System On Chip (SoC)

A system on a chip or system on chip (SoC or SOC) is an integrated circuit (also known as an "IC" or "chip") that integrates all components of a computer or other electronic systems. It may contain digital, analogue, mixed-signal, and often radio-frequency functions—all on a single substrate. SoCs are very common in the mobile computing market because of their low power consumption. A typical application is in the area of embedded systems.

SoC integrates a microcontroller (or microprocessor) with advanced peripherals like graphics processing unit (GPU), Wi-Fi module, or coprocessor. If the definition of a microcontroller is a system that integrates a microprocessor with peripheral circuits and memory, the SoC is to a microcontroller what a microcontroller is to processors, remembering that the SoC does not necessarily contain built-in memory.

The term SoC has a bit of a marketing tinge to it.

System on chip (SoC) implies that a single chip contains silicon, which provides hardware support for a lot of functions. Previously, these functions would require multiple chips (a chip set).

Here's a menu of functionality, which can be found in SoC

(Again, this is a menu. I'm not implying that every one of these features has to be included to a chip to qualify as a SoC. Neither this menu is complete.)

- Computing: CPU, DSP core, MPEG codec. Some of the recent SoC have multiple cores.
- Memory: RAM, non-volatile memory
- Wired communication: USB slave (and master in some cases too), Ethernet MAC (and PHY in some cases too)
- Wireless communication: Bluetooth, Wi-Fi
- Embedded buses: CAN, I2C, SPI
- Analog capabilities: A/D, D/A

Embedded Systems

An embedded system is a computer system with a dedicated function within a larger mechanical or electrical system, often with real-time computing constraints. It is embedded as part of a complete device often including hardware and mechanical parts. Embedded systems control many devices in common use today. Ninety-eight percent of all microprocessors are manufactured as components of embedded systems. Modern embedded systems are often based on microcontrollers (i.e. CPU's with integrated memory or peripheral interfaces).

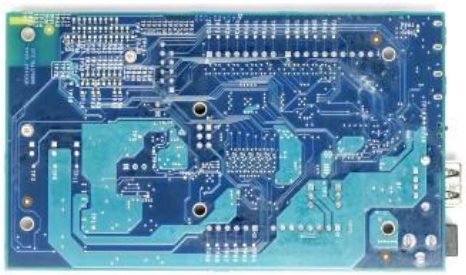
Intel® Edison Module

The Intel® Edison module is a SoC (System on Chip) that includes an Intel® Atom™ 500MHz dual-core, dual-threaded CPU and an Intel® Quark™ 100MHz microcontroller.

Key features:

- Integrated Wi-Fi, Bluetooth 4.0 LE;
- Support for Yocto Linux, Python, Node.js and Wolfram

Intel® Edison Kit for Arduino Front



Intel® Edison Kit for Arduino Back

Intel® Edison Kit for Arduino provides the Arduino 1.0 pinout and standard connectors such as a micro USB connected to a UART, a USB OTG port that can be switched between a second micro USB device connector, a standard size USB host Type-A connector, a uSD card holder, and a DC power jack.

Like an Arduino Uno, the Intel® Edison Kit for Arduino makes possible to have provides 20 digital input/output pins, of which 6 can be used as analogue inputs. The Intel® Edison has 4 PWM outputs which can be configured via jumpers to any of the 6 pins supporting PWM on the Arduino Uno (pins 3, 5, 6, 9, 10, or 11).

The Intel® Edison Kit for Arduino is designed to be hardware and software pin-compatible with Arduino shields designed for the Arduino Uno R3. Digital pins 0 to 13 (and the adjacent AREF and GND pins), Analog inputs 0 to 5, the power header, ICSP header, and the UART port pins (0 and 1), are all in the same locations as on the Arduino Uno R3.

The digital IOs and analogue pins can be configured to operate at either 5V or 3.3V. The outputs can source or sink 24 mA at 3.3V and 32 mA at

Intel® Edison Breakout Board Front *Intel® Edison Breakout Board Back*

The Intel® Edison Breakout Board is designed to expose the native 1.8V I/O of the Intel® Edison module. The board consists of power supply, battery recharger, USB OTG power switch, UART to USB bridge, USB OTG port, and I/O header.

