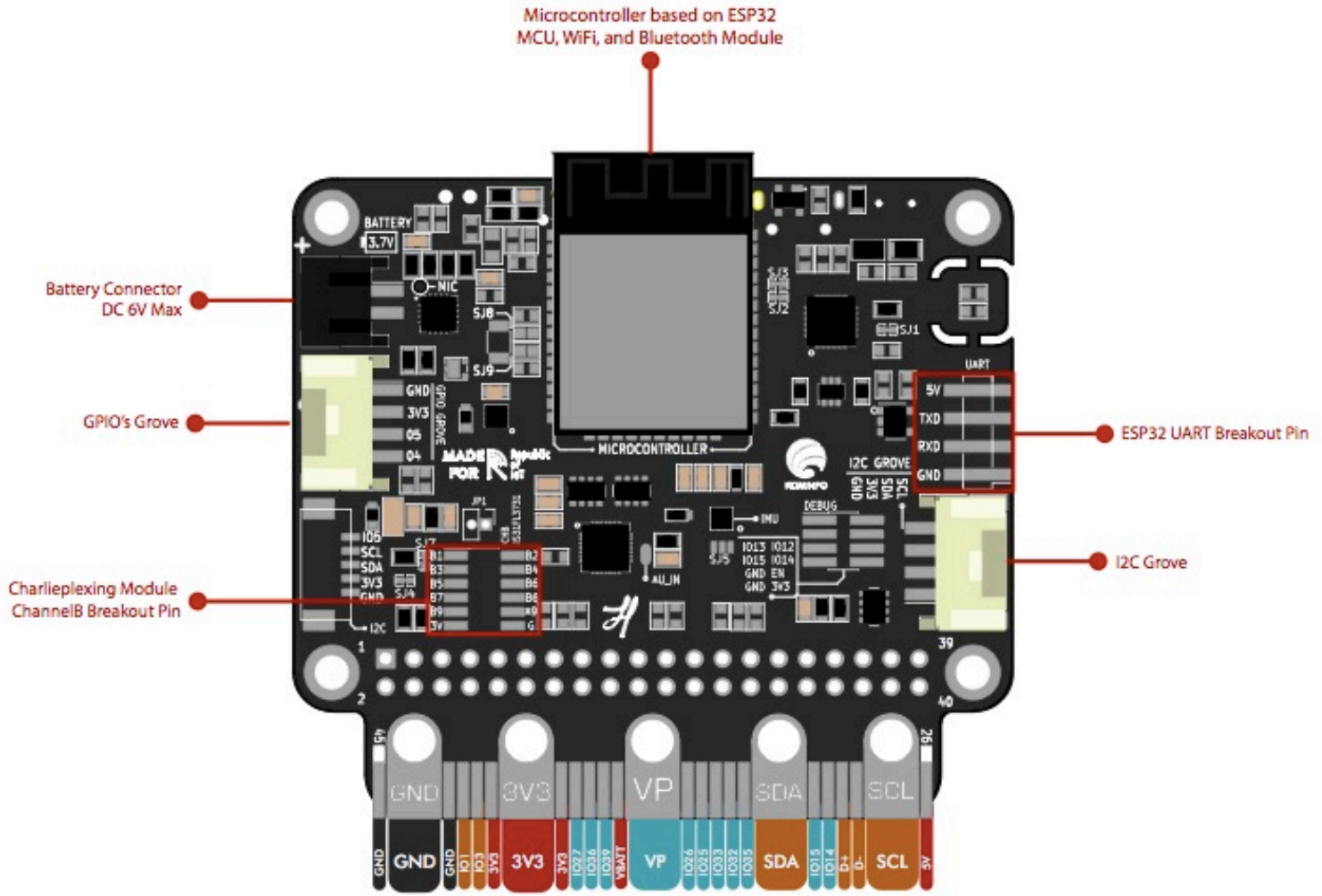
Pesky ESP32

and many more…

ESPectro32 (Front)



ESPectro32 (Back)



ESP8266 vs ESP32

Specifications	ESP8266	ESP32
MCU	Xtensa Single-Core 32-bit L 106	Xtensa Dual-Core 32-bit LX6 600 DMIPS
802.11 b/g/n Wi-Fi	Yes, HT20	Yes, HT40
Bluetooth	N/A	Bluetooth 4.2 and below
Typical Frequency	80 MHz	160 MHz
SRAM	160 kBytes	512 kBytes
Flash	SPI Flash up to 16 MBytes	SPI
GPIO	17	36
Hardware / Software PWM	None / 8 Channels	1 / 16 Channels
SPI / I2C / I2S / UART	2/1/2/2	4/2/2/2
ADC	10-bit	12-bit
CAN	N/A	1
Ethernet MAC Interface	N/A	1
Touch Sensor	N/A	Yes
Temperature Sensor	N/A	Yes
Working Temperature	-40° C – 125° C	-40° C – 125° C

ESP32 CPU

- Dual High Performance LX6 CPU : PRO (PROTOCOL CPU) and APP (APPLICATION CPU). We can refer to this cores as Core 0 and Core 1
- One (ULP) Ultra Low Power Processor – During Sleep
- How can we run our firmware on two cores simultaneously?

KEEP

CALM

IT IS

DEMO

TIME

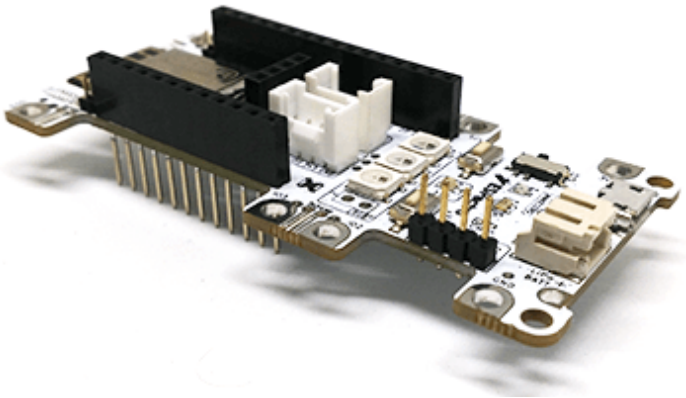
Test Scenario

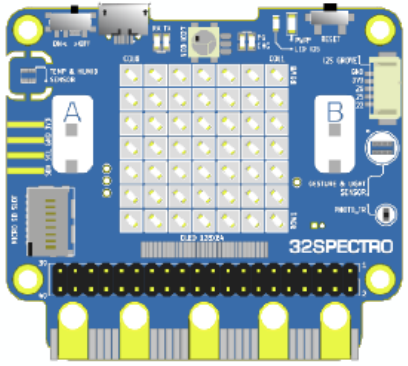
We will use Arduino-ESP32 for the following scenario :

- Check on which core the application runs
- Move application execution to another core
- Single Core Load Test : Arduino Mega, ESP8266 and ESP32
- Run tasks simultaneously on different core and how to synchronize task using semaphore
- ESP32 : Single Core vs Dual Core Speed Up

Hardware and tools used in the demo

- Arduino Mega + ESpectro + ESPectro32
- VS Code + PlatformIO





Codes in this session

https://github.com/albert.suwandhi/RIoT_Strikes_Back_2017

Thank You