A Quick Introduction: MCU

Gunesh.raj@gmail.com


NOT THIS MCU
Disney ©

MCU – Micro Controller Unit

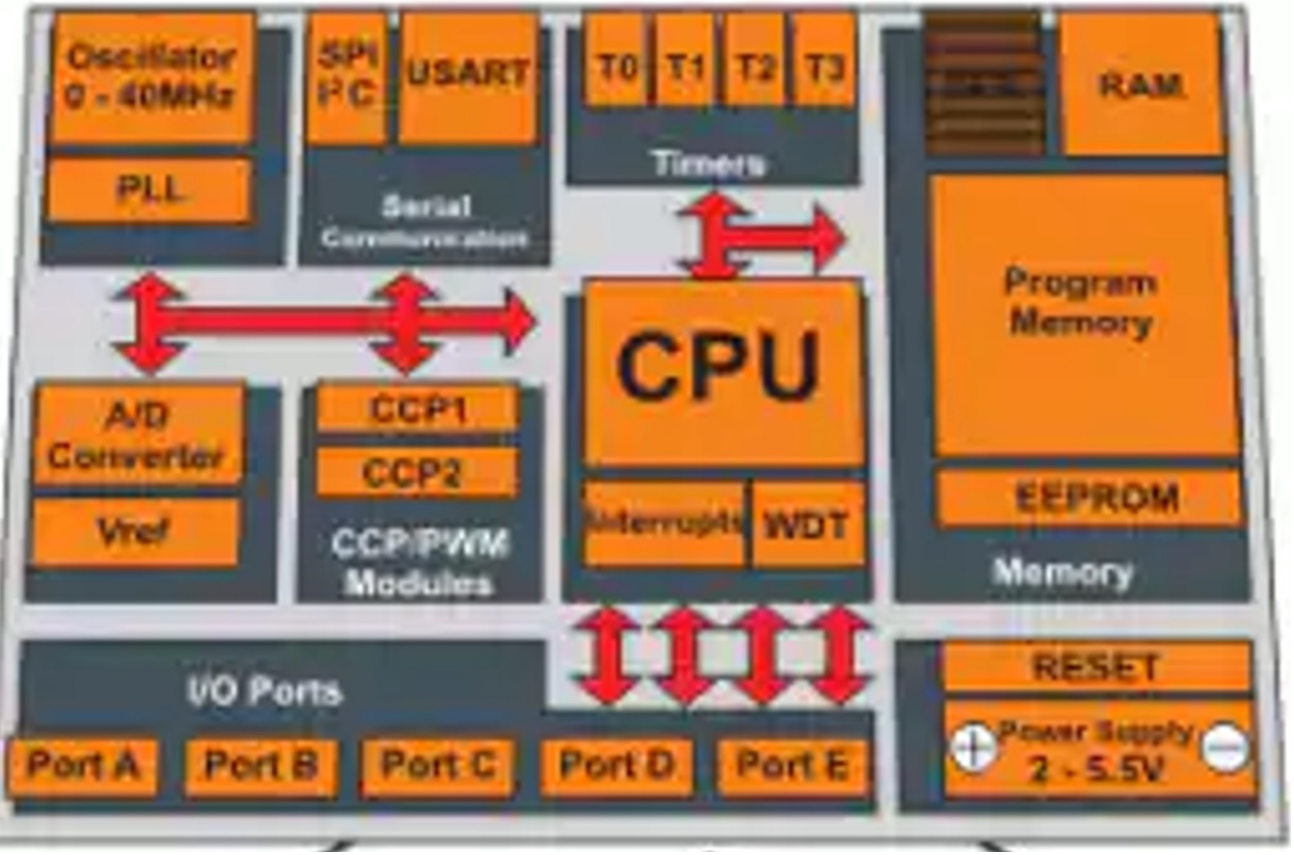
- uC Symbol, Runs from 1.8V – 20V
- Low powered, usually 8, 16 Bits Processor, 1 or 2 Cores, 1KHz – 133MHz
- SRAM for the Executable Code to Run
- EEPROM for some Storage
- Flash for the Executable Code
- ARM, RISC, x86 Chipsets & Custom ones
- Arduino ATMEGA, ESP, STM, RP Core are popular
- Analog IO, Digital IO, Supports: High, Low, Pull up, Pull Down, Falling.
- Supports I2C, SPI and various protocols.

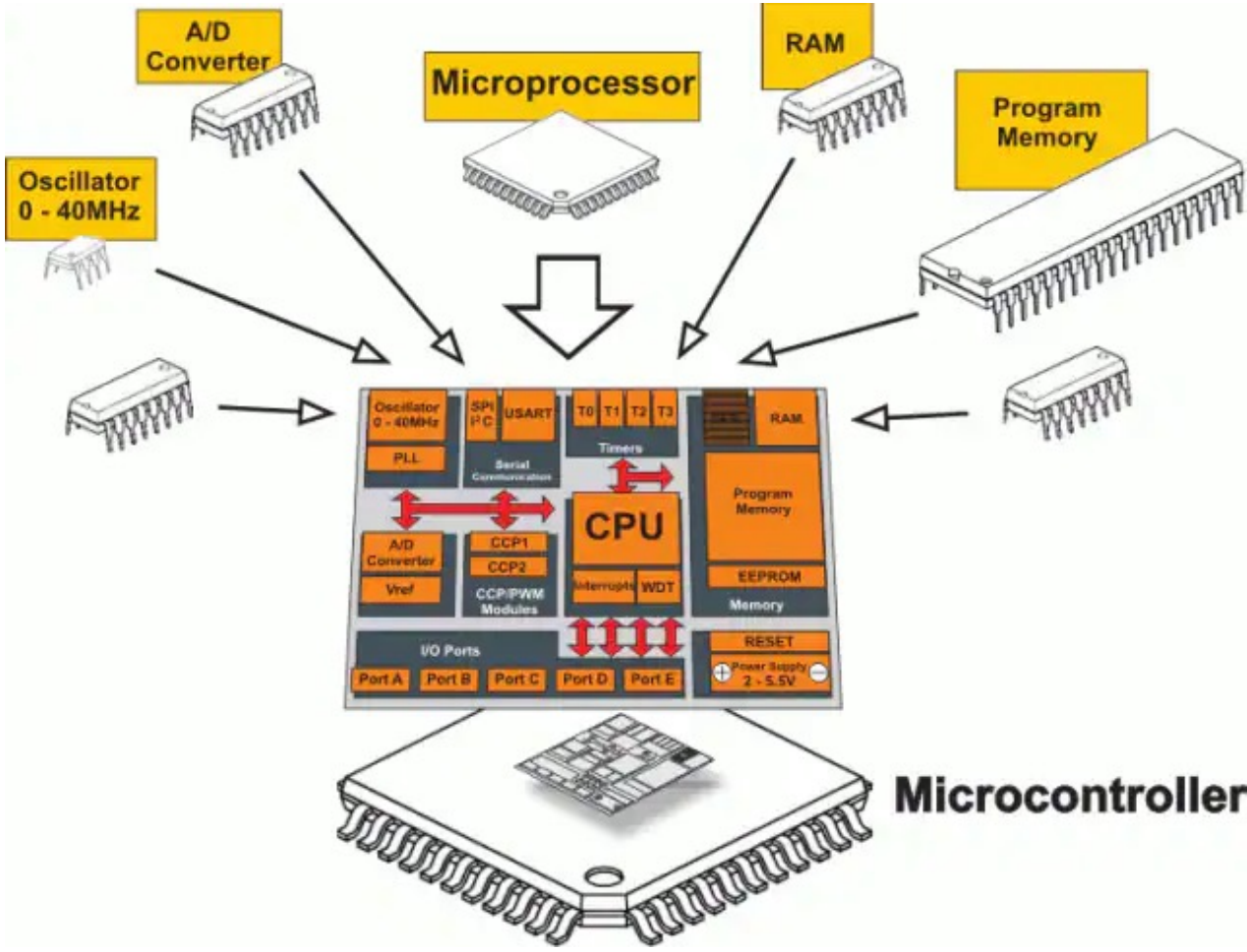
Also, MCU?

- MCU is not a PLC.
  - MCU is like Bare Metal C Stuff, ~~PLC is like Drag and Drop VB (Mostly).~~ PLC is suited mostly for Industrial Usage usually with Ladder Programming or SCADA.
- MCU Could run in Real Time for both Cores.
- Programming is mostly in C, Python, Rust, JavaScript and others are getting popular
- Arduino IDE, Thorny, Platform IO, Cube, Microchip Studio.
- Some has Excellent Datasheet – Read them first.
- Code -> Cross Compile to ARCH -> Flash to MCU
- Can’t run a Full Blown OS, but still can achieve multi tasking.

CPU = MCU – MEMORY MANAGEMENT

What's in the MCU?





Why MCU?

