EDGE IMPULSE

Making things smarter

Copyright © Edge Impulse Inc

GLOBAL AGENDA

Embedded Programming

Lectures (1 & 2):	01/10/2021
Lab 1:	01/10/2021

IoT communication protocols

Lecture 1:	06/10/2021
Lab 1:	15/10/2021

Introduction to Embedded Machine Learning

Lecture 3:	12/11/2021
Lab 2:	12/11/2021

WHO AM I ?

Louis Moreau (@luisomoreau)

I studied in HEI Lille for 6 years 🤔

I started my career connecting
African rhinoceroses 🦏
@Sigfox (~3 years)

Then, I worked as a **Full-Stack Engineer**
@Goodeed (@KissKissBankBank) (~ 6 months)

After I did some **Technical Marketing Management**
@Scaleway (~1.5 year)

Now am a **User Success Engineer**
@Edge Impulse

AGENDA

Embedded Programming 1/3

1. Introduction
2. Microcontrollers
3. Basics of embedded programming
4. Understand an Arduino Program

Embedded Programming 1/3

1. Introduction

Introduction

What is Embedded Programming?

Embedded programming is a specific type of programming that supports the creation of consumer facing or business facing devices **that don't operate on traditional operating systems** the way that full-scale laptop computers and mobile devices do. The idea of embedded programming is part of what drives the evolution of the digital appliances and equipment in today's IT markets.

TL;DR

Some experts define embedded programming as the dominant methodology for **microcontroller programming**.

Source: https://www.techopedia.com/definition/29945/embedded-programming

Embedded Programming 1/3

1. Introduction

Introduction

Low-end MCU | MCU + FPU | High-end MCU | CPU | CPU + GPU

Source: https://www.techopedia.com/definition/29945/embedded-programming

Embedded Programming 1/3

1. Introduction
2. Microcontrollers

@luisomoreau

MCU

What is a microcontroller?

A microcontroller unit (MCU) is a **small, self-contained computer** that is housed on a **single integrated circuit, or microchip**. They differ from your desktop computer in that they are **typically dedicated to a single function**, and are most often embedded in other devices (e.g. cellphones; household electronics, **IoT devices**).

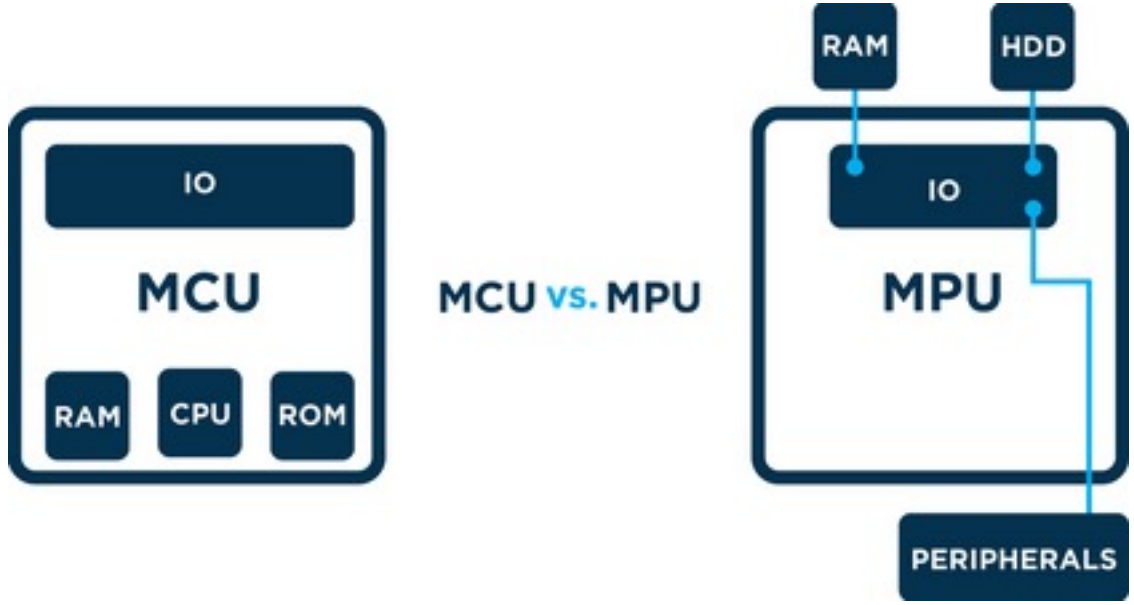


Image source: https://www.particle.io/

Embedded Programming 1/3

1. Introduction
2. Microcontrollers

@luisomoreau

MCU

What are the elements of a microcontroller?

Memory

A microcontroller's memory is used to store the data that the processor receives and uses to respond to instructions that it's been programmed to carry out. A microcontroller has two main memory types:

RAM

Data memory,
which is required for **temporary data storage** while the instructions are being executed. Data memory is volatile, meaning the data it holds is temporary and is only maintained if the device is connected to a power source.

ROM

Program memory,
which stores **long-term information** about the instructions that the CPU carries out. Program memory is non-volatile memory, meaning it holds information over time without needing a power source.

Embedded Programming 1/3

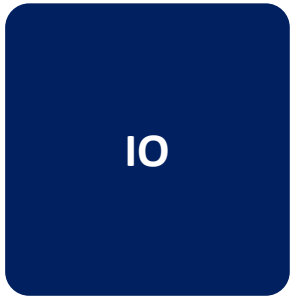
1. Introduction
2. Microcontrollers

@luisomoreau

MCU

What are the elements of a microcontroller?

I/O Peripherals:



The **input** and **output** devices are the interface for the processor to the outside world. The input ports receive information and send it to the processor in the form of binary data. The processor receives that data and sends the necessary instructions to output devices that execute tasks external to the microcontroller.

https://internetofthingsagenda.techtarget.com/definition/microcontroller

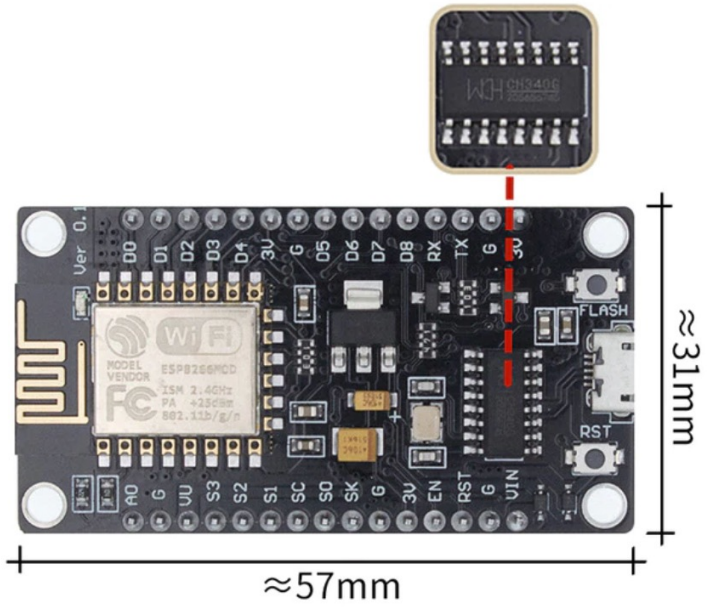
Embedded Programming 1/3

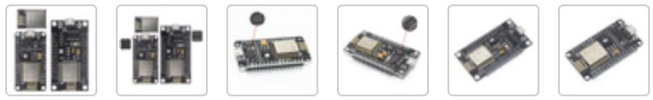
1. Introduction
2. Microcontrollers

@luisomoreau

NodeMCU ESP8266

What about the NodeMCU ESP8266?



