

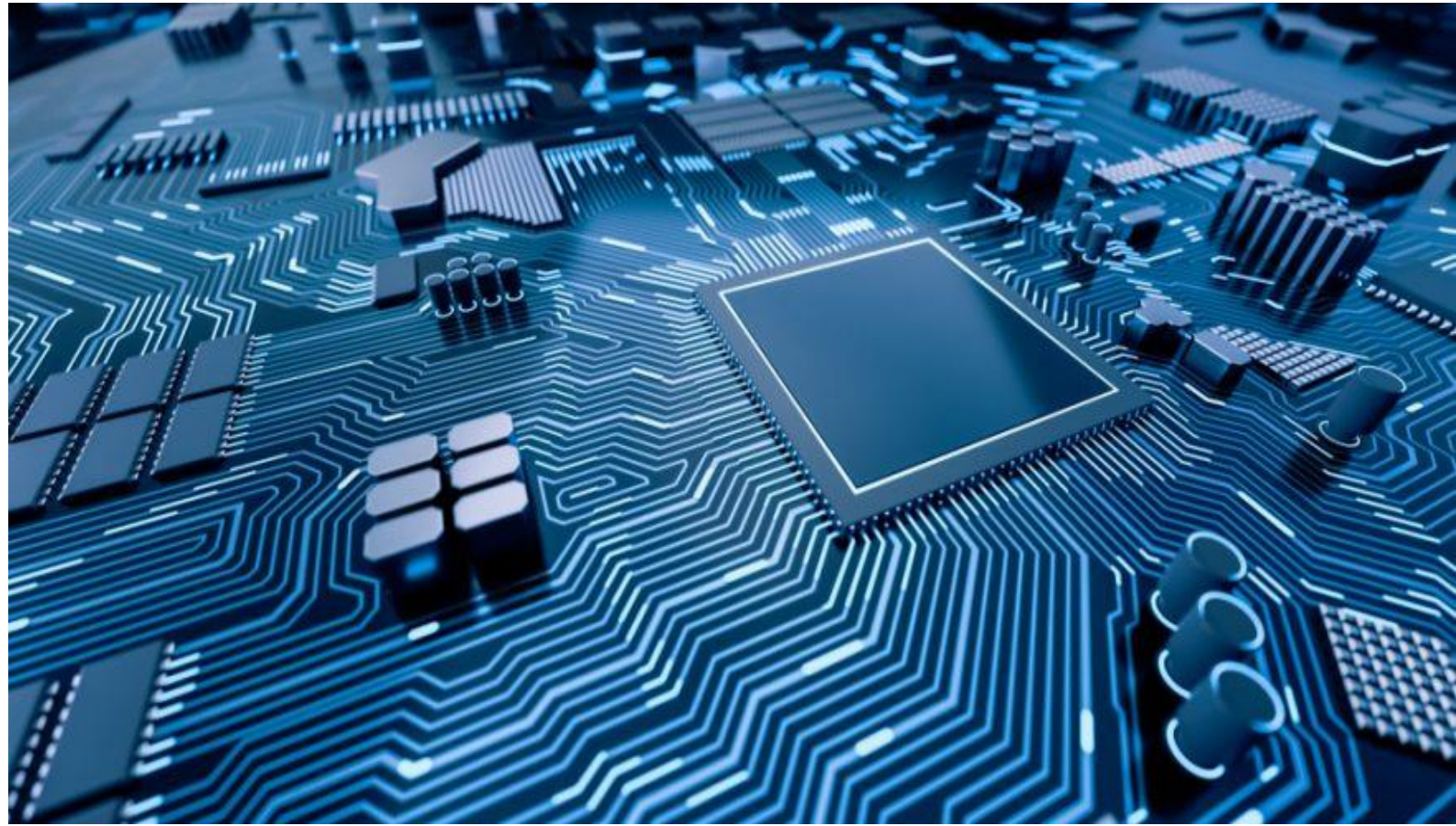


Đại học Công Nghệ - ĐHQGHN



KHOA ĐIỆN TỬ - VIỄN THÔNG

Bộ môn Mạng truyền thông



Internet of Things: Practice

- GV: Nguyễn Ngọc Tân
- Email: nguyen.tan17089@gmail.com
- SĐT: 0904.183.123

CONTENTS

❖ Getting Started with Embedded Programming

- Introduction to Embedded systems
 - Working with ESP8266
- Embedded programming

❖ Controlling Actuators

- Controlling LED light, DC motor, relay, v.v.

❖ Getting Sensor Data

- Temperature/humidity sensors, gas sensor, light sensor, sonar sensor, v.v.

