

PSI3542 2023

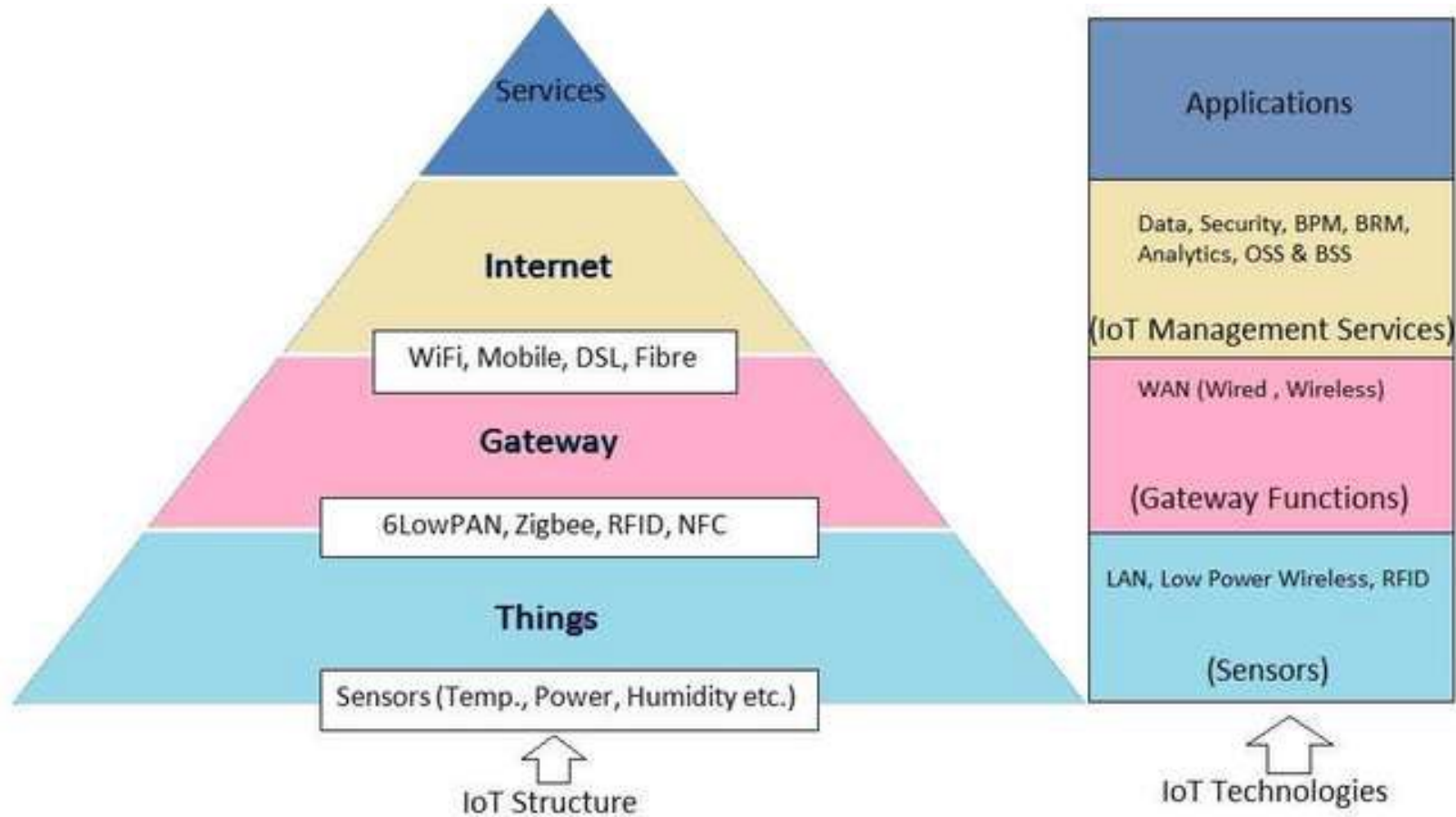
SISTEMAS EMBARCADOS DISTRIBUIDOS

AULA 05 – SW e HW de Dispositivos IoT. Simulador ESP32 WOKI

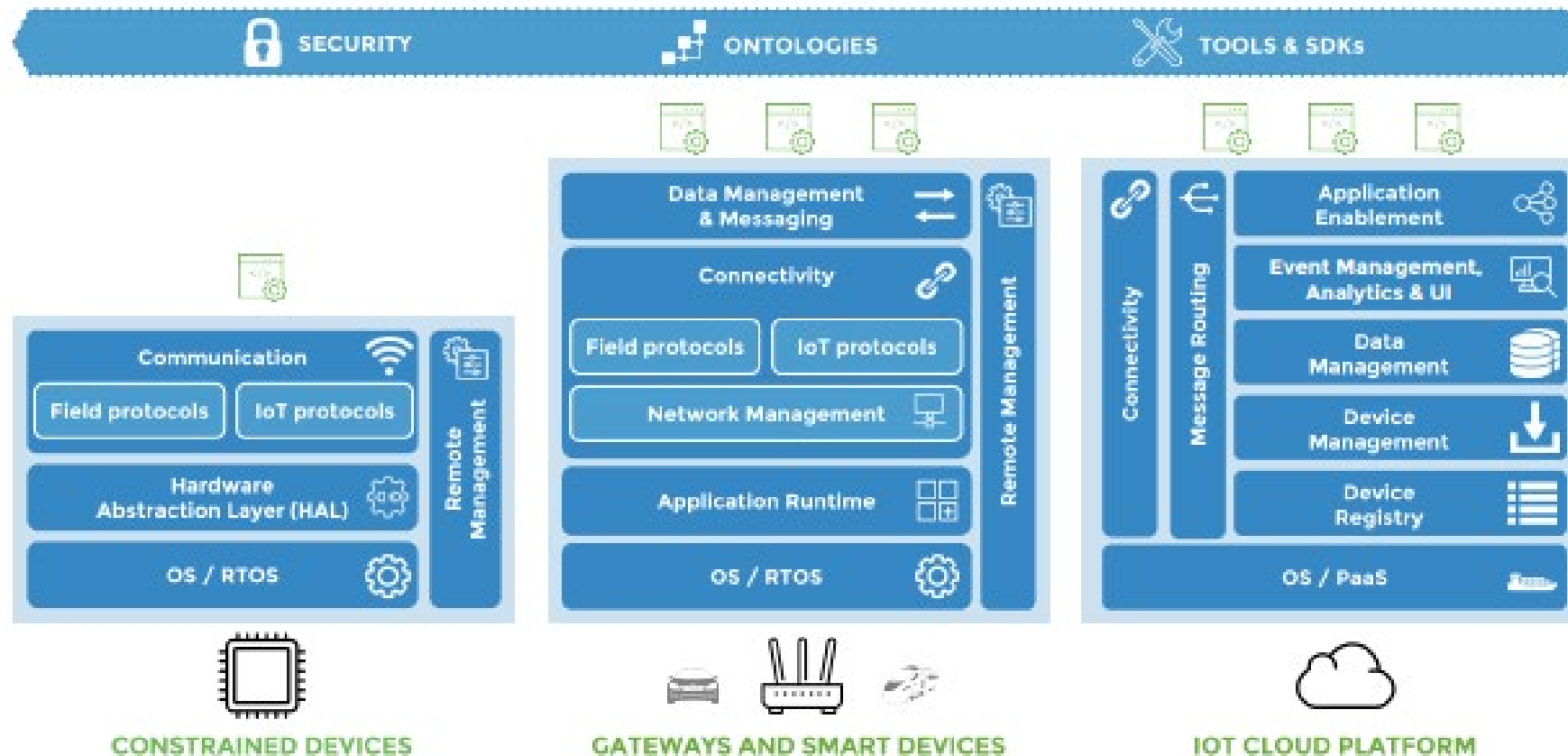
Sergio Takeo Kofuji

kofuji@usp.br

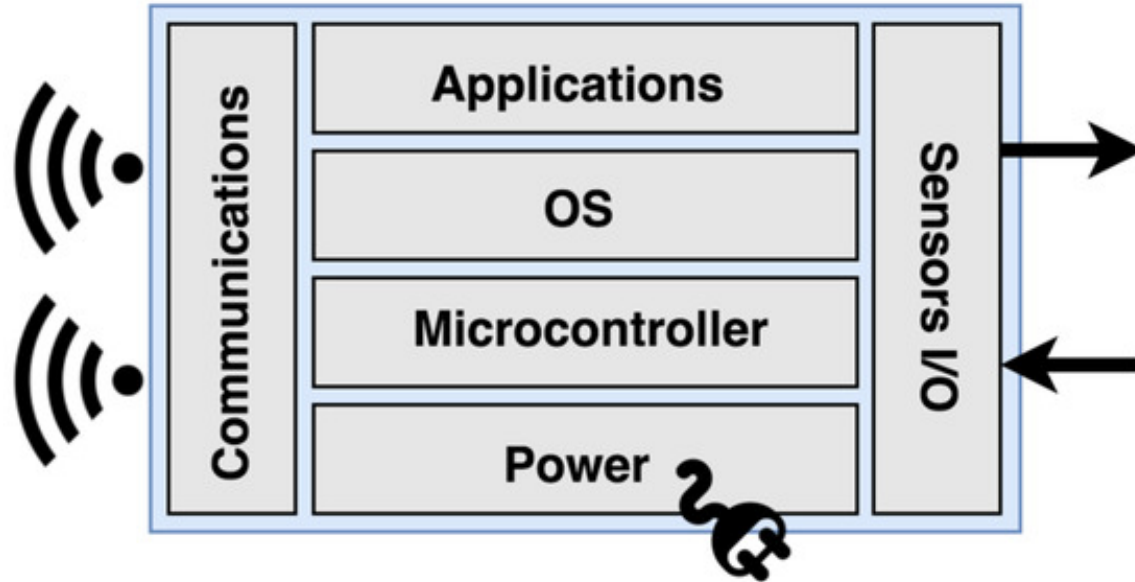
Architettura da IoT



Arquitetura de Software da IoT

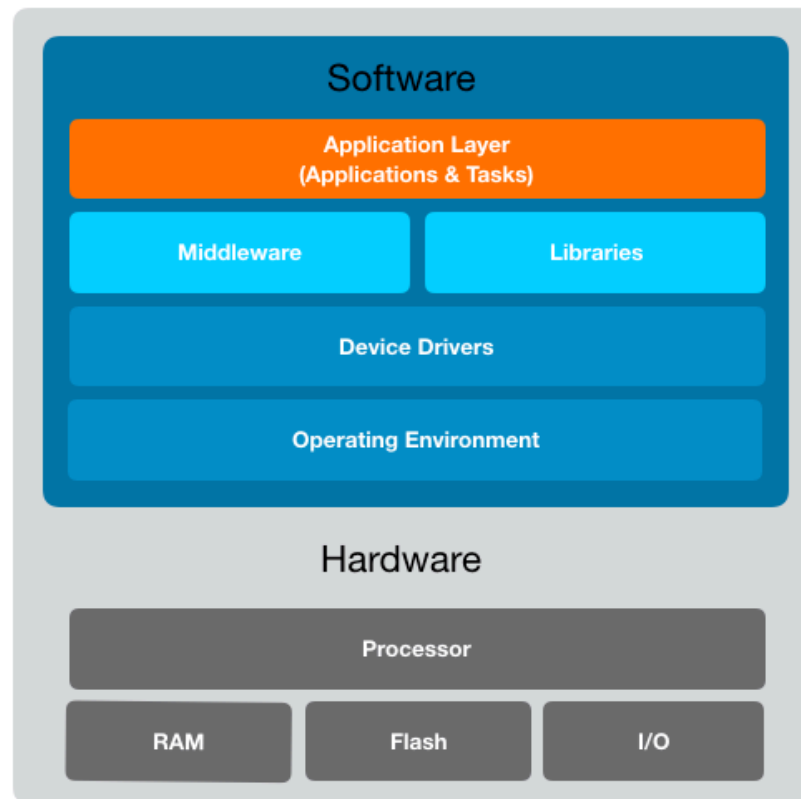


Arquitetura básica de um dispositivo de IoT



Software Embarcado Dispositivo IoT

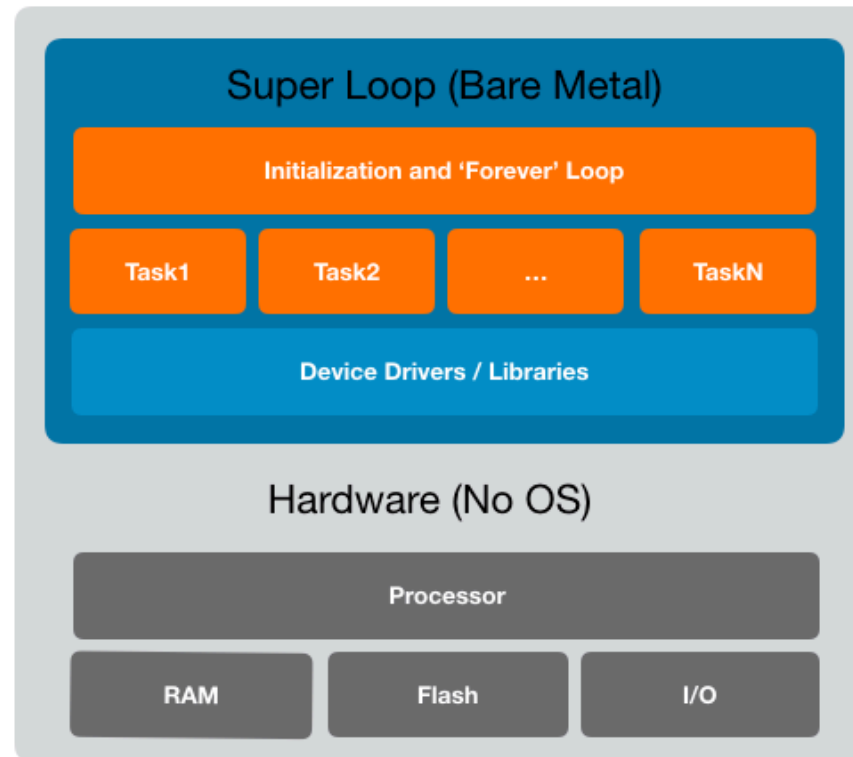
Figure 1 – Generalized Layers of Software



<http://locolabs.com/how-to-choose-the-right-firmware-architecture-for-your-iot-device-2/>

