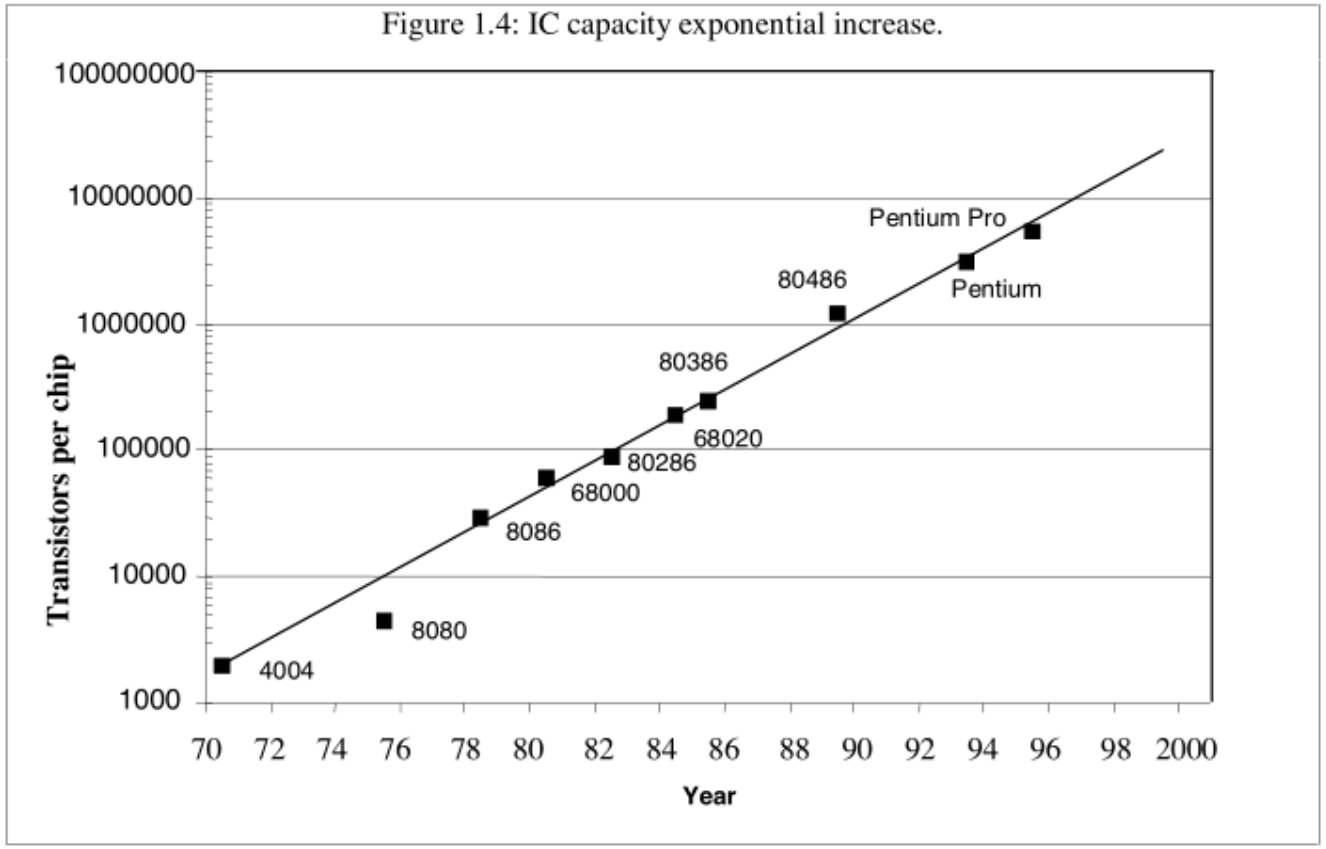
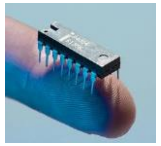
ESP32 IoT Webserver

Hasbi Ash Shiddieqy

Teknologi IC

- Gordon Moore, co-founder of Intel, predicted in 1965 that the transistor density of semiconductor chips would double roughly every 18-24 months. His very accurate prediction is known as "Moore's Law." He recently predicted about another decade before such growth slows down [1]