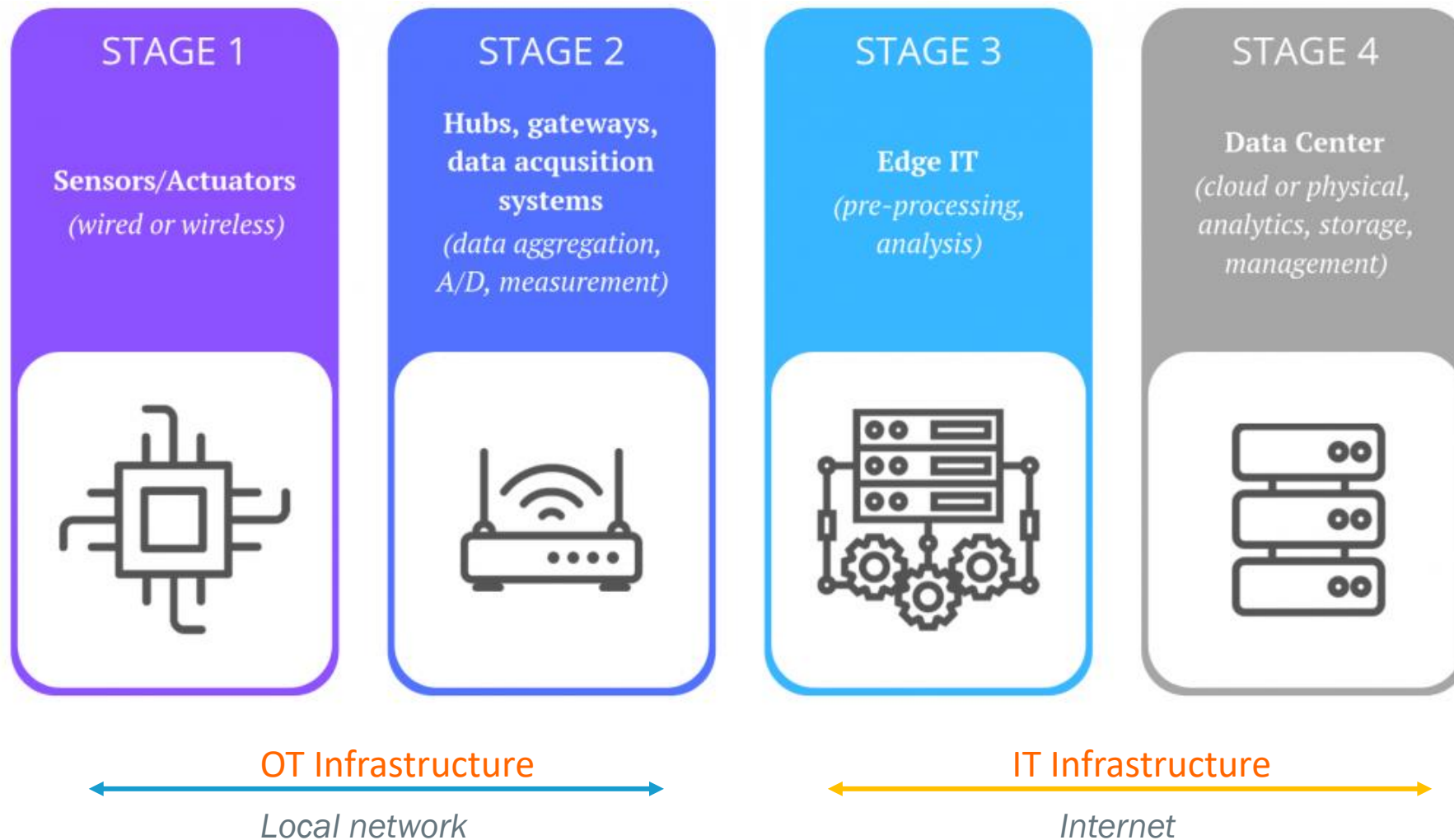
❖ IoT Communication Protocols

- ESP8266: Connecting to Wi-Fi
- HTTP, MQTT protocols

❖ Open IoT Platforms

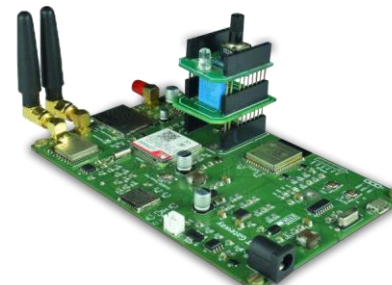
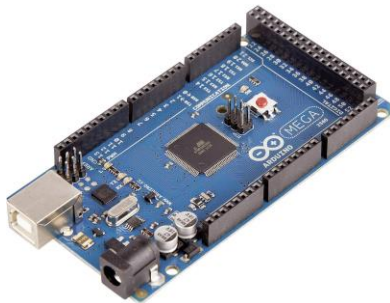
- Blynk, Thingsboard, v.v.

Horizontal Architecture of IoT system



Getting started with Embedded Programming

- ❖ Introduction to Embedded systems
 - Working with ESP8266
- ❖ Embedded programming





**What is an
embedded
system?**

Introduction to Embedded systems

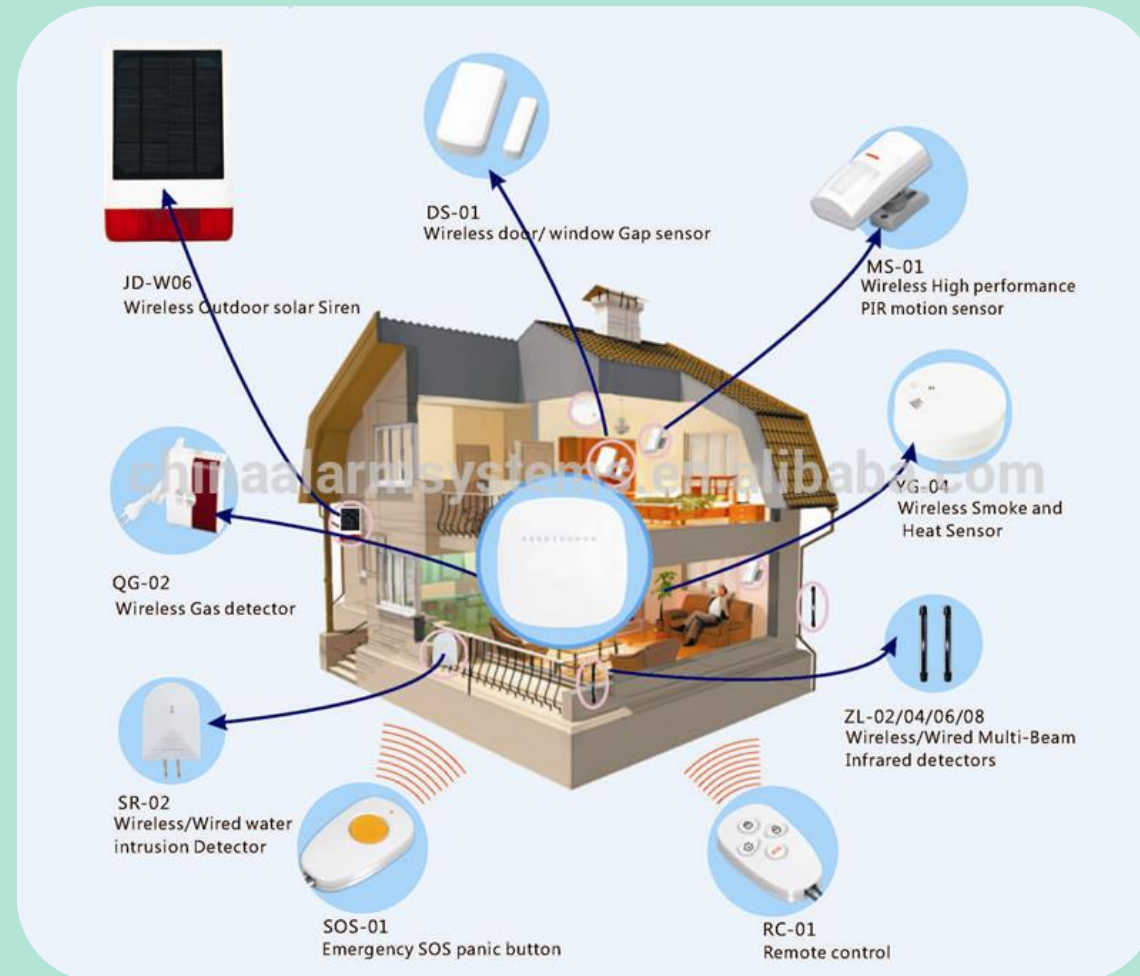
❖ Definitions

- An **embedded system** is one that has **computer hardware with software embedded in it** as one of its components.
- An **embedded system** is a **special-purpose computer system** designed to perform certain dedicated functions. In some cases, embedded systems are **part of a larger system or product**, as is the case of an antilock braking system in a car.