Summary

SoC	Dual-core, dual-threaded Intel® Atom™ CPU and a 32-bit Intel® Quark™ microcontroller
Operating Voltage	1.8V (Breakout Board) 3.3V / 5V (Kit for Arduino
Input Voltage	7-15V (Both Breakout Board and Kit for Arduino
Digital I/O Pins	20 (of which 6 provide analogue input and 4 provide PWM output) (Kit for Arduino)
Flash Storage	4 GB eMMC
RAM	1 GB LPDDR3 POP
Clock Speed	500 MHz (Intel® Atom™ CPU) 100 MHz (Quark™ microcontroller)
Length	35.5mm (Edison) 61mm (Breakout Board) 127mm (Kit for Arduino)
Width	25mm (Edison) 29mm (Breakout Board) 72mm (Kit for Arduino)
Height	4mm (Edison) 12mm (Breakout Board) 12mm (Kit for Arduino)

Beagle Bone Black

The Beagle Bone Black is a low-cost credit-card-sized development platform with good support from a fast growing community. The BeagleBone Black differs slightly from the regular version by providing you with an on-board micro HDMI port, 512MB of DDR3L DRAM, 4GB on-board flash memory, an AM3358 processor at 1GHz, and making JTAG optional with a user supplied header. Ultimately, the BeagleBone Black is still perfect for physical computing and smaller embedded applications.

Processor: AM335x 1GHz ARM® Cortex-A8

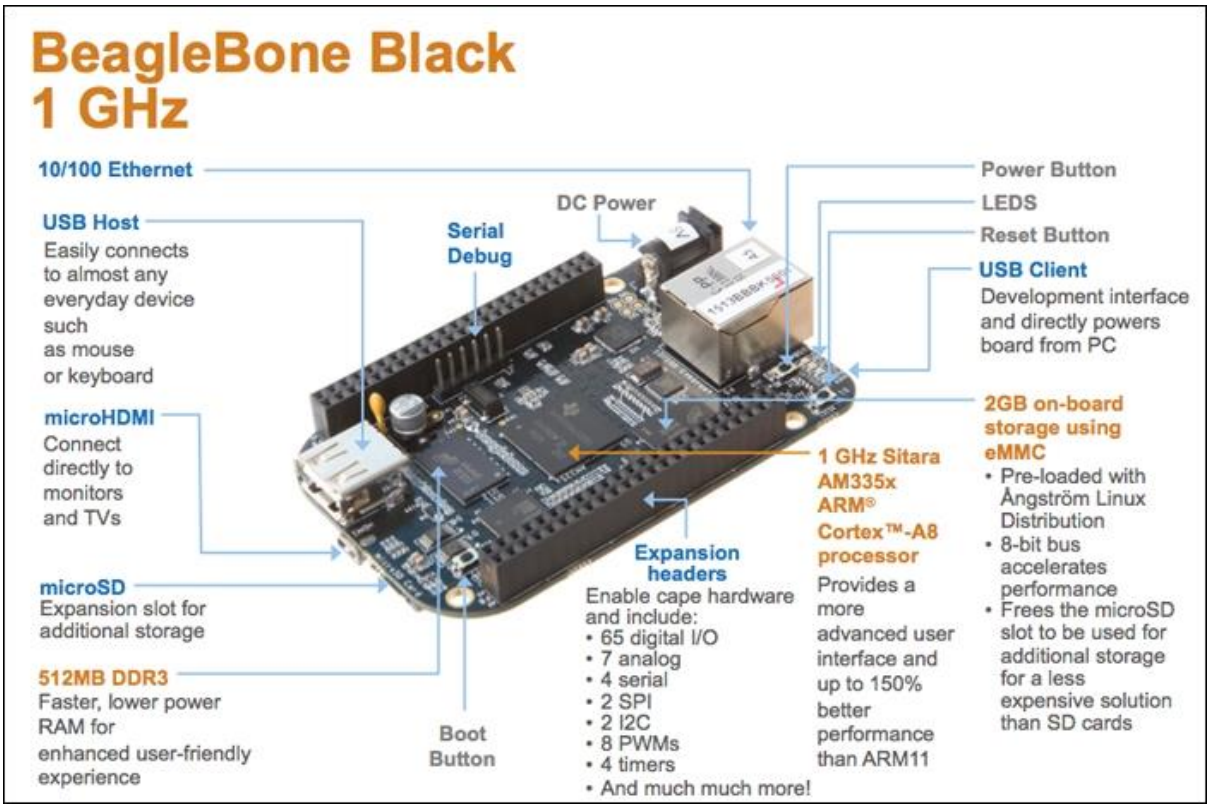
- 512MB DDR3 RAM
- 4GB 8-bit eMMC on-board flash storage
- 3D graphics accelerator
- NEON floating-point accelerator
- 2x PRU 32-bit microcontrollers

Software Compatibility

- Debian
- Android
- Ubuntu
- Cloud9 IDE on Node.js w/ BoneScript library
- Plus much more

Connectivity

- USB client for power & communications
- USB host
- Ethernet
- HDMI
- 2x 46 pin headers



Raspberry Pi

The **Raspberry Pi** is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation.