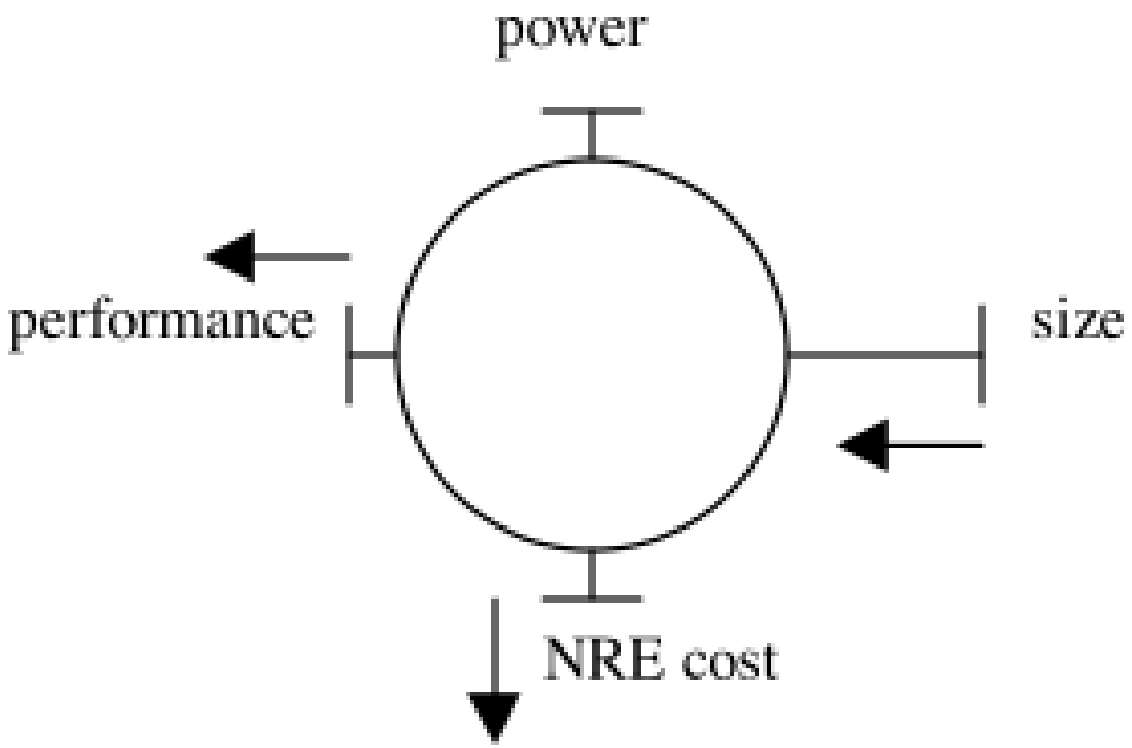
[1] Gordon moore paper 1 and paper 2

Macam-macam Teknik Implementasi Sistem Digital

Mikroprosesor adalah salah satu **cara implementasi sistem digital**. Berikut ini teknik lain yang ada:

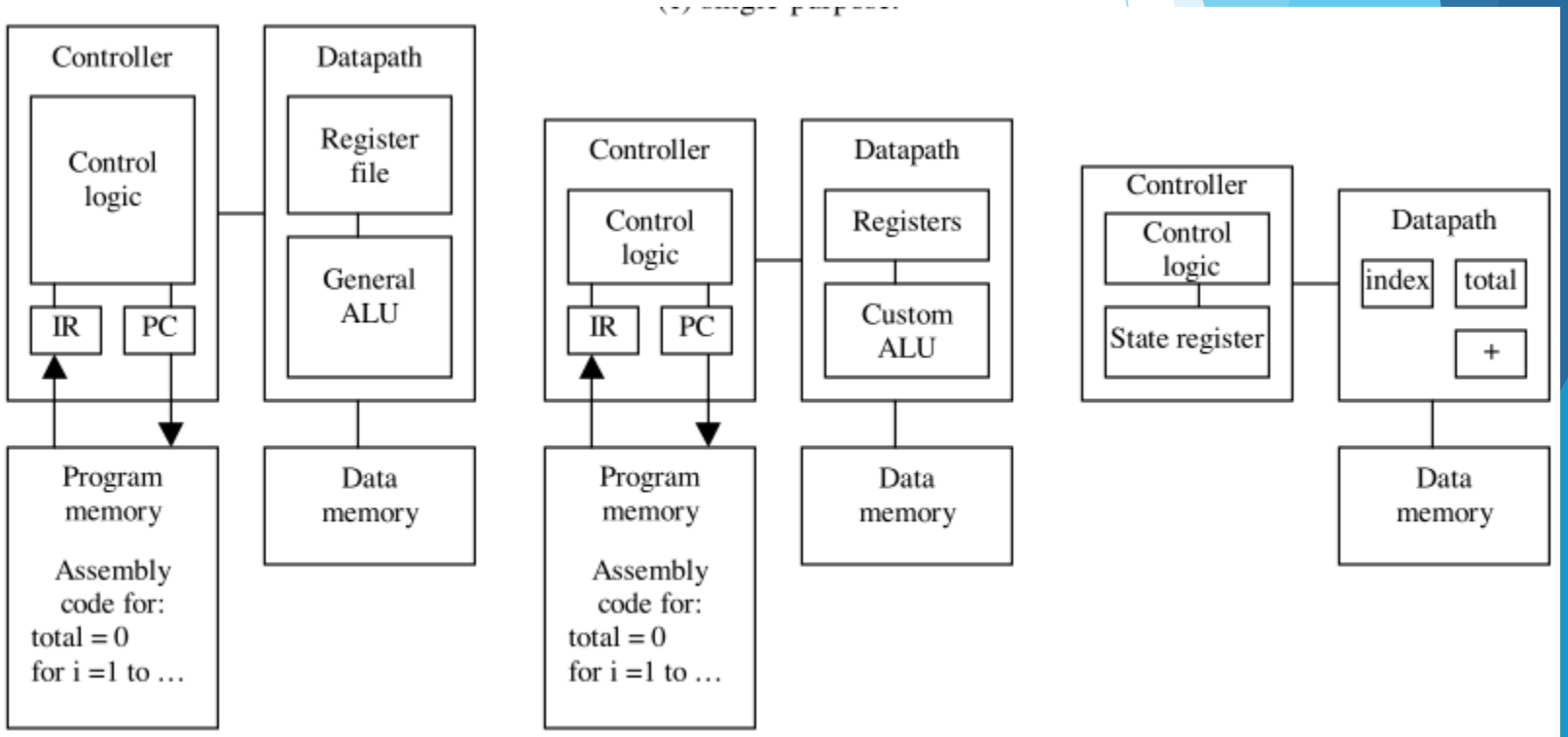
- Transistor diskrit
- Rangkaian Logika (gerbang AND,OR,Flip flop, dsb)
- Rangkaian digital, dengan Register Transfer Level (RTL) -> VHDL, Verilog. Hardware dengan FPGA/ASIC (kuliah sistem digital, perancangan ASIC)
- Mikroprosesor/Mikrokontroler + Software
- SoC (System on Chip): 1 keping silikon berisi CPU+ memori + periferal