Wide range of Applications

Smart Home



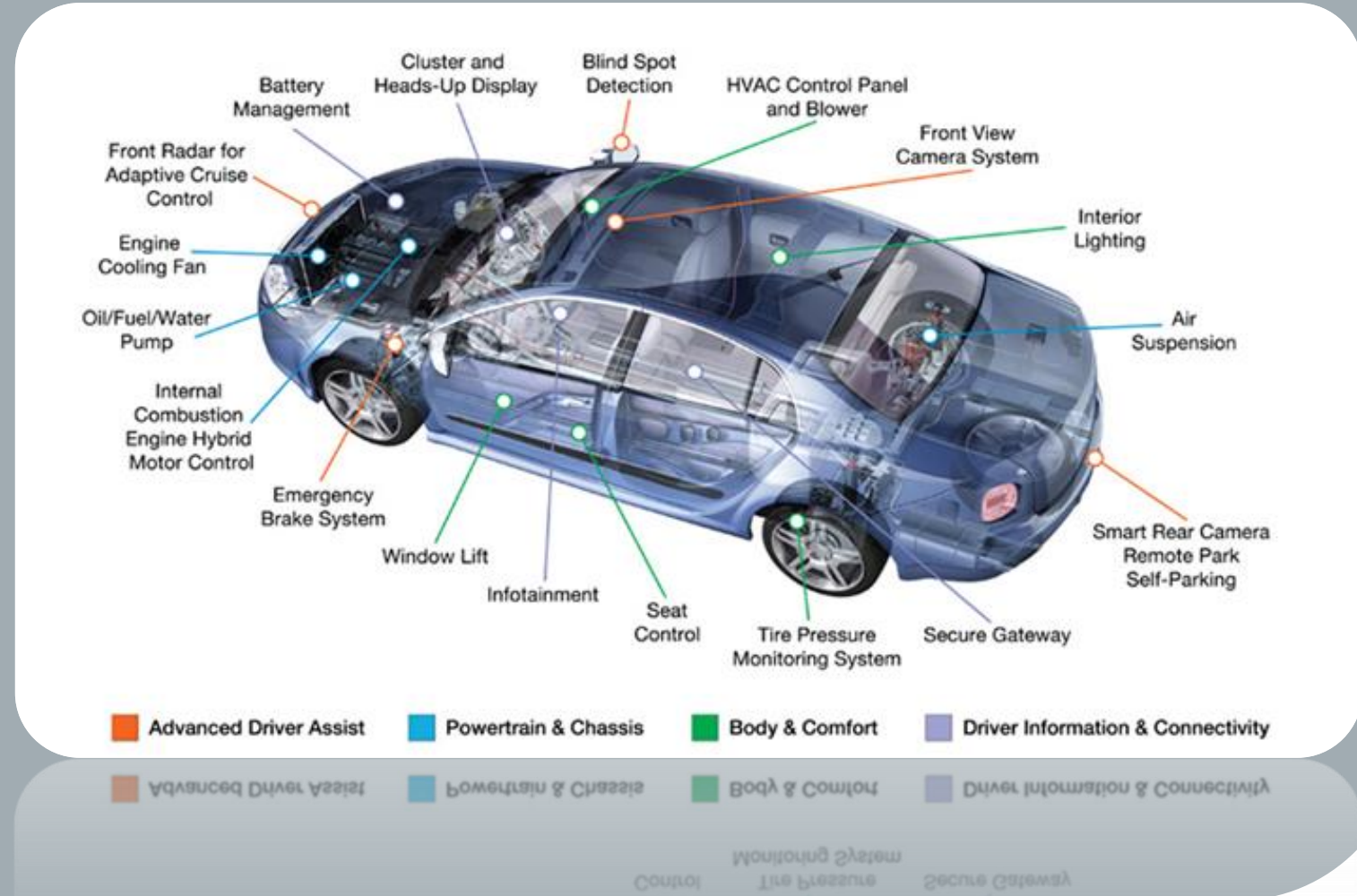
Wide range of Applications



In Kitchen

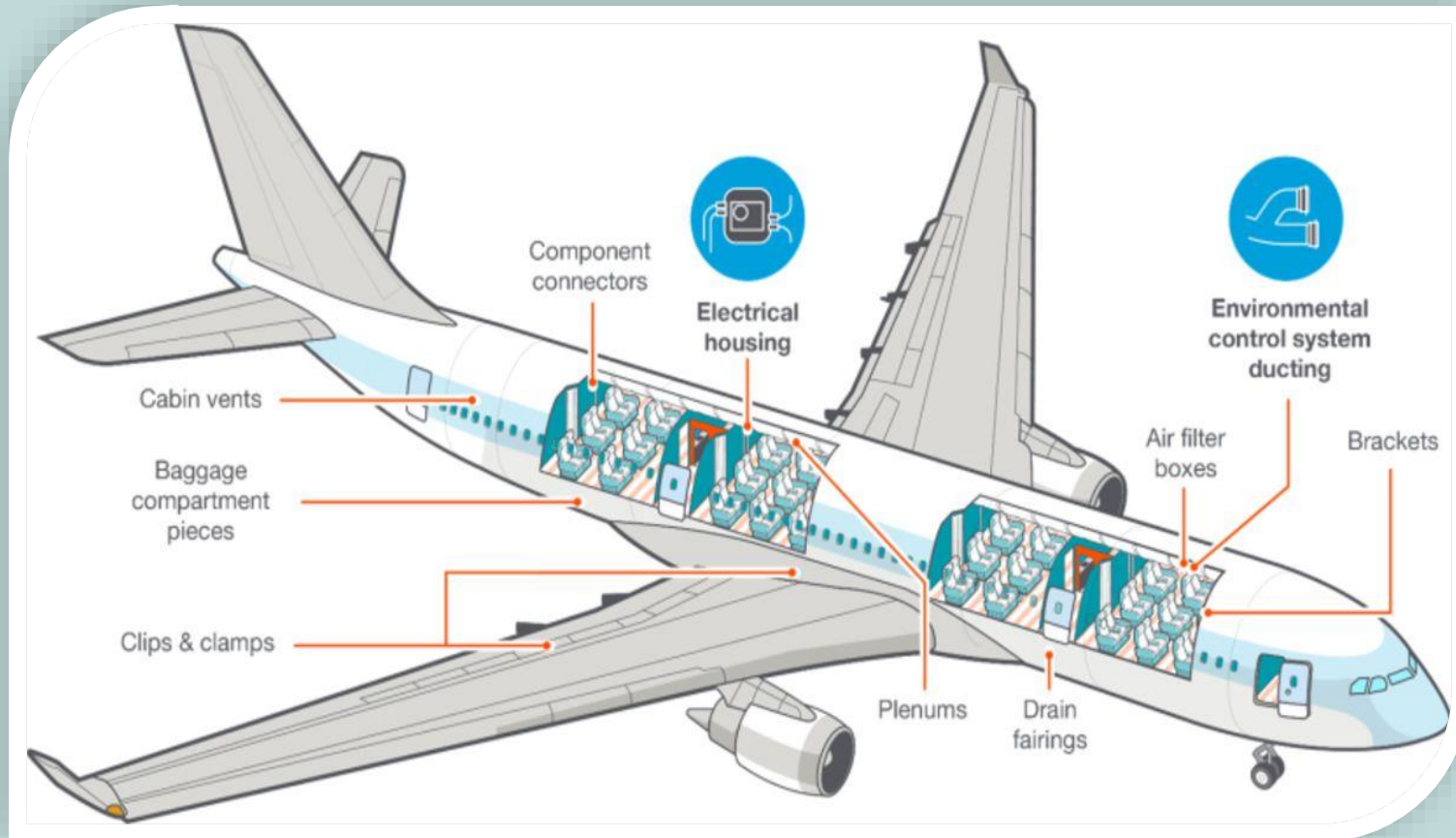
Wide range of Applications

In Cars



Wide range of Applications

In Aircraft



Wide range of Applications



In Factory

Characteristics

❖ Special-purpose

- Typically, is designed to execute a single program, repeatedly
- It used to be single-purpose
- Nowadays, multi-functioned, but single-purpose