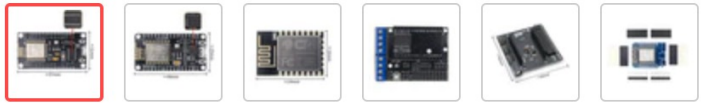


Wireless module CH340/CP2102 NodeMcu V3 V2 Lua WIFI Internet of Things development board based ESP8266 ESP-12E with pcb Antenna

★★★★★ 4.9 ⌄ 4275 Reviews 8393 orders

€ 1,67

Color: V3 Nodemcu-CH340



Quantity:

1 1% off (20 pieces or more)
68154 pieces available

Shipping: € 1,19 Free shipping for orders over € 45,99 via the selected shipping method

to France via Cainiao Super Economy Global ⌄

Estimated Delivery on 10/11

Buy Now | Add to Cart | ♡ 12.4K

90-Day Buyer Protection
Money back guarantee

Link to product: https://www.aliexpress.com/item/32665100123.html

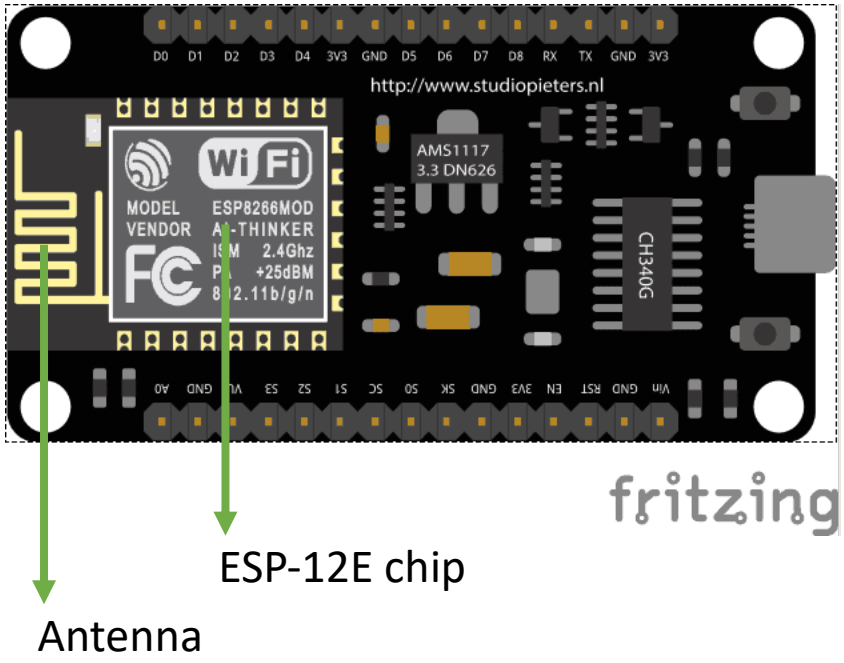
MCU main components

ESP-12E Module

The development board equips the ESP-12E module containing ESP8266 chip having Tensilica Xtensa® 32-bit LX106 RISC microprocessor which operates at 80 to 160 MHz adjustable clock frequency and supports RTOS.

ESP-12E Chip

- Tensilica Xtensa® 32-bit LX106
- 80 to 160 MHz Clock Freq.
- 128kB internal RAM
- 4MB external flash
- 802.11b/g/n **Wi-Fi** transceiver



MCU main components

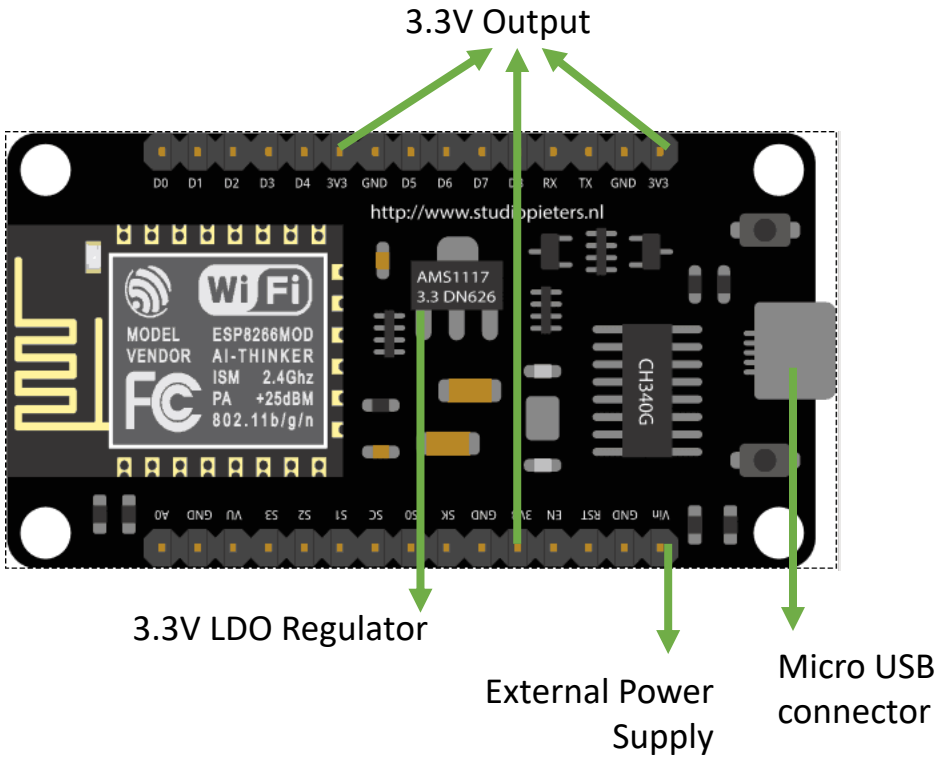
Example with the Nodemcu ESP8266

Power requirement

As the operating voltage range of ESP8266 is **3V to 3.6V**, the board comes with a LDO voltage regulator to keep the voltage steady at 3.3V. It can reliably supply up to 600mA, which should be more than enough when ESP8266 pulls as much as **80mA during RF transmissions**. The board can also supply power to external components (3V3 pin).

Power Requirement

- Operating Voltage: 2.5V to 3.6V
- On-board 3.3V 600mA regulator
- 80mA Operating Current
- 20 µA during Sleep Mode



MCU main components

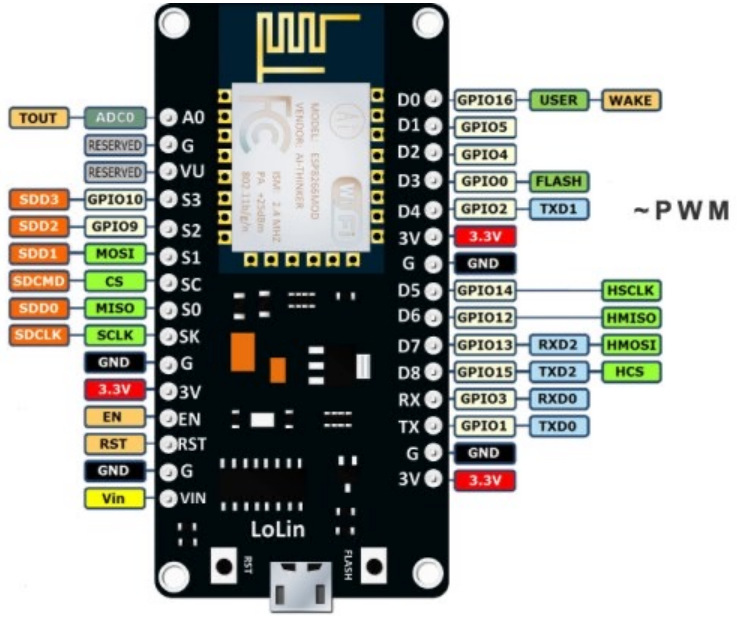
Example with the Nodemcu ESP8266

Peripherals and I/O

The ESP8266 NodeMCU has total 17 GPIO pins broken out to the pin headers on both sides of the development board. These pins can be assigned to all sorts of peripheral duties.