Design metric competition



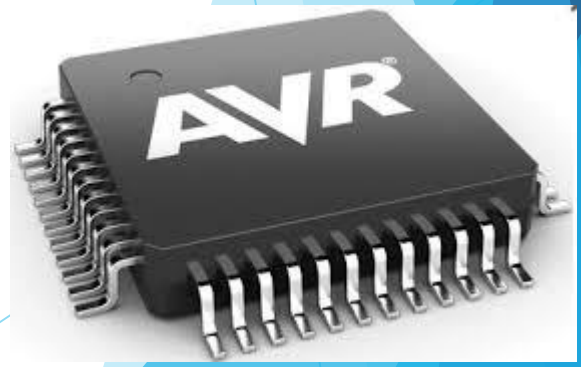
Mikroprosesor/Mikrokontroler

- General purpose
- Application specific
- Single purpose



Mikroprocessor vs Mikrokontroller

- Ketika mikroprosesor dan rangkaian pendukung tergabung, komponen peripheral I/O dan memory (program maupun data) dikumpulkan akan membentuk sebuah komputer kecil yang spesifik untuk akusisi data dan aplikasi kontrol, yang disebut mikrokomputer
- Kalau seluruh komponen tersebut dikumpulkan dalam satu chip maka disebut mikrokontroler



Pemilihan Mikrokontroler

- CPU→ ALU,Register, dekoder instruksi,dll
- Program Memory dan RAM
- Clock Oscillator
- Rangkaian Reset dan Brownout Detector
- Serial Port
- Digital I/O Port dan Analog I/O Port
- Timer dan WDT
- Real Time Clock

Perbandingan mikrokontroller

	8051	PIC	AVR	ARM
Bus width	8-bit for standard core	8/16/32-bit	8/32-bit	32-bit mostly also available in 64-bit
Communication Protocols	UART, USART,SPI,I2C	PIC, UART, USART, LIN, CAN, Ethernet, SPI, I2S	UART, USART, SPI, I2C, (special purpose AVR support CAN, USB, Ethernet)	UART, USART, LIN, I2C, SPI, CAN, USB, Ethernet, I2S, DSP, SAI (serial audio interface), IrDA
Speed	12 Clock/instruction cycle	4 Clock/instruction cycle	1 clock/ instruction cycle	1 clock/ instruction cycle
Memory	ROM, SRAM, FLASH	SRAM, FLASH	Flash, SRAM, EEPROM	Flash, SDRAM, EEPROM
ISA	CLSC	Some feature of RISC	RISC	RISC
Memory Architecture	Von Neumann architecture	Harvard architecture	Modified	Modified Harvard architecture
Power Consumption	Average	Low	Low	Low
Families	8051 variants	PIC16,PIC17, PIC18, PIC24, PIC32	Tiny, Atmega, Xmega, special purpose AVR	ARMv4,5,6,7 and series
Community	Vast	Very Good	Very Good	Vast
Manufacturer	NXP, Atmel, Silicon Labs, Dallas, Cyprus, Infineon, etc.	Microchip Average	Atmel	Apple, Nvidia, Qualcomm, Samsung Electronics, and TI etc.
Cost (as compared to features provide)	Very Low	Average	Average	Low
Other Feature	Known for its Standard	Cheap	Cheap, effective	High speed operation Vast
Popular Microcontrollers	AT89C51, P89v51, etc.	PIC18fXX8, PIC16f88X, PIC32MXX	Atmega8, 16, 32, Arduino Community	LPC2148, ARM Cortex-M0 to ARM Cortex-M7, etc.

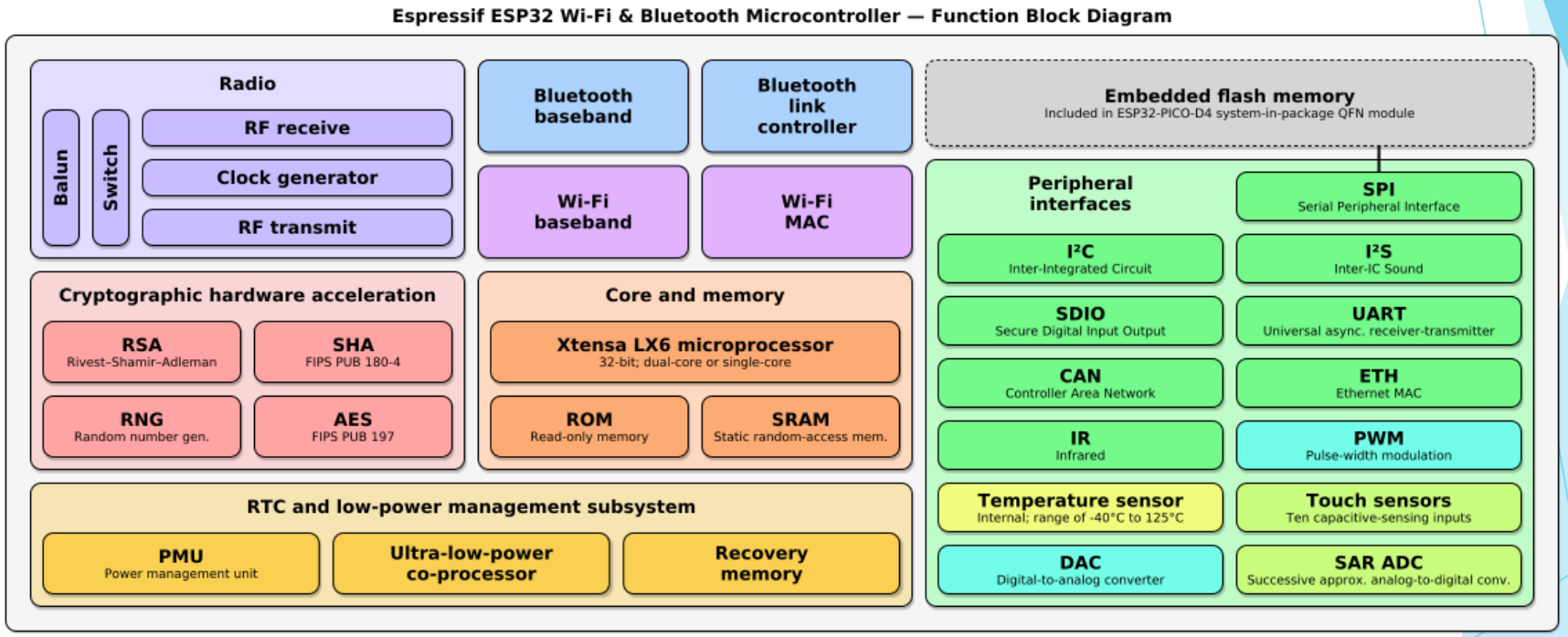
Goal

- Memahami system kerja ESP32
- Mengunakan Input Output pada ESP32
- Menghubungkan wifi pada ESP32
- Membuat webserver sederhana