Characteristics

❖ Tightly constrained

- Low cost
- Simple systems
- Fewer components based
- Performs functions fast enough
- Minimum power



Characteristics

❖ Reactive and real-time

- Reactive: Continually reacts to external events
- Real-time: Must compute certain results in real-time



Characteristics

❖ Hardware and software co-exist

- The software written for embedded systems is often called firmware
- Is stored in read-only memory or Flash memory chips rather than a disk drive
- Using a microcontroller rather than a microprocessor



Characteristics

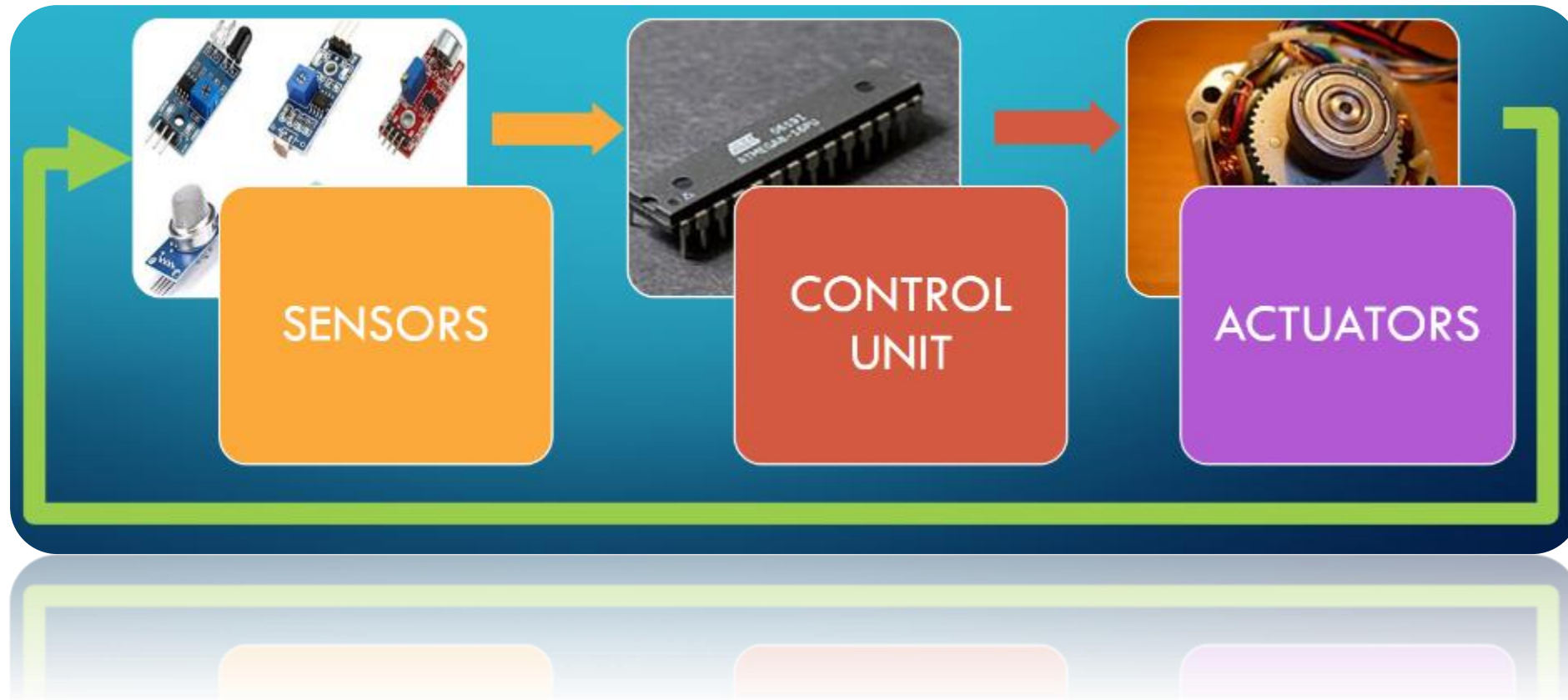
❖ Why a microcontroller

- A microcontroller is a single silicon chip with memory and all Input/Output peripherals on it. Hence a microcontroller is also popularly known as a single chip computer. Normally, a single microcomputer has the following features:

- Arithmetic and logic unit
- Memory for storing program
- EEPROM for nonvolatile data storage
- RAM for storing variables and special function registers
- Input/output ports
- Timers and counters
- Analog to digital converter
- Circuits for reset, power up, serial programming, debugging
- Instruction decoder and a timing and control unit
- Serial communication port

So, its no wonder to say that the microcontroller is the most sought-after device for designing an efficient embedded system.