Multiplexed I/Os

- 1 ADC channels
- 2 UART interfaces
- 4 PWM outputs
- SPI, I2C & I2S interface



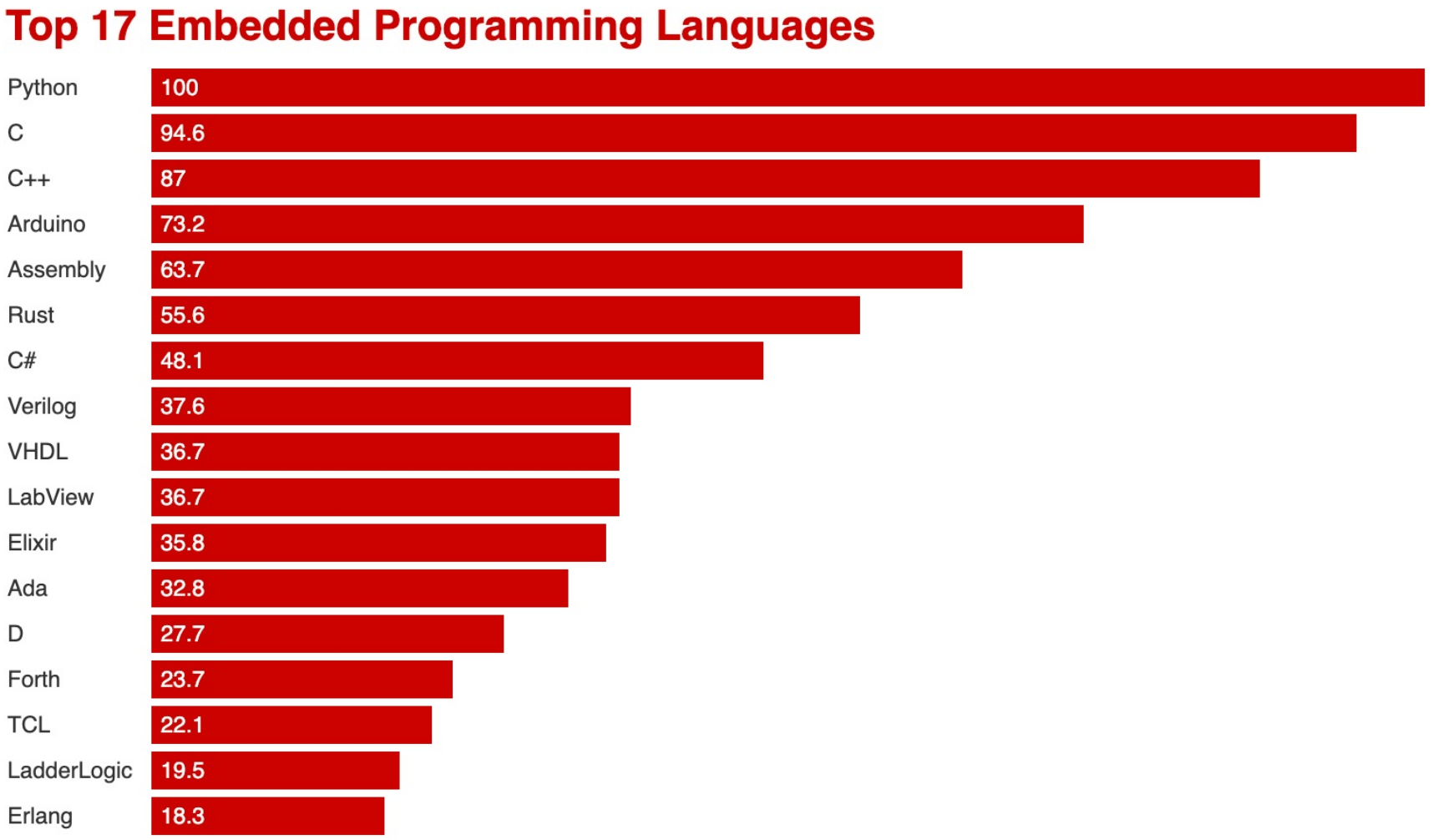
Embedded Programming 1/3

1. Introduction
2. Microcontrollers
3. Basics of Embedded Programming

@luisomoreau

Embedded Programming

Programming Languages:



Source: IEEE Spectrum

Embedded Programming

Programming Languages: MicroPython

MicroPython is a lean and efficient implementation of the Python 3 programming language that includes a small subset of the Python standard library and is optimised to run on microcontrollers and in constrained environments.

Code example 1

```
import pyb
# turn on an LED
pyb.LED(1).on()
# print some text to the serial console
print('Hello MicroPython!')
```

Source: https://micropython.org/

Embedded Programming 1/3

1. Introduction
2. Microcontrollers
3. Basics of Embedded Programming

@luisomoreau

Embedded Programming

Programming Languages: MicroPython

Code example 2 (blink)

```
from machine import Pin
# create an I/O pin in output mode
p = Pin('X1', Pin.OUT)
# toggle the pin
p.high()
p.low()
import time
while True:
        led.high()
        time.sleep(0.5)
        led.low()
        time.sleep(0.5)
```

Embedded Programming 1/3

1. Introduction
2. Microcontrollers
3. **Basics of Embedded Programming**

@luisomoreau

Embedded Programming

Programming Languages: C

C is a powerful general-purpose programming language. It can be used to develop software like operating systems, databases, compilers, and so on.

Code example 1 (with a STM32Cube microcontroller)

```
/* Infinite loop */

// GPIO_PIN_SET => 1
// GPIO_PIN_RESET => 0

static unsigned short pin_state = 0;

while (1) {
/* USER CODE END WHILE */
/* USER CODE BEGIN 3 */

pin_state = !pin_state;

// write pin state
HAL_GPIO_WritePin(LED1_GPIO_Port, LED1_Pin, pin_state);

// synchronous delay for 500 ms
HAL_Delay(500);
}
```

Source: https://medium.com/@rlamarrr/introduction-to-stm32cube-blinking-an-led-61469168f9e4

Embedded Programming 1/3

1. **Introduction**
2. **Microcontrollers**
3. **Basics of Embedded Programming**

@luisomoreau

Embedded Programming

Programming Languages: C

Project example: Part of the rhino tracker

https://www.hackster.io/luisomoreau/sigfox-gps-tracker-26ae94

- Overview
- Things
- Story
  - What will be covered by this tutorial
  - What is Sigfox
  - Any skills required
  - Step 1: Hardware
  - Step 2: Get started with Telecom Design SDK
  - Step 3: (Re)set the device with the original firmware
  - Step 4: Send your first Sigfox message
  - Step 5: View your message in Sigfox backend
  - Step 6: Import project
  - Step 7: Compile the sources
  - Step 8: Code explanation
  - Step 9: Flash project to the device
  - Step 10: View the geolocalisation
  - Step 11: Bravo
- Code
- Credits
- Comments (4)

Louis Moreau
Published August 25, 2016

Sigfox GPS Tracker

Make your own GPS tracker using Sigfox. So, ready to track your dog, bike, boat or even your shipment?

Advanced · Full instructions provided · 5 hours · 6,438

Embedded Programming 1/3

1. Introduction
2. Microcontrollers
3. Basics of Embedded Programming

@luisomoreau

Embedded Programming

Programming Languages: Assembly

Assembly language is **a low-level programming language** for a computer or other programmable device specific to a particular computer architecture in contrast to most high-level programming languages, which are generally portable across multiple systems. Assembly language is converted into executable machine code by a utility program referred to as an assembler.

Code example

```
main:
  sbi 0x04, 5              ; PORTB5 output
loop:                      ; main loop begin
  sbi 0x05, 5              ; PORTB5 high
  call delay_1000ms        ; delay 1s
  cbi 0x05, 5              ; 5 PORTB5 low
  call delay_1000ms        ; delay 1s
  rjmp  loop               ; main loop


delay_1000ms:              ; subroutine for 1s delay
                           ; initialize counters
  ldi r18, 0xFF            ; 255
  ldi r24, 0xD3            ; 211
  ldi r25, 0x30            ; 48
inner_loop:
  subi  r18, 0x01          ; 1
  sbci  r24, 0x00          ; 0
  sbci  r25, 0x00          ; 0
  brne  inner_loop
  ret
```

Embedded Programming 1/3

1. **Introduction**
2. **Microcontrollers**
3. **Basics of Embedded Programming**

@luisomoreau

Embedded Programming

Programming Languages: C++ / Arduino

The Arduino language is a **subset of C/C++**, where you can also use assembly for ultra-low level code. Due to their simplicity, the programs you write using the Arduino IDE are called **sketches**.
A compiler is used to transform code into object files.

Code example

```
int LED =13;  // The digital pin to which the LED is connected

void setup ( ) {
pinMode (LED, OUTPUT); //Declaring pin 13 as output pin
}

void loop( ) // The loop function runs again and again
{

digitalWrite (LED, HIGH); //Turn ON the LED
delay(1000); //Wait for 1sec

digitalWrite (LED, LOW); // Turn off the LED
delay(1000); // Wait for 1sec

}
```