Perbandingan Prosesor

Model	Clock	Flash	SRAM
ATMega328 (Arduino Nano)	16 Mhz	32 kB	2 kB
STM32F103C8T (Blue Pill)	72 Mhz	64 kB	20 kB
LPC1769 (LPCXpresso)	100 MHz	512 kB	64 kB
ESP32	240 MHz (600 MIPS)	External ~16 MB (tipikal 4 MB)	520 kB
ESP8266	80 ~ 160 MHz	External ~ 16 MiB	80 kB

Diagram Espressif ESP32



ESP32 System Structure

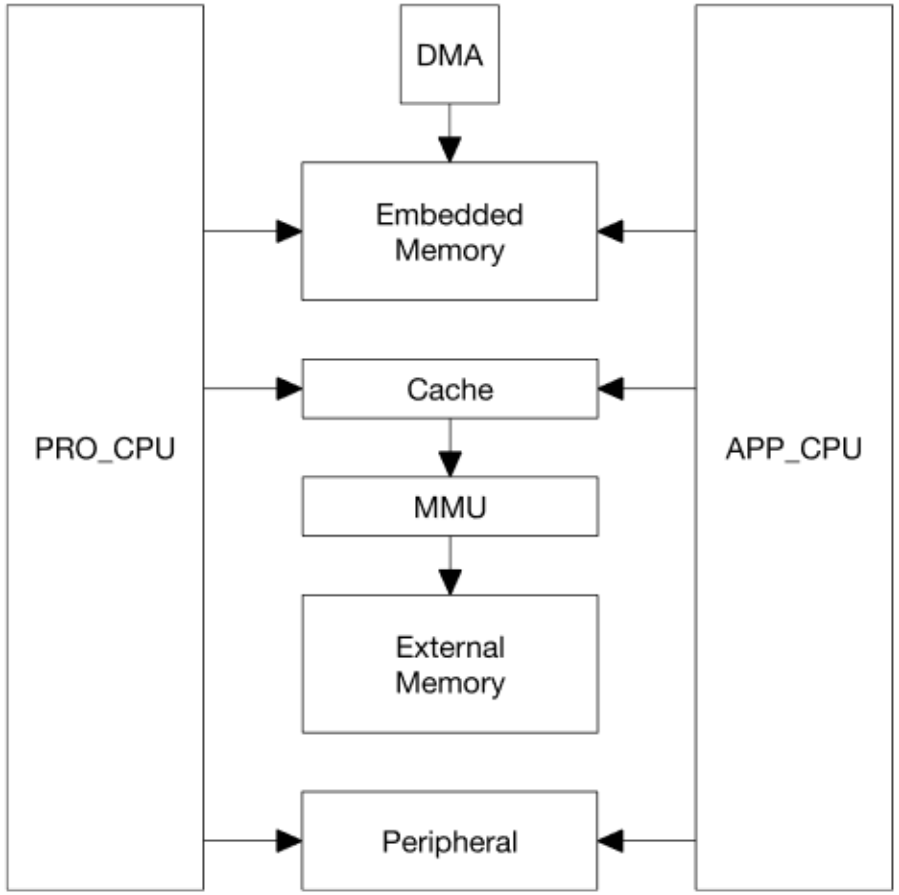
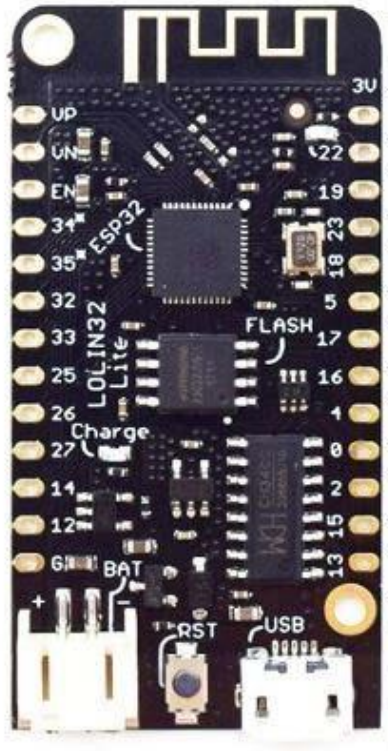