Specs

- Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
- 1GB RAM
- BCM43438 wireless LAN and Bluetooth Low Energy (BLE) on board
- 40-pin extended GPIO
- 4 USB 2 ports
- 4 Pole stereo output and composite video port
- Full size HDMI
- CSI camera port for connecting a Raspberry Pi camera
- DSI display port for connecting a Raspberry Pi touchscreen display
- Micro SD port for loading your operating system and storing data
- Upgraded switched Micro USB power source up to 2.5A

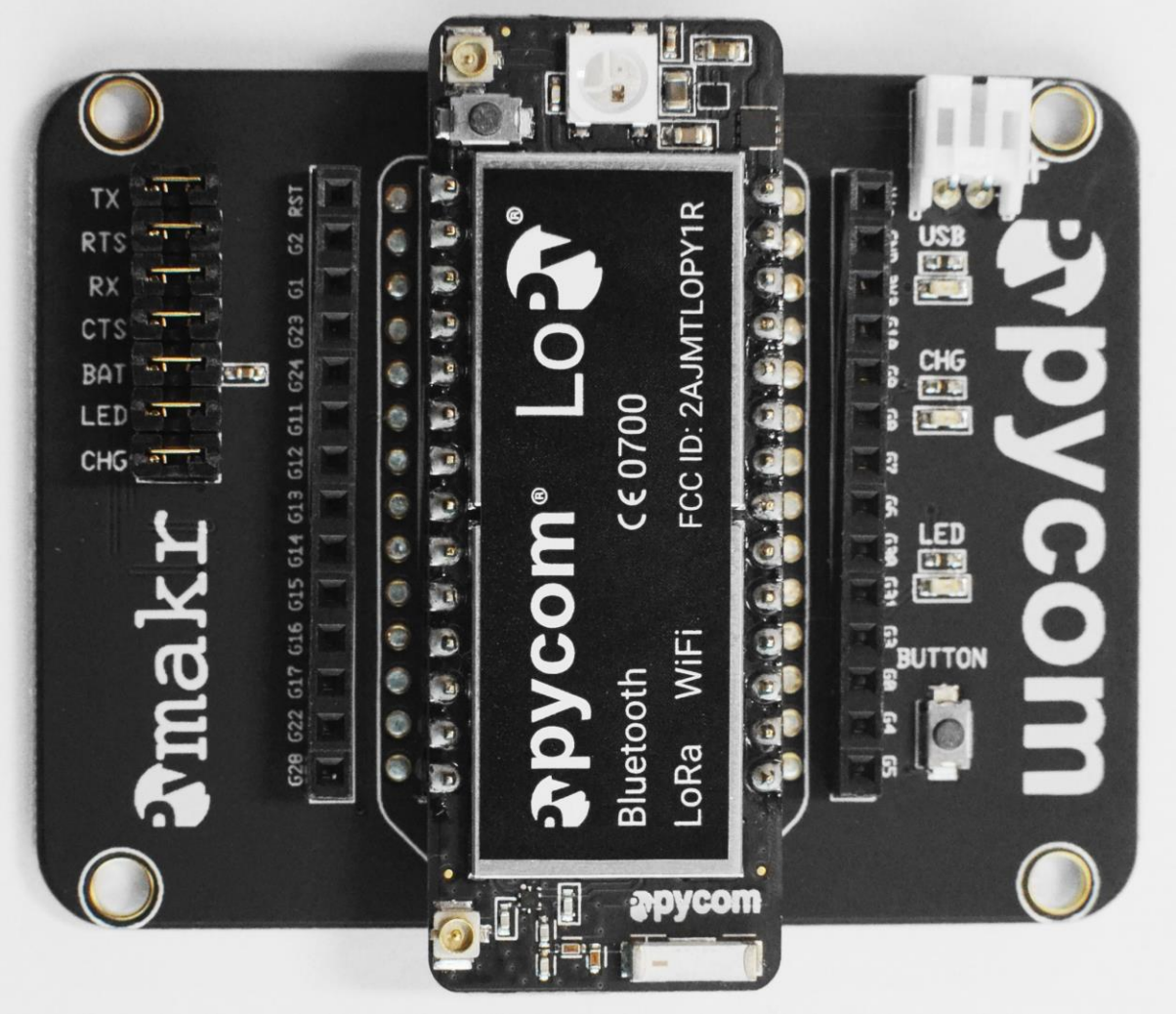
LoPy

With LoRa, Wi-Fi and BLE, the Pycom LoPy is the only triple bearer MicroPython enabled microcontroller on the market today – the perfect enterprise grade IoT platform for your connected Things. With the latest Espressif ESP32 chipset the LoPy offers a perfect combination of power, friendliness and flexibility. Create and connect your things everywhere. Fast.

