Imperial College
London

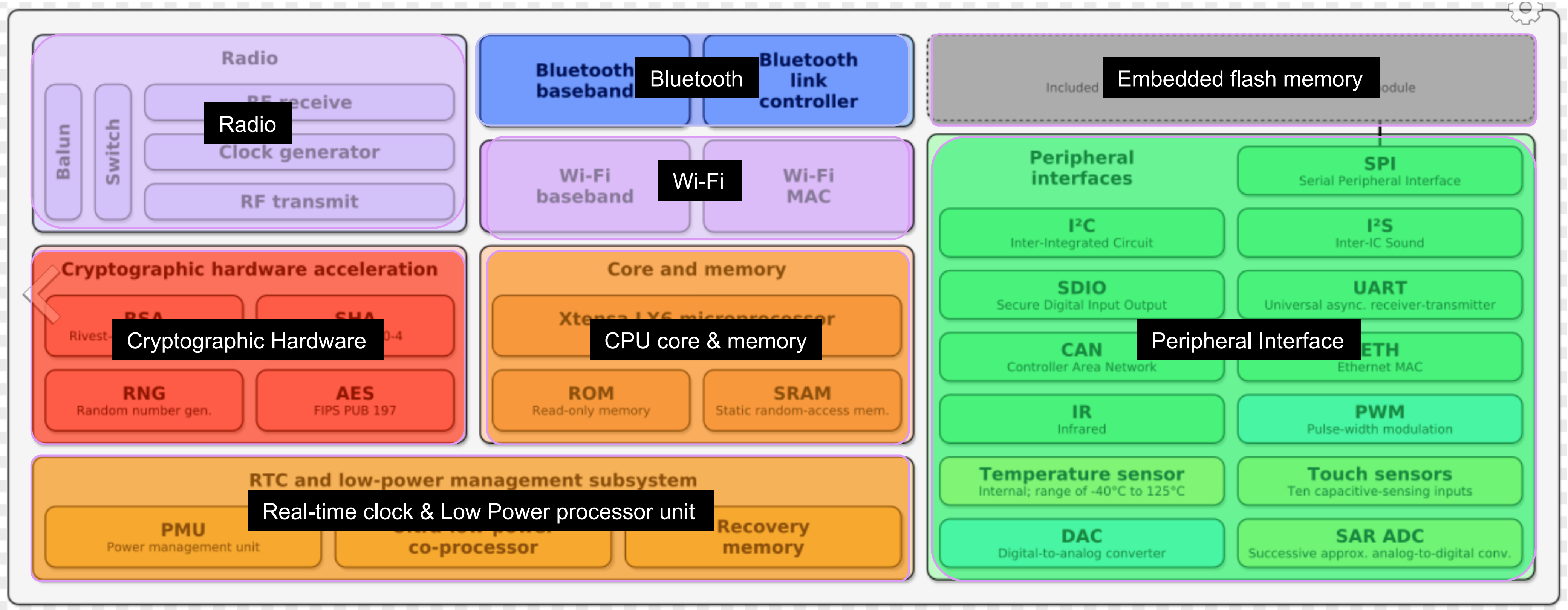
Topic 15

Using MicroPython on ESP32

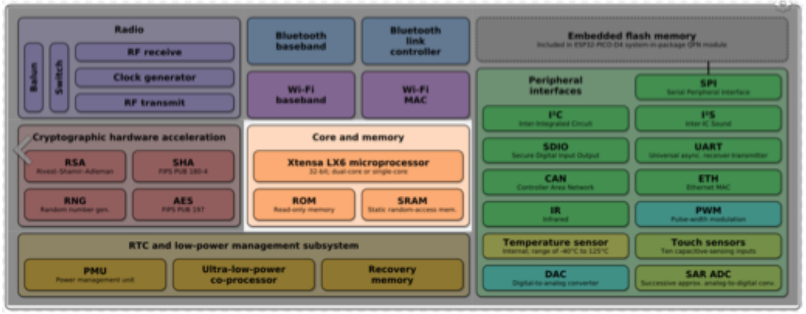
Professor Peter YK Cheung
Dyson School of Design Engineering

URL: www.ee.ic.ac.uk/pcheung/teaching/DE1_EE/
E-mail: p.cheung@imperial.ac.uk

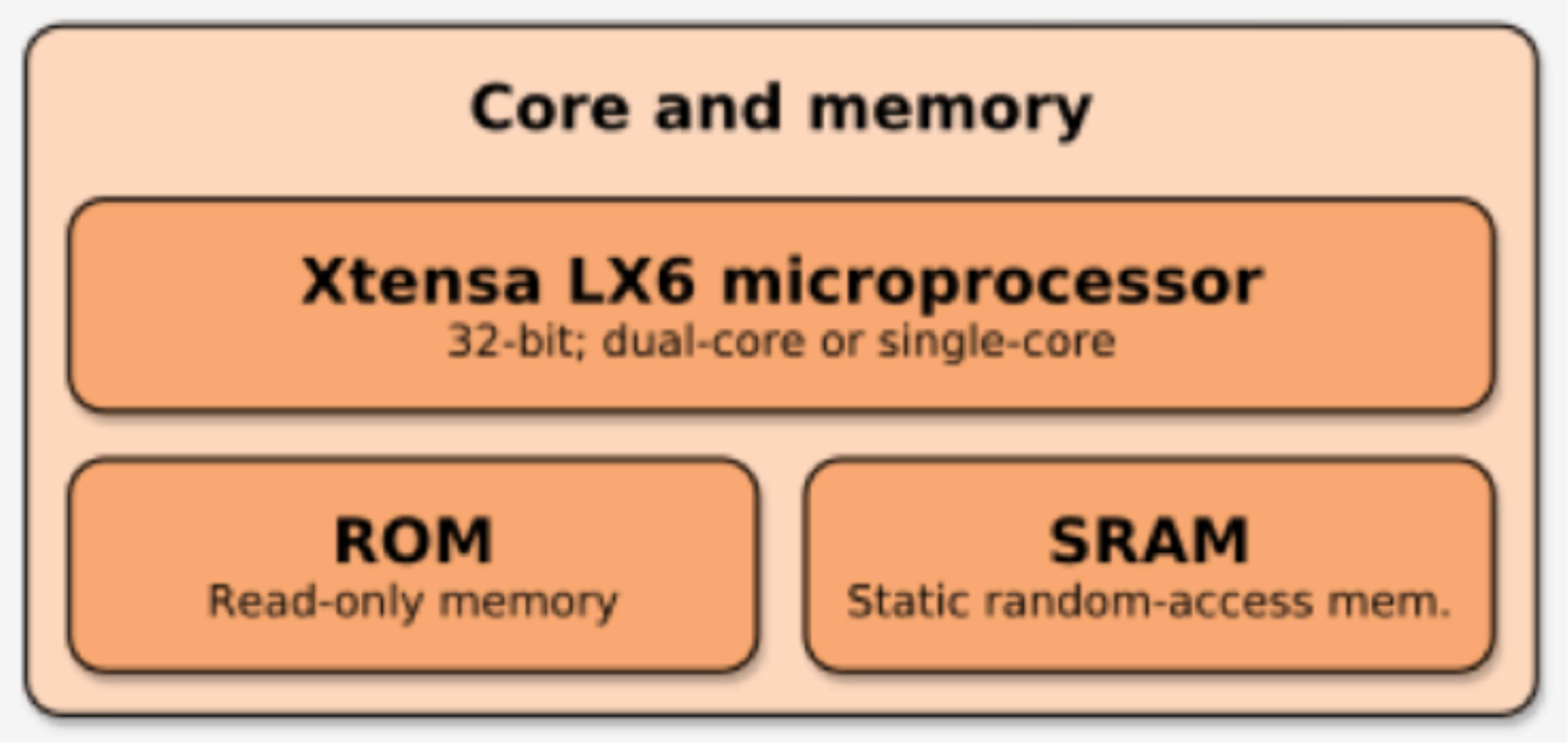
ESP32 IoT Microcontroller (1)