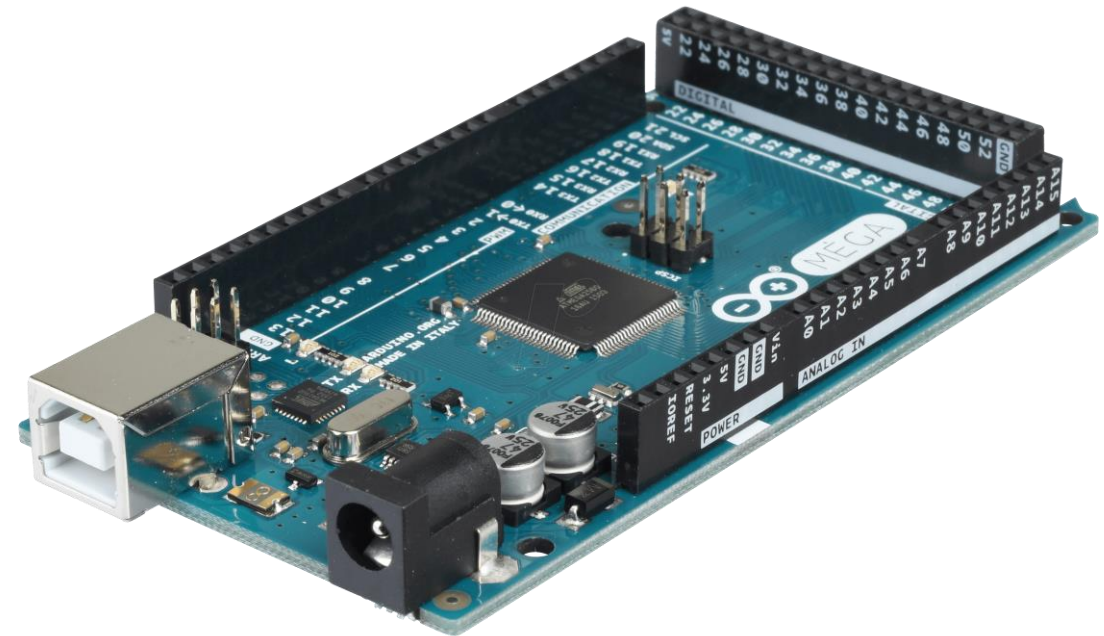
Block Description of an Embedded System



Common Development KITs

❖ Arduino MEGA 2560 R3

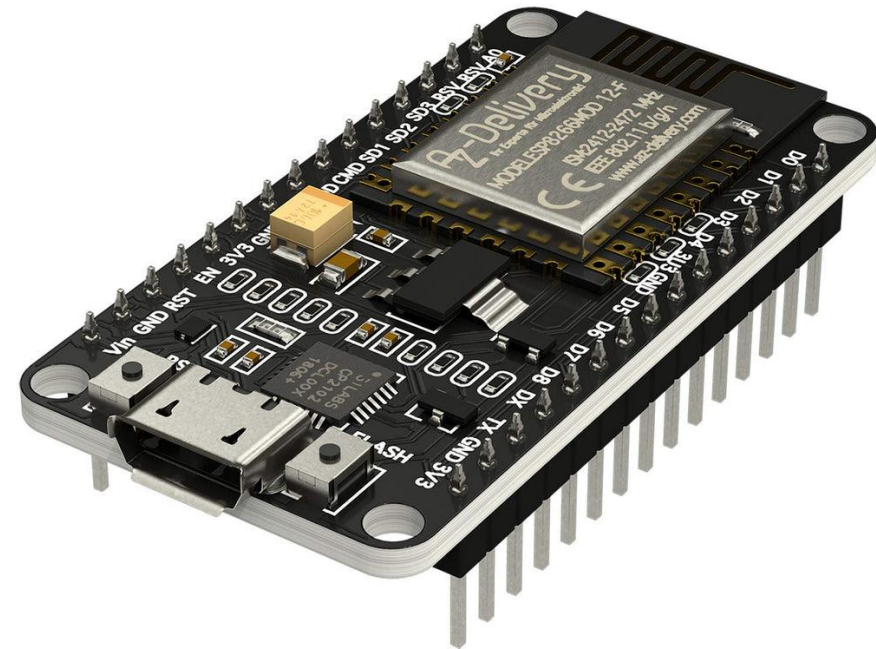
- Operating Voltage: 5V
- Input Voltage (recommended): 7 → 12V
- Input Voltage (limit): 6 → 20V
- Digital I/O Pins: 54 (of which 15 provide PWM output)
- Analog Input Pins: 16
- DC Current per I/O Pin: 20 mA
- DC Current for 3.3V Pin: 50 mA
- Flash Memory: 256 KB of which 8 KB used by bootloader
- SRAM: 8 KB
- EEPROM: 4 KB
- Clock Speed: 16 MHz



Common Development KITs

❖ NodeMCU ESP8266

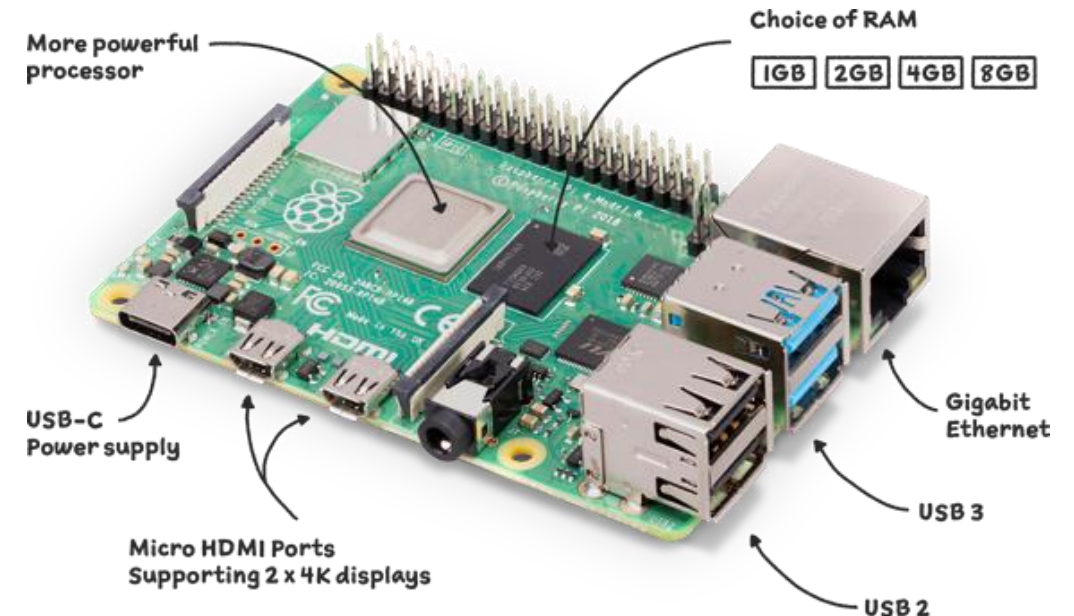
- Operating Voltage: 3.3V
- Input Voltage (recommended): 4.5V \rightarrow 10V
- Input Voltage (limit): 6 \rightarrow 20V
- Digital I/O Pins: 11
- Analog Input Pins: 1
- Flash Memory: 4 MB
- SRAM: 64 KB
- Clock Speed: 80 MHz
- Wi-Fi built in: 802.11b/g/n



Common Development KITs

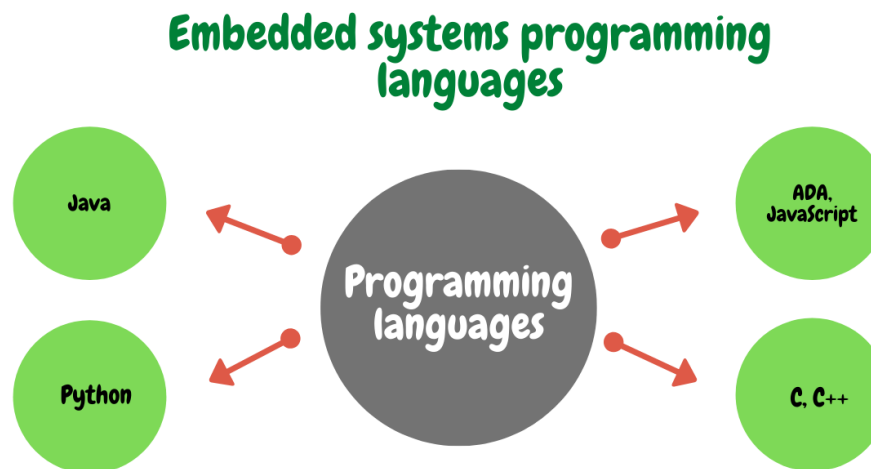
❖ Raspberry Pi 4 Model B:

- CPU: Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
- RAM: 1/2/4/8GB LPDDR4-3200 SDRAM
- Wi-Fi: 802.11ac (2.4 GHz & 5 GHz)
- Bluetooth: ver. 5.0 BLE
- LAN: GigaEthernet
- Input Voltage: 5V DC
- Input Current: minimum 2.5A
- Digital I/O Pins: 40
- 2 USB 3.0 ports; 2 USB 2.0 ports
- 2 × micro-HDMI ports (up to 4kp60 supported)
- 2-lane MIPI DSI display port
- 2-lane MIPI CSI camera port
- 4-pole stereo audio and composite video port



Languages for Programming Embedded Systems

- Assembly language was the pioneer for programming embedded systems till recently. Nowadays, there are many more languages to program these systems such as C, C++, Ada, Forth, Python, and Java.
- The majority of software for embedded systems is still done in C language (~45%). It is due to the fact that C is very close to assembly programming and it allows very easy access to underlying hardware.
- C++ is also increasing its presence in embedded systems thanks to the benefits of Object-oriented programming.