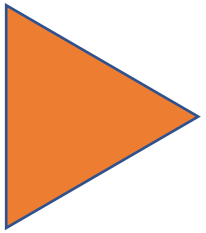
INPUT

SENSORS

BEACONS

SIGNAL VARIATION

INTERUPTS



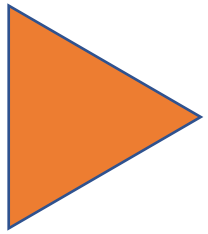
PROCESS

COMPUTE

SLEEP

HIBERNATE

WATCHDOG TIMER



OUTPUT

POWER RELAY

ANALOG

DIGITAL / PWM

SELF TIMER

GYRO, OPTO, REED, SWITCH, VARIABLE RESISTORS, CAMERA, GPS, WATER SENSOR, SOUND, SHOCK, HEAT, POWER, COMMUNICATION

LED, MOTOR, AUDIO, SWITCHES, LASER, DISPLAYS, COMMUNICATIONS, RELAYS

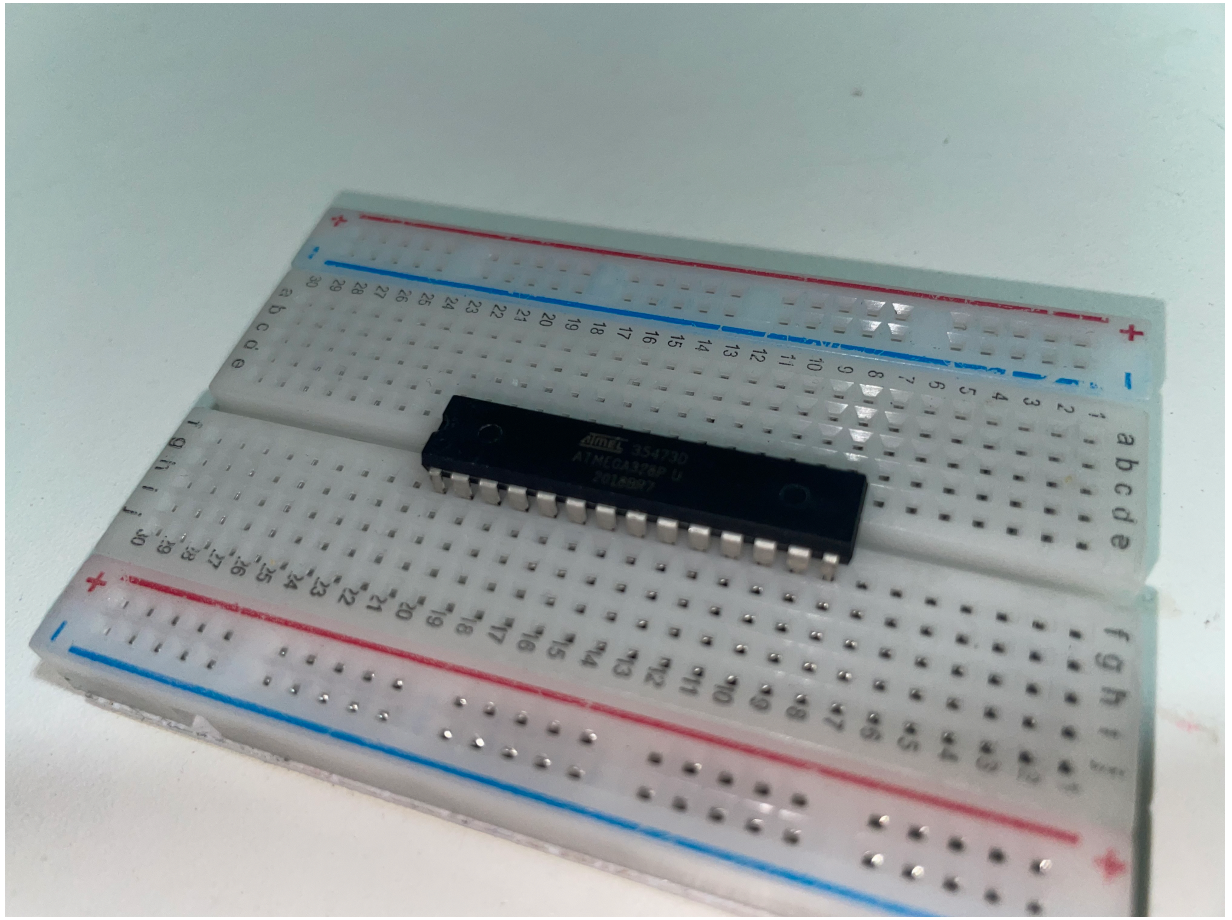
- **Low Powered, efficient, Battery Powered**
- **Do a simple tasks (If this Then A, or B)**
- **Easy to replicate, Manufacture**
- **Cheaper alternative to a PC Arch or a PLC**
- **Lots of IO Peripherals to use**

- **No OS needed, Just a Boot Loader**
- **RTOS Support**
- **IOT is the Future, AI Models can work**
- **Fine Tuning a process**

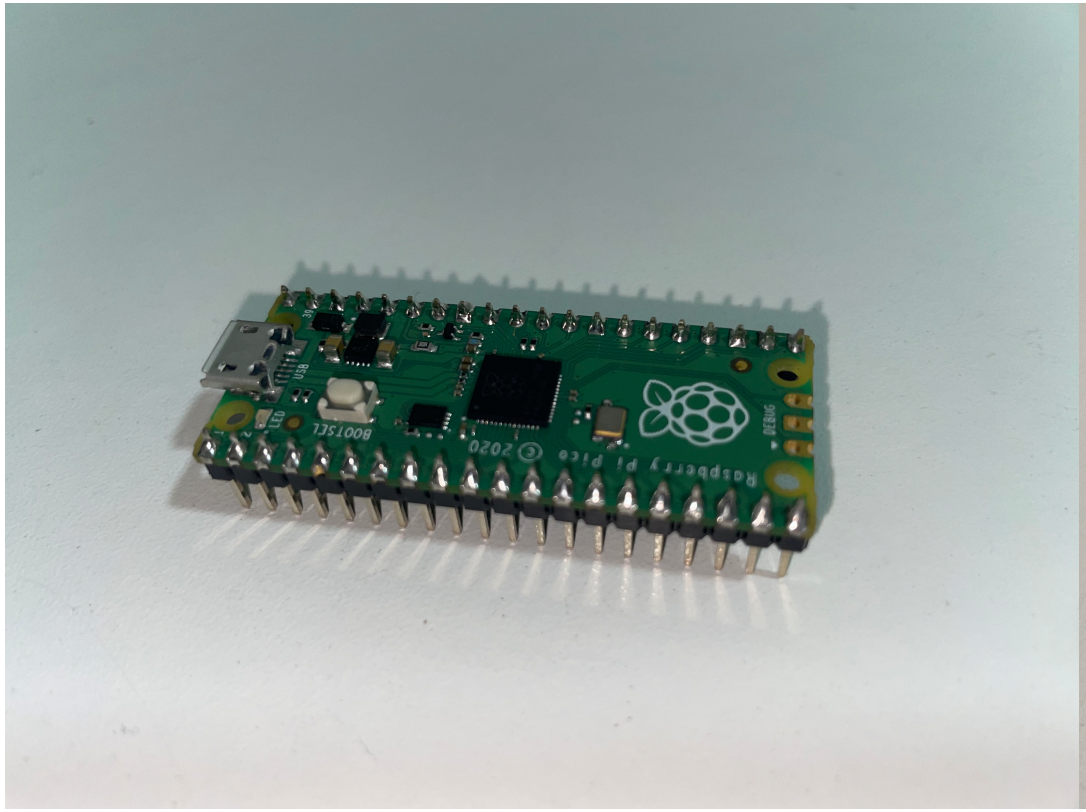
MCU PROCESSORS AND BOARDS

- BARE METAL SHIPS
- MCU WITH BUILT IN WIFI
- MCU WITH ANOTHER COMPONENT
- MCU WITH ANOTHER MCU
- MCU WITH A FORM OF ANOTHER MCU BOARD