ESP32 CPU Core & Memory

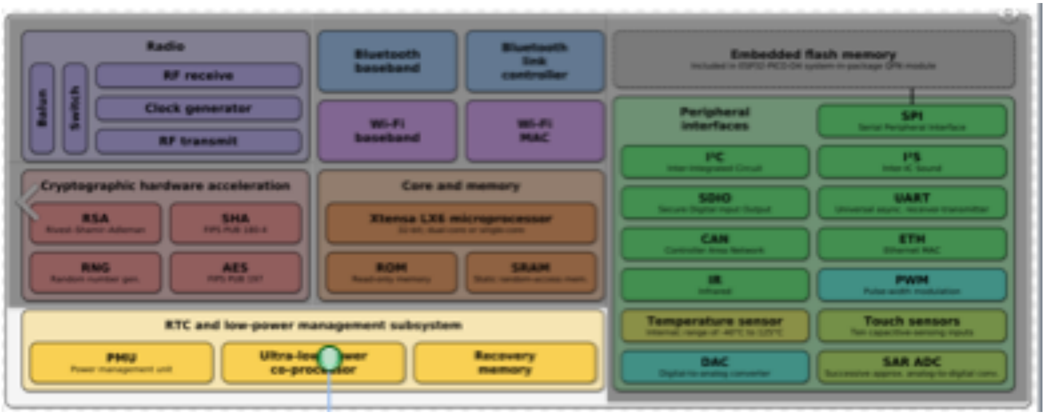


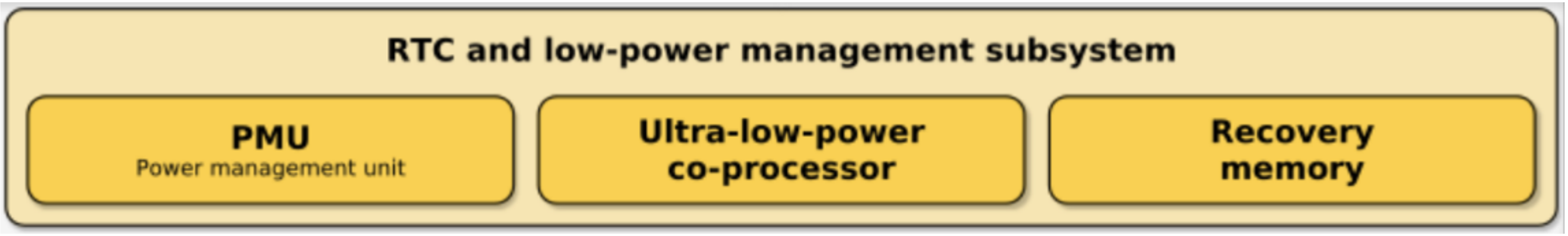
Radio
Balun
Switch
RF receive
Clock generator
RF transmit
Bluetooth baseband
Bluetooth link controller
Wi-Fi baseband
Wi-Fi MAC
Embedded flash memory
Included in ESP32-PICO-D4 system-in-package QFN module
Cryptographic hardware acceleration
RSA
Rivest-Shamir-Adleman
SHA
FIPS PUB 180-4
RNG
Random number gen.
AES
FIPS PUB 197
Core and memory
Xtensa LX6 microprocessor
32-bit; dual-core or single-core
ROM
Read-only memory
SRAM
Static random-access mem.
RTC and low-power management subsystem
PMU
Power management unit
Ultra-low-power co-processor
Recovery memory
Peripheral interfaces
SPI
Serial Peripheral Interface
I²C
Inter-Integrated Circuit
I²S
Inter-IC Sound
SDIO
Secure Digital Input Output
UART
Universal async. receiver-transmitter
CAN
Controller Area Network
ETH
Ethernet MAC
IR
Infrared
PWM
Pulse-width modulation
Temperature sensor
Internal; range of -40°C to 125°C
Touch sensors
Ten capacitive-sensing inputs
DAC
Digital-to-analog converter
SAR ADC
Successive approx. analog-to-digital conv.



Core and memory
Xtensa LX6 microprocessor
32-bit; dual-core or single-core
ROM
Read-only memory
SRAM
Static random-access mem.