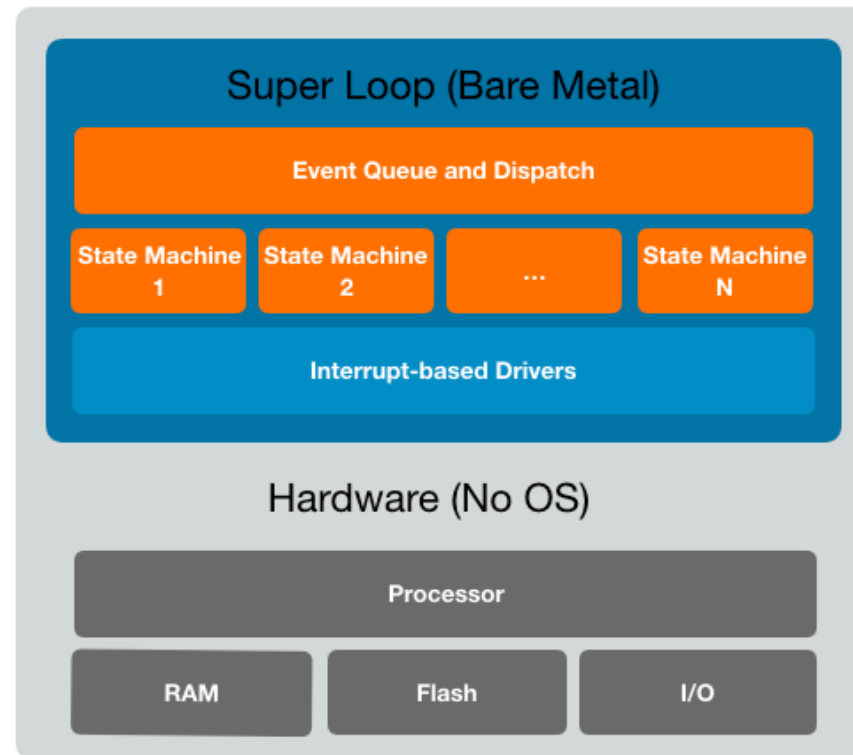
Super loop

Figure 2 – Example of a Super Loop Software Architecture on Bare Metal



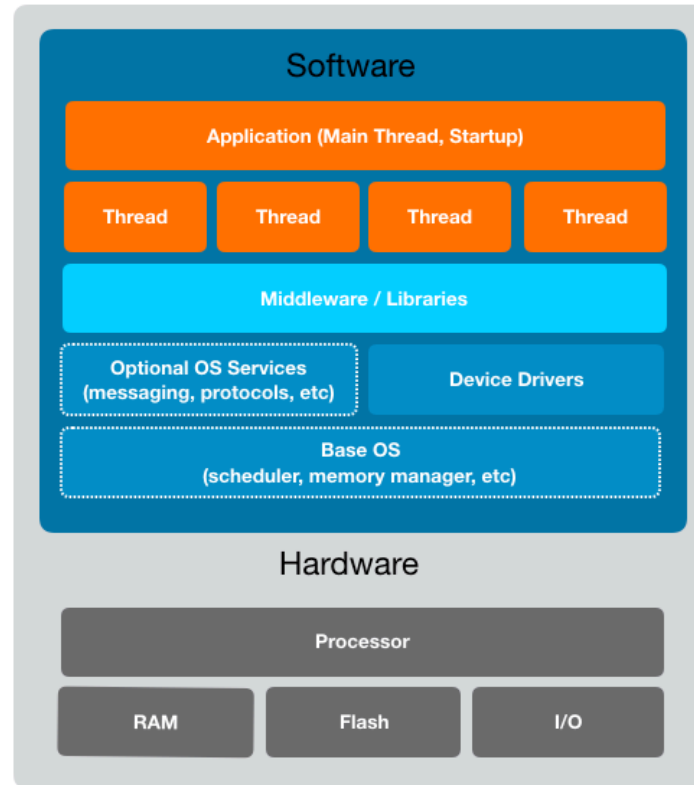
Event-Driven

Figure 3 – Example of an Event-Driven Software Architecture on Bare Metal

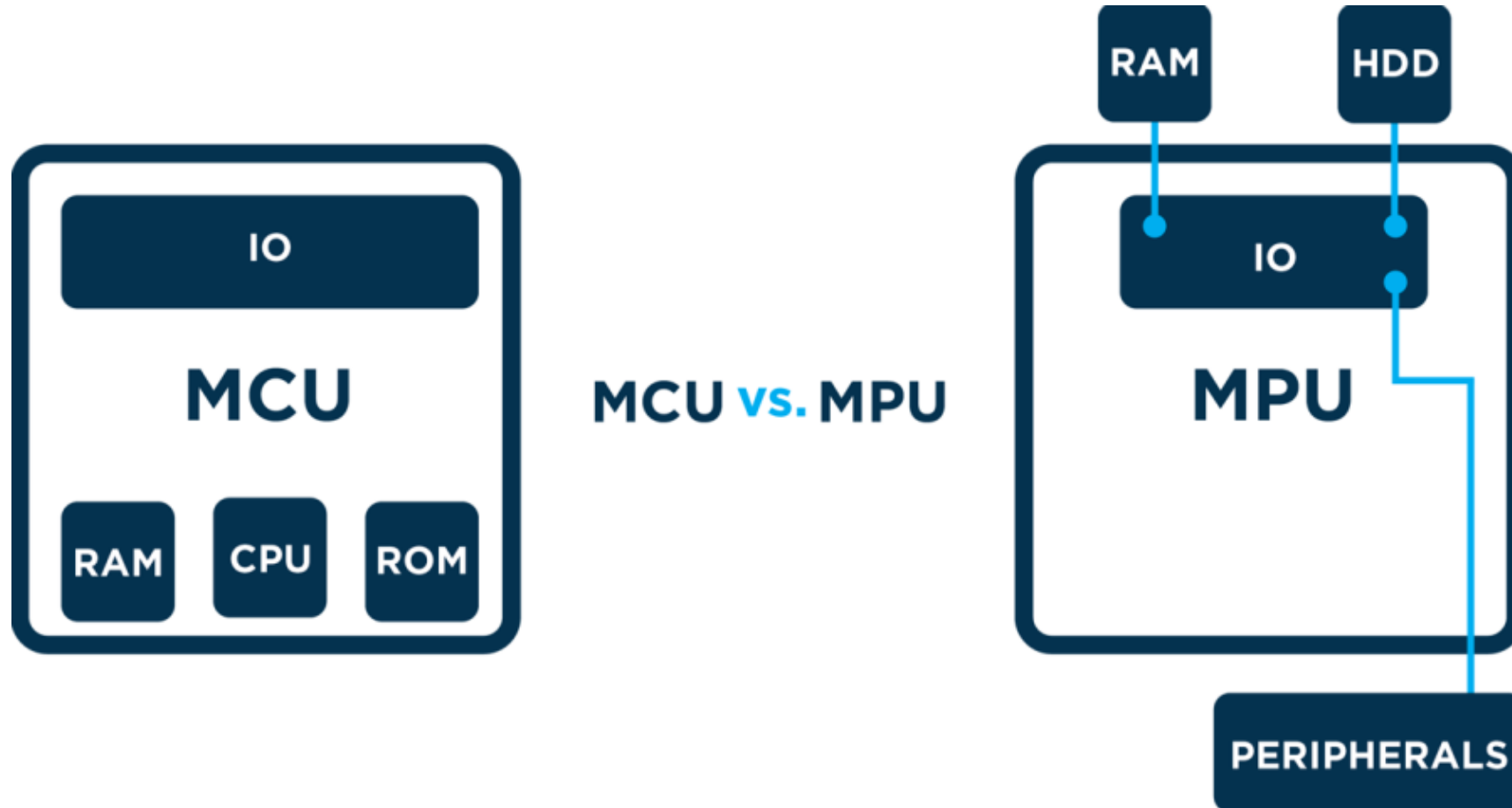


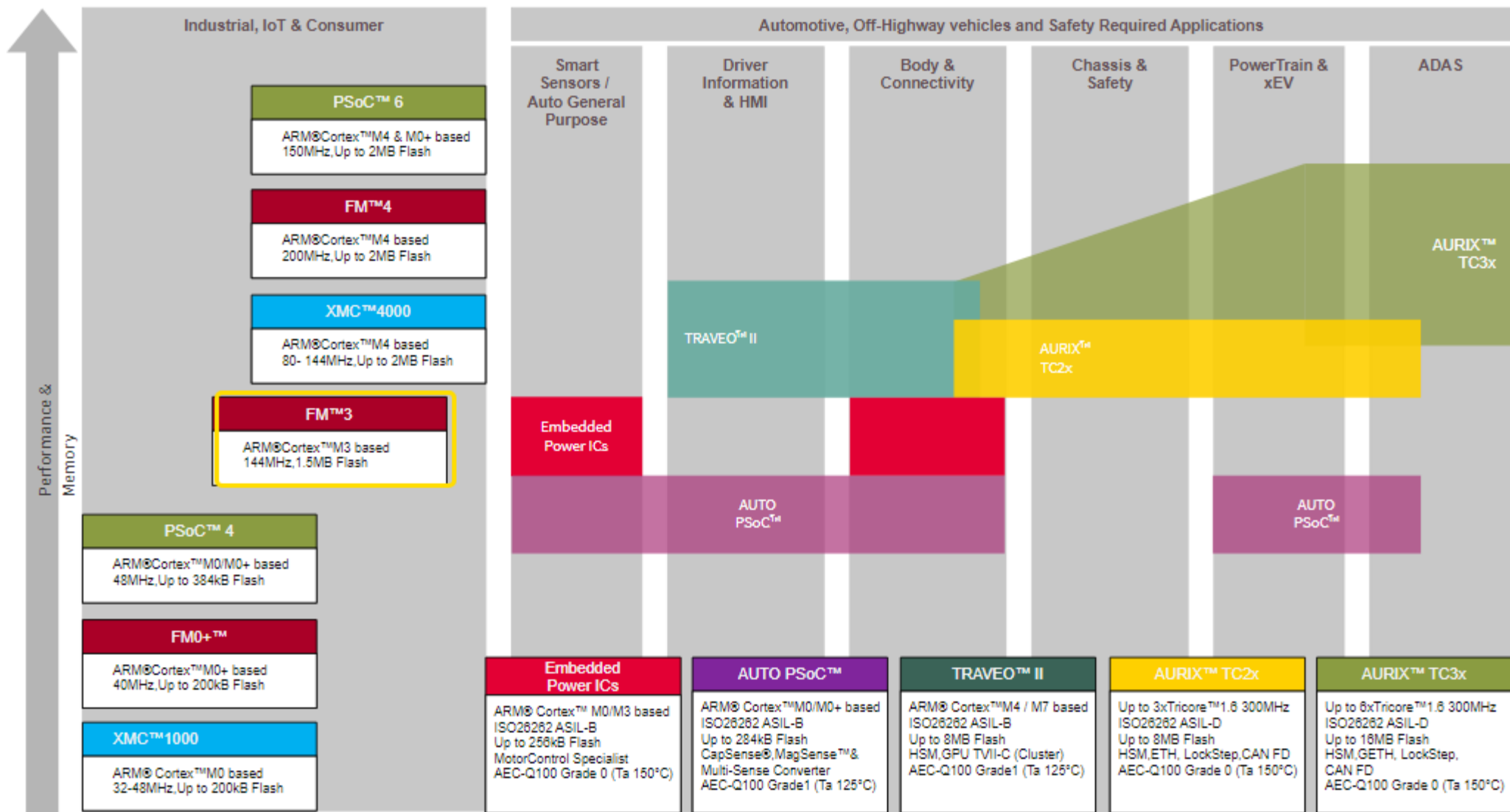
Embedded OS

Figure 4 – Example Software Architecture When Using an Embedded OS or RTOS



Microcontroladores vs Microprocesadores
































Alguns microcontroladores para IoT

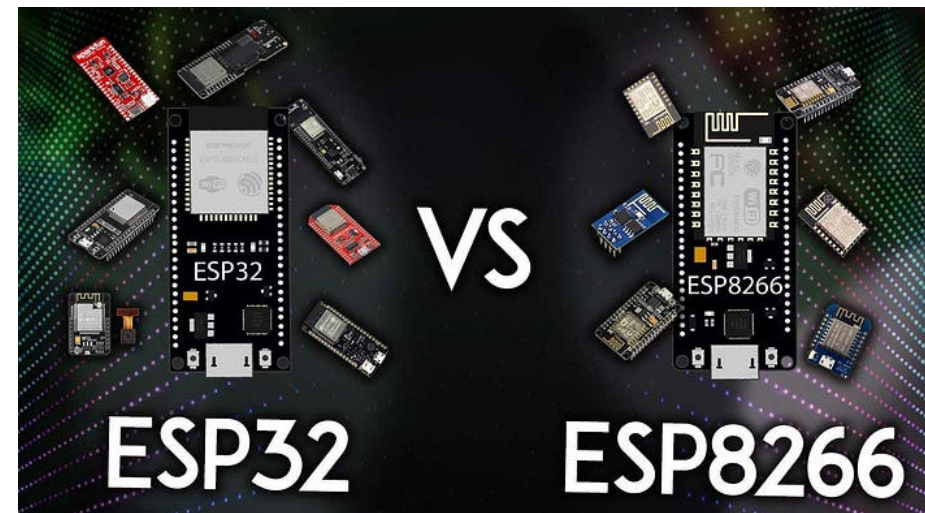
- Atmel AVR: ATmega32, ATmega16, AVR128
- Microchip PIC: PIC18 (incluindo PIC12 e PIC16), PIC32
- Espressif ESP: ESP8266, ESP32
- STMicroelectronics: STM32

Algumas Placas de Desenvolvimento para IoT

Raspberry Pi Boards

 <p>Raspberry Pi 4 Model B Your tiny, dual-display, desktop computer</p> <p>More info ></p>	 <p>Raspberry Pi 3 Model A+ Our third-generation single-board computer, now in the A+ format</p> <p>More info ></p>	 <p>Raspberry Pi 3 Model B+ The final revision of our third-generation single-board computer</p> <p>More info ></p>	 <p>Raspberry Pi 3 Model B Our third-generation single-board computer</p> <p>More info ></p>
 <p>Raspberry Pi 2 Model B The Raspberry Pi 2 Model B is the second-generation Raspberry Pi</p> <p>More info ></p>	 <p>Raspberry Pi 1 Model B+ The Model B+ is the final revision of the original Raspberry Pi</p> <p>More info ></p>	 <p>Raspberry Pi 1 Model A+ The Model A+ is the low-cost variant of the Raspberry Pi</p> <p>More info ></p>	 <p>Raspberry Pi Zero W Single-board computer with wireless and Bluetooth connectivity</p> <p>More info ></p>