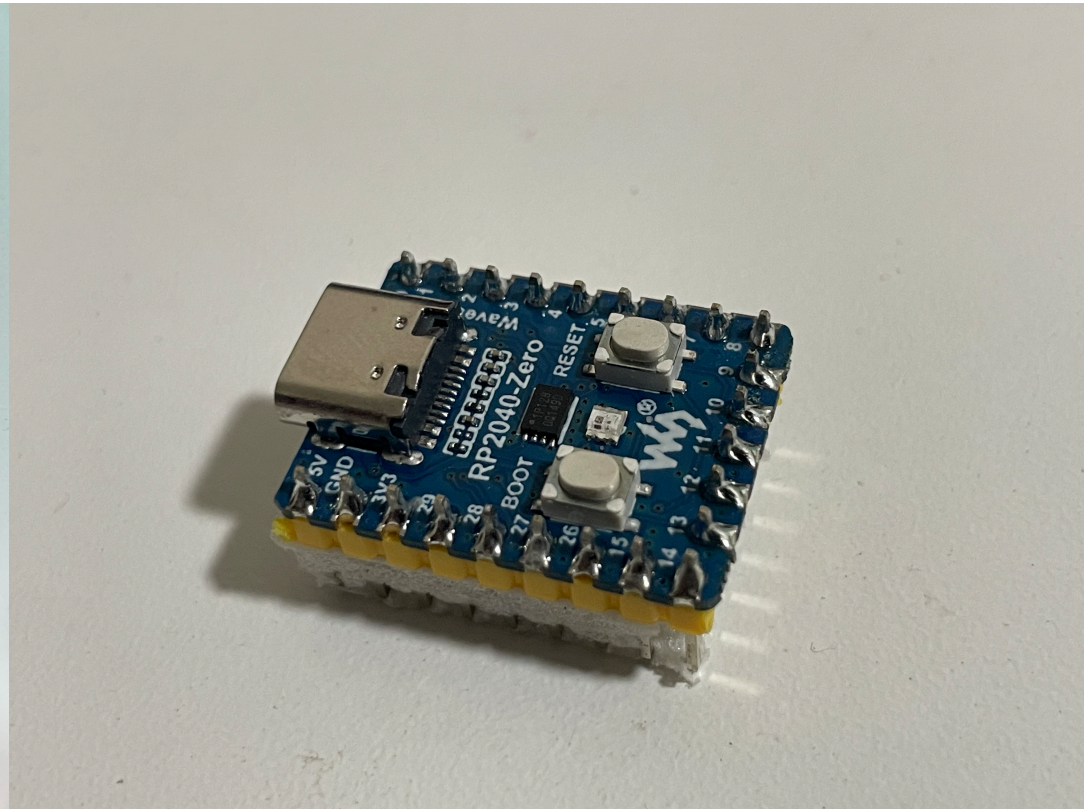
ARDUINO BARE METAL



RP2040 – RASPBERRY FOUNDATION

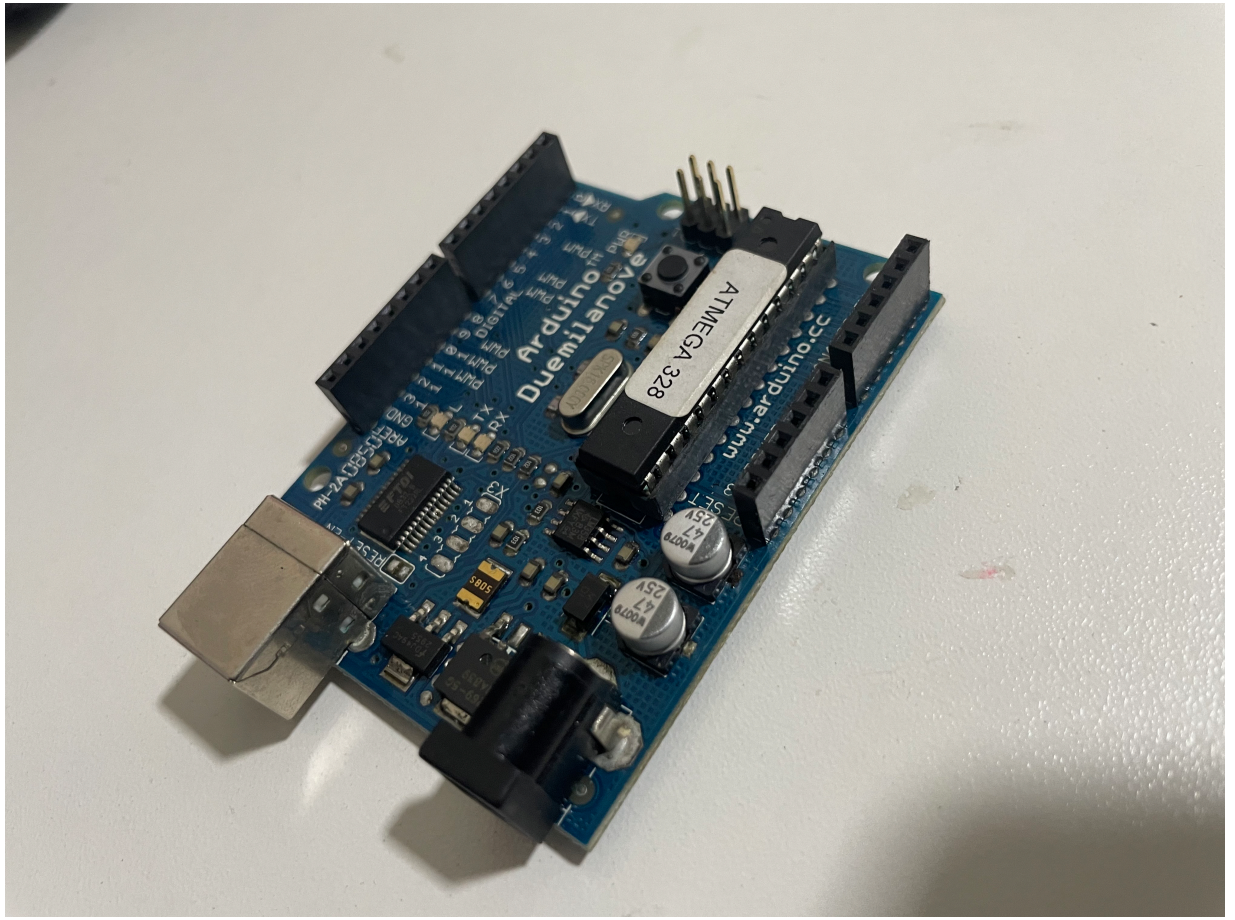


ORIGINAL



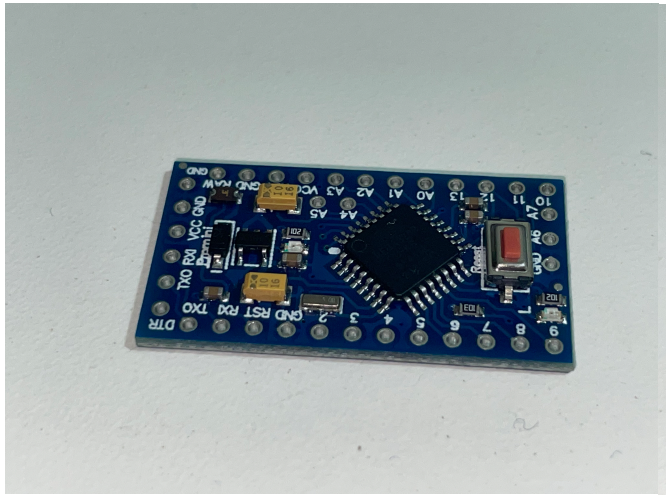
USING THE RP40 CHIP

ARDUINO DUEMINALOVE

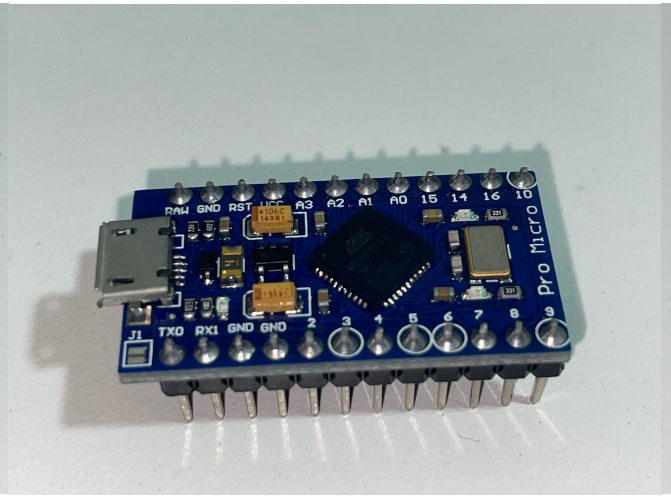


ONE OF THE OLDER ONES, 2009 WITH THE 328 CHIP

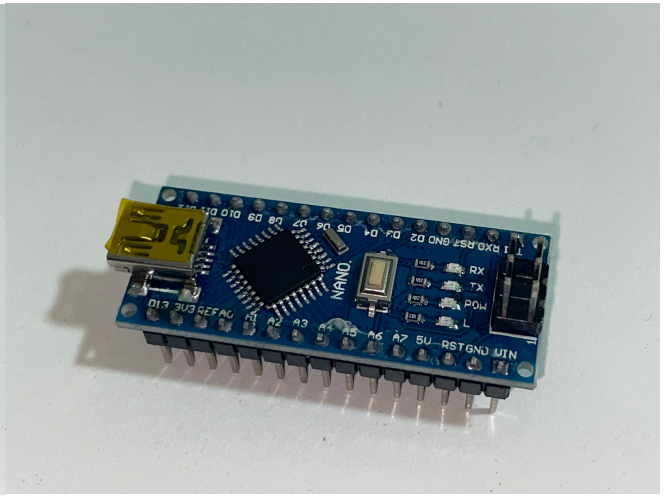
MINI ARDUINOS



PRO MINI

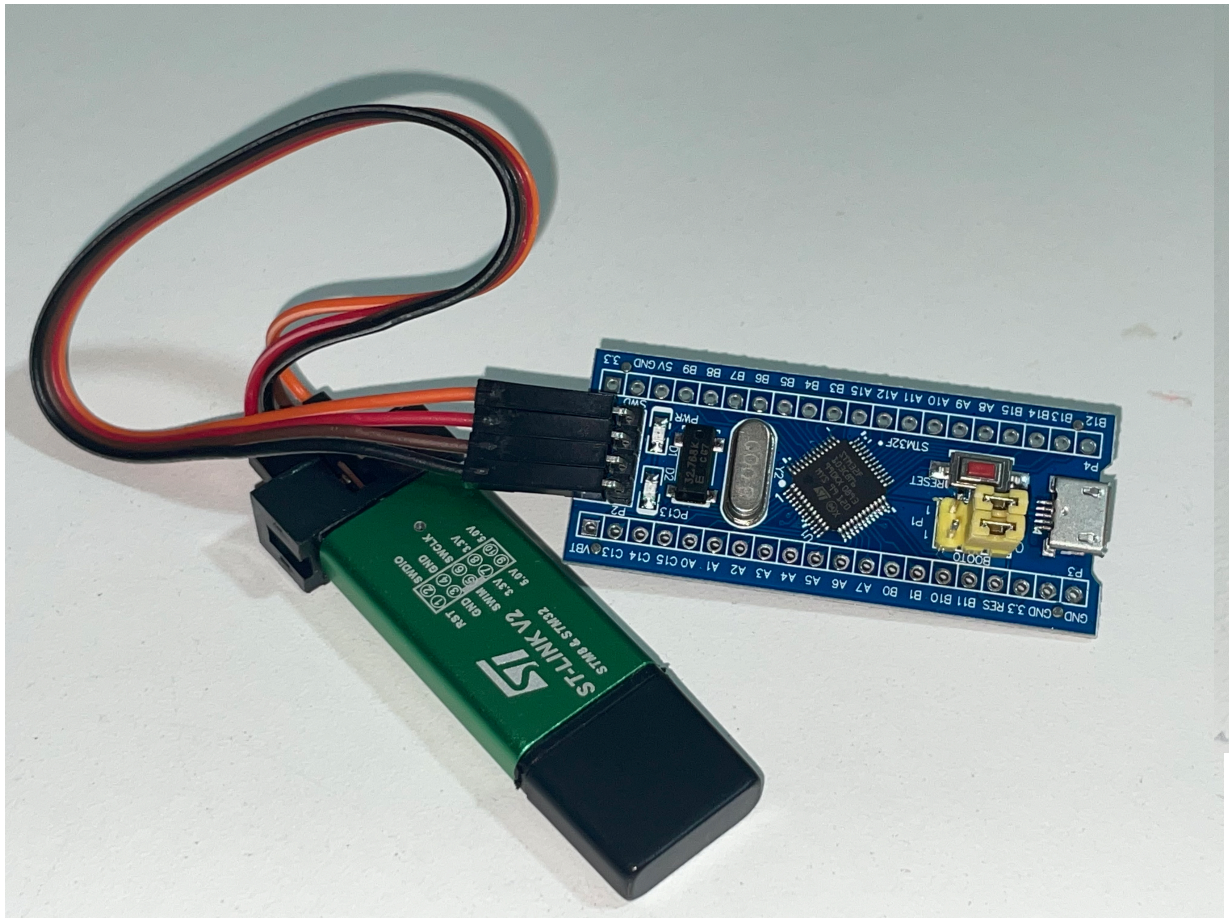


PRO MICRO WITH USB

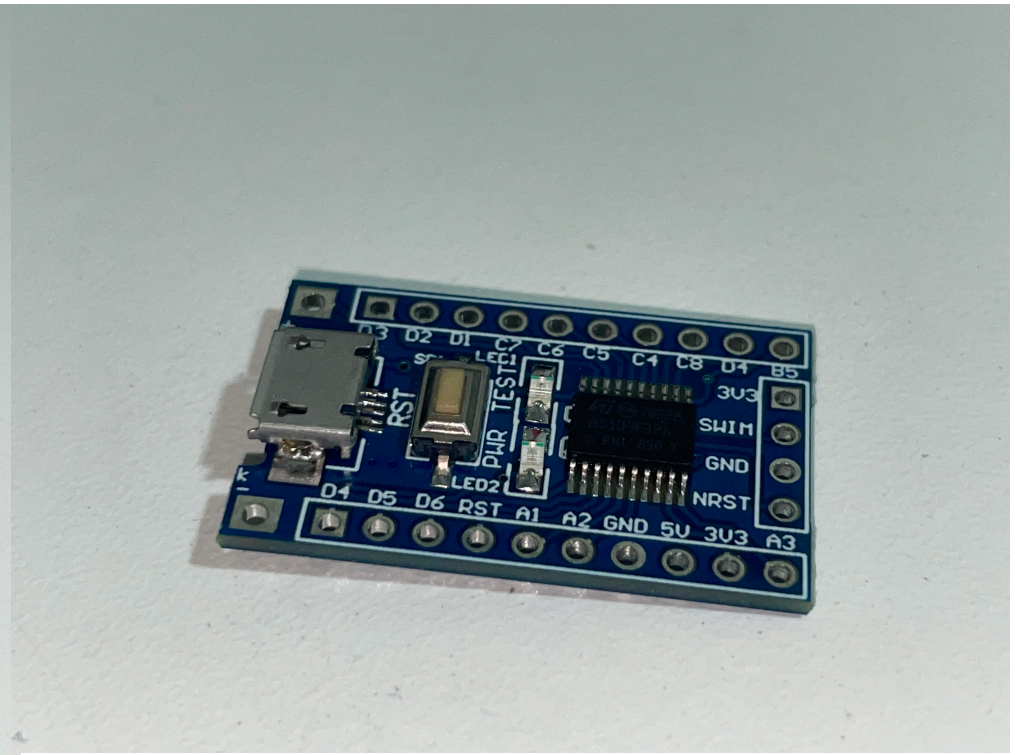


NANO WITH USB

STM 8 AND STM 32

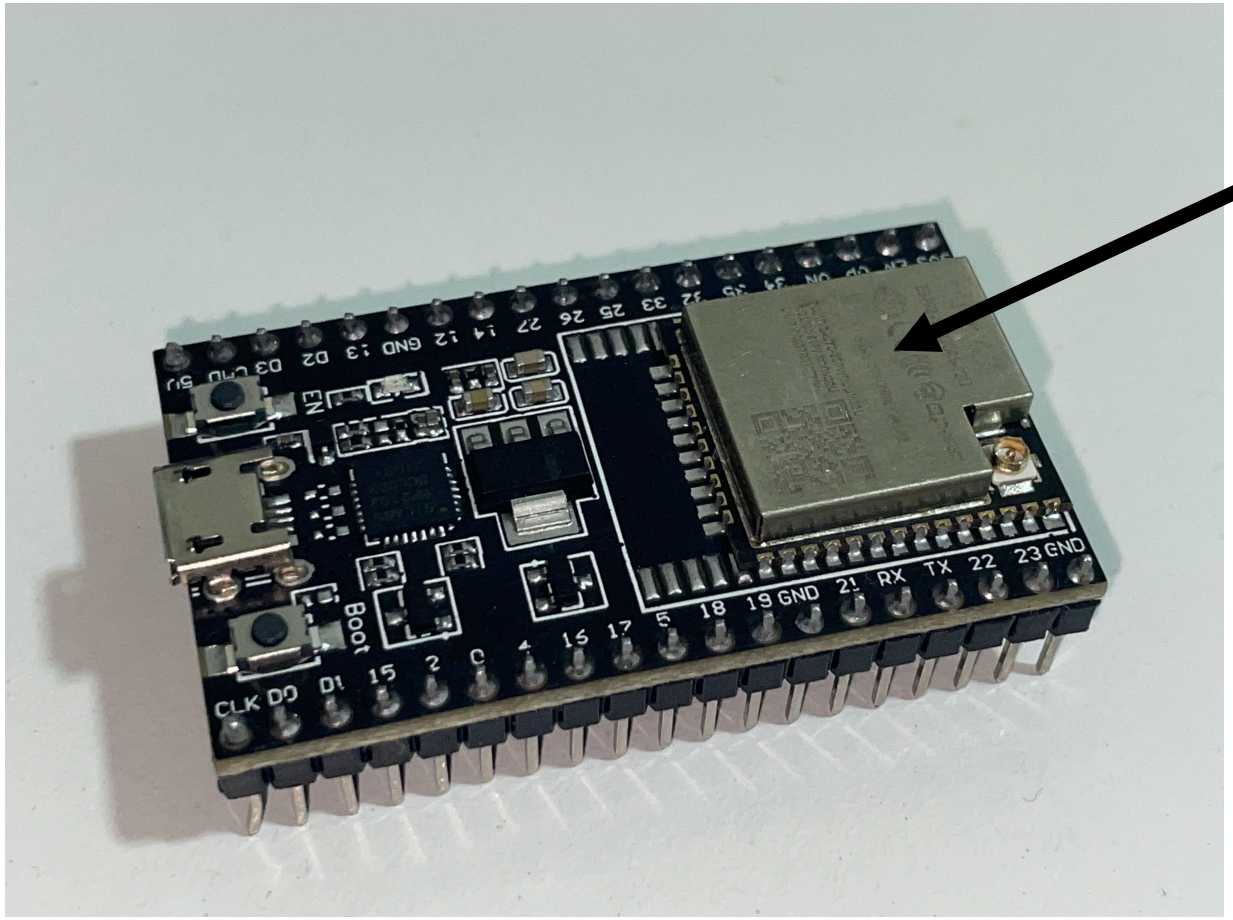


THE FAMOUS BLUE PILL



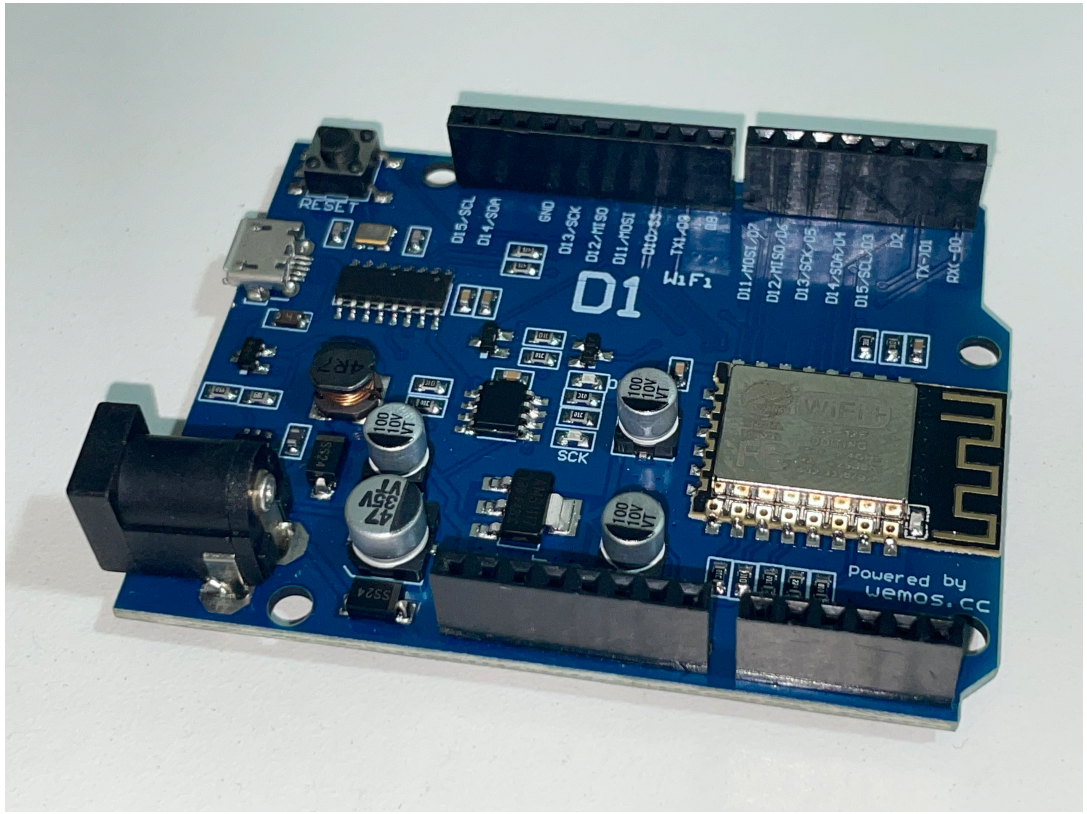
THE LOW POWERED FAVOURITE

ESP32

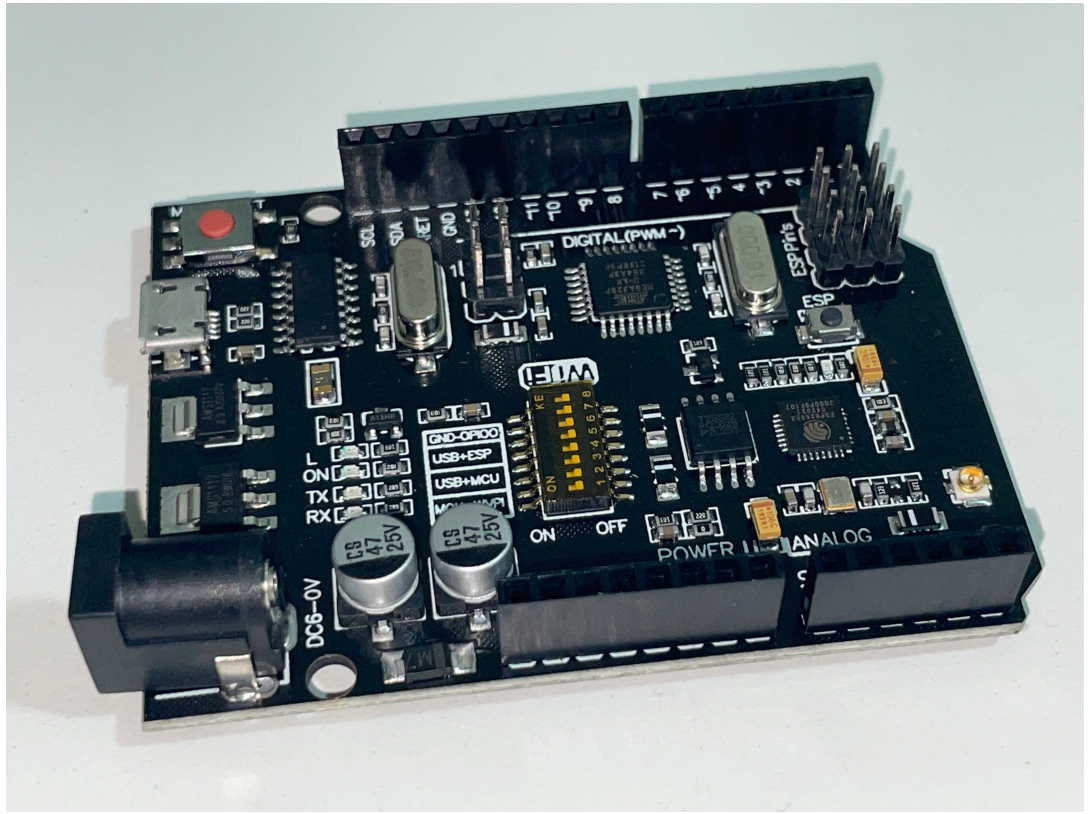


IT’S A CHIP WITH WIFI AND BLUETOOTH BUILT IN

ESP8266 AND ATMEGA + ESP32

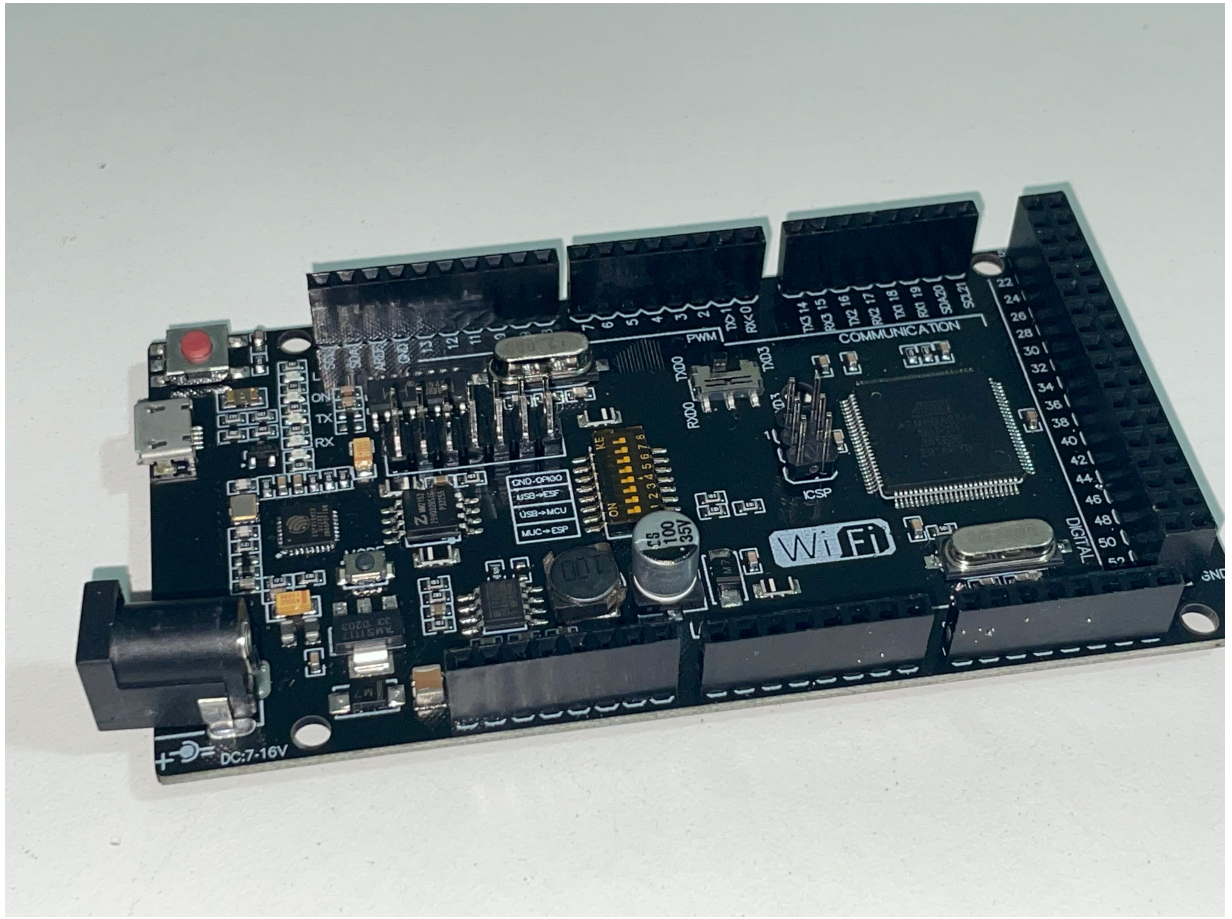