ARDUINO IDE & First Embedded Program

❖ ARDUINO IDE

- The open-source Arduino Software (IDE) makes it easy to write code and upload it to any Arduino board.
- Download it at: <https://www.arduino.cc/en/software>, or use Online version at: <https://create.arduino.cc/editor>.

Downloads

**Arduino IDE 1.8.19**

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.


Refer to the [Getting Started](#) page for Installation instructions.

SOURCE CODE

Active development of the Arduino software is [hosted by GitHub](#). See the instructions for [building the code](#). Latest release source code archives are available [here](#). The archives are PGP-signed so they can be verified using [this gpg key](#).

DOWNLOAD OPTIONS

Windows Win 7 and newer
Windows ZIP file

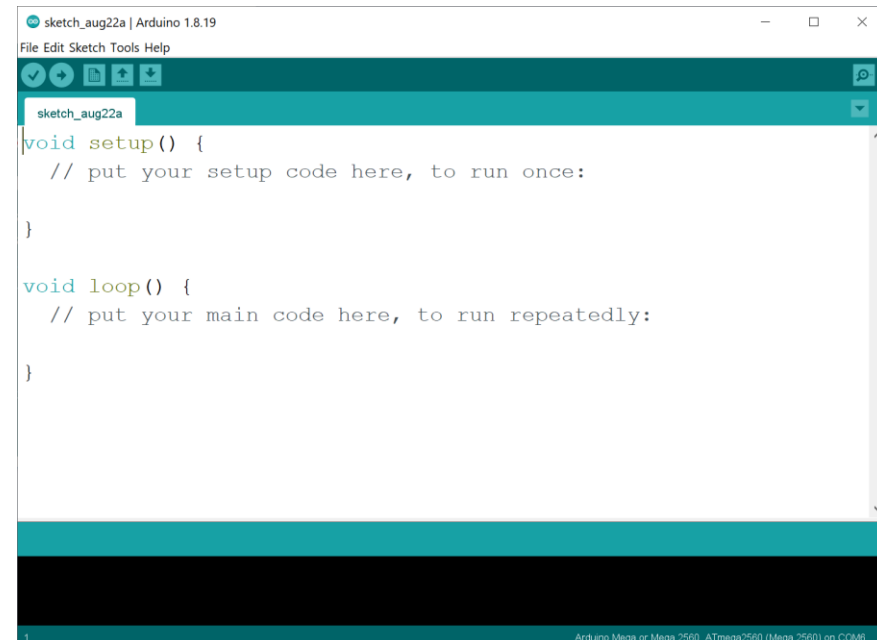
Windows app Win 8.1 or 10 

Linux 32 bits
Linux 64 bits
Linux ARM 32 bits
Linux ARM 64 bits

Mac OS X 10.10 or newer

[Release Notes](#)

[Checksums \(sha512\)](#)



```
sketch_aug22a
void setup() {
  // put your setup code here, to run once:
}

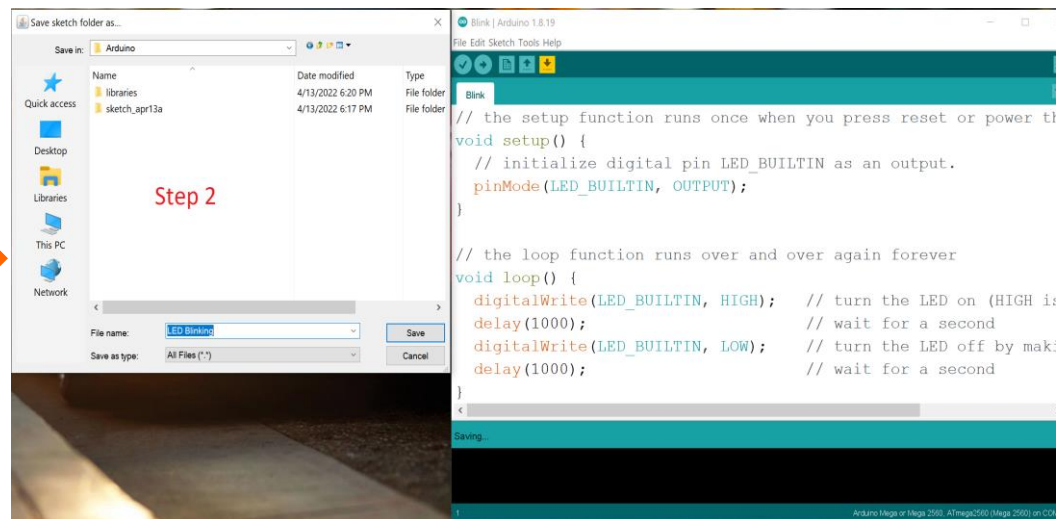
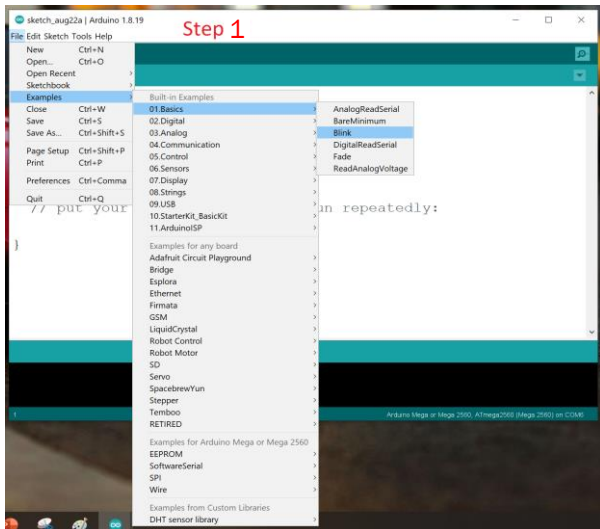
void loop() {
  // put your main code here, to run repeatedly:
}
```