Figure 1: System Structure

https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf

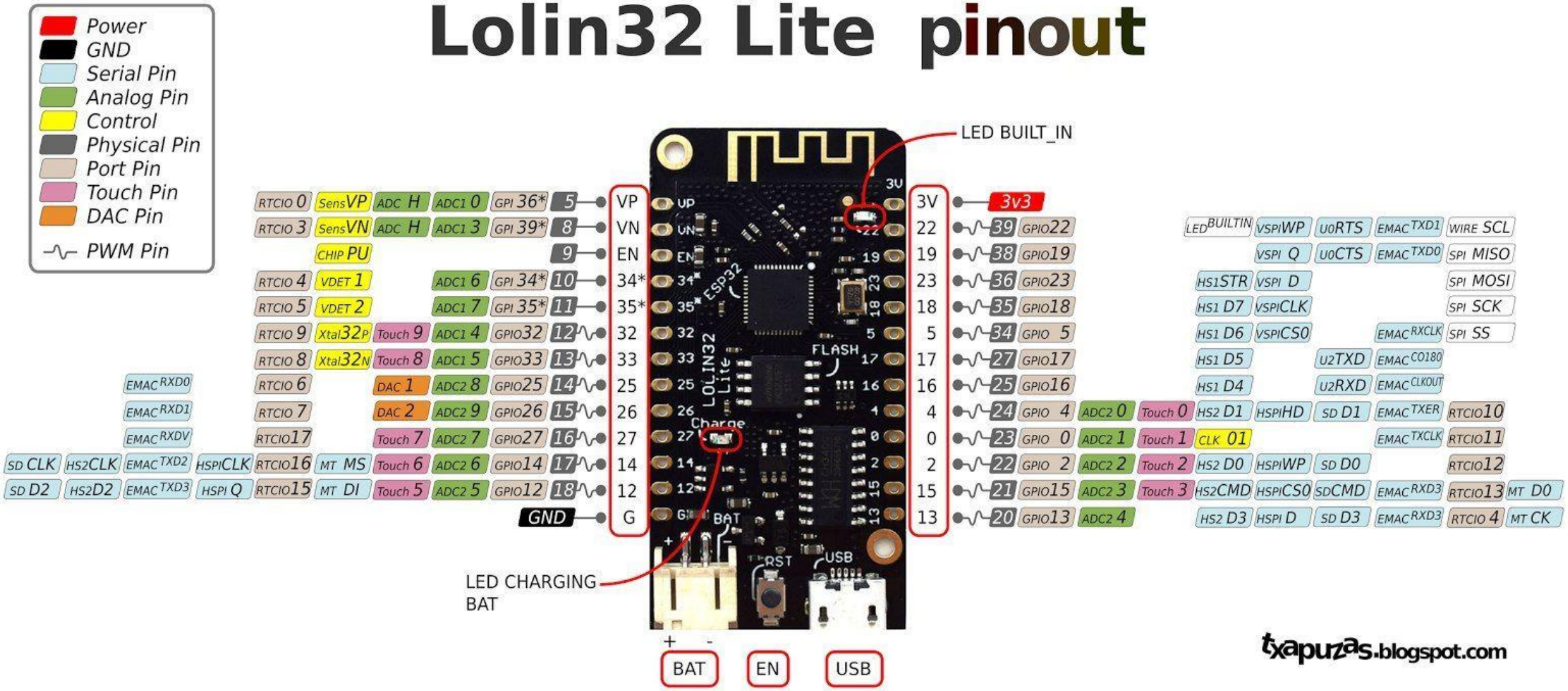
Pinout Lolin32 Lite (ESP32)

						VP
						VN
					CHIP_PU	EN
			RTC4	ADC1:6	GPIO34	34
				ADC1:7	GPIO35	35
			RTC9	ADC1:4	GPIO32	32
			RTC8	ADC1:5	GPIO33	33
		DAC1	RTC6	ADC2:8	GPIO25	25
		DAC2	RTC7	ADC2:9	GPIO26	26
	TOUCH7		RTC17	ADC2:7	GPIO27	27
SPICLK	TOUCH6		RTC16	ADC2:6	GPIO14	14
	TOUCH5		RTC15	ADC2:5	GPIO12	12
						GND



3V3							
22	GPIO22	HS2_CMD	VSPIWP				
19	GPIO19	HS2_DATA2	VSPIQ				
23	GPIO23						
18	GPIO18	HS1_DATA7	VSPICLK				
5	GPIO5	HS1_DATA6	VSPICS0				
17	GPIO17	HS1_DATA5					
16	GPIO16	HS1_DATA4					
4	GPIO4	ADC2:0	TOUCH0	RTC10	SPIHD		
0	GPIO0	ADC2:1	TOUCH1	RTC11	CLK_OUT1		
2	GPIO2	ADC2:2	TOUCH2	RTC12	SPIWP		
15	GPIO15	ADC2:3	TOUCH3	RTC13	SPICS0	U0RTS	MTDO
13	GPIO0	ADC2:4	TOUCH4	RTC14	SPIO	U0CTS	MTCK

Lolin32 Lite pinout



Komponen Utama Lolin32

- **ESP32-D0WDQ6** (ESP32-D0WDQ6, Dual core, No embedded flash, Wi-Fi b/g/n + BT/BLE Dual Mode, QFN 6*6“)
- **W25Q32FVSS** (3V 32M-bit Serial Flash Memory With Dual/Quad SPI & QPI)
  UMH3N (General purpose dual digital transistor)
- **CH340C** (USB to serial chip)
- **ME6211** (High speed LDO regulators)
- **TP4054** (TP4054 Standalone Linear Li-lon Battery Charger with Thermal Regulation in SOT) .
- **LED** onboard (pin 22, GPIO22, active low)