ESP8266 WITH ARDUINO UNO FORM FACTOR



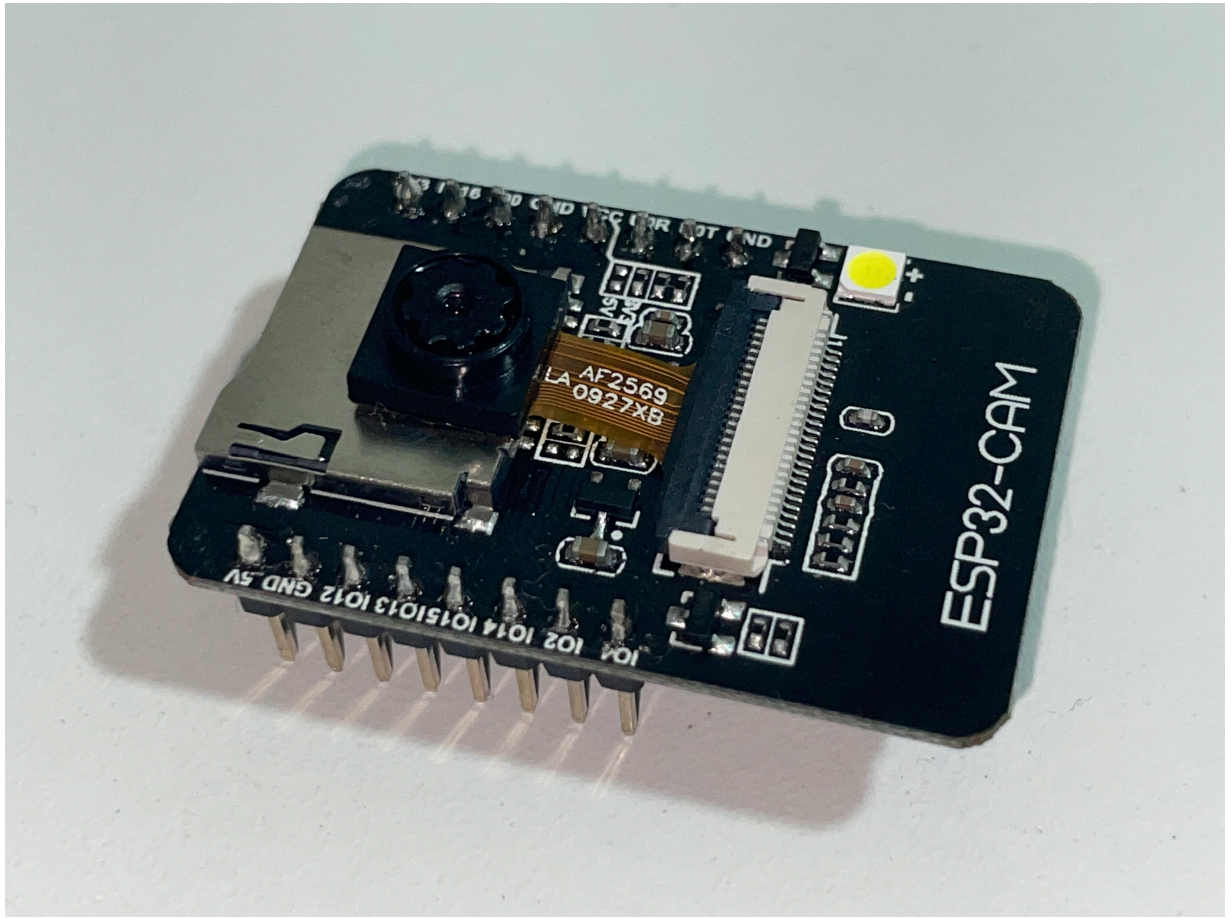
COMBO OF ATM AND ESP32 IN UNO FORM

ATMEGA WITH ESP32



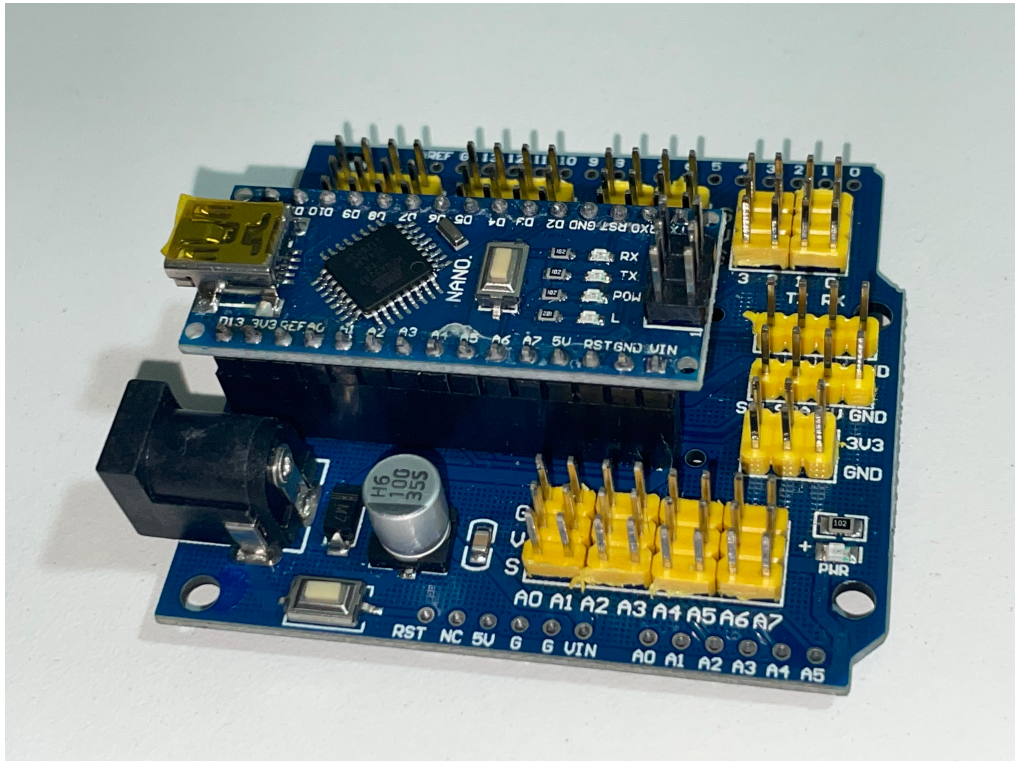
ATMEGA WITH ESP32

ESP32 WITH A CAMERA

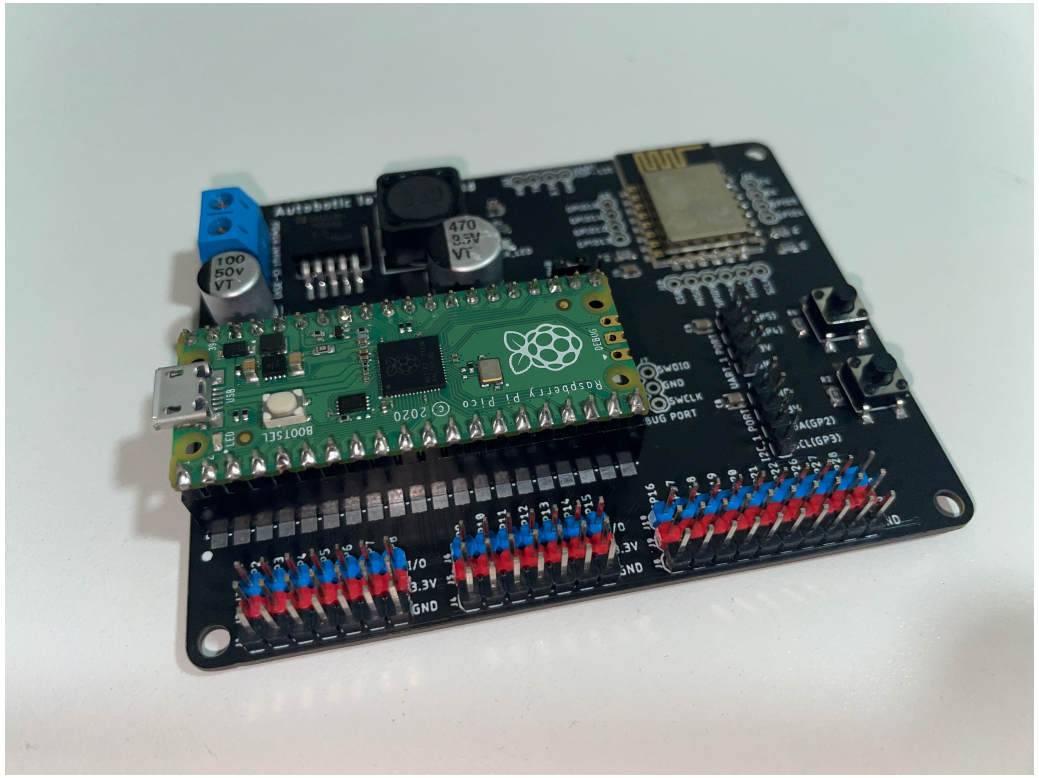


ESP32 WITH A CAMERA – WIRELESS STREAMING

EXPANSION BOARDS



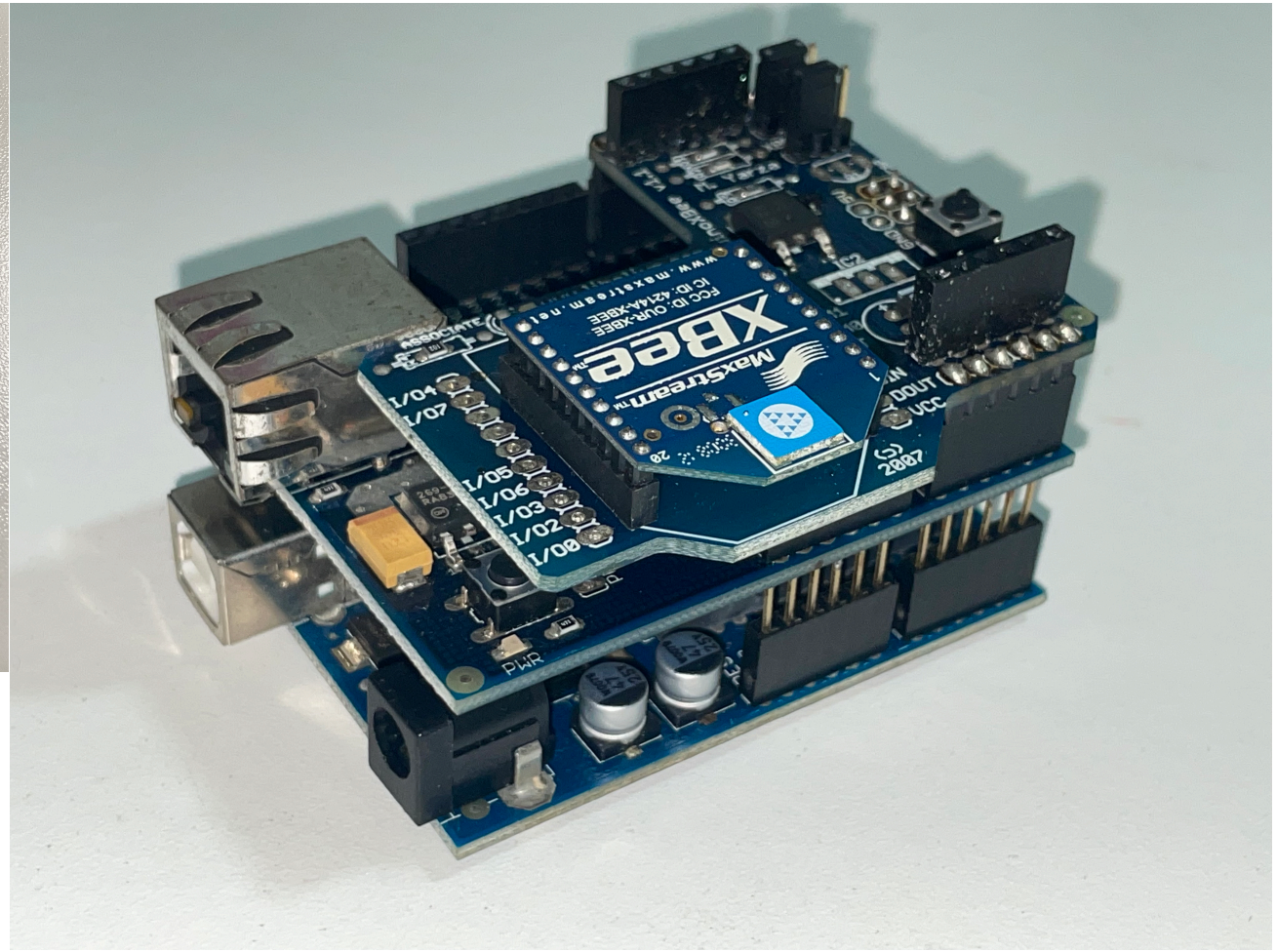
NANO EXPANSION



RP2040 EXPANSION

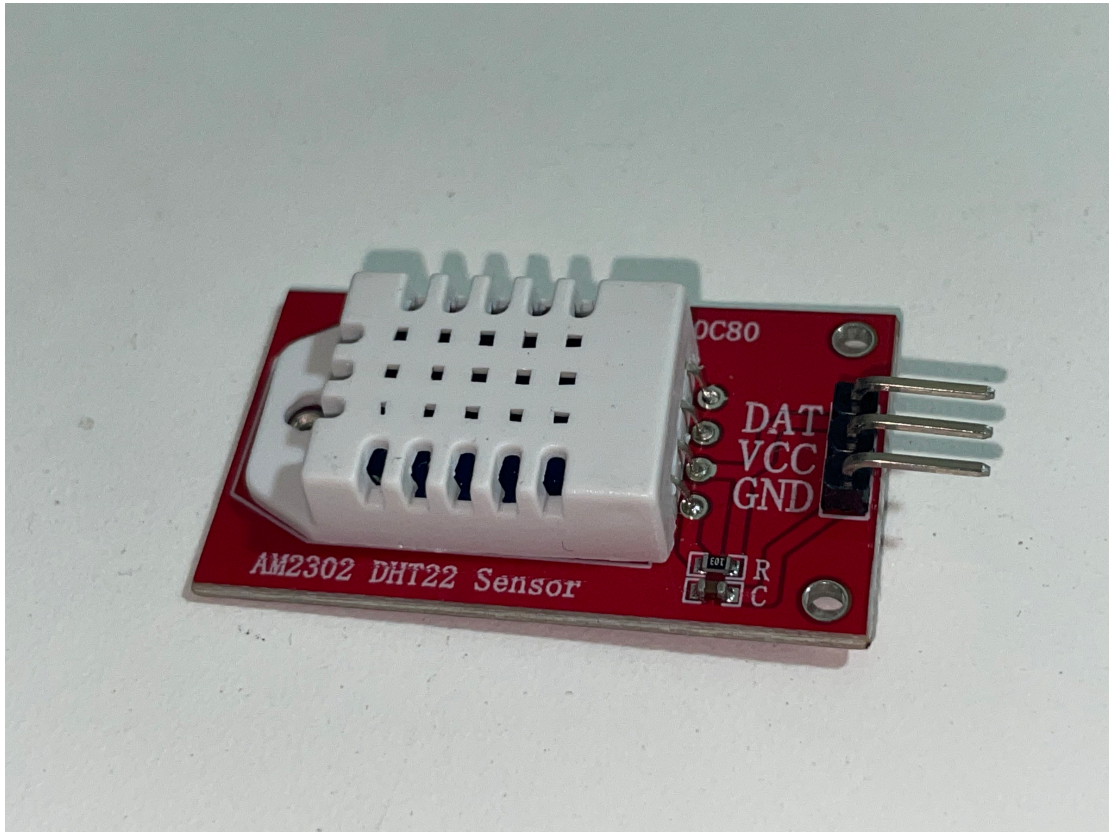
ADD ON BOARDS

ETHERNET
WIRELESS XBEE

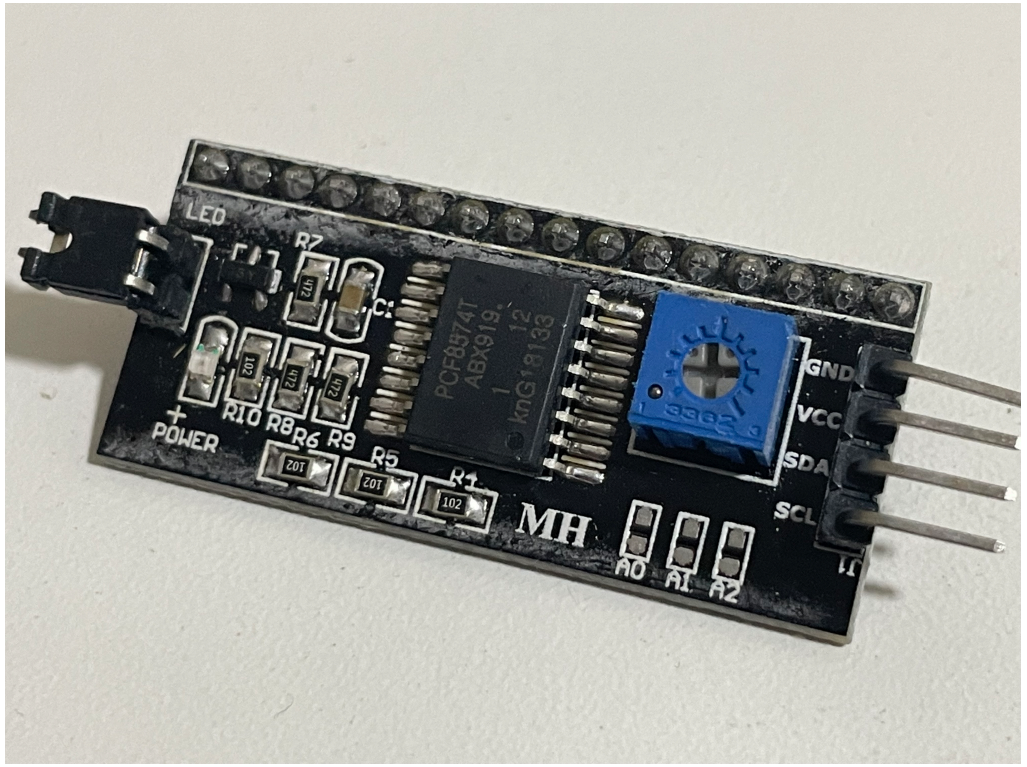


FULLY STACKED UP – NO IT DOESN'T WORK LIKE THIS ☹

SENSORS AND INPUTS

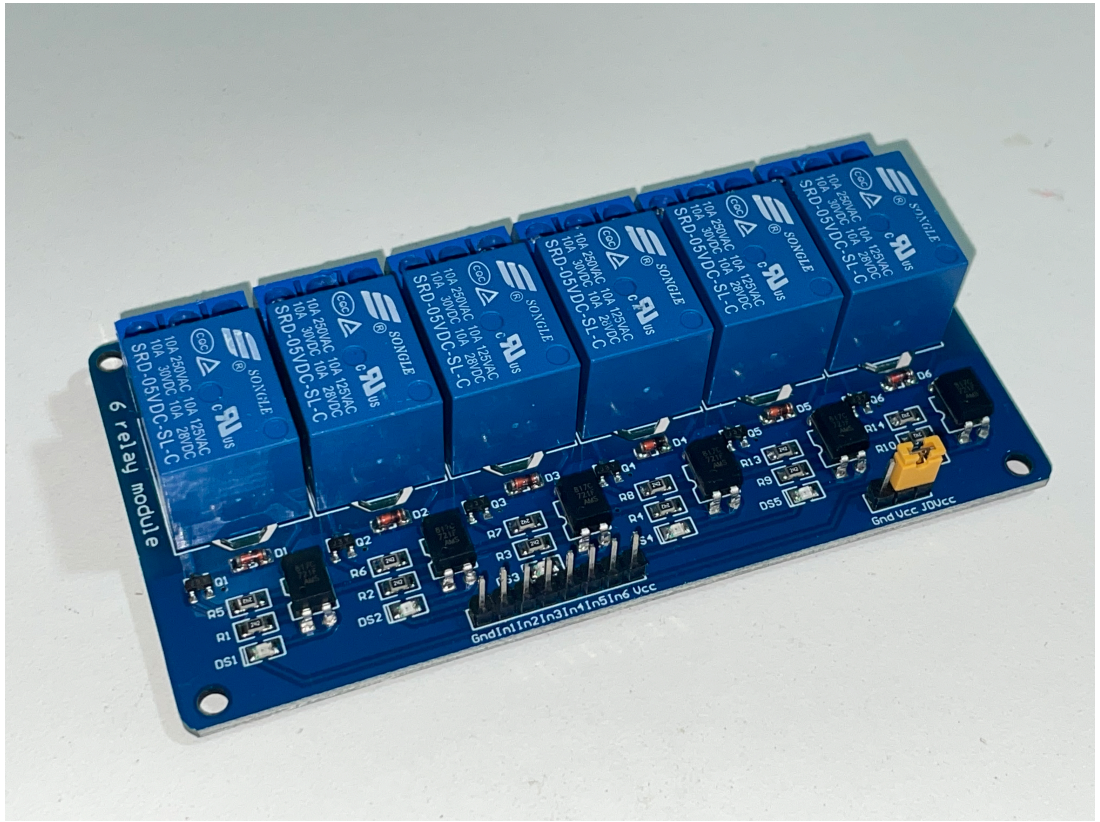


TEMPERATURE AND HUMIDITY

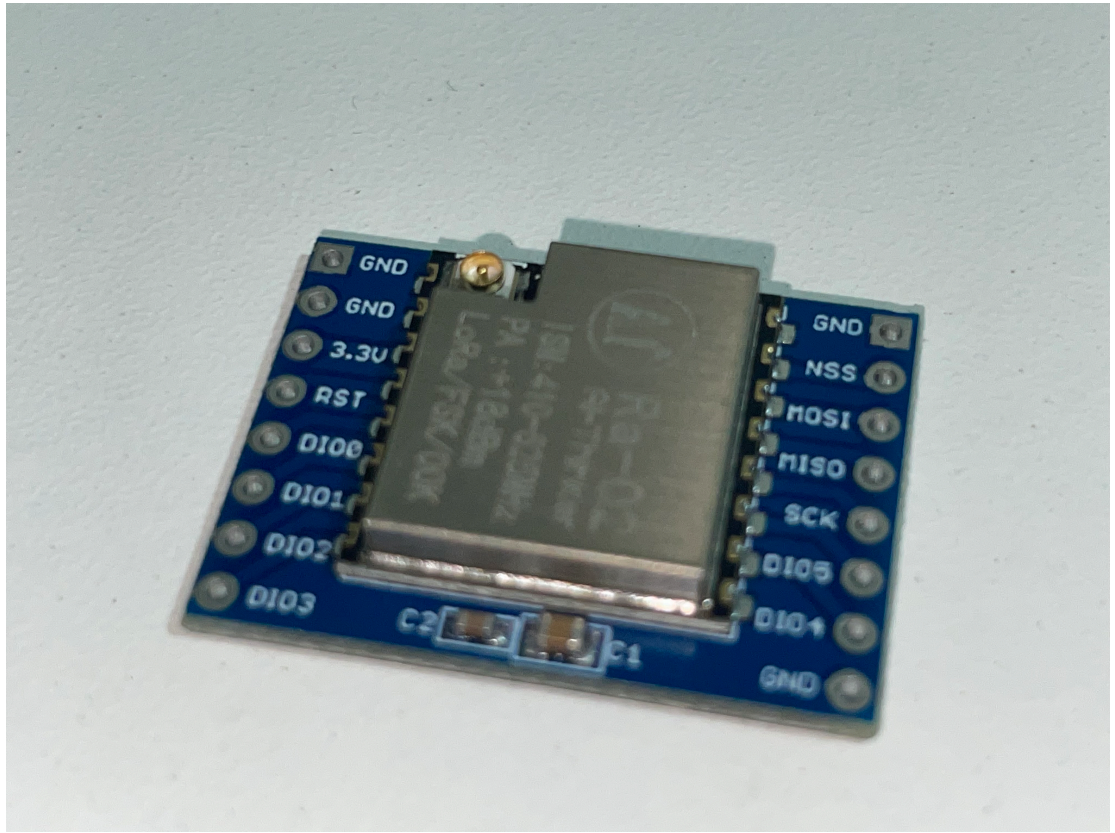


I2C CONVERTER

MODULES AND CONNECTIVITY

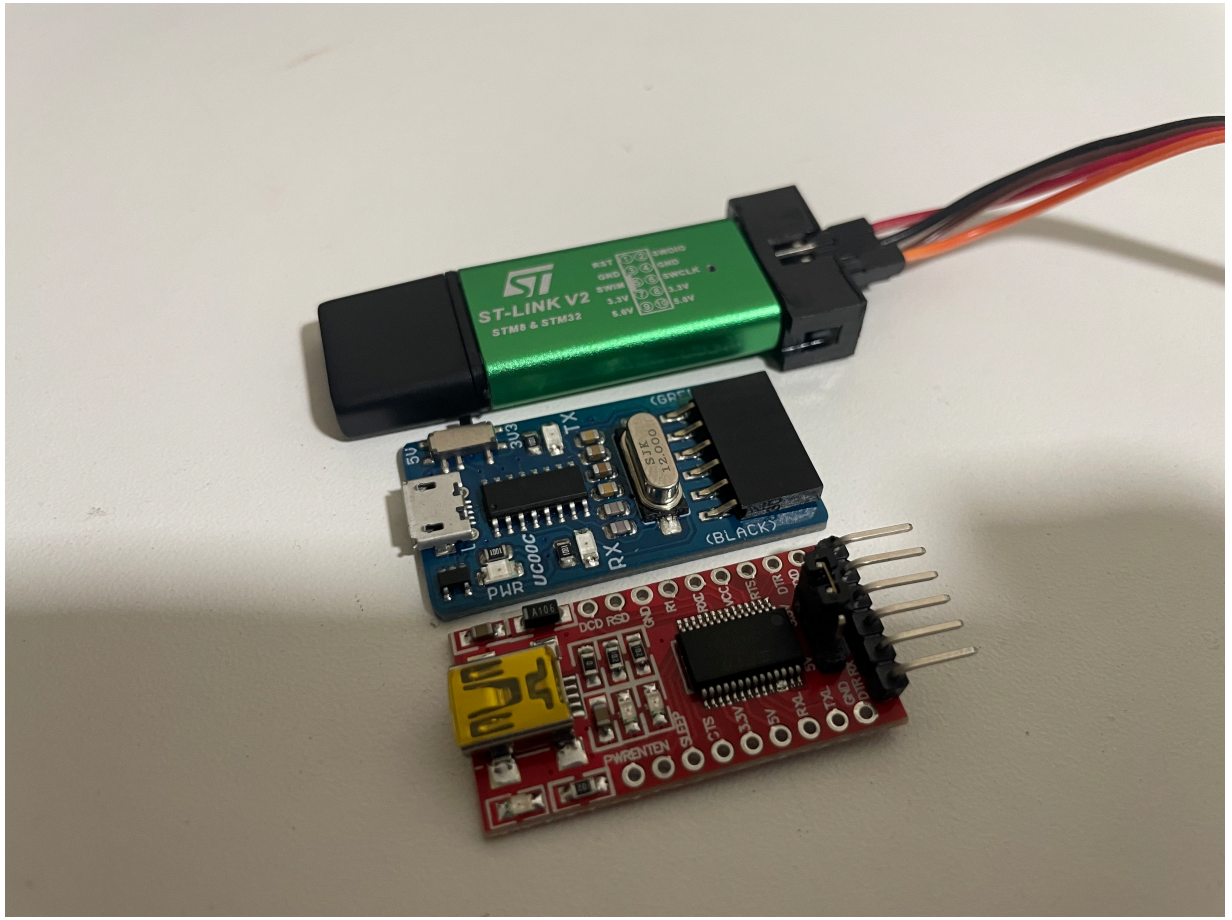