Arduino Mega or Mega 2560, ATmega2560 (Mega 2560) on COM6

ARDUINO IDE & First Embedded Program

❖ First Embedded Program – LED Blinking

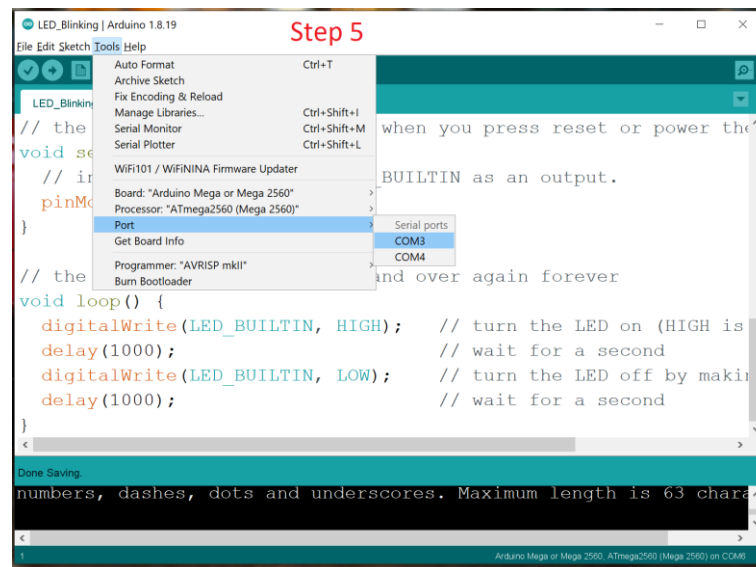
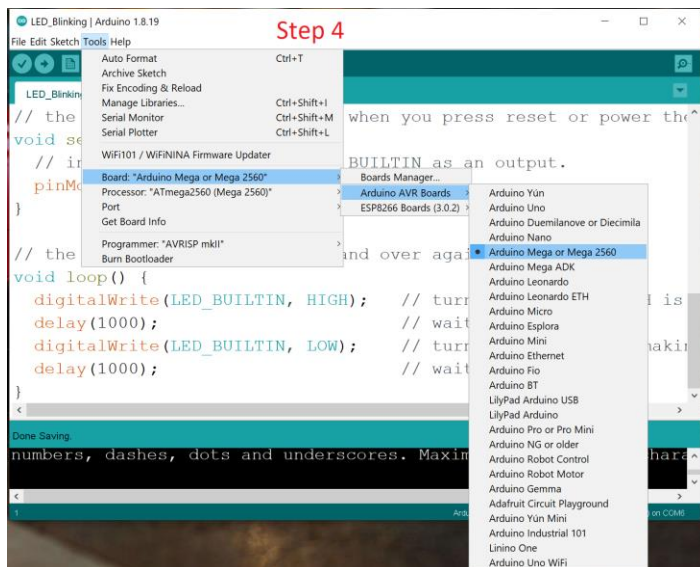
- Step 1: Open Arduino IDE → File → Examples → 01. Basics → Blink
- Step 2: Save as the new Arduino IDE window with the name “LED Blinking”
- Step 3: Connect the Arduino board with your computer via a USB cable



ARDUINO IDE & First Embedded Program

❖ First Embedded Program – LED Blinking

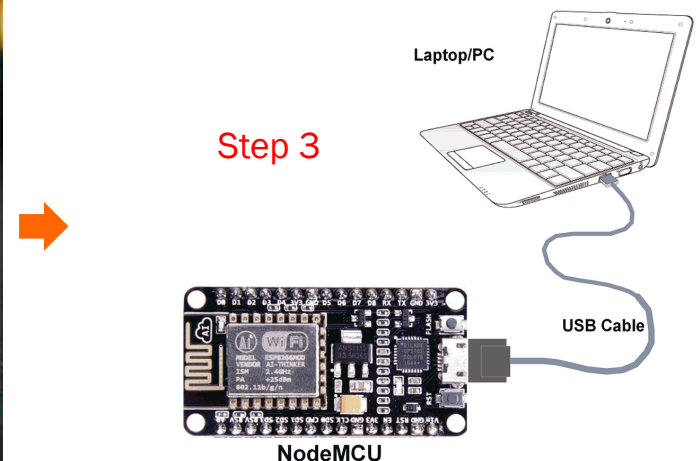
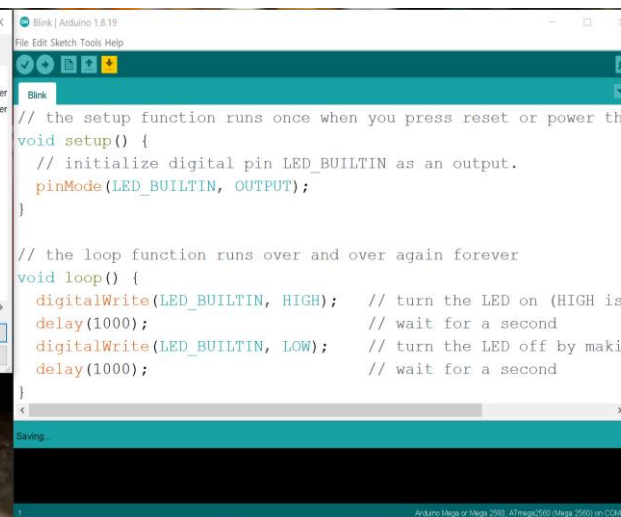
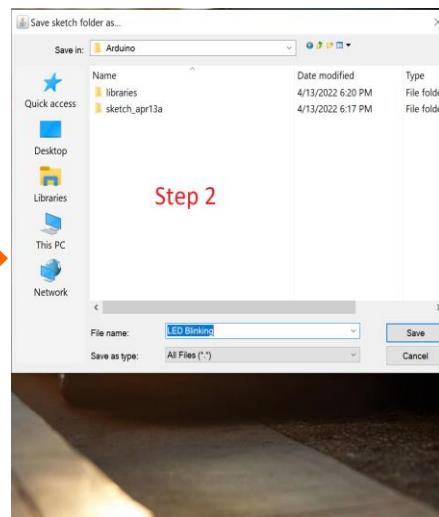
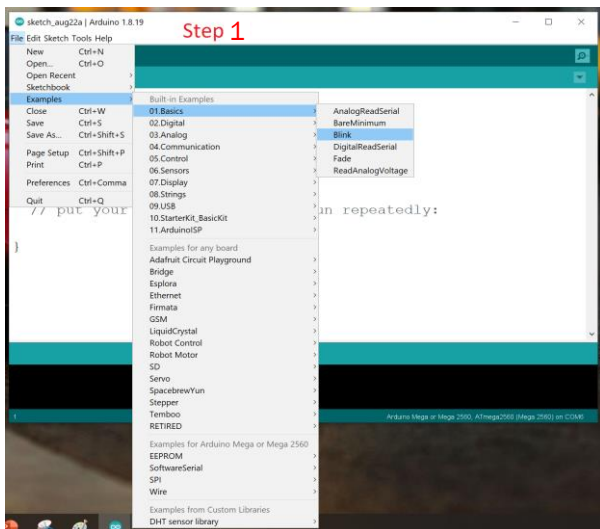
- Step 4: Select “Tools → Board → Arduino AVR Boards → Arduino Mega or Mega 2560”
- Step 5: Select “Tools → Port → COM?”
- Step 6: Press “Upload” button to compile + upload the source code to Arduino board.
- Step 7: Observe LED blinking on Arduino board when getting the status “Done uploading”



ARDUINO IDE & First Embedded Program

❖ First Embedded Program – LED Blinking (for ESP8266)