ESP32 RTC and Power Management Sub-system





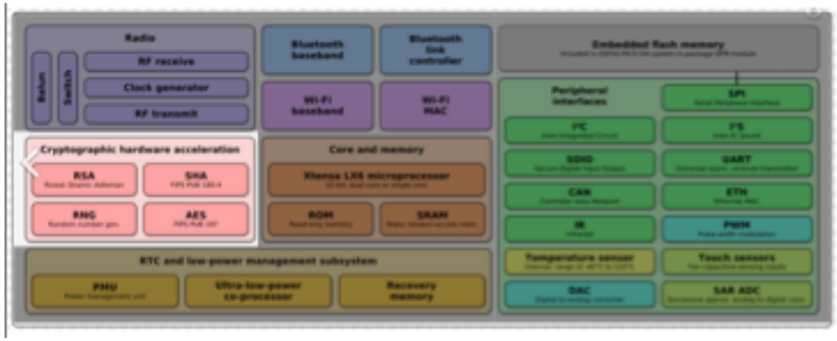
RTC and low-power management subsystem

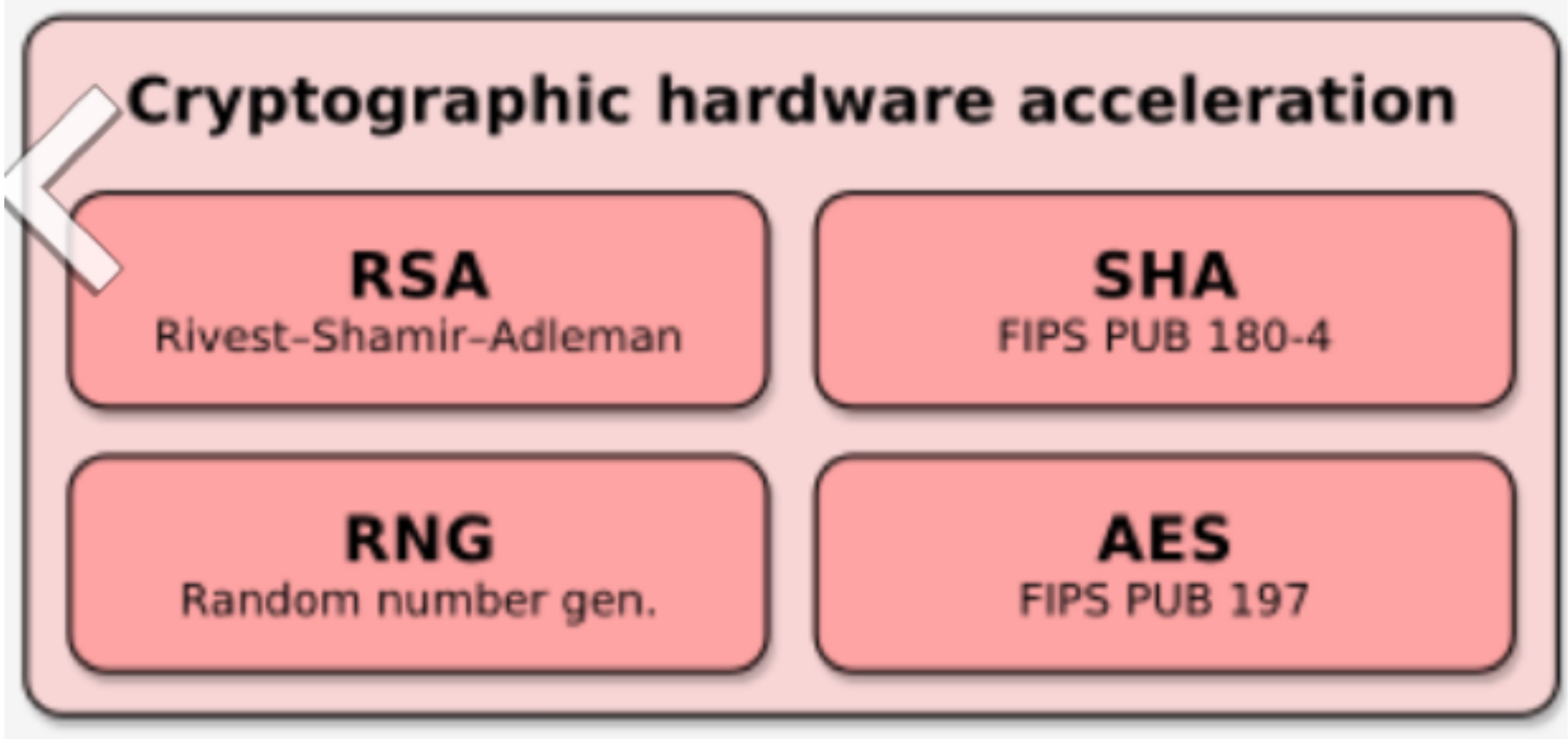
PMU
Power management unit

Ultra-low-power co-processor

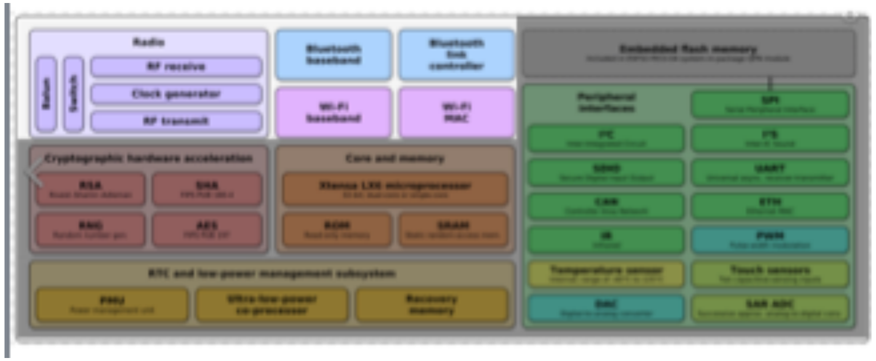
Recovery memory

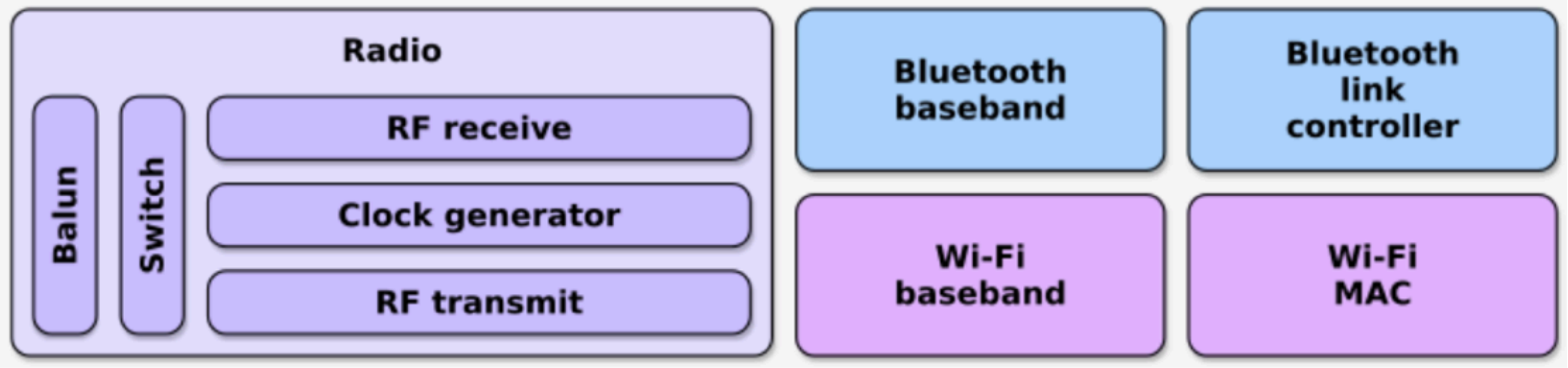
ESP32 Crypto Hardware



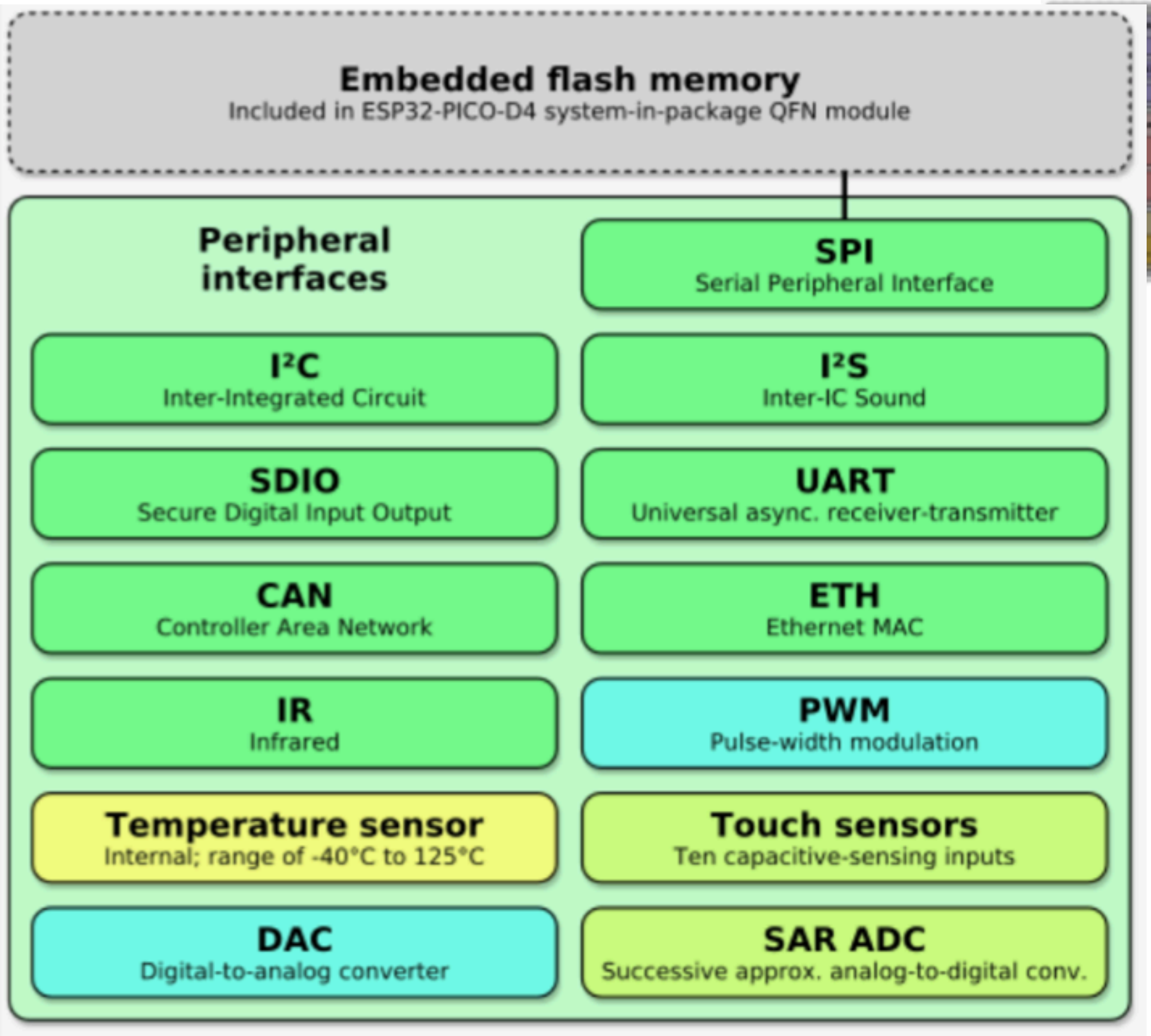


ESP32 wireless links





ESP32 Peripheral Interfaces & SPI RAM

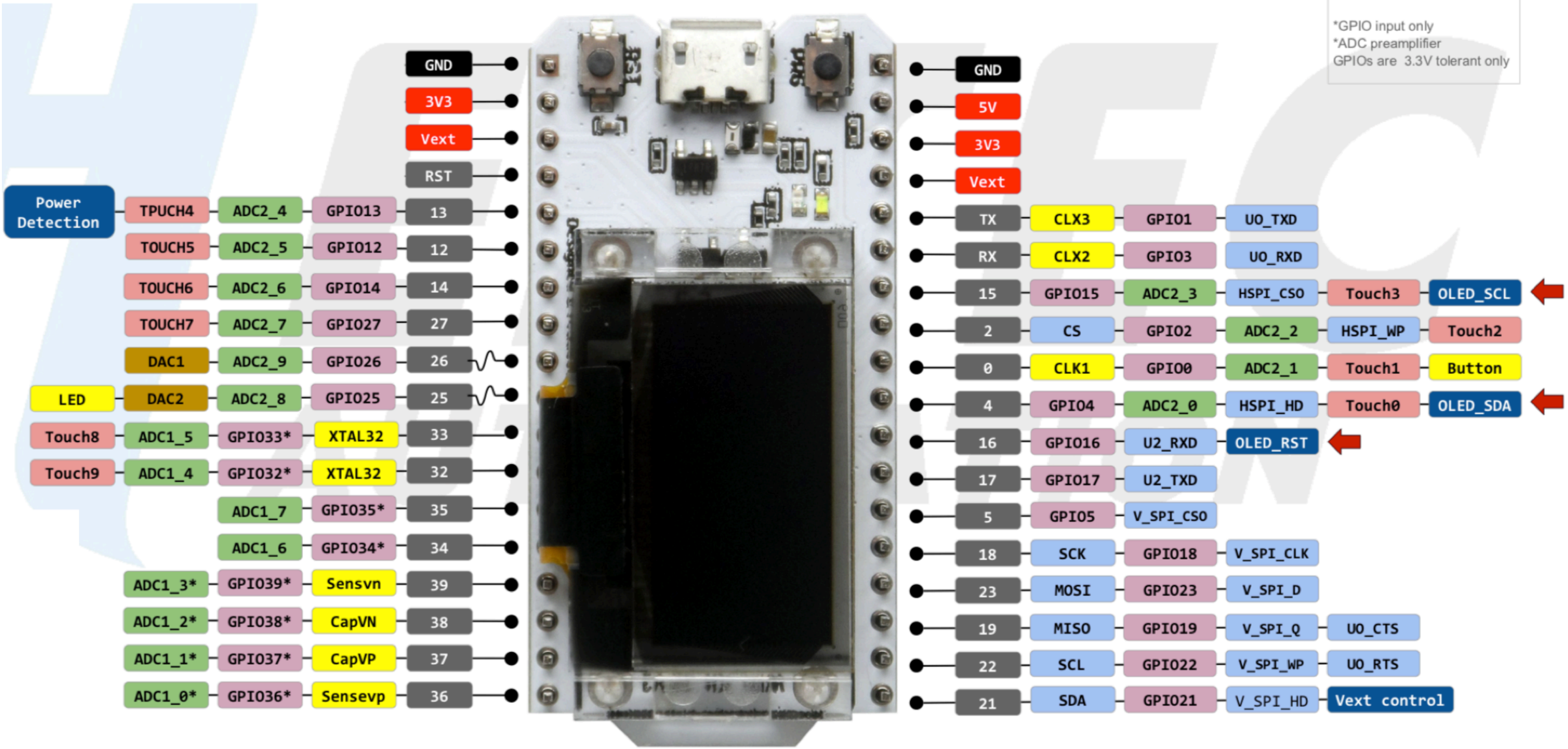


Embedded flash memory
Included in ESP32-PICO-D4 system-in-package QFN module
Peripheral interfaces
SPI
Serial Peripheral Interface
I²C
Inter-Integrated Circuit
I²S
Inter-IC Sound
SDIO