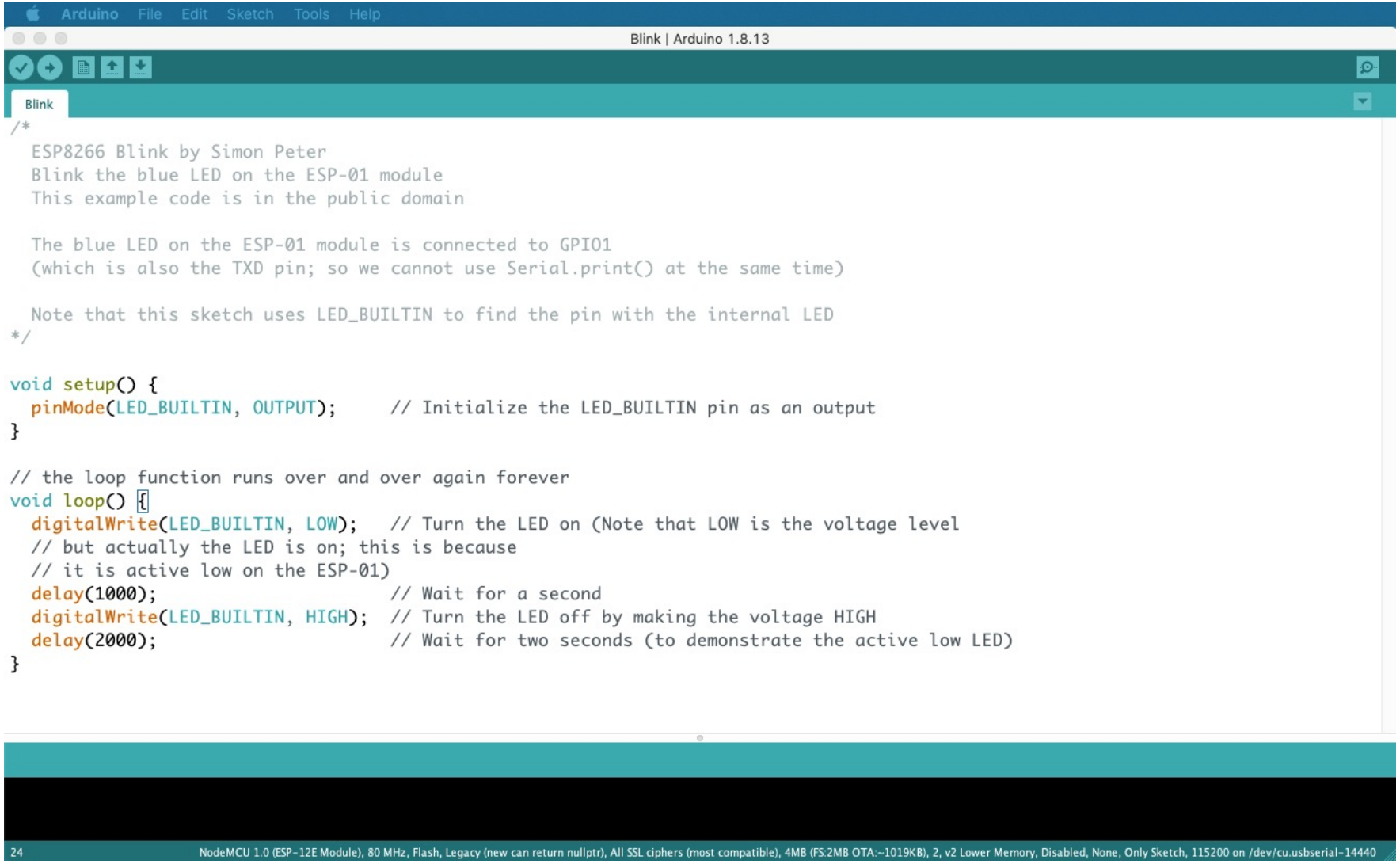
Embedded Programming 1/3

1. Introduction
2. Microcontrollers
3. Basics of Embedded Programming
4. Understand an Arduino Program

@luisomoreau

Arduino Programs

Arduino IDE



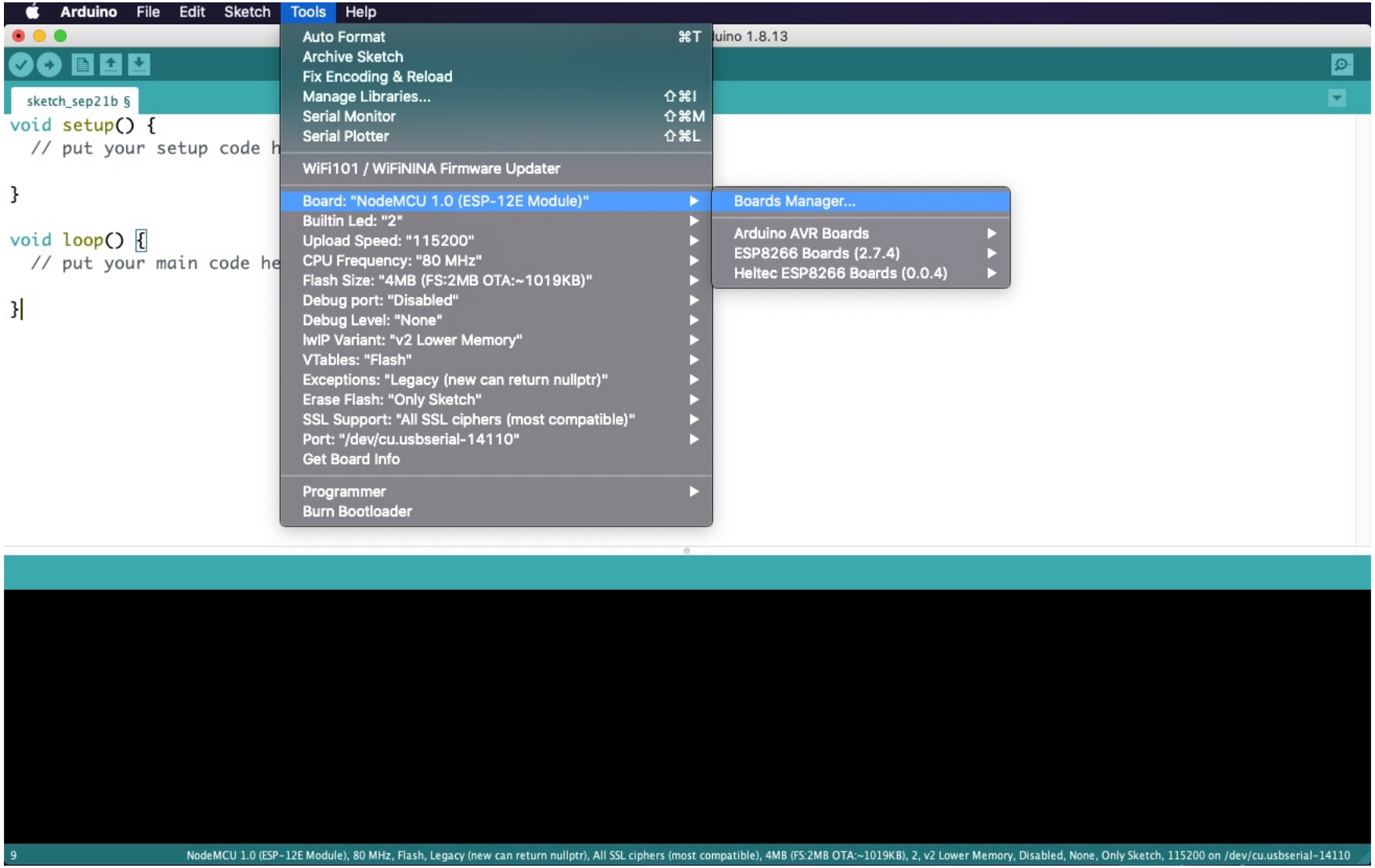
Embedded Programming 1/3

1. Introduction
2. Microcontrollers
3. Basics of Embedded Programming
4. Understand an Arduino Program

@luisomoreau

Arduino Programs

Arduino IDE – Board Manager



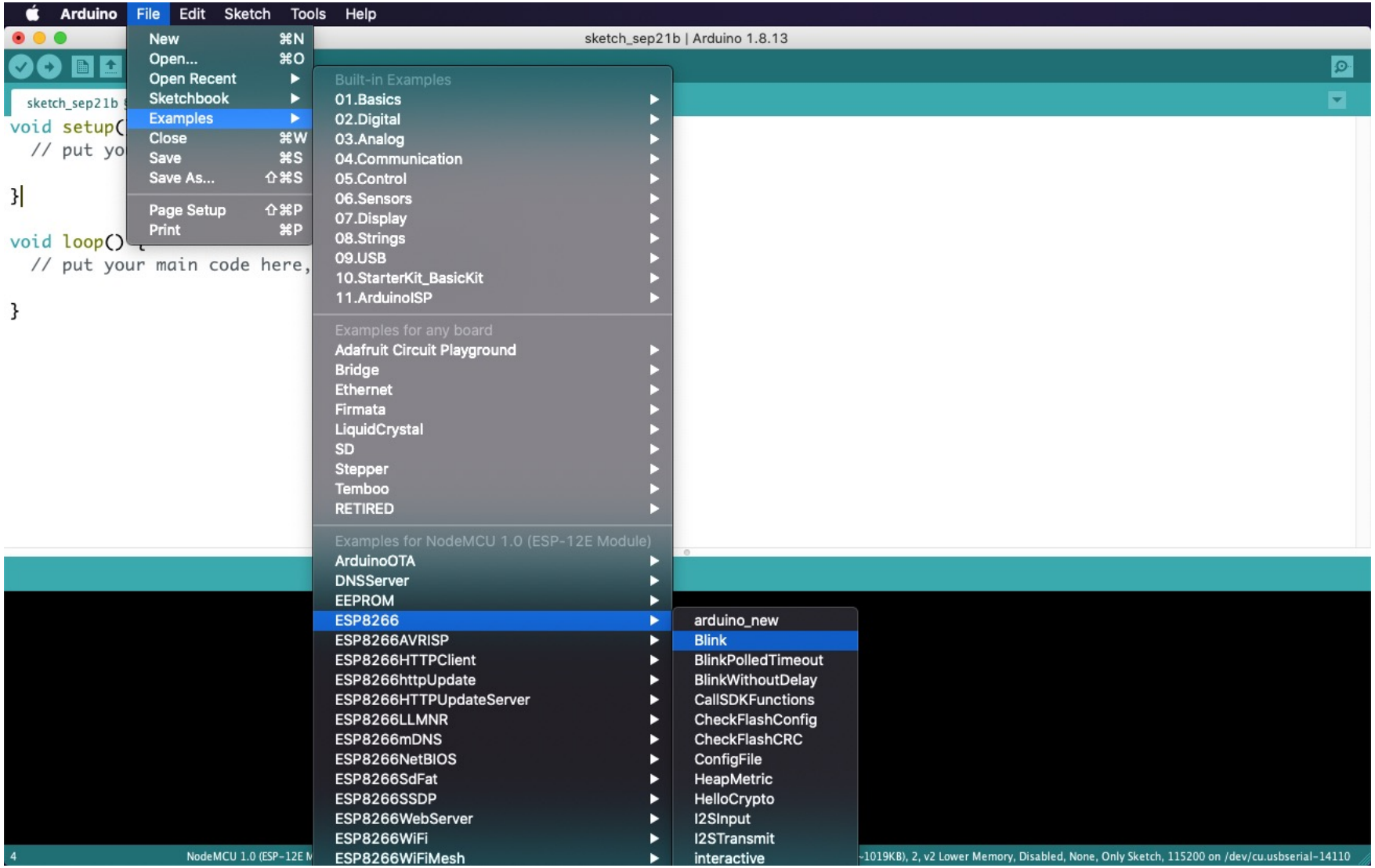
Embedded Programming 1/3

1. Introduction
2. Microcontrollers
3. Basics of Embedded Programming
4. Understand an Arduino Program