6 CHANNEL RELAY



WIRELESS COMMUNICATION

PROGRAMMERS



STM, AND FTDI ALIKE PROGRAMMERS / USB INTERFACE

IDE AND HOW TO PROGRAM

- Get a Data Cable for your PC to your Device
- Use a Native / FTDI like converter if any
- Arduino IDE – Get the 2.0 (If you need a memory hog, or choose v1 without Auto Complete)
- Add on the Board if required
- Add on the Libraries if required
- Write your Code – Test to Compile
- Flash your Compiled Code to your Device
- Arduino boards have Auto Flash, ESP32 and RP2040 requires the Boot
- Restart your device if required
- Get Serial Data out output (2-way coms over UART)
- Debug with Arduino (with some boards only)

SAMPLE CODE – BLINK.INO

```
void setup() {
pinMode(LED_BUILTIN, OUTPUT);
}

void loop() {
digitalWrite(LED_BUILTIN, HIGH);
delay(1000); // wait for a second
digitalWrite(LED_BUILTIN, LOW);
delay(1000); // wait for a second
}
```

ON RESET

- **LED PIN IS FOR DECORATION**

FOREVER

- **POUR SOME SUGA ON IT**
- **SLEEP 1 s**
- **TURN IT OFF**
- **SLEEP 1 s**

BLINK.C

```c
#include <avr/io.h>

#define F_CPU 16000000
#define BLINK_DELAY_MS 5000

#include <util/delay.h>

int main (void)
{
// Arduino digital pin 13 (pin 5 of PORTB) for output
DDRB |= 0B100000; // PORTB5

while(1) {
    // turn LED on
    PORTB |= 0B100000; // PORTB5
    _delay_ms(BLINK_DELAY_MS);

    // turn LED off
    PORTB &= ~ 0B100000; // PORTB5
    _delay_ms(BLINK_DELAY_MS);
    }
}
```

COMPILE AND RUN

- AVRDUDE – your tool for Arduino
- IDE

- Log

```
avrdude: AVR device initialized and ready to accept instructions

Reading | ################################################## | 100% 0.00s