Secure Digital Input Output
UART
Universal async. receiver-transmitter
CAN
Controller Area Network
ETH
Ethernet MAC
IR
Infrared
PWM
Pulse-width modulation
Temperature sensor
Internal; range of -40°C to 125°C
Touch sensors
Ten capacitive-sensing inputs
DAC
Digital-to-analog converter
SAR ADC
Successive approx. analog-to-digital conv.

Heltec ESP32 module

Notes:

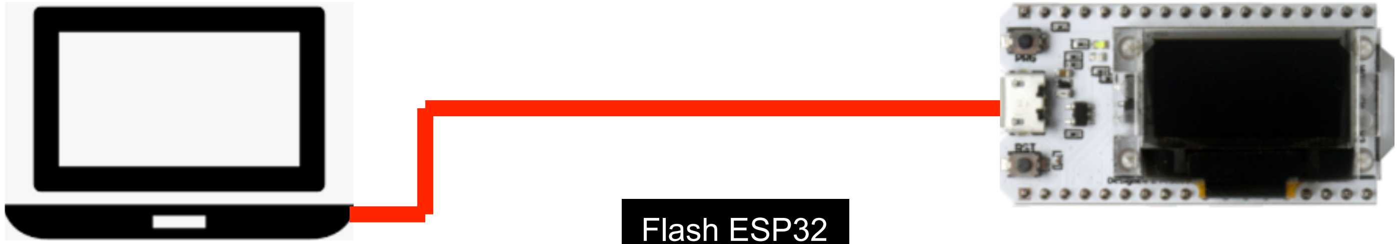
*GPIO input only
*ADC preamplifier
GPIOs are 3.3V tolerant only



					GND
					3V3
					Vext
					RST
Power Detection	TPUCH4	ADC2_4	GPIO13		13
	TOUCH5	ADC2_5	GPIO12		12
	TOUCH6	ADC2_6	GPIO14		14
	TOUCH7	ADC2_7	GPIO27		27
	DAC1	ADC2_9	GPIO26		26
LED	DAC2	ADC2_8	GPIO25		25
Touch8	ADC1_5	GPIO33*	XTAL32		33