 <p>Raspberry Pi Zero Our lowest-cost single-board computer</p> <p>More info ></p>			

 <p>Arduino Uno</p>	 <p>Arduino Leonardo</p>	 <p>Arduino Due</p>	 <p>Arduino Yún</p>	 <p>Arduino Tre</p>	 <p>Arduino Micro</p>
 <p>Arduino Robot</p>	 <p>Arduino Esplora</p>	 <p>Arduino Mega ADK</p>	 <p>Arduino Ethernet</p>	 <p>Arduino Mega 2560</p>	 <p>Arduino Mini</p>
 <p>LilyPad Arduino USB</p>	 <p>LilyPad Arduino Simple</p>	 <p>LilyPad Arduino SimpleSnap</p>	 <p>LilyPad Arduino</p>	 <p>Arduino Nano</p>	 <p>Arduino Pro Mini</p>



VS

ESP32

ESP8266

The image shows two development boards, ESP32 and ESP8266, surrounded by various components like resistors, capacitors, and sensors. The ESP32 board is on the left and the ESP8266 board is on the right. The background is a dark, grid-like pattern with glowing dots.

ESP32, ESP8266 e Arduino R3

	ESP32	ESP8266	ARDUINO UNO R3
Cores	2	1	1
Arquitetura	32 bits	32 bits	8 bits
Clock	160MHz	80MHz	16MHz
WiFi	Sim	Sim	Não
Bluetooth	Sim	Não	Não
RAM	512KB	160KB	2KB
FLASH	16Mb	16Mb	32KB
GPIO	36	17	14
Interfaces	SPI / I2C / UART / I2S / CAN	SPI / I2C / UART / I2S	SPI / I2C / UART
ADC	18	1	6
DAC	2	0	0



Arduino RS232^[38]
(male pins)



Arduino Diecimila^[39]



Arduino Duemilanove^[40]
(rev 2009b)



Arduino Uno R2^{[41][42]}



Arduino Uno SMD
R3^[43]



Arduino Leonardo^[44]



Arduino micro [Archived](#)
2020-10-29 at the [Wayback Machine](#)(AtMega 32U4)



Arduino pro micro
(AtMega32U4)



Arduino Pro^[45]
(No USB)



Arduino Mega^[46]



Arduino Nano^[47]
(DIP-30 footprint)

ARDUINO



Arduino LilyPad 00^[48]
(rev 2007) (No USB)



Arduino Robot^[49]



Arduino Esplora^[50]



Arduino Ethernet^[51]
(AVR + W5100)