@luisomoreau

Arduino Programs

Arduino IDE – Examples



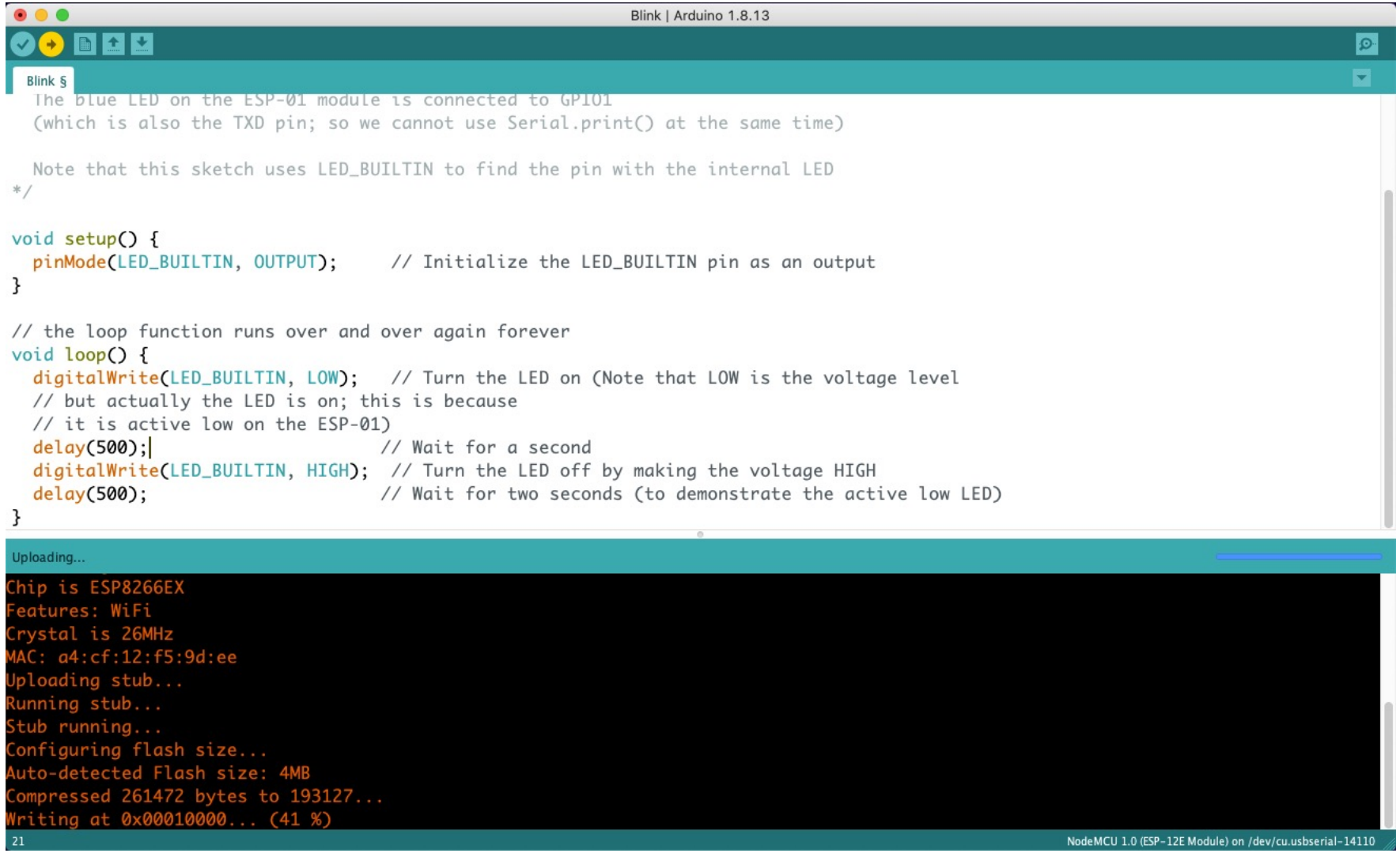
Embedded Programming 1/3

1. Introduction
2. Microcontrollers
3. Basics of Embedded Programming
4. Understand an Arduino Program

@luisomoreau

Arduino Programs

Arduino IDE – Upload your code



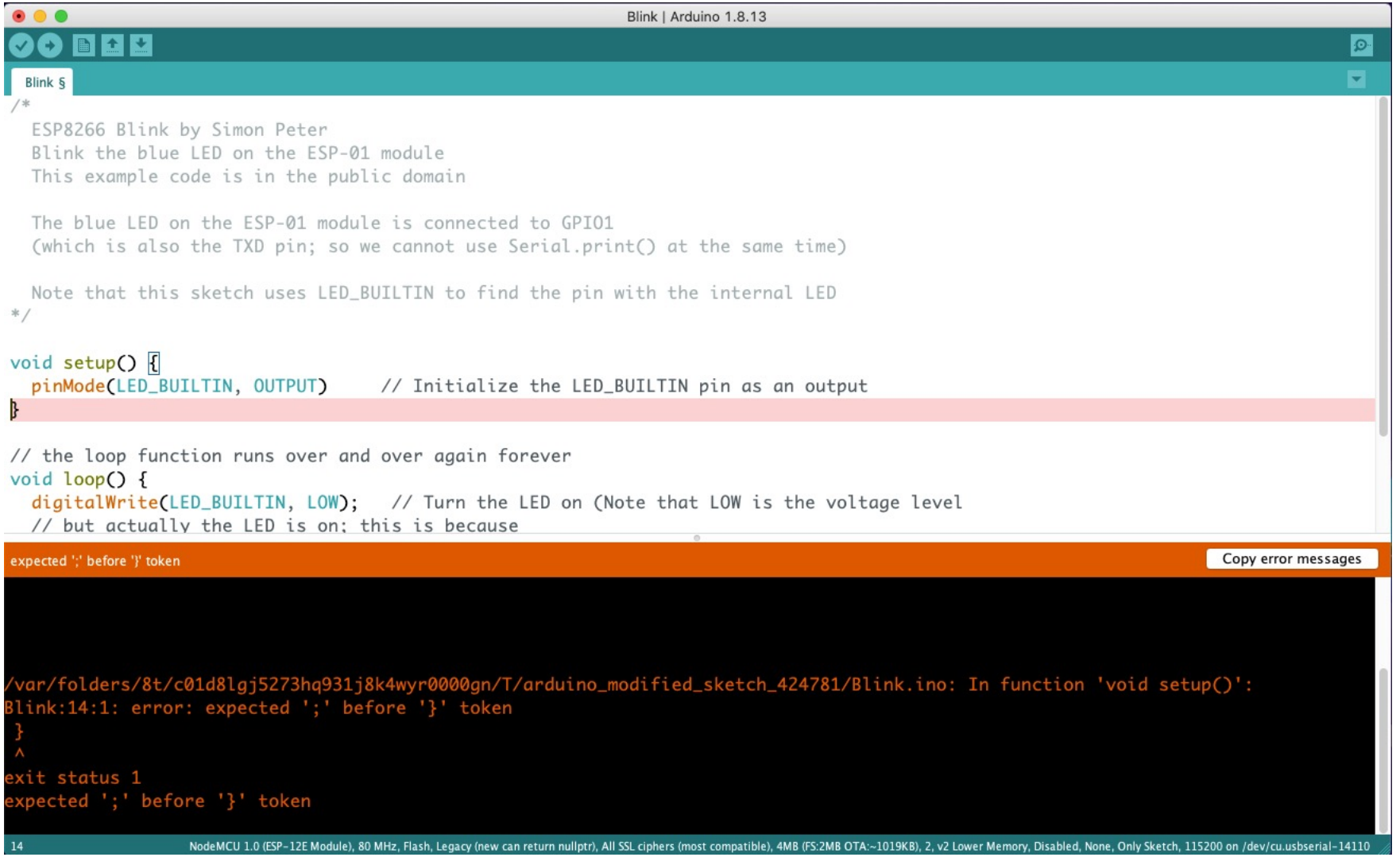
Embedded Programming 1/3

1. Introduction
2. Microcontrollers
3. Basics of Embedded Programming
4. Understand an Arduino Program

@luisomoreau

Arduino Programs

Arduino IDE – Upload your code - Error



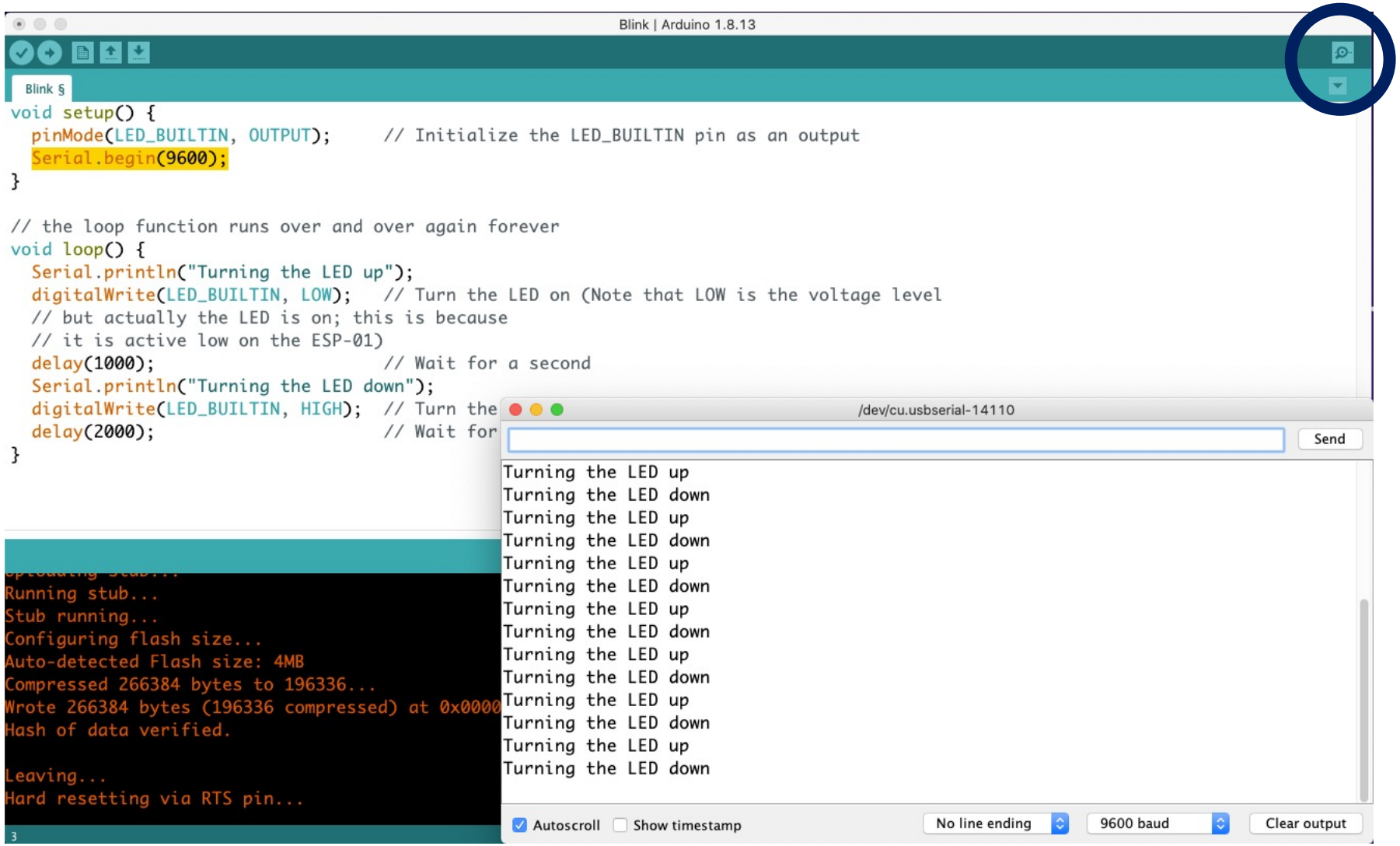
Embedded Programming 1/3