Komponen Utama Lolin32

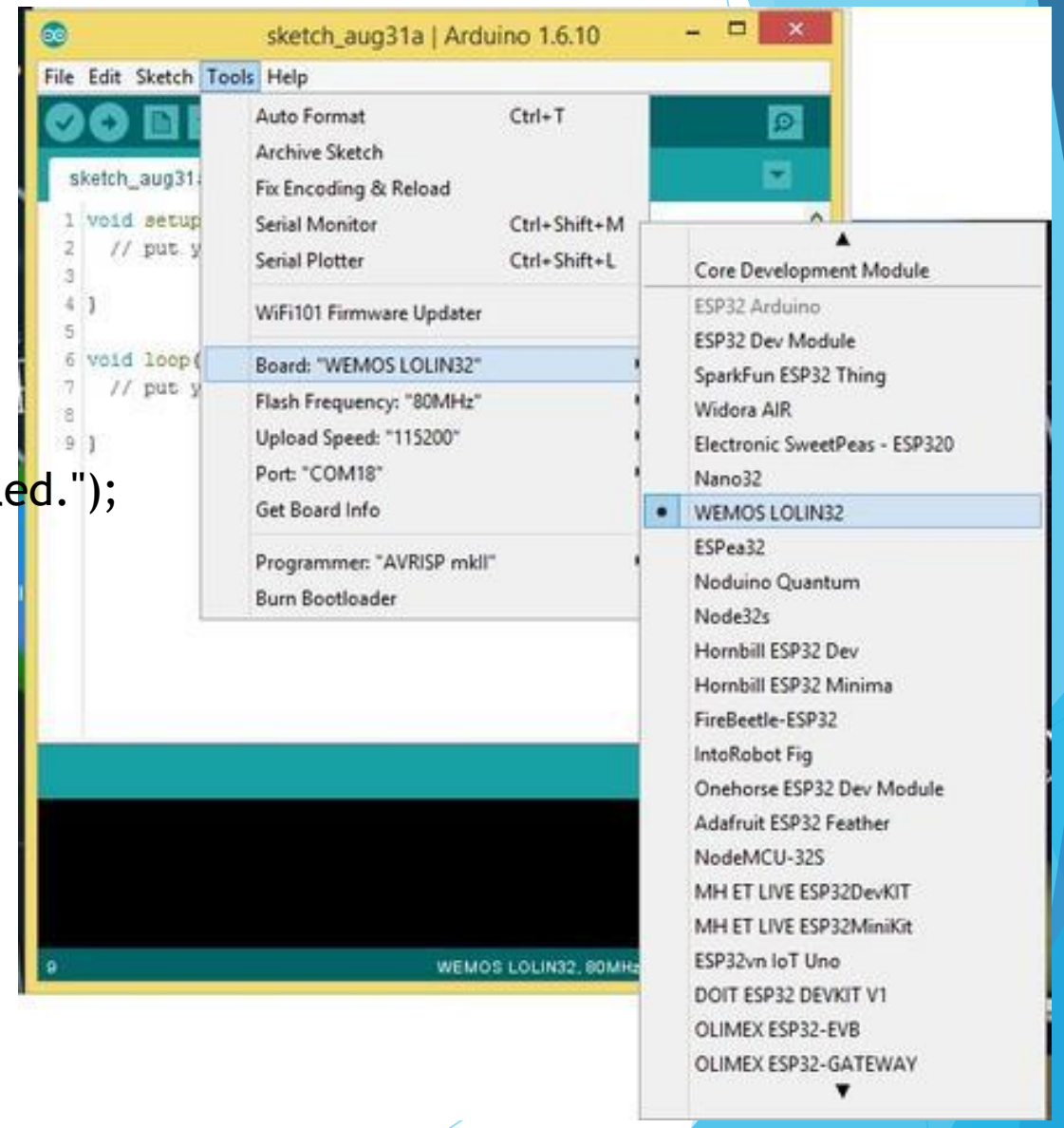
- **ESP32-D0WDQ6** (ESP32-D0WDQ6, Dual core, No embedded flash, Wi-Fi b/g/n + BT/BLE Dual Mode, QFN 6*6“)
- **W25Q32FVSS** (3V 32M-bit Serial Flash Memory With Dual/Quad SPI & QPI) https://www.winbond.com/resource-files/w25q32fv%20revi%2010202015.pdf
- **UMH3N** (General purpose dual digital transistor) http://rohmfs.rohm.com/en/products/databook/datasheet/discrete/transistor/digital/emh3t2r-e.pdf , https://www.mouser.com/ds/2/348/umh3ntn-e-1018108.pdf
- **CH340C** (USB to serial chip) https://www.mpja.com/download/35227cpdata.pdf
- **ME6211** (High speed LDO regulators) https://datasheet.lcsc.com/szlcsc/Nanjing-Micro-One-Elec-ME6211C33M5G-N_C82942.pdf
- **TP4054** (TP4054 Standalone Linear Li-Ion Battery Charger with Thermal Regulation in SOT) . https://www.piekarz.pl/pl/pdf.php?id=30089 (English) , https://datasheet.lcsc.com/szlcsc/Nanjing-Extension-Microelectronics-TP4054_C32574.pdf (Chinese)

Menambahkan library Lolin32

- Install UART diriver usb to seial
- Download board dari github esp32 copy to `C:\Users\[UserName]\Documents\Arduino\hardware\espressif\`
- Run `C:\Users\[User Name]\Documents\Arduino\hardware\espressif\tools\get.exe`
- Pop akan terbuka dan tunggu hingga process instalasi selesai
- Compile dan run simple code