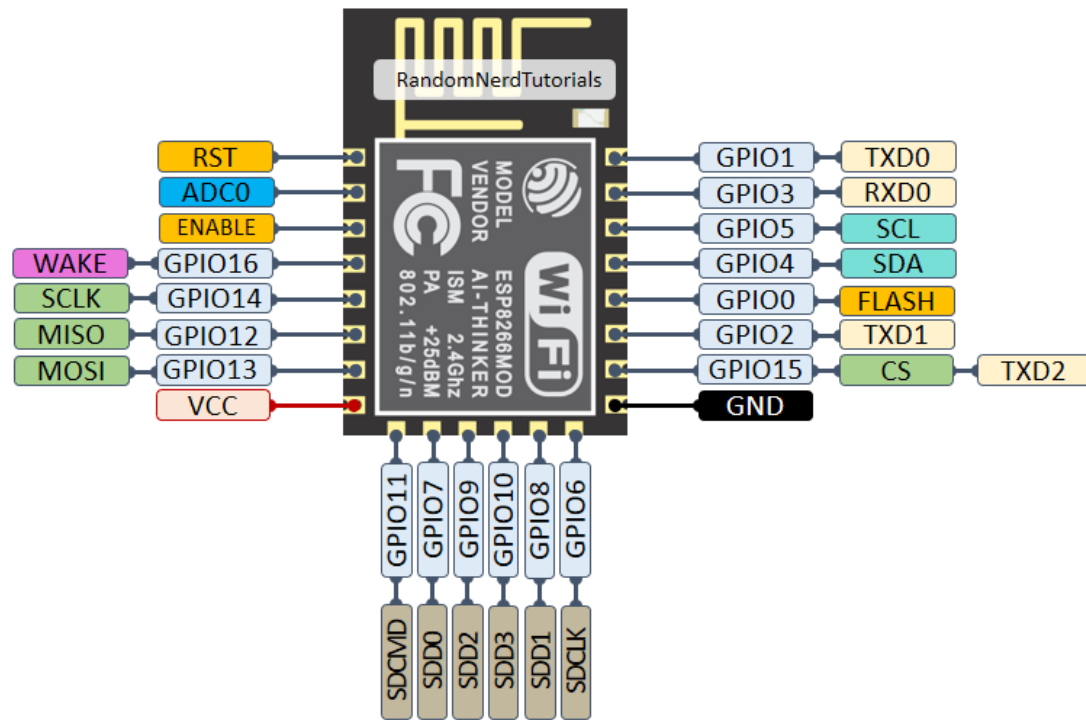


Arduino Yún^[52]
(AVR + AR9331)



Arduino Due^[53]
(ARM Cortex-M3 core)

SOC 8266 E MÓDULOS ESP8266



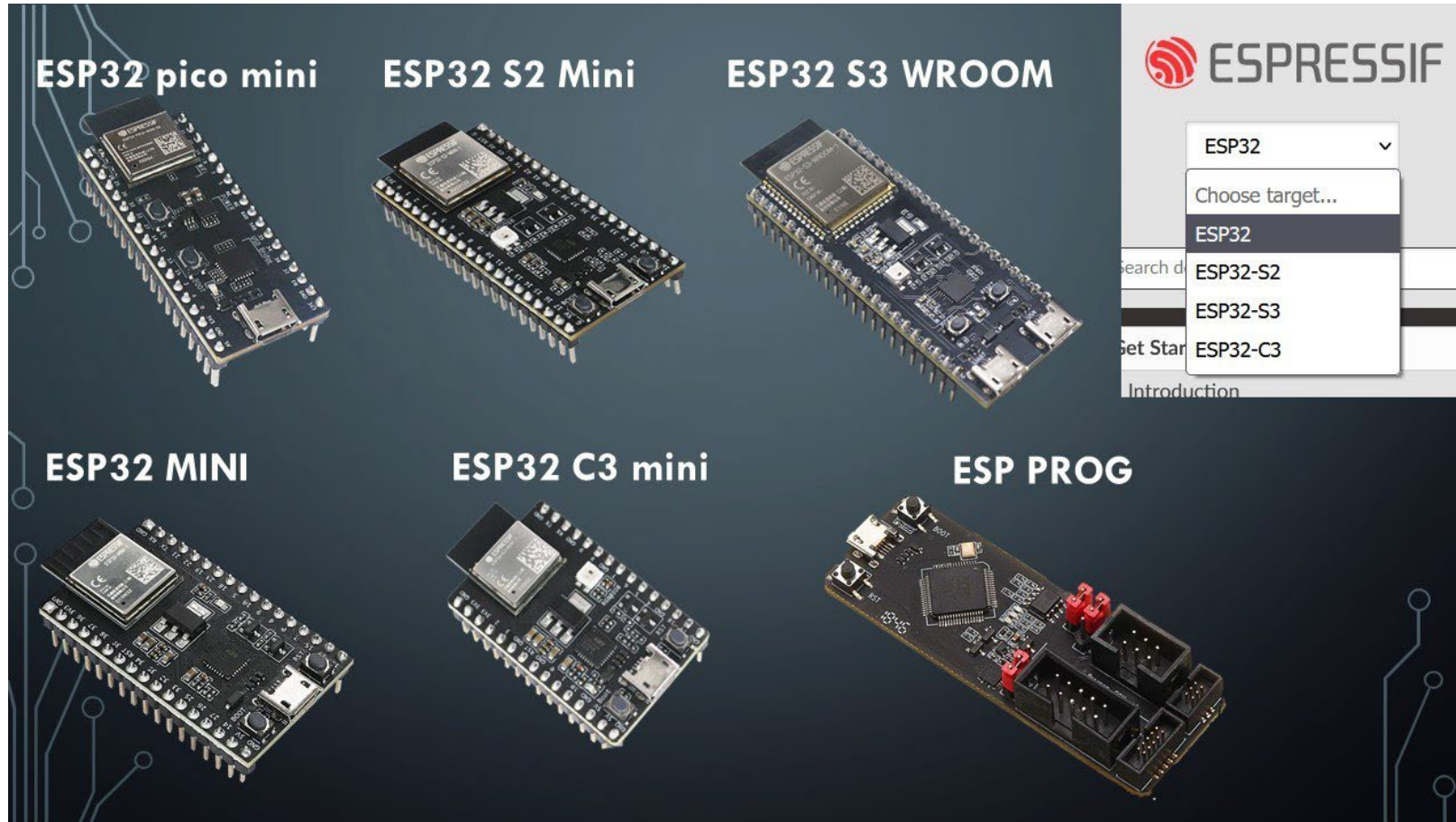
PLACAS DE DESENV. COM ESP8266

	Huzzah ESP8266	NodeMCU v2	Generic ESP	imp001	Edison	Core	Photon	WiPy	Oak
Company	Adafruit	NodeMCU	Espressif	Electric Imp	Intel	Particle (Spark)	Particle	WiPy	Digistump
Microcontroller	Xtensa lx106 micro 32-bit	Xtensa lx106 micro 32-bit	Xtensa lx106 micro 32-bit	Cortex M3	Atom Z3400 + Quark	STM32F103 Cortex M3	STM32F205 Cortex M3	CC3200 onboard Cortex M4	Xtensa lx106 micro 32-bit
Wifi Module	ESP-12	ESP-12E	ESP-12	Unknown	BCM43340	TI CC3000	BCM43362	TI CC3200	ESP8266EX
Wifi Standards	802.11 b/g/n	802.11 b/g/n	802.11 b/g/n	802.11 b/g/n	802.11 a/b/g/n	802.11 b/g	802.11 b/g/n	802.11 b/g/n	802.11 b/g/n
Concurrent Sockets	5	5	5	1 to cloud	4	4	5	8	5
Access Point Mode	WiFi Direct, Soft-AP	WiFi Direct, Soft-AP	WiFi Direct, Soft-AP	None	WiFi Direct	No	Soft-AP	AP, Station, WiFi Direct	WiFi Direct, Soft-AP
True SSL (HTTPS)	No	No	No	Yes	Yes with wolfSSL	No	No	Yes (2 concurrent)	No
Arduino IDE Programmable	Yes	Yes	Yes	No	Yes	No	No	No	No
Manufacturer Cloud	No	No	No	Yes	No	Yes	Yes	No	Yes
Size	25 x 38 mm	49 x 24.5 mm	24 x 16 mm	24 x 32 mm	35.5 x 25.0 mm	36.58 x 20.3 mm	36.58 x 20.32 mm	25 x 45 mm	23.4 x 30 mm
Transmit Current	~250 mA	~250 mA	215 mA (typ)	400 ma (max)	120 mA unofficial	300 mA (max)	430 mA (peak)	229 mA	220 mA (max)
Digital Pins	9	13	9	6 shared	20	10	18	25	11
Analog Pins	1	1	1	6 in & out, shared	6	8	10	3	1
Antenna Type	PCB	PCB	PCB	PCB	Chip or uFL	Chip or uFL	Chip and uFL	Chip and uFL	Chip
Flash	4 MB	4 MB	Can Vary	not stated	4 GB	2MB + 128 KB	1 MB	2 MB	1 MB
RAM	128 KB	128 KB	128 KB	128 KB	1 GB	20 KB	128 KB	256 KB	64 KB
Supply Voltage	3 to 12 volts	5 volts	3.3 volts	2.7 to 3.3 volts	7 to 15 volts	3.6 to 6 volts	3.6 to 5.5 volts	3.6 to 5.5 volts	3.3 to 4.5 volts
Signal Voltage	3.3 volts	3.3 volts	3.3 volts	3.3 volts	1.8 volts	3.3 volts	3.3 volts	3.3 volts	3.3 volts
Cost (US dollars)	\$9.95	\$12.95	\$3.37 to \$6.95 +	29.95 +	49.95 +	\$39.00	\$19.00	29 Euro (About \$33)	\$10.00
Available	Now	Now	Now	Now	Now	Now	July 2015	August 2015	September 2015

SOC ESP32 E MÓDULOS ESP32

Feature	ESP32 Series	ESP32-S2 Series	ESP32-C3 Series
Launch year	2016	2020	2020
Variants	See ESP32 Datasheet (PDF)	See ESP32-S2 Datasheet (PDF)	See ESP32-C3 Datasheet (PDF)
Core	Xtensa® dual-core 32-bit LX6 with 600 MIPS (in total); 200 MIPS for ESP32-U4WDH/ESP32-S0WD (single-core variants); 400 MIPS for ESP32-D2WD	Xtensa® single-core 32-bit LX7 with 300 MIPS	32-bit single-core RISC-V
Wi-Fi protocols	802.11 b/g/n, 2.4 GHz	802.11 b/g/n, 2.4 GHz	802.11 b/g/n, 2.4 GHz
Bluetooth®	Bluetooth v4.2 BR/EDR and Bluetooth Low Energy	✘	Bluetooth 5.0
Typical frequency	240 MHz (160 MHz for ESP32-S0WD, ESP32-D2WD, and ESP32-U4WDH)	240 MHz	160 MHz