Touch9	ADC1_4	GPIO32*	XTAL32		32
		ADC1_7	GPIO35*		35
		ADC1_6	GPIO34*		34
	ADC1_3*	GPIO39*	Sensvn		39
	ADC1_2*	GPIO38*	CapVN		38
	ADC1_1*	GPIO37*	CapVP		37
	ADC1_0*	GPIO36*	Sensevp		36

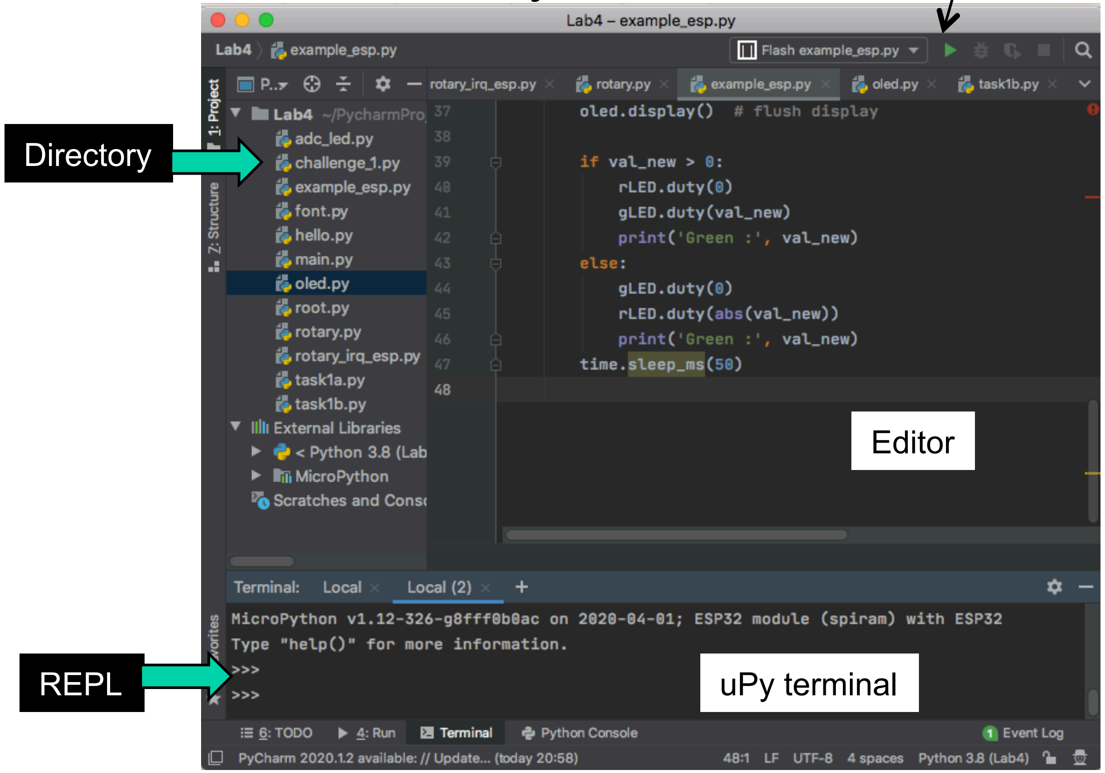
GND					
5V					
3V3					
Vext					
TX	CLX3	GPIO1	U0_TXD		
RX	CLX2	GPIO3	U0_RXD		
15	GPIO15	ADC2_3	HSPI_CS0	Touch3	OLED_SCL
2	CS	GPIO2	ADC2_2	HSPI_WP	Touch2
0	CLK1	GPIO0	ADC2_1	Touch1	Button
4	GPIO4	ADC2_0	HSPI_HD	Touch0	OLED_SDA
16	GPIO16	U2_RXD	OLED_RST		
17	GPIO17	U2_TXD			
5	GPIO5	V_SPI_CS0			
18	SCK	GPIO18	V_SPI_CLK		
23	MOSI	GPIO23	V_SPI_D		
19	MISO	GPIO19	V_SPI_Q	U0_CTS	
22	SCL	GPIO22	V_SPI_WP	U0_RTS	
21	SDA	GPIO21	V_SPI_HD	Vext control	