Features

- Powerful CPU, BLE and state of the art Wi-Fi radio. 1KM Wi-Fi Range
- Can also double up as a Nano LoRa gateway
- MicroPython enabled, the Linux of IoT for fast deployment
- Fits in a standard breadboard (with headers)
- Ultra-low power usage (850uA with the Wi-Fi connection active): a fraction compared to other connected micro controllers
- 100% Python programmable
- Lots of GPIOs, interfaces and peripherals
- Hardware floating point acceleration
- Python multi-threading

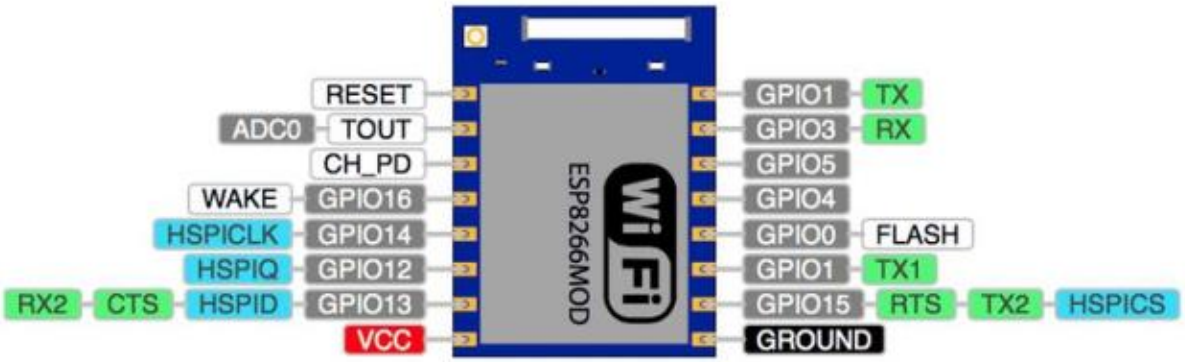
The LoPy can act as both a LoRa Nano Gateway and a multi-bearer (LoRa, Wi-Fi and BLE) development platform. It is programmable with MicroPython and the Pymakr IDE for fast IoT application development, easy programming in-field, and extra resilience with network failover. The best blend of speed to deployment and access to new LPWAN networks rolling out across Europe, USA, Africa and India.