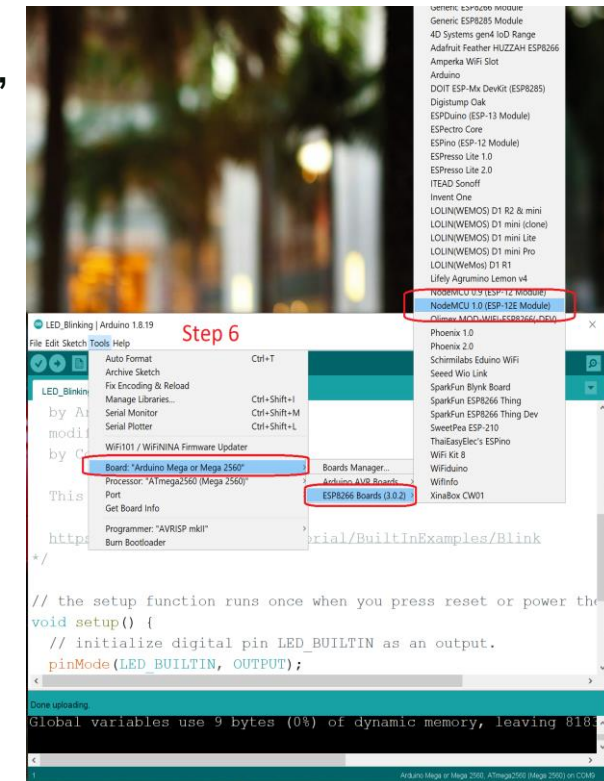
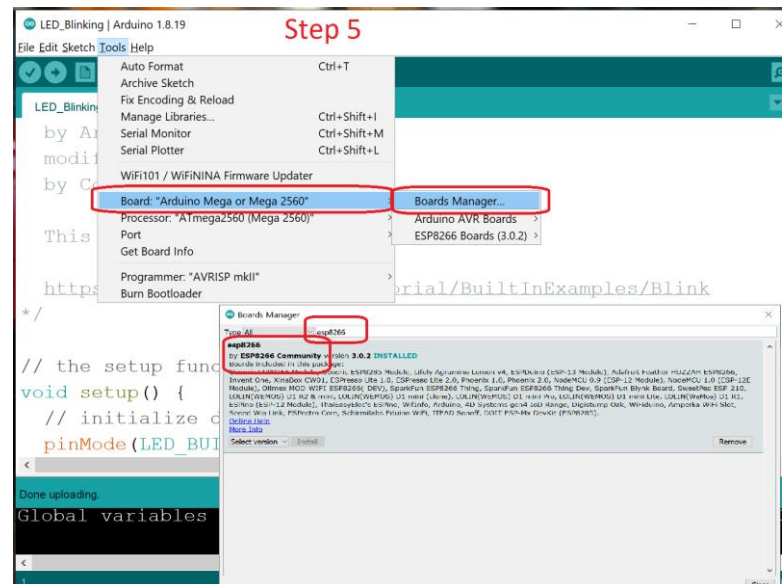
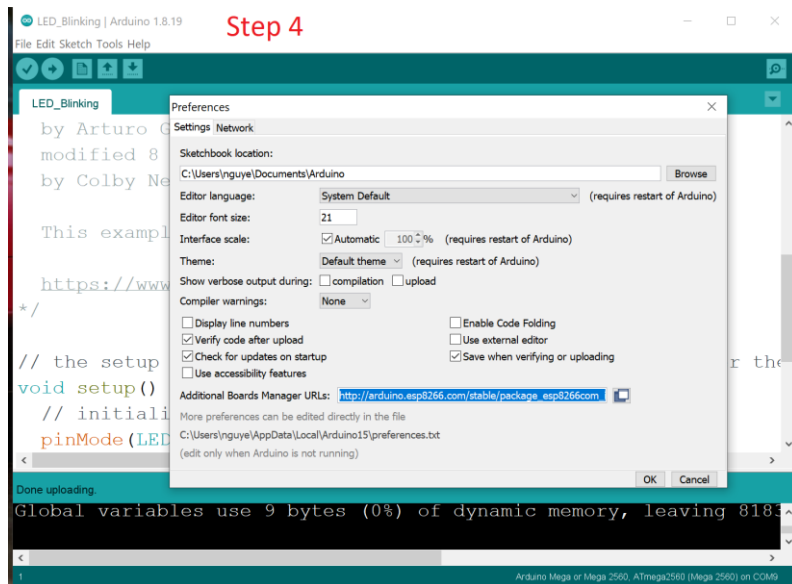
- Step 1: Open Arduino IDE → File → Examples → 01. Basics → Blink
- Step 2: Save as the new Arduino IDE window with the name “LED Blinking”
- Step 3: Connect the Arduino board with your computer via a USB cable



ARDUINO IDE & First Embedded Program

❖ First Embedded Program – LED Blinking (for ESP8266)

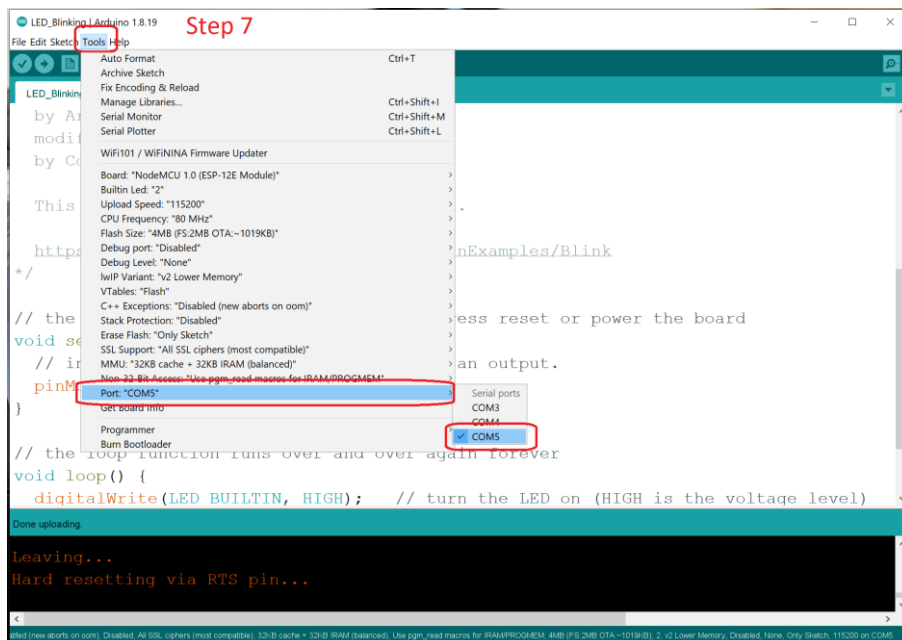
- Step 4: Select “File → Preferences → Additional Board Manager URLs”. Add the following link to this textbox: http://arduino.esp8266.com/stable/package_esp8266com_index.json. Then press “OK”
- Step 5: Select “Tools → Board → Board Manager”. Search “esp8266”, then install the searched package “ESP8266 by ESP8266 Community”
- Step 6: Select “Tools → Board → ESP8266 Boards → NodeMCU 1.0 (ESP-12 Module)”



ARDUINO IDE & First Embedded Program

❖ First Embedded Program – LED Blinking (for ESP8266)

- Step 7: Select “Tools → Port → COM?”
- Step 8: Press “Upload” button to compile + upload the source code to Arduino board.
- Step 9: Observe LED blinking on Arduino board when getting the status “Done uploading”





THANK YOU!