ARDUINO CRASHCOURSE – ZERO TO HERO!

WHO AM I?

- Jacob Bechmann Pedersen
  - Electronics Engineer (AU, 2019)
  - Started using Arduino in 2014
  - Volunteer in Coding Pirates 2016-2018
  - Teaching at MakerCamp since2018
    - 12-16 y/o"Inventors"
  - Taught Arduino for IDA, StudyNow, Herning Municipality, etc.
  - Full-time embedded electronicsengineer at DTU
    - Robots and autonomous systems

PURPOSE OF THIS WORKSHOP

- Give you a QUICK understanding of the basics and essentials of Arduino
  - Hands-on successes
  - Experience some difficulties
- Inspiration on how to use Arduino for yourself
- Try some IoT capabilities of ESP32 on the Arduino platform

BEFORE WE GET STARTED

- The code and examples throughout the workshop are available on:
  - https://github.com/iakop/ArduinoCrashcourseIoT
  - It's a good idea to keep this site open throughout

AGENDA

- 16:00-17:30 : Arduino Basics
- 17:30-18:00 : Setup of ESP32 on Arduino
- 18:00-18:30 : Dinner - Pizza and softdrinks
- 18:30-19:45 : IoT on ESP32
- 19:45-20:00 : Summary, evaluation, and thanks for now!


ARDUINO BASICS

WHAT IS ARDUINO?

- Most used platform for MCU prototyping
  - MCU is short for Micro Controller Unit
  - A tiny computersystem, controlling electronics
- Simple, intuitive programming
- Many code examples available
  - Many software libraries too!

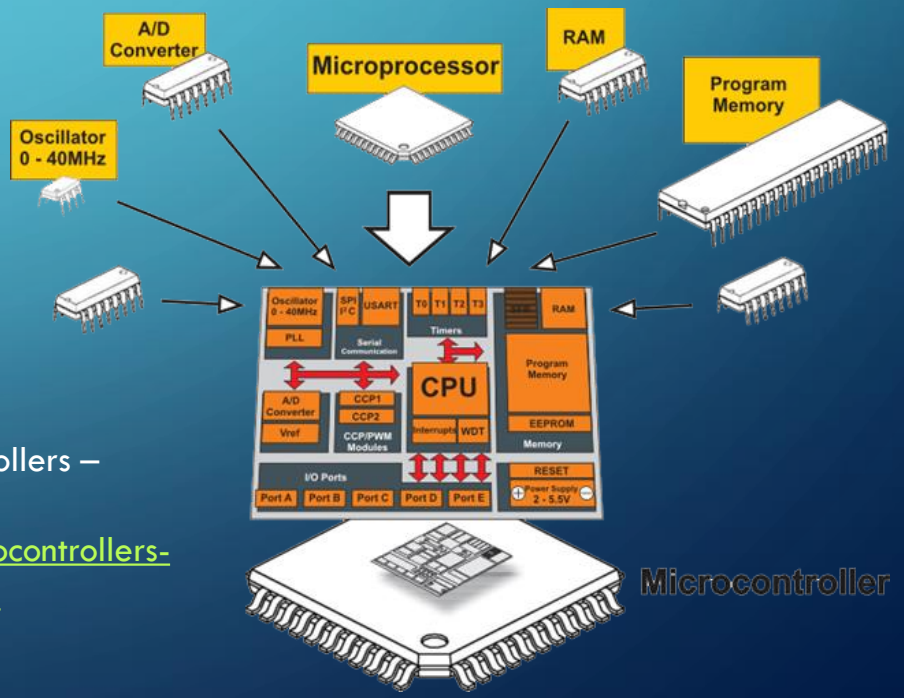
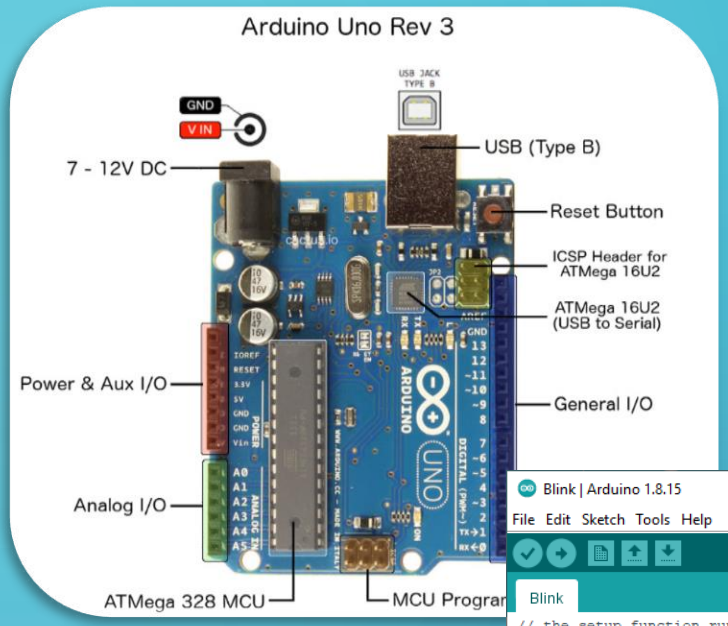


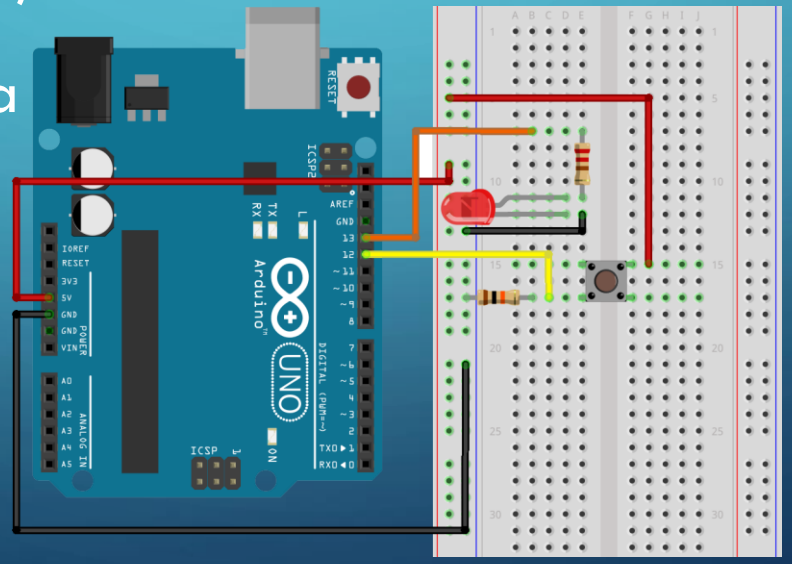
Figure: Introduction to the World of MicroControllers – Mikroelektronika
URL: https://www.mikroe.com/ebooks/pic-microcontrollers-programming-in-c/introduction-to-the-world-of-microcontrollers

THE ARDUINO PLATFORM