ESP32 with MicroPython (uPy)



Flash ESP32 via esptool

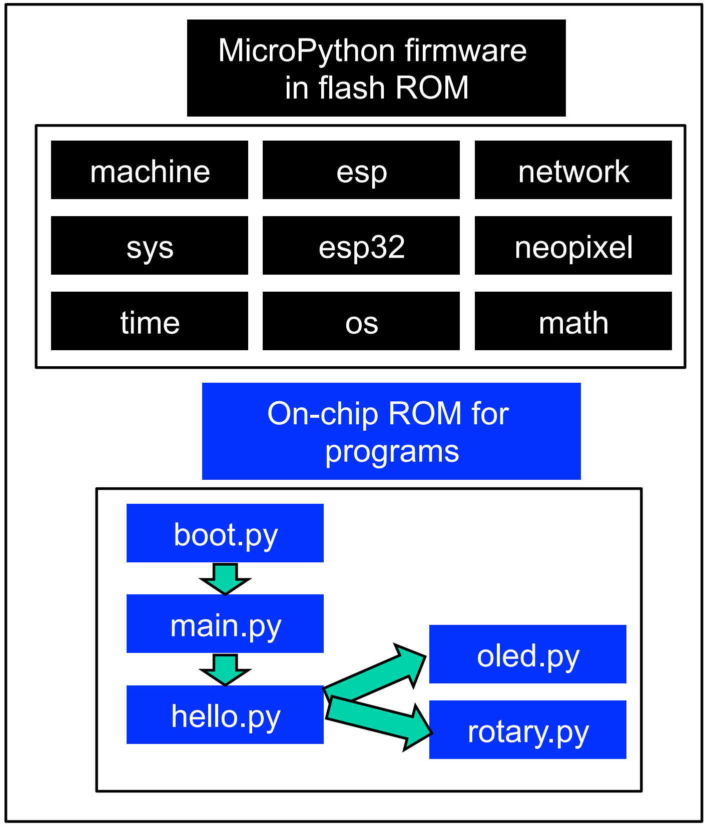
PyCharm IDE



```
oled.display()  # flush display

if val_new > 0:
    rLED.duty(0)
    gLED.duty(val_new)
    print('Green :', val_new)
else:
    gLED.duty(0)
    rLED.duty(abs(val_new))
    print('Green :', val_new)
time.sleep_ms(50)
```

```
MicroPython v1.12-326-g8fff0b0ac on 2020-04-01; ESP32 module (spiram) with ESP32
Type "help()" for more information.
>>>
>>>
```