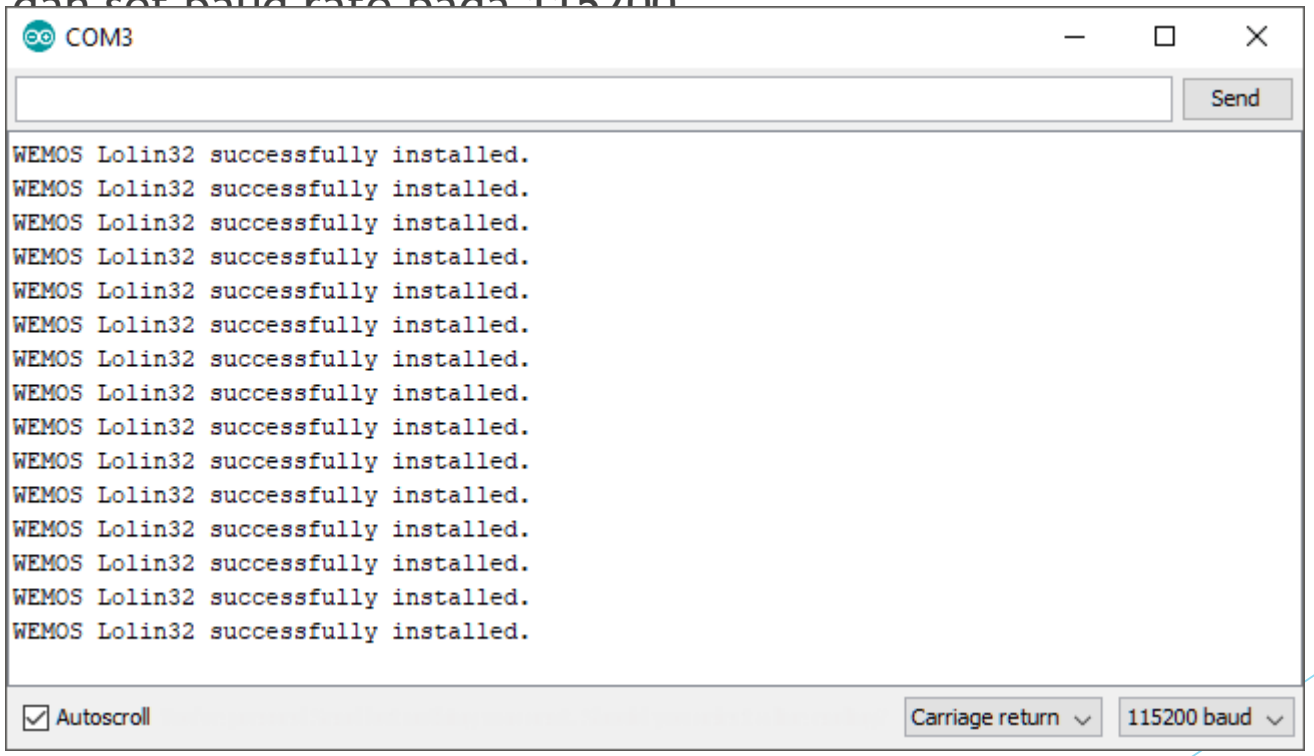
Simple Code

```
void setup() {
Serial.begin(115200);
            }
void loop() {
 Serial.println("WEMOS Lolin32 successfully installed.");
 delay(500);
}
```



Testprogram

- Open Serial Monitor dan set baud rate pada 115200



Blink Test pada LED (built in LED)

```
int ledPin = 22;

void setup()
{
    pinMode(ledPin, OUTPUT);
    Serial.begin(115200);
}
```

```
void loop()
{
    Serial.println("Hello ESP32!");
    digitalWrite(ledPin, HIGH);
    delay(500);
    digitalWrite(ledPin, LOW);
    delay(500);
}
```

Input Touch sensor

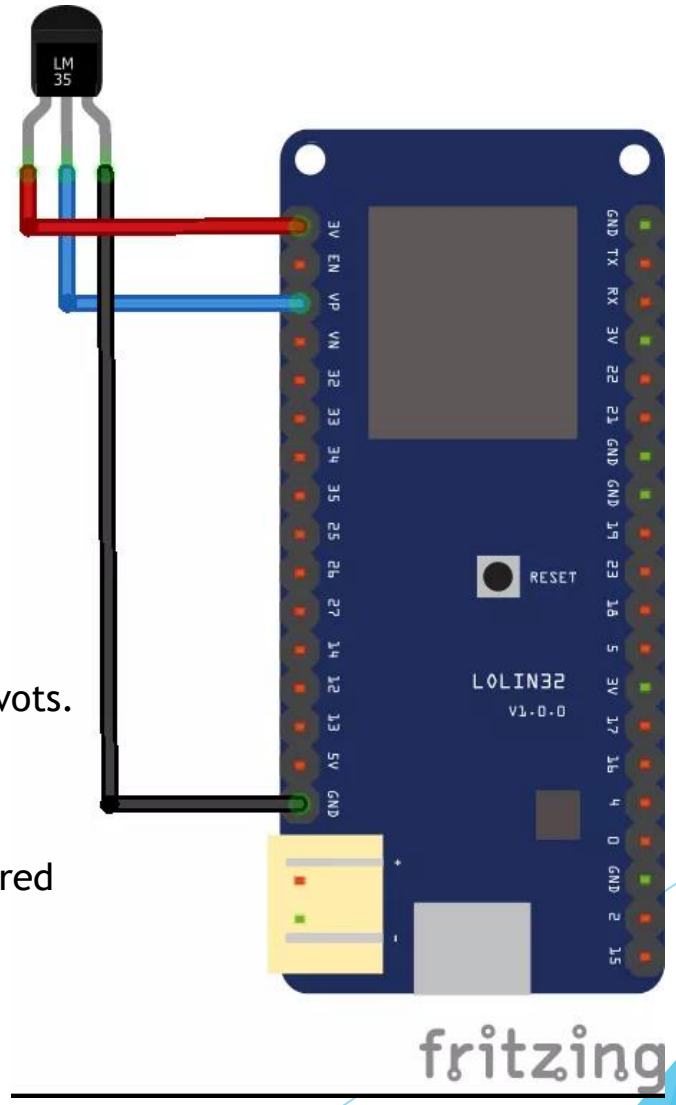
```
#define TOUCH_PIN T0 //connected to 4
#define LED_PIN 22 //connected to 22
int touch_value = 100;

void setup()
{
Serial.begin(115200);
Serial.println("ESP32 Touch Test");
pinMode(LED_PIN, OUTPUT);
digitalWrite (LED_PIN, LOW);
}
```

using T0

```
void loop(){
touch_value = touchRead(TOUCH_PIN);
Serial.println(touch_value); // get value

if (touch_value < 50){
digitalWrite (LED_PIN, HIGH);
}
else{
digitalWrite (LED_PIN, LOW);
}
delay(500);
}
```