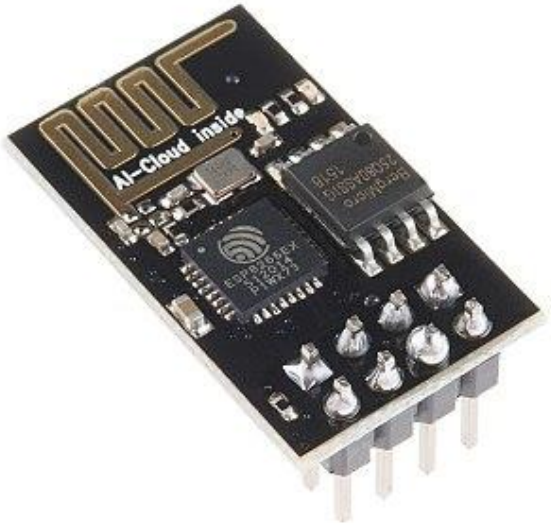


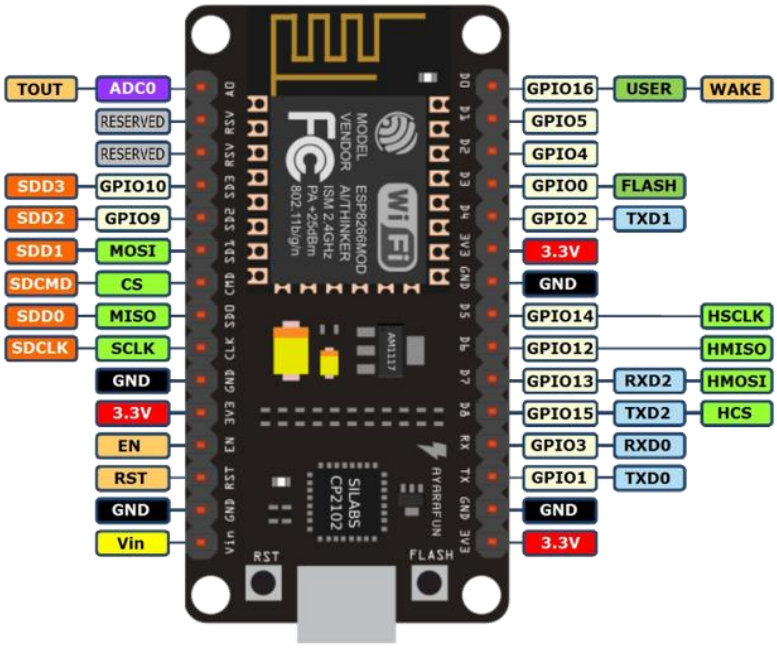
ESP8266

The ESP8266 Wi-Fi Module is a self-contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your Wi-Fi network. The ESP8266 is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor. Each ESP8266 module comes pre-programmed with an AT command set firmware, meaning, you can simply hook this up to your Arduino device and get about as much Wi-Fi-ability as a Wi-Fi Shield offers (and that's just out of the box)! The ESP8266 module is an extremely cost effective board with a huge, and ever growing, community.

This module has a powerful enough on-board processing and storage capability that allows it to be integrated with the sensors and other application specific devices through its GPIOs with minimal development up-front and minimal loading during runtime. Its high degree of on-chip integration allows for minimal external circuitry, including the front-end module, is designed to occupy minimal PCB area. The ESP8266 supports APSD for VoIP applications and Bluetooth co-existence interfaces, it contains a self-calibrated RF allowing it to work under all operating conditions, and requires no external RF parts.


RESET
ADC0
TOUT
CH_PD
WAKE
GPIO16
HSPICLK
GPIO14
HSPIQ
GPIO12
RX2
CTS
HSPID
GPIO13
VCC
ESP8266MOD
Wi Fi
GPIO1
TX
GPIO3
RX
GPIO5
GPIO4
GPIO0
FLASH
GPIO1
TX1
GPIO15
RTS
TX2
HSPICS
GROUND


AI-Cloud inside


TOUT
ADC0
RESERVED
RESERVED
SDD3
GPIO10
SDD2
GPIO9
SDD1
MOSI
SDCMD
CS
SDD0
MISO
SDCLK
SCLK
GND
3.3V
EN
RST
GND
Vin
GPIO16
USER
WAKE
GPIO5
GPIO4
GPIO0
FLASH
GPIO2
TXD1
3.3V
GND
GPIO14
HSCLK
GPIO12
HMISO
GPIO13
RXD2
HMOSI
GPIO15
TXD2
HCS
GPIO3
RXD0
GPIO1
TXD0
GND
3.3V
SILABS CP2102
RST
FLASH

A Mention of Python

Python is a powerful high-level, object-oriented programming language created by Guido van Rossum.

It has simple easy-to-use syntax, making it the perfect language for someone trying to learn computer programming for the first time.

Python is used in almost everything! From data Science to Web programming. Python is increasingly being used to program embedded systems. It is worthwhile to learn Python. The link below gives a more detailed description of Python in embedded systems comparing it with the more commonly used languages of C/C++.

https://opensource.com/life/16/8/python-vs-cc-embedded-systems

I have included a simple Python Introductory book in the class folder. Read it! Even if you won't use it to program IoT components. Its syntax is quite different from that of C/C++ which ensures you don't get confused with the 2 languages.

Practical

Now Let's Do Some Practical's with The ESP8266 MOD12!

We have not started to learn about communication in IoT but we would be underutilising the ESP8266 if we only used it for the functions that the standard Atmega 328P can do. So we will read some values and print these values on an online dashboard. I have a predesigned circuit on a proto-board, but the same can be replicated on the breadboard. The circuit description is found in the folder named 'documents' which you will find in the class main folder. The code is also provided in the class folder. If it seems too overwhelming, try to look at each of the different functionalities separately(e.g. reading the humidity and temp sensor, reading the light sensor, sending the values etc) then work your way up to how you may make one program to do all those individual functions.

Please head over to io.adafruit.com and register an account and let's explore the ESP8266!