1. Introduction
2. Microcontrollers
3. Basics of Embedded Programming
4. Understand an Arduino Program

@luisomoreau

Arduino Programs

Arduino IDE – Serial Monitor

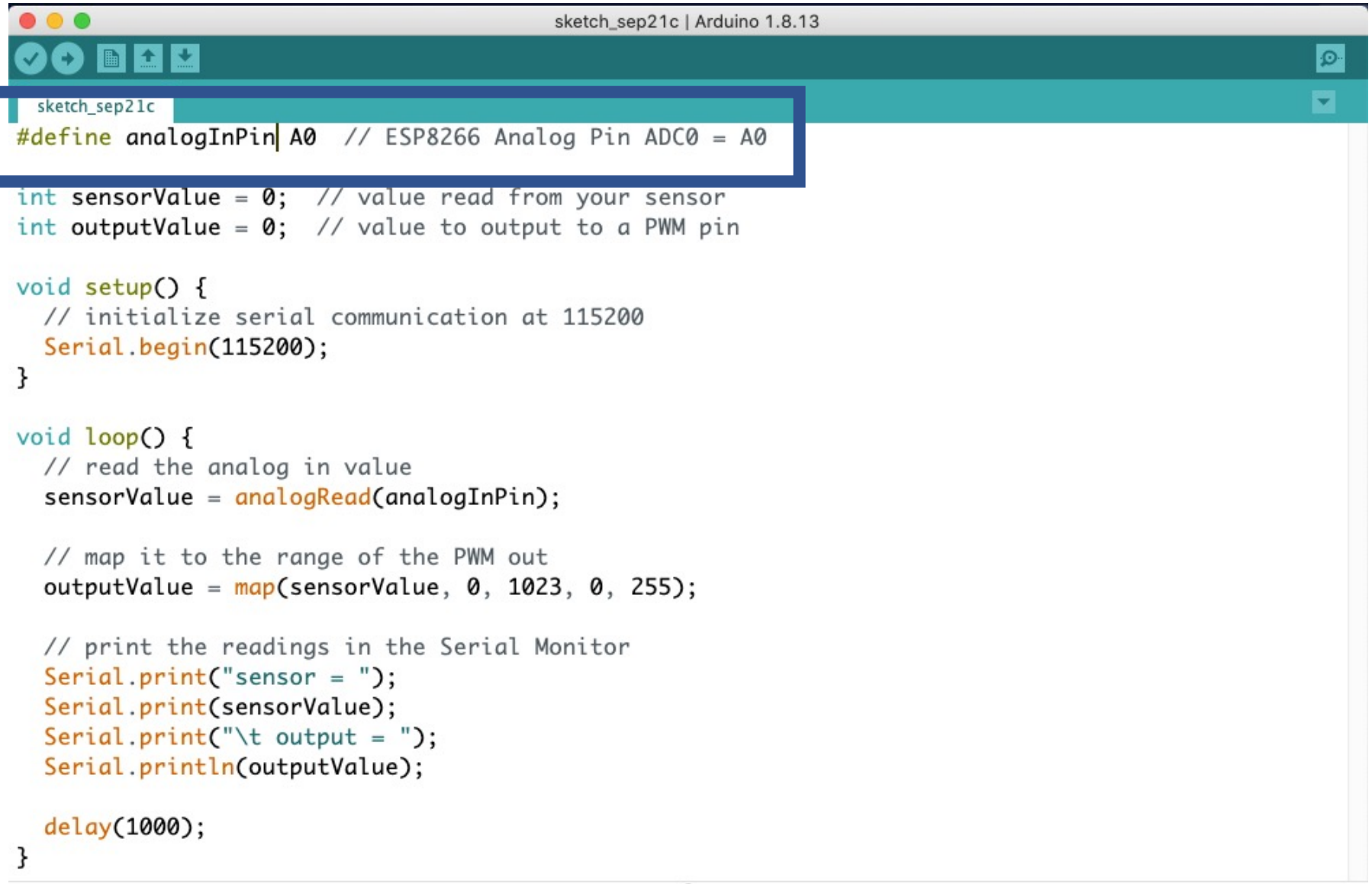


Embedded Programming 1/3

1. Introduction
2. Microcontrollers
3. Basics of Embedded Programming
4. Understand an Arduino Program

Arduino Programs

Arduino IDE – Pin definition

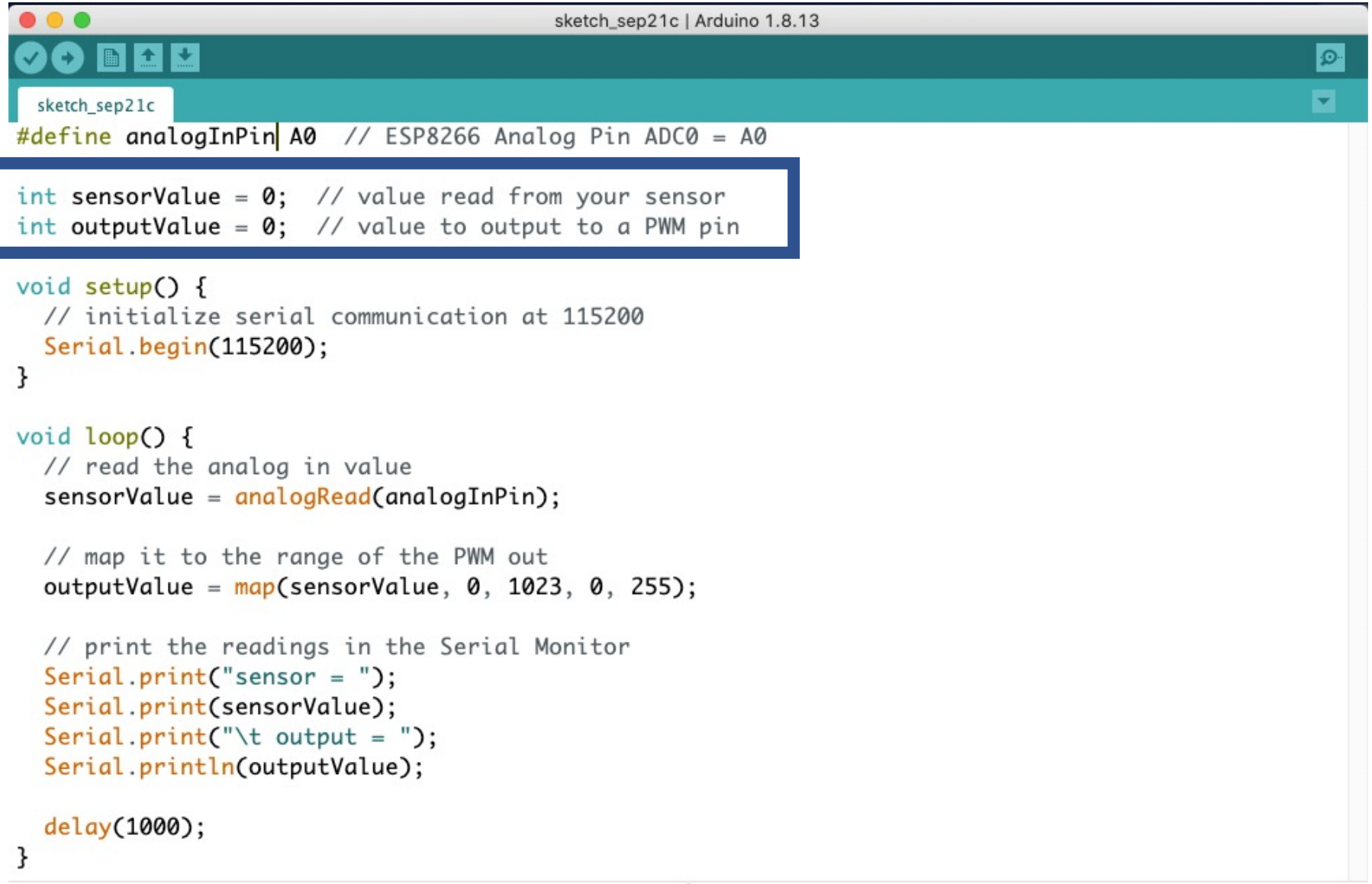


Embedded Programming 1/3

1. Introduction
2. Microcontrollers
3. Basics of Embedded Programming
4. Understand an Arduino Program

Arduino Programs

Arduino IDE – Variables declaration




```
#define analogInPin A0  // ESP8266 Analog Pin ADC0 = A0

int sensorValue = 0;  // value read from your sensor
int outputValue = 0;  // value to output to a PWM pin

void setup() {
  // initialize serial communication at 115200
  Serial.begin(115200);
}

void loop() {
  // read the analog in value
  sensorValue = analogRead(analogInPin);

  // map it to the range of the PWM out
  outputValue = map(sensorValue, 0, 1023, 0, 255);

  // print the readings in the Serial Monitor
  Serial.print("sensor = ");
  Serial.print(sensorValue);
  Serial.print("\t output = ");
  Serial.println(outputValue);

  delay(1000);
}
```