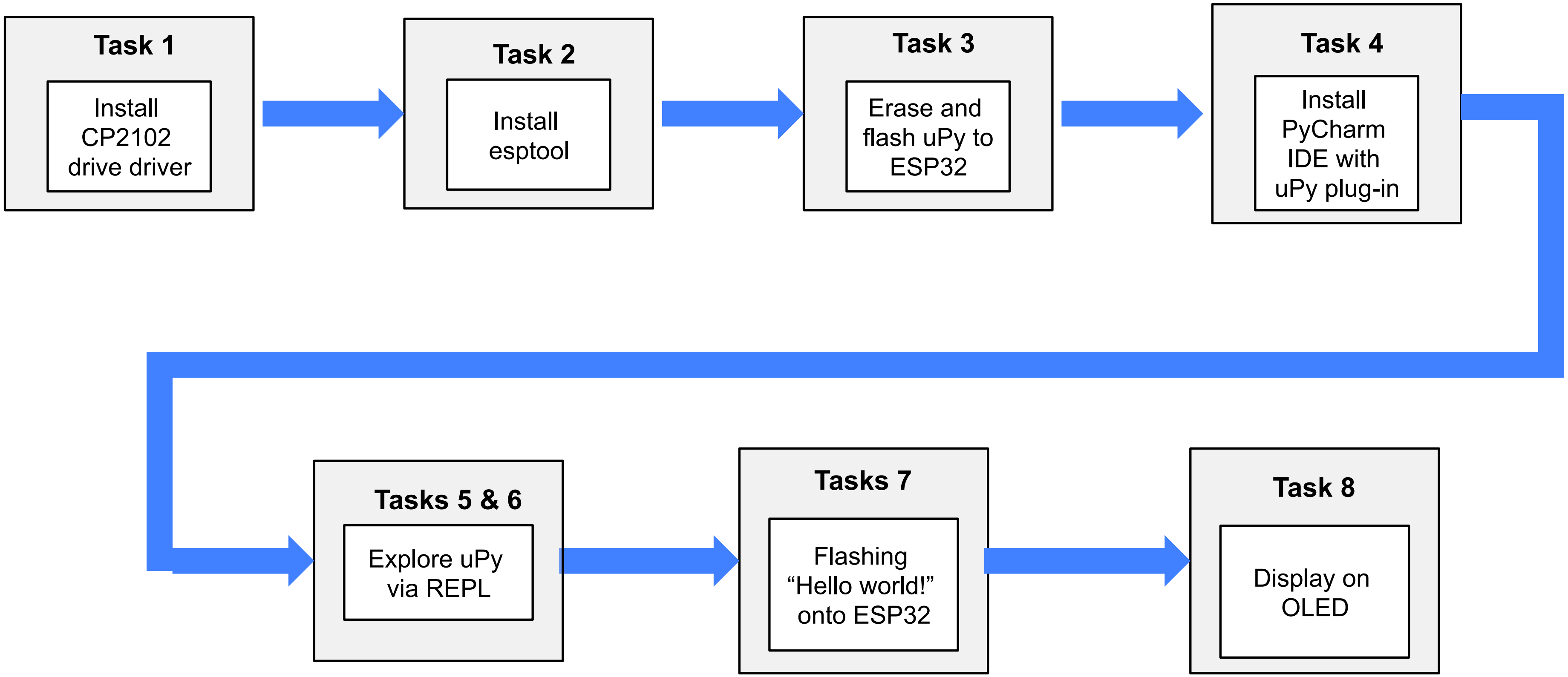
MicroPython firmware in flash ROM

machine	esp	network
sys	esp32	neopixel
time	os	math

On-chip ROM for programs

boot.py → main.py → hello.py → oled.py, rotary.py

Lab 4A – Setting up the MicroPython environment



Task 1

Install CP2102 drive driver

Task 2

Install esptool

Task 3

Erase and flash uPy to ESP32

Task 4

Install PyCharm IDE with uPy plug-in

Tasks 5 & 6

Explore uPy via REPL

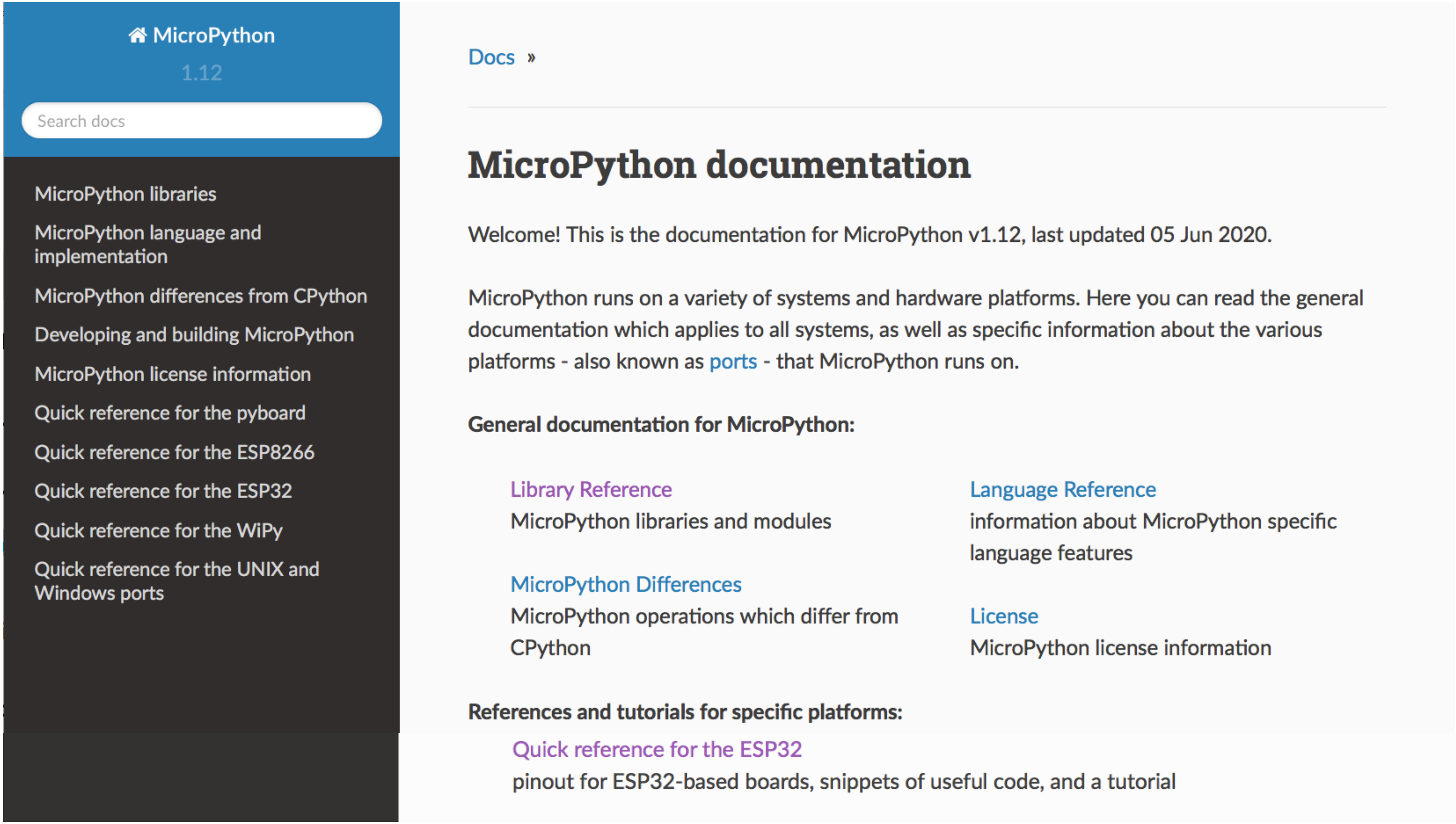
Tasks 7

Flashing “Hello world!” onto ESP32

Task 8

Display on OLED

MicroPython Documentation



MicroPython libraries

MicroPython language and implementation

MicroPython differences from CPython

Developing and building MicroPython

MicroPython license information

Quick reference for the pyboard

Quick reference for the ESP8266

Quick reference for the ESP32

Quick reference for the WiPy

Quick reference for the UNIX and Windows ports

Docs »

MicroPython documentation

Welcome! This is the documentation for MicroPython v1.12, last updated 05 Jun 2020.

MicroPython runs on a variety of systems and hardware platforms. Here you can read the general documentation which applies to all systems, as well as specific information about the various platforms - also known as ports - that MicroPython runs on.