PLACAS DE DESENV. COM ESP32

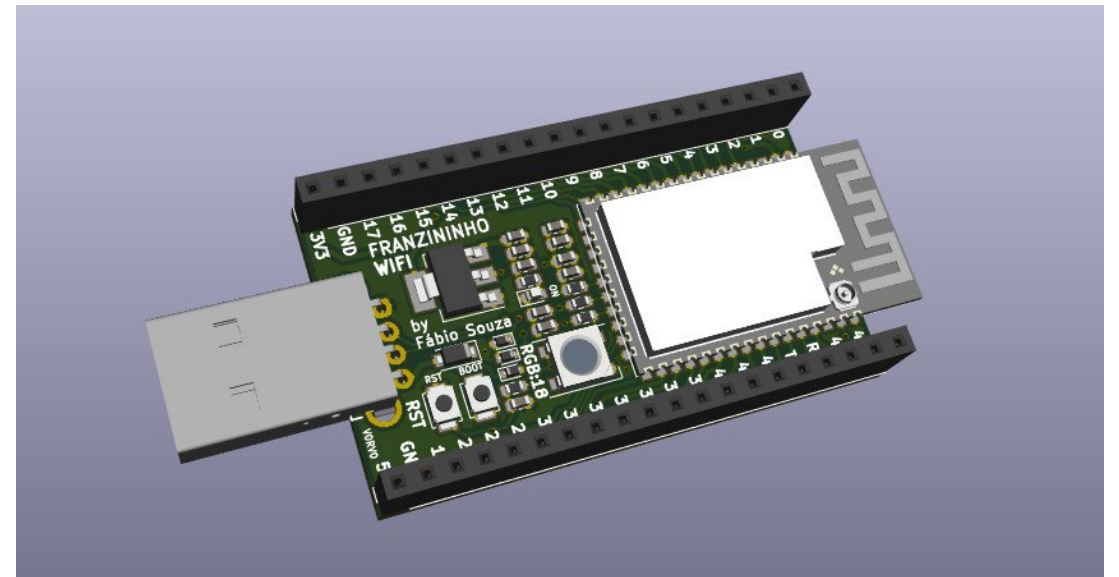
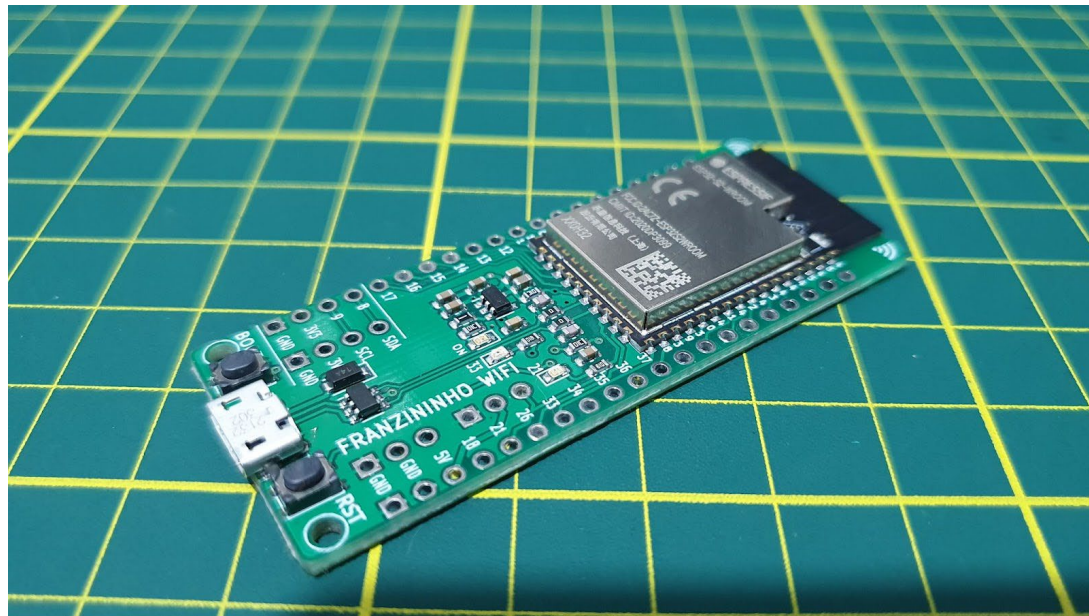


PLACAS FRANZININHO

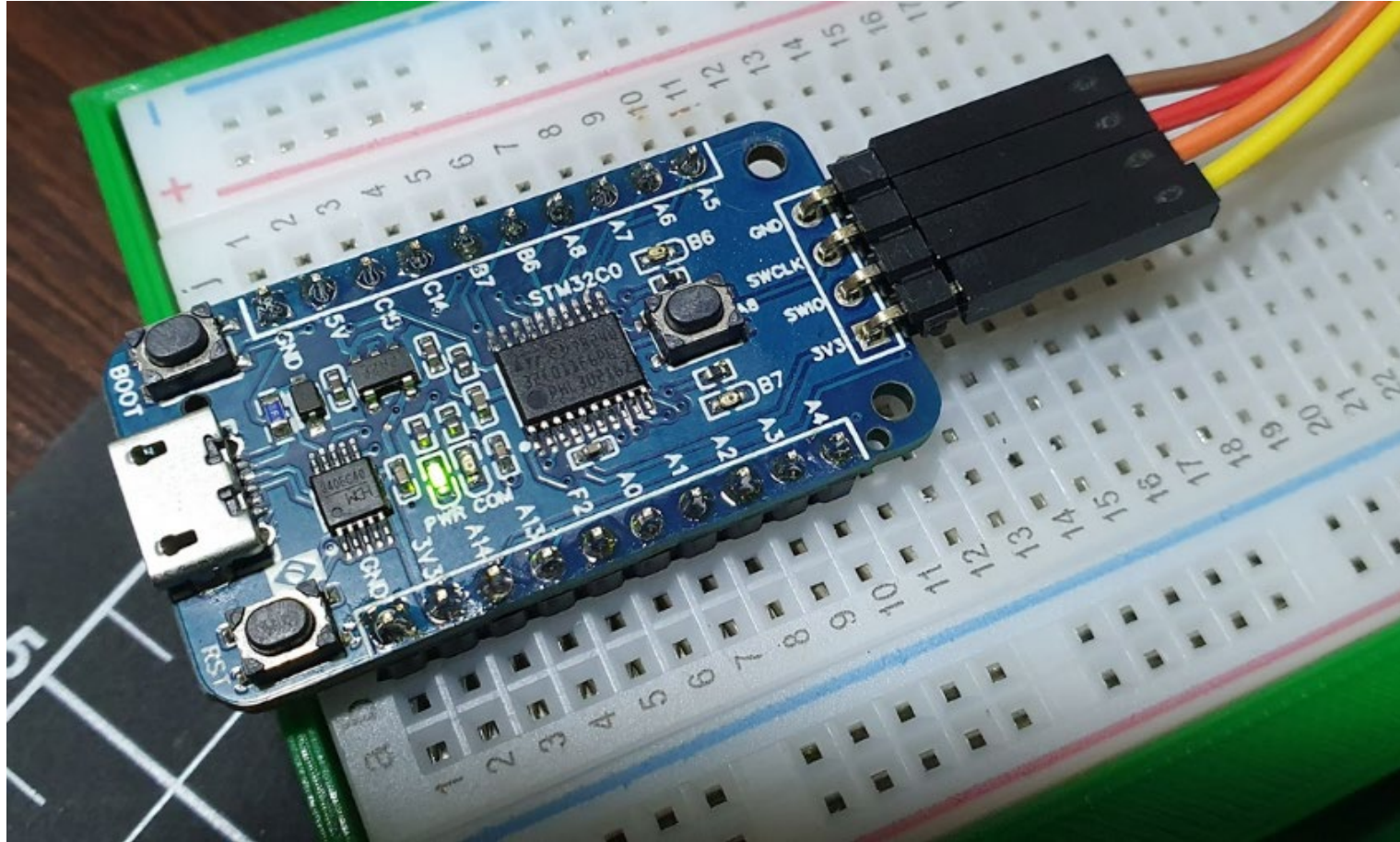
- OPEN HARDWARE
- <https://docs.franzininho.com.br/docs/>



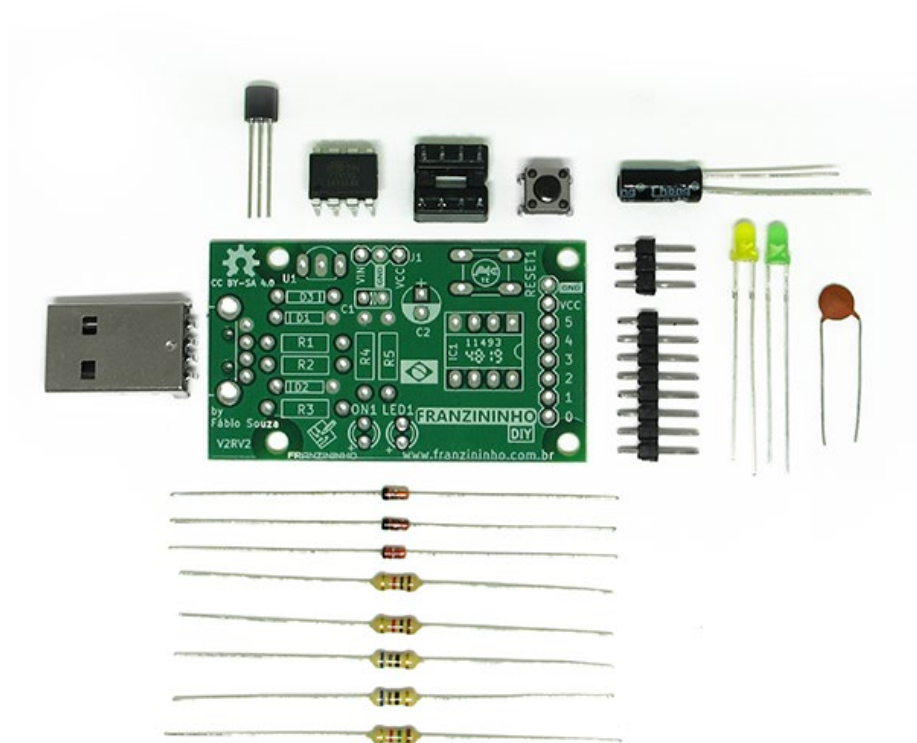
PLACA FRANZININHO WIFI E WIFI DEV



PLACA FRANZININHO C0 (PROCESSADOR STM32C0)