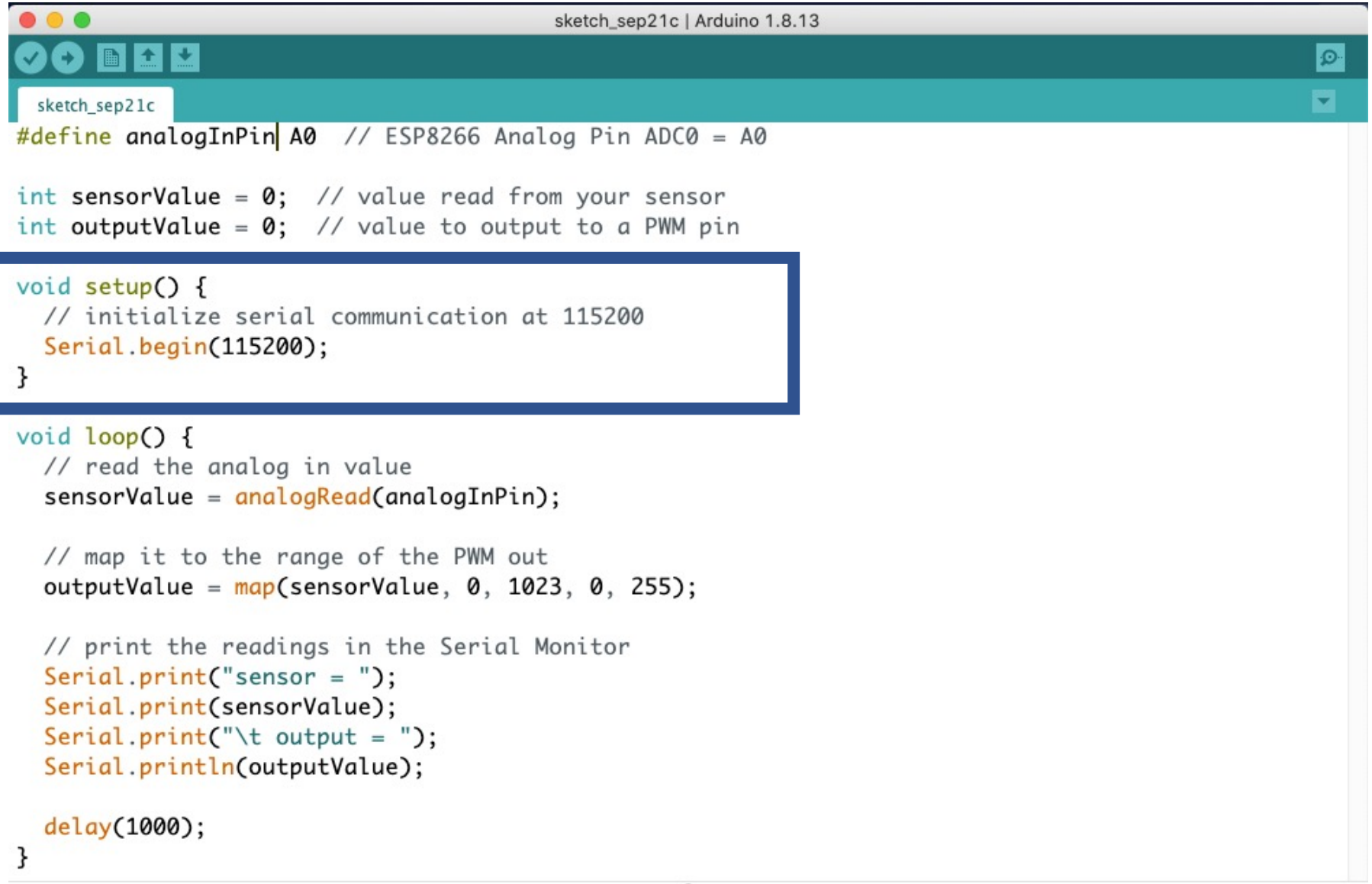
Embedded Programming 1/3

1. Introduction
2. Microcontrollers
3. Basics of Embedded Programming
4. Understand an Arduino Program

@luisomoreau

Arduino Programs

Arduino IDE – Setup function



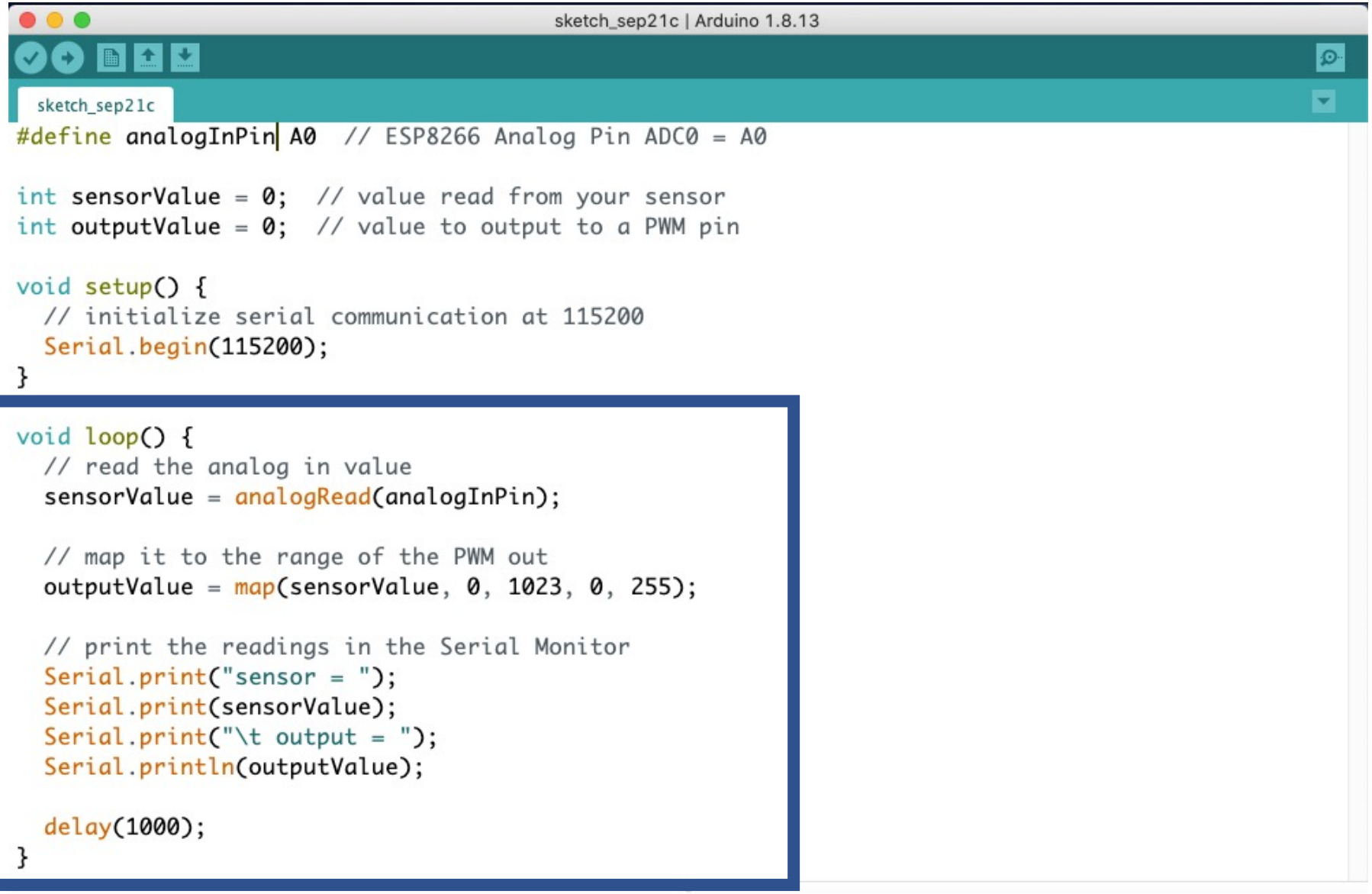
Embedded Programming 1/3

1. Introduction
2. Microcontrollers
3. Basics of Embedded Programming
4. Understand an Arduino Program

@luisomoreau

Arduino Programs

Arduino IDE – Loop function



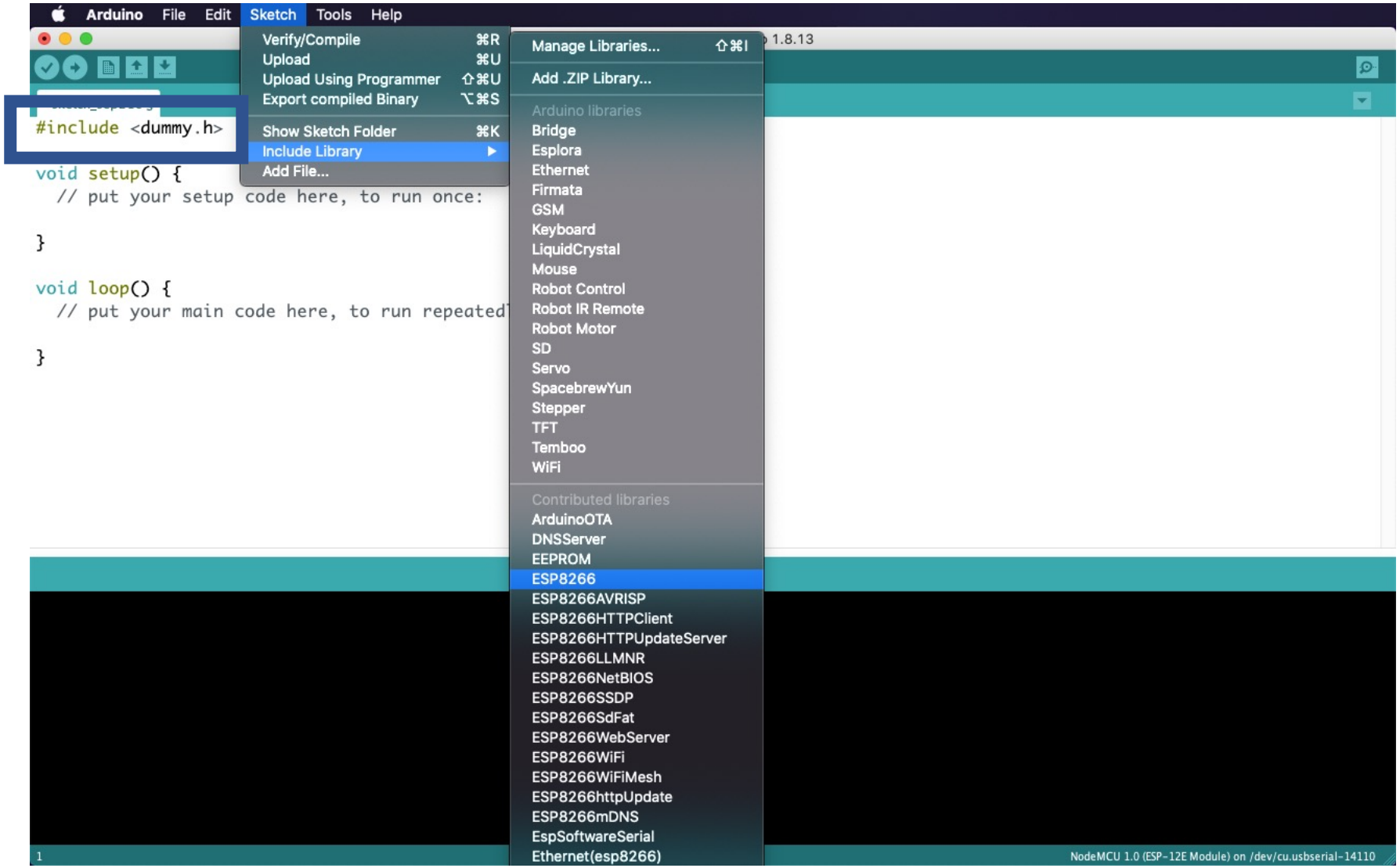
Embedded Programming 1/3

1. Introduction
2. Microcontrollers
3. Basics of Embedded Programming
4. Understand an Arduino Program

@luisomoreau

Arduino Programs

Arduino IDE – Include a library



Arduino Programs

Next steps:

http://github.com/luisomoreau/Embedded-Programming-Lab-1

During the first lab, we will dive deeper into the following topics:

- **General-purpose input/output (GPIO)**
- **Analog to Digital Converter (ADC)**
- **Pulse Width Modulation (PWM)**
- **Interrupts**
- **HTTP Client**
- **HTTP Server**

Thank you!

EDGE IMPULSE

hello@edgeimpulse.com

Edge Impulse
3031 Tisch Way 110 Plaza West
San Jose, CA 95128 USA