ESP32 dan lm35 sensor

```
const int analogIn = A0;

int RawValue= 0;
double Voltage = 0;
double tempC = 0;
double tempF = 0;

void setup(){
Serial.begin(115200);
}

void loop(){

RawValue = analogRead(analogIn);
Voltage = (RawValue / 2048.0) * 3300; // 5000 to get millivots.
tempC = Voltage * 0.1;
Serial.print("Raw Value = " ); // shows pre-scaled value
Serial.print(RawValue);
Serial.print("\t milli volts = "); // shows the voltage measured
Serial.print(Voltage,0); //
Serial.print("\t Temperature in C = ");
Serial.print(tempC,1); delay(500);
}
```

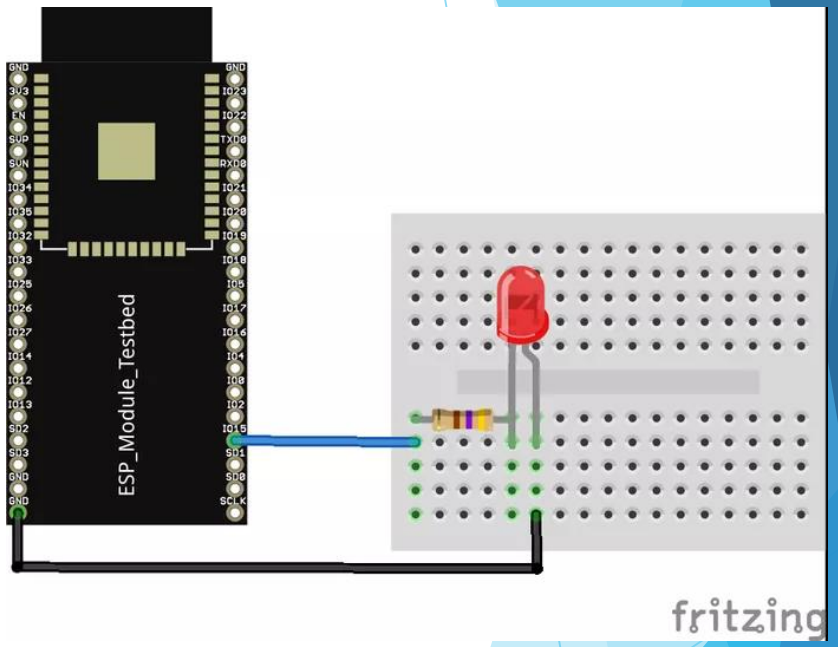


WEBserver Lolin32

```
#include <WiFi.h>

// Replace with your network credentials
const char* ssid     = "HotspotITB";
const char* password = "hotspotitb";
const int analogIn = A0;

int RawValue= 0;
double Voltage = 0;
double tempC = 0;

 WiFiServer server(80);

const int led =  15;      // the number of the LED pin

// Client variables
char linebuf[80];
int charcount=0;

void setup() {
  // initialize the LED as an output:
  pinMode(led, OUTPUT);
  //Initialize serial and wait for port to open:
  Serial.begin(115200);
  while(!Serial) {
  }
   // We start by connecting to a WiFi network
  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);

  // attempt to connect to Wifi network:
  while(WiFi.status() != WL_CONNECTED)
  {
    // Connect to WPA/WPA2 network.
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
  server.begin();
}
 void loop()
{
  // listen for incoming clients
  WiFiClient client = server.available();
  if (client)  {
    Serial.println("New client");
    memset(linebuf,0,sizeof(linebuf));
    charcount=0;
    // an http request ends with a blank line
    boolean currentLineIsBlank = true;
    while (client.connected())
    {
      if (client.available())
      {
        char c = client.read();
        Serial.write(c);
        //read char by char HTTP request
        linebuf[charcount]=c;
        if (charcount<sizeof(linebuf)-1) charcount++;

        if (c == '\n' && currentLineIsBlank)
        {
          RawValue = analogRead(analogIn);
          Voltage = (RawValue / 2048.0) * 3300; // 5000 to get millivots.
          tempC = Voltage * 0.1;
          // send a standard http response header
          client.println("HTTP/1.1 200 OK");
          client.println("Content-Type: text/html");
          client.println("Connection: close");  // the connection will be closed after completion of the response
          client.println();
          client.println("<!DOCTYPE HTML><html><head>");
          client.println("<meta name=\"viewport\" content=\"width=device-width, initial-scale=1\"></head>");
          client.println("<h1>ESP32 - Web Server</h1>");
          client.println("<p>LED <a href=\"on\"><button>ON</button></a> <a href=\"off\"><button>OFF</button></a></p>");
          client.println(tempC);
          client.println("*C</p><p>");
          client.println("</html>");
          break;
        }
        if (c == '\n')
        {
          // you're starting a new line
          currentLineIsBlank = true;
          if (strstr(linebuf,"GET /on") > 0)  {
            Serial.println("LED ON");
            digitalWrite(led, HIGH);
          }
          else if (strstr(linebuf,"GET /off") > 0) {
            Serial.println("LED OFF");
            digitalWrite(led, LOW);
          }

          // you're starting a new line
          currentLineIsBlank = true;
          memset(linebuf,0,sizeof(linebuf));
          charcount=0;
        }
        else if (c != '\r')        {
          // you've gotten a character on the current line
          currentLineIsBlank = false;
        }
      }
    }
    // give the web browser time to receive the data
    delay(1);
     // close the connection:
    client.stop();
    Serial.println("client disconnected");
  }
}
```

Web test


192.168.1.7/on
Apps
TV Guide & Schedule
Sky Guide TV Listing
OLED Kit for
ESP32 - Web Server
LED
ON
OFF
22.19 *C


ESP_Module_Testbed
fritzing

Reference

- http://www.esp32learning.com
- https://www.connectedcities.com.ph/blogs/tutorial/getting-started-with-wemos-lolin32
- https://github.com/espressif/arduino-esp32