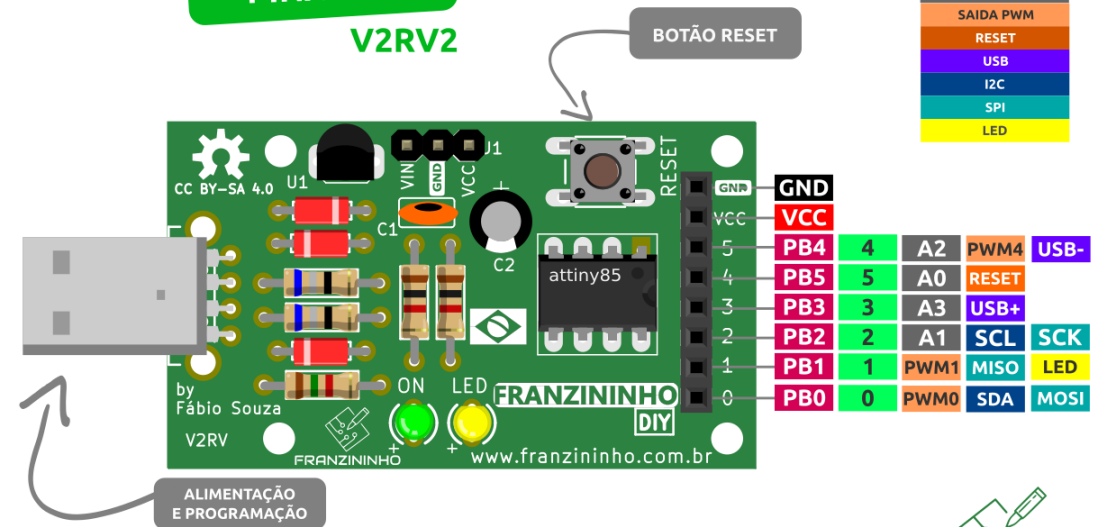
PLACA FRANZININHO DIY



FRANZININHO DIY

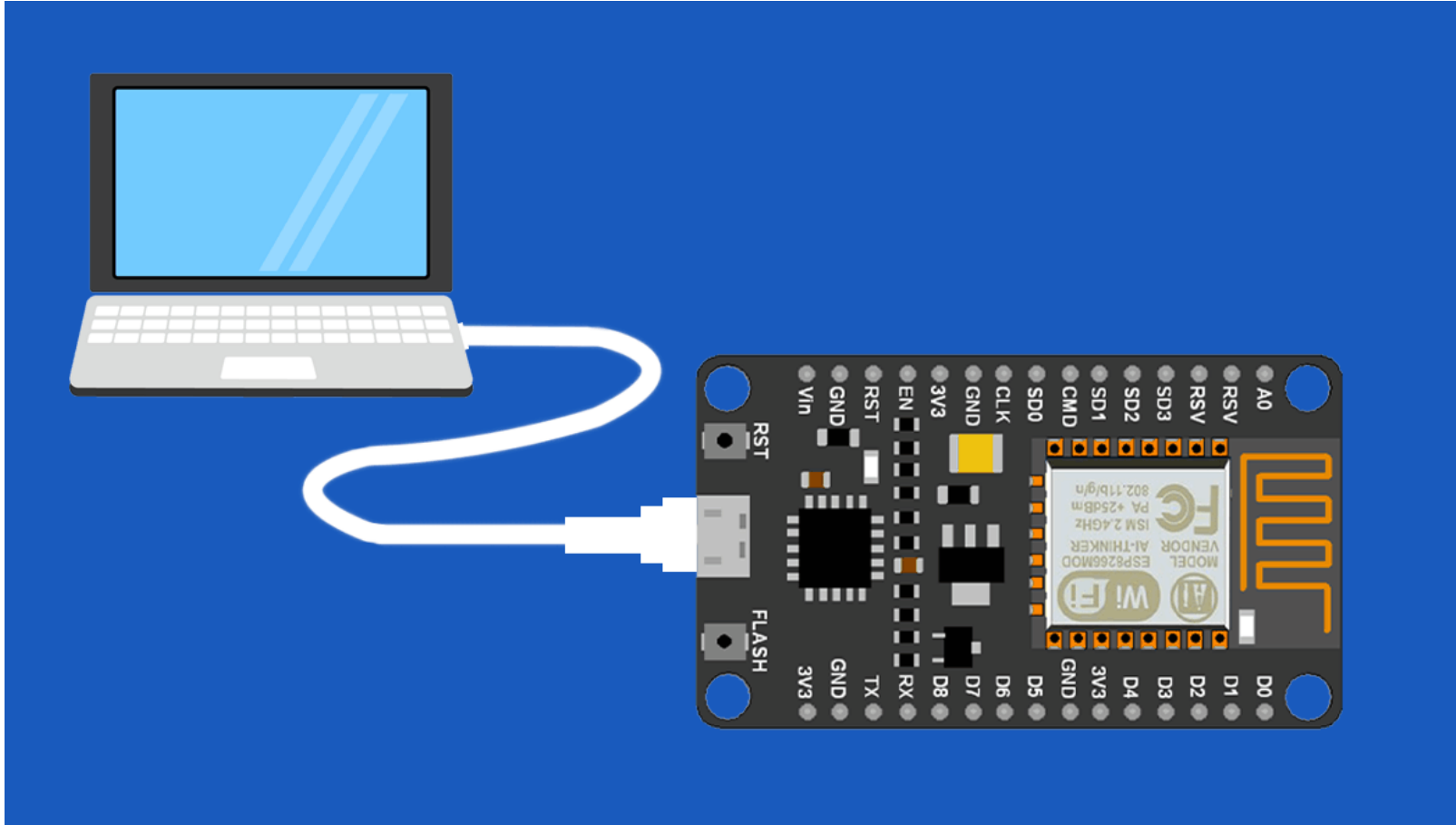
PINAGEM

V2RV2



DESENVOLVIMENTO DE SOFTWARE

DESENVOLVIMENTO

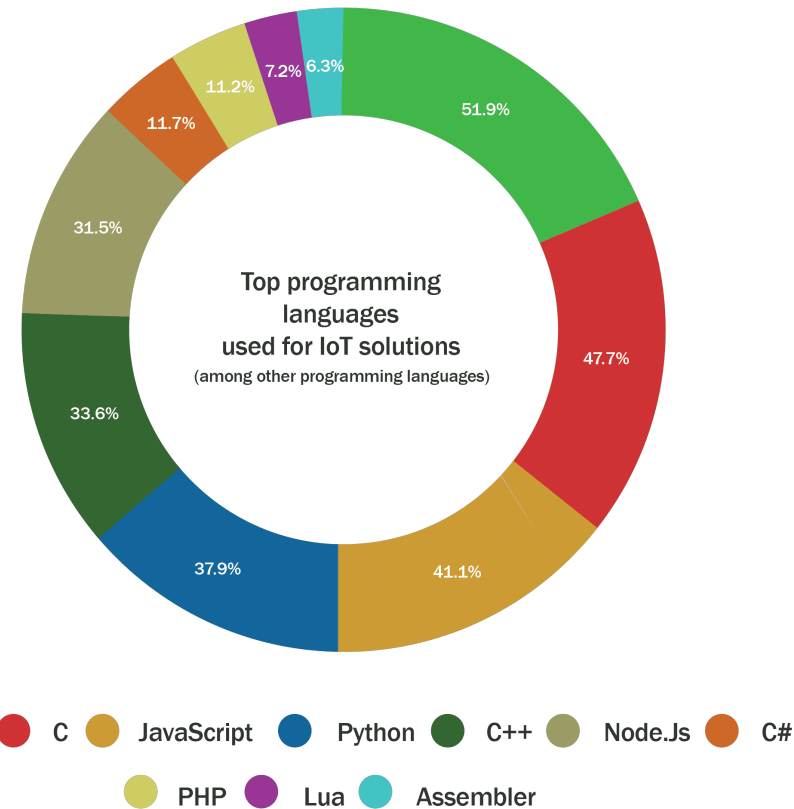


PLATAFORMAS DE DESENVOLVIMENTO

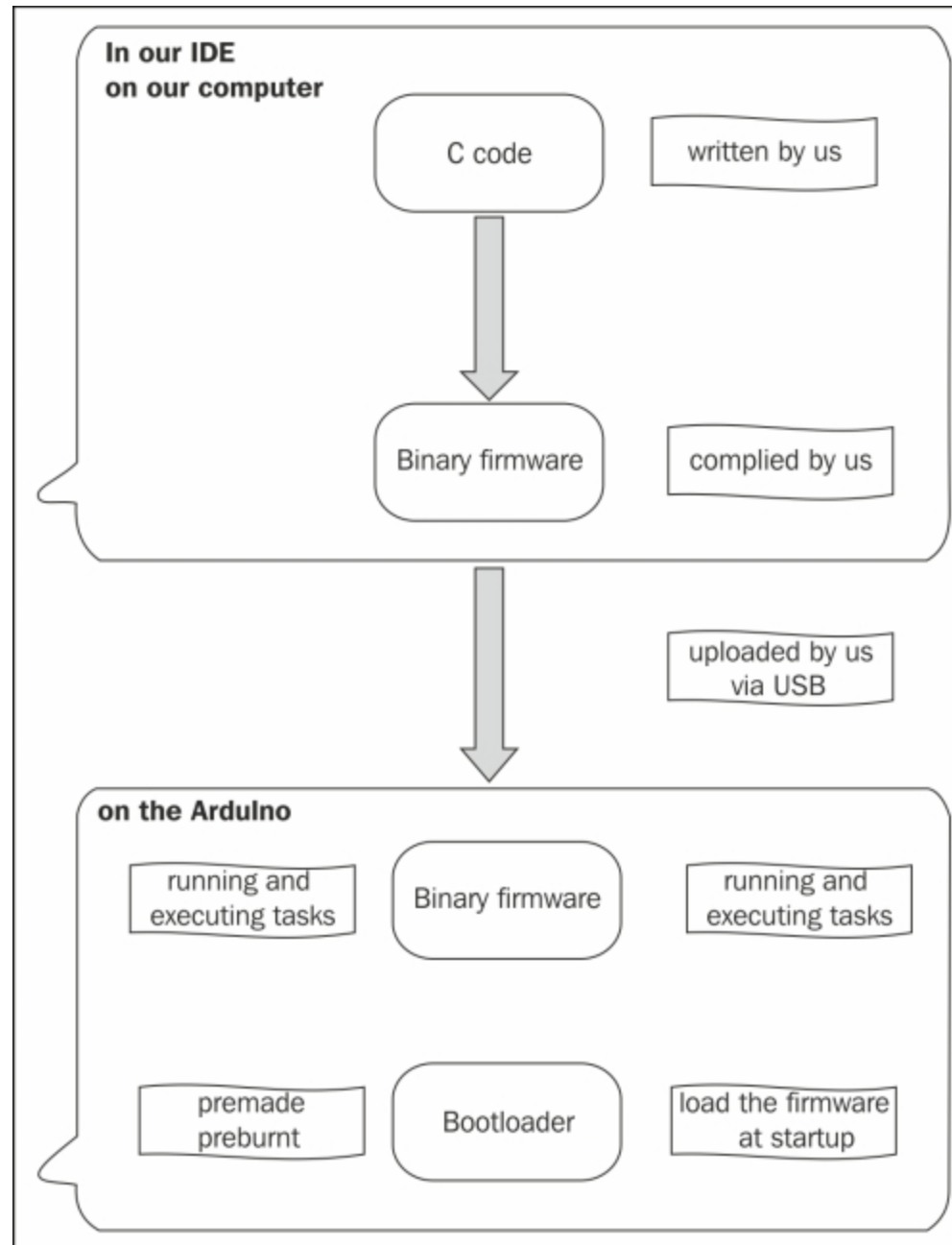
- PLATAFORMA PARA DESENVOLVIMENTO:
 - ARDUINO IDE
 - VISUAL STUDIO CODE
 - PLATFORM COMPUTING
 - ESPRESSIF IDE

LINGUAGENS DE PROGRAMAÇÃO

- ASSEMBLY
- C/C++
- PYTHON
- MICROPYTHON
- CIRCUITPYTHON
- JAVA
- JAVASCRIPT
- GO
- LUA
- RUST
- ETC.