General documentation for MicroPython:

Library Reference
MicroPython libraries and modules

MicroPython Differences
MicroPython operations which differ from CPython

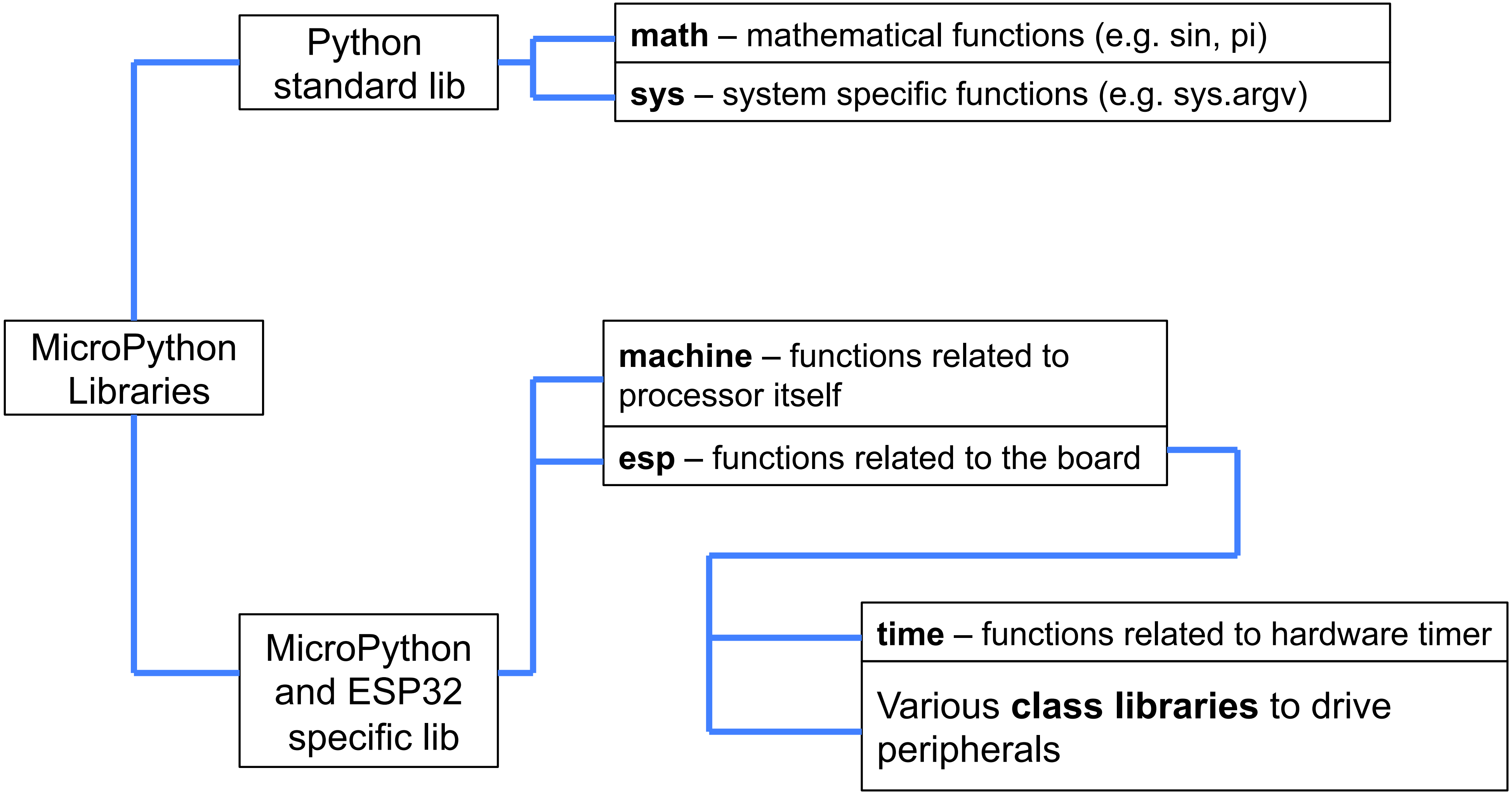
Language Reference
information about MicroPython specific language features

License
MicroPython license information

References and tutorials for specific platforms:

Quick reference for the ESP32
pinout for ESP32-based boards, snippets of useful code, and a tutorial

MicroPython Library Functions



MicroPython Libraries

Python standard lib

math – mathematical functions (e.g. sin, pi)

sys – system specific functions (e.g. sys.argv)

MicroPython and ESP32 specific lib

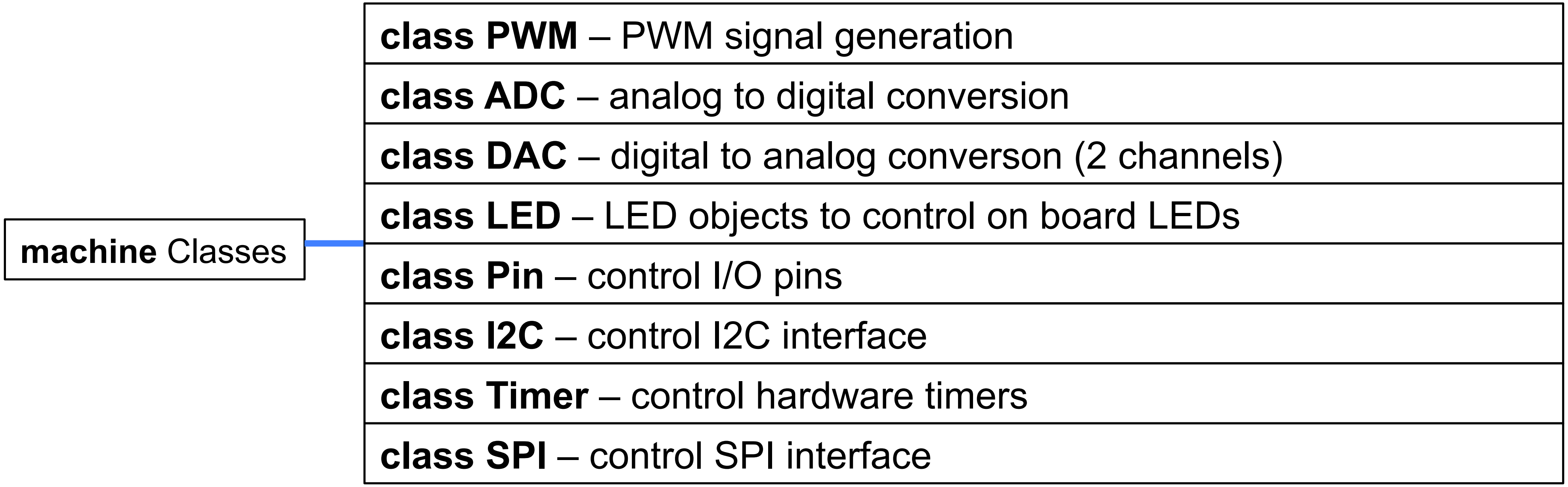
machine – functions related to processor itself

esp – functions related to the board

time – functions related to hardware timer

Various **class libraries** to drive peripherals

pyb - Class Library



machine Classes

class PWM – PWM signal generation

class ADC – analog to digital conversion

class DAC – digital to analog converson (2 channels)

class LED – LED objects to control on board LEDs

class Pin – control I/O pins

class I2C – control I2C interface

class Timer – control hardware timers

class SPI – control SPI interface