avrdude: Device signature = 0x1e950f (probably m328p)
avrdude: reading input file "/private/var/folders/7q/w6mdcsyx6_z71xzq1hqd4lk4000
avrdude: writing flash (748 bytes):

Writing | ################################################## | 100% 0.24s

avrdude: 748 bytes of flash written
avrdude: verifying flash memory against /private/var/folders/7q/w6mdcsyx6_z71xzq
avrdude: load data flash data from input file /private/var/folders/7q/w6mdcsyx6_
avrdude: input file /private/var/folders/7q/w6mdcsyx6_z71xzq1hqd4lk40000gn/T/arc
avrdude: reading on-chip flash data:

Reading | ################################################## | 100% 0.17s

avrdude: verifying ...
avrdude: 748 bytes of flash verified

avrdude done.  Thank you.
```

SOURCE CODES

- FILE NAME . INO
- The files are all in 1 Folder, The project is the folder
- The libraries needed to be added manually sometimes
- The INO files (sometimes other names) are converted to C
- Check the FLASH remaining memory always
- Turn on Verbose Output – warnings can also cause issues sometimes, but mostly ignored by noobs like me.
- Libraries have sometimes the same name – be careful what file you import

WHY ARDUINO

- 1 IDE and Source Code Format / Core Libraries for all the supporting boards, SAVES TIME!!!
- Supports Custom features for certain boards
- Encapsulation & Portability – where you can Write for ATM but port to ESP32 or ESP32
- Lots of Libraries to support almost every common peripheral
- Easily obtainable and searchable with active community
- Open Sourced, Easy for Beginers.

DEMO

Blink Code on Arduino IDE
FLASHED to a Pro Mini 3v3