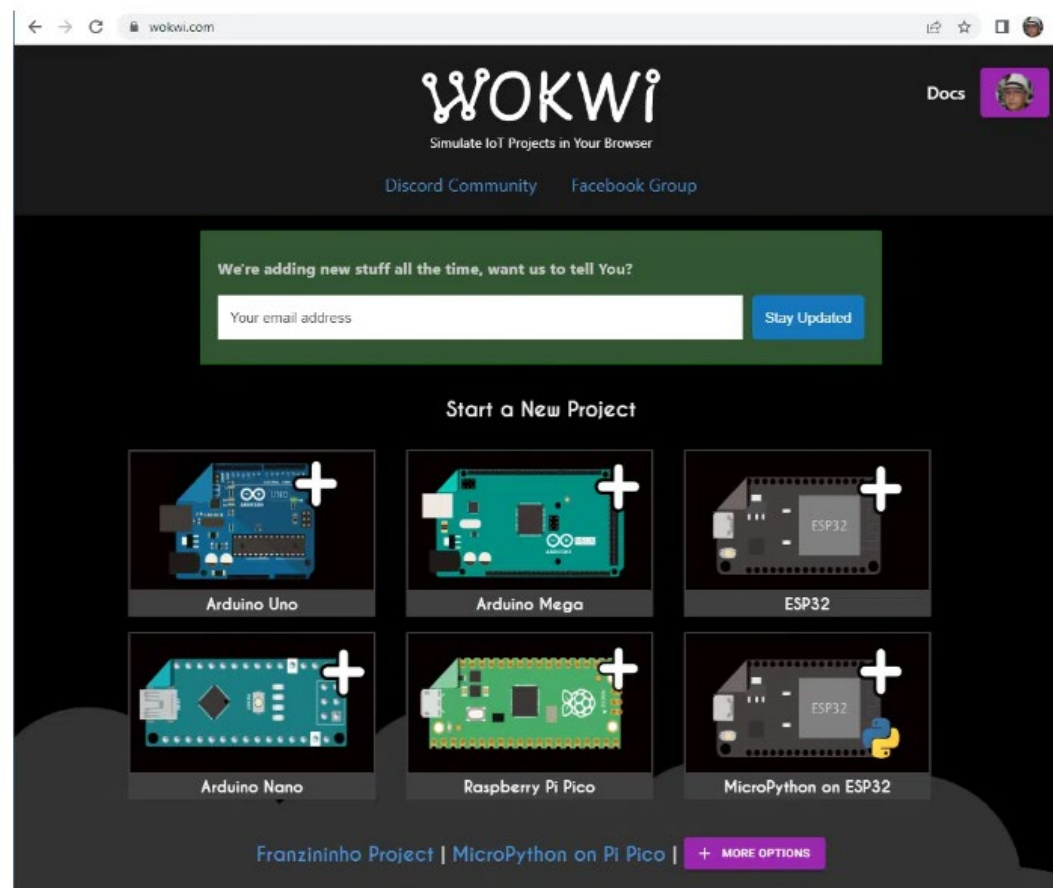


(*Statistics - Eclipse IoT Working Group. IEEE IoT & AGILE IoT)

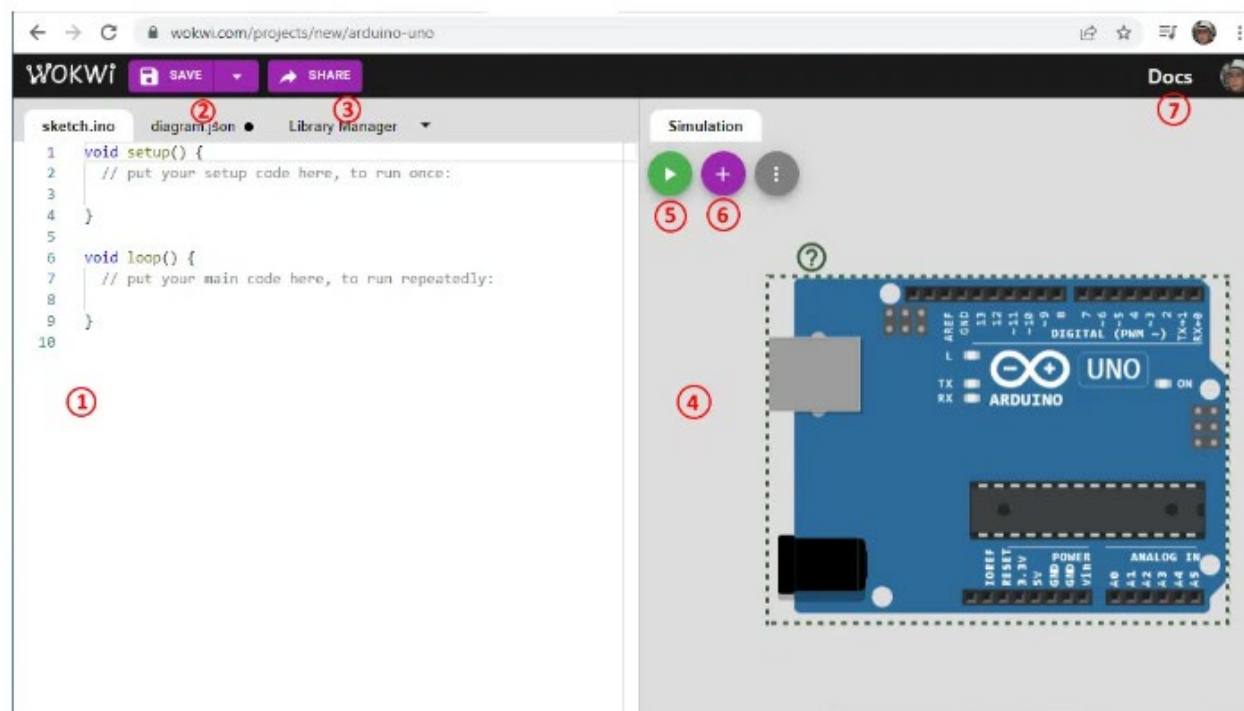


<https://subscription.packtpub.com/book/iot-and-hardware/9781849517584/1/ch01vl1sec11/understanding-arduino-software-architecture>

Emulador/Simulador WOKWI



Tela Principal



- 1: Área de edição do programa
- 2: Edição do diagrama
- 3: Gerenciamento de bibliotecas

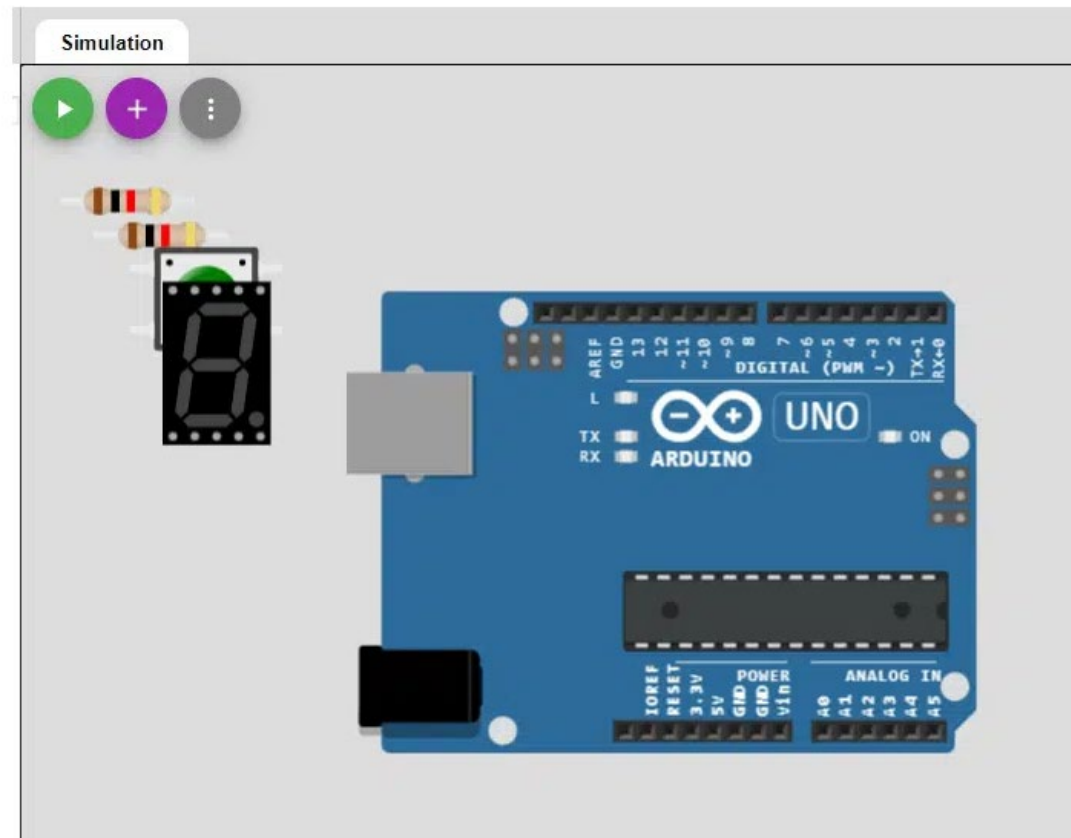
- 4: Área de montagem
- 5: Compila e inicia a simulação
- 6: Acrescenta componentes
- 7: Ajuda

Exemplo de simulação

- <https://www.makerhero.com/blog/wokwi-simulador-de-arduino/>
- <https://www.makerhero.com/blog/como-usar-um-display-7-segmentos-com-o-arduino/>

Exemplo de simulação

<https://wokwi.com/projects/326402385232527956>



Componentes do projeto

<https://www.makerhero.com/blog/wokwi-simulador-de-arduino/>

```
sketch.ino  diagram.json  Library Manager  ▾  
  
1  {  
2  "version": 1,  
3  "author": "Daniel Quadros",  
4  "editor": "wokwi",  
5  "parts": [  
6    { "type": "wokwi-arduino-uno", "id": "uno", "top": -93.33, "left": -41.33, "attrs": {} },  
7    {  
8      "type": "wokwi-resistor",  
9      "id": "r1",  
10     "top": -142.36979166666666,  
11     "left": -166.33333333333334,  
12     "attrs": { "value": "220" }  
13   },  
14   {  
15     "type": "wokwi-resistor",  
16     "id": "r2",  
17     "top": -127.36979166666666,  
18     "left": -151.33333333333334,  
19     "attrs": { "value": "1000" }  
20   },  
21   {  
22     "type": "wokwi-pushbutton",  
23     "id": "btn1",  
24     "top": -112.36979166666666,  
25     "left": -136.33333333333334,  
26     "attrs": { "color": "green" }  
27   },  
28   {  
29     "type": "wokwi-7segment",  
30     "id": "sevsegi",  
31     "top": -97.36979166666666,  
32     "left": -121.33333333333334,  
33     "attrs": {}  
34   }  
35 ],  
36 "connections": []  
37 }
```

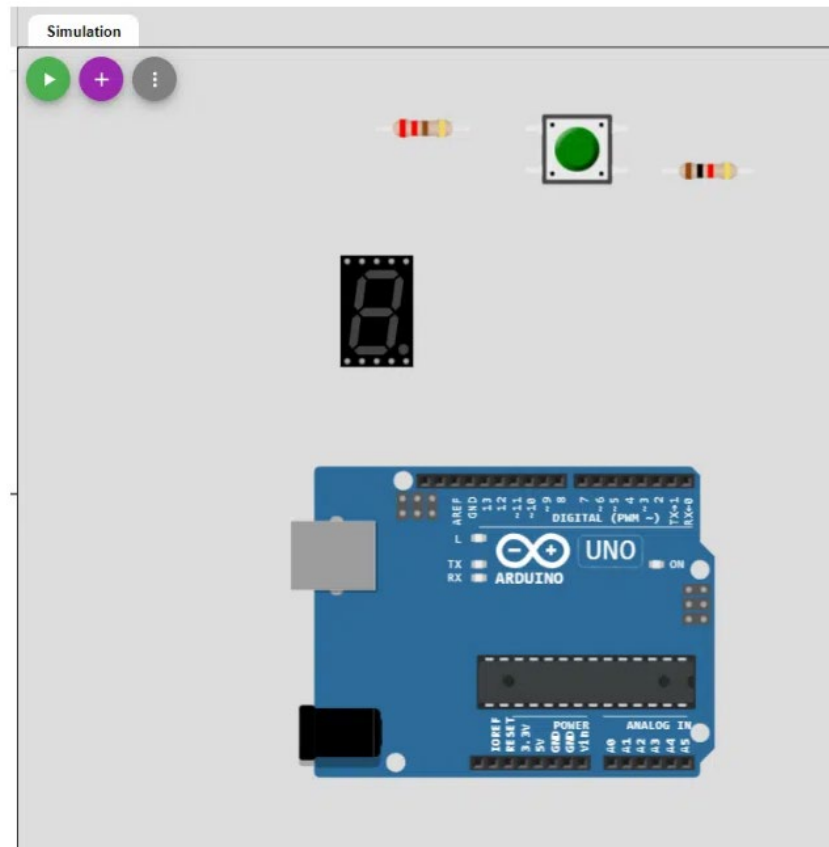
Modificando o atributo value do resistor R1 (de 1000 para 220)

<https://www.makerhero.com/blog/wokwi-simulador-de-arduino/>

```
28   {  
29     "type": "wokwi-7segment",  
30     "id": "sevseg1",  
31     "top": -97.36979166666666,  
32     "left": -121.33333333333334,  
33     "attrs": { "common": "cathode" }  
34   }
```

Acrescentando atributo "cathode"

<https://www.makerhero.com/blog/wokwi-simulador-de-arduino/>

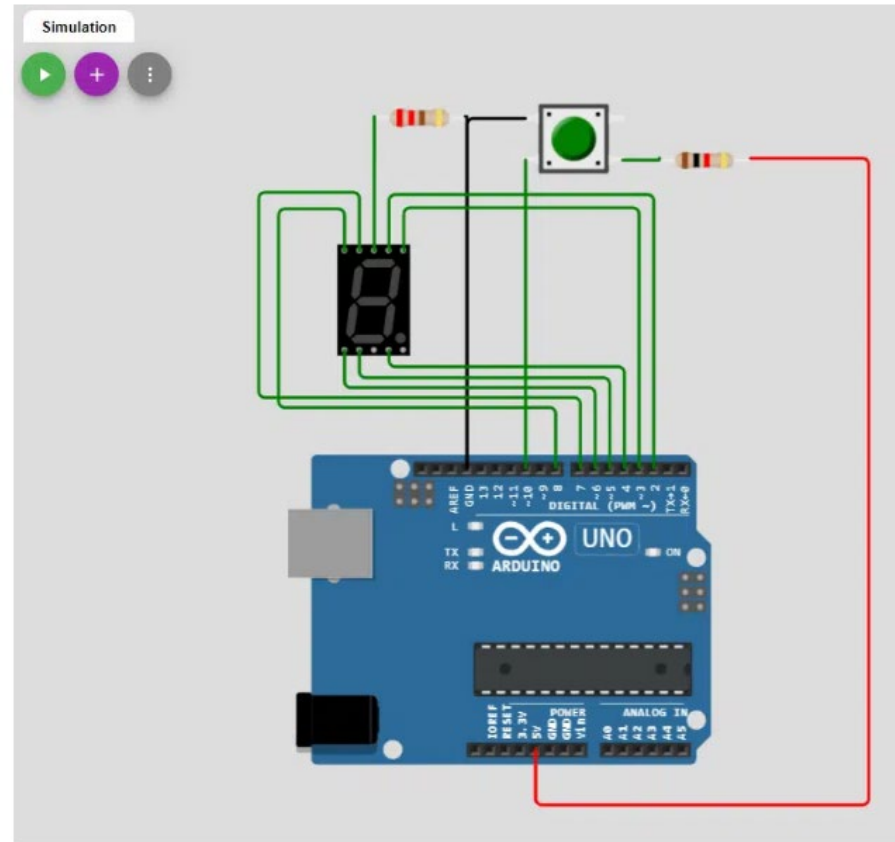


Posicionamento dos Componentes no simulador Wokwi

<https://www.makerhero.com/blog/wokwi-simulador-de-arduino/>

uno:5V	r2:2
uno:GND.1	btn1:1.l
uno:GND.1	r1:2
btn1:2.r	r2:1
sevseg1:COM.2	r1:1
uno:10	btn1:2.l
sevseg1:A	uno:2
sevseg1:B	uno:3
sevseg1:C	uno:4
sevseg1:D	uno:5
sevseg1:E	uno:6
sevseg1:F	uno:7
sevseg1:G	uno:8

<https://www.makerhero.com/blog/wokwi-simulador-de-arduino/>



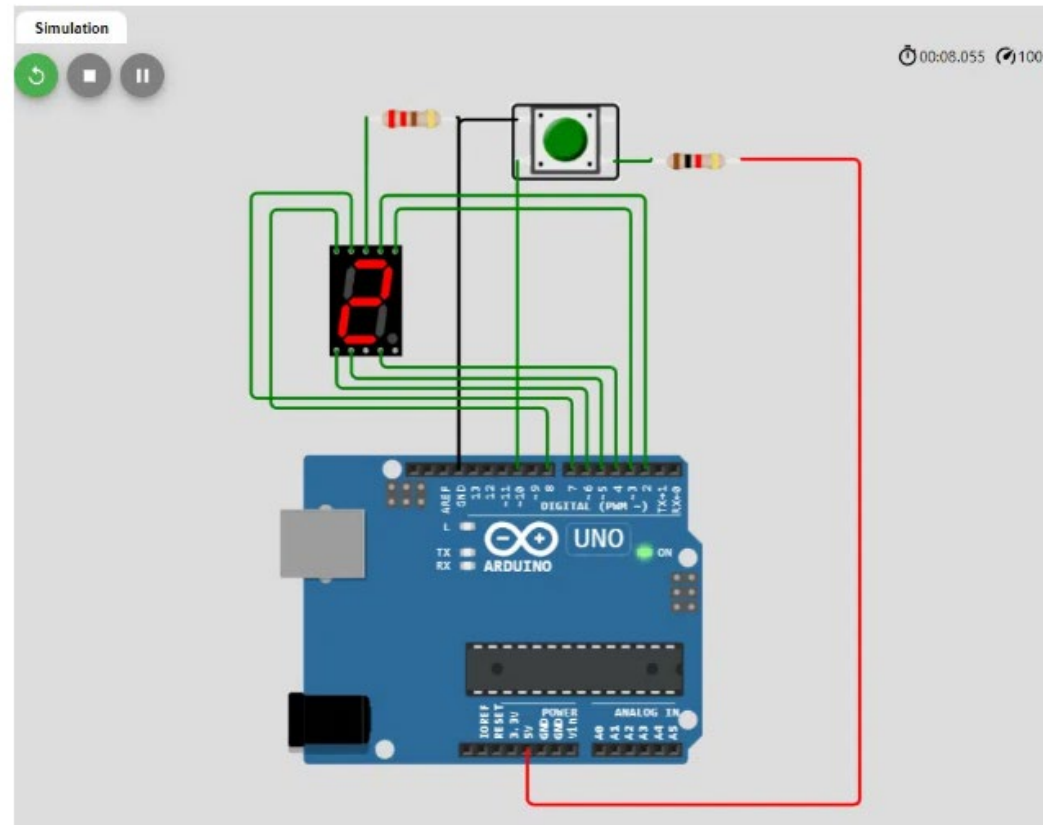
Esquemático do Projeto

```

1 int conta; //
2
3 byte displaySeteSeg[10][7] = {
4
5     { 1,1,1,1,1,1,0 }, //DIGITO 0
6     { 0,1,1,0,0,0,0 }, //DIGITO 1
7     { 1,1,0,1,1,0,1 }, //DIGITO 2
8     { 1,1,1,1,0,0,1 }, //DIGITO 3
9     { 0,1,1,0,0,1,1 }, //DIGITO 4
10    { 1,0,1,1,0,1,1 }, //DIGITO 5
11    { 1,0,1,1,1,1,1 }, //DIGITO 6
12    { 1,1,1,0,0,0,0 }, //DIGITO 7
13    { 1,1,1,1,1,1,1 }, //DIGITO 8
14    { 1,1,1,1,0,1,1 } //DIGITO 9
15
16 };
17
18 void setup(){
19
20     //Definindo pinos como saída
21     pinMode(2, OUTPUT);
22     pinMode(3, OUTPUT);
23     pinMode(4, OUTPUT);
24     pinMode(5, OUTPUT);
25     pinMode(6, OUTPUT);
26     pinMode(7, OUTPUT);
27     pinMode(8, OUTPUT);
28     pinMode(9, OUTPUT);
29     pinMode(10,INPUT);
30
31     //inicializa display com número 0
32     conta = 0;
33     ligaSegmentosDisplay(conta);
34
35 }
36
37 void loop() {
38
39     if(digitalRead(10)== LOW){ //Se o botão for pressionado soma +1 no display
40
41         delay(500);
42
43         conta++; //incremento
44
45         if(conta == 10){ //Se passar de 9 reseta o display para 0
46             conta = 0;
47             ligaSegmentosDisplay(conta);
48         }
49         else{ //Se não mostra no display o próximo número
50             ligaSegmentosDisplay(conta);
51         }
52
53     }
54
55 }
56
57 void ligaSegmentosDisplay(byte digito){ //função para acionar o display com o digito corresponde
58
59     byte pino = 2;
60
61     //Liga os segmentos de acordo com o digito correspondente
62     for (byte contadorSegmentos = 0; contadorSegmentos < 7; ++contadorSegmentos){
63         digitalWrite(pino, displaySeteSeg[digito][contadorSegmentos]);
64         ++pino;
65     }
66 }

```

<https://www.makerhero.com/blog/wokwi-simulador-de-arduino/>



Simulação após apertar o botão duas vezes

Dúvidas?

kofuji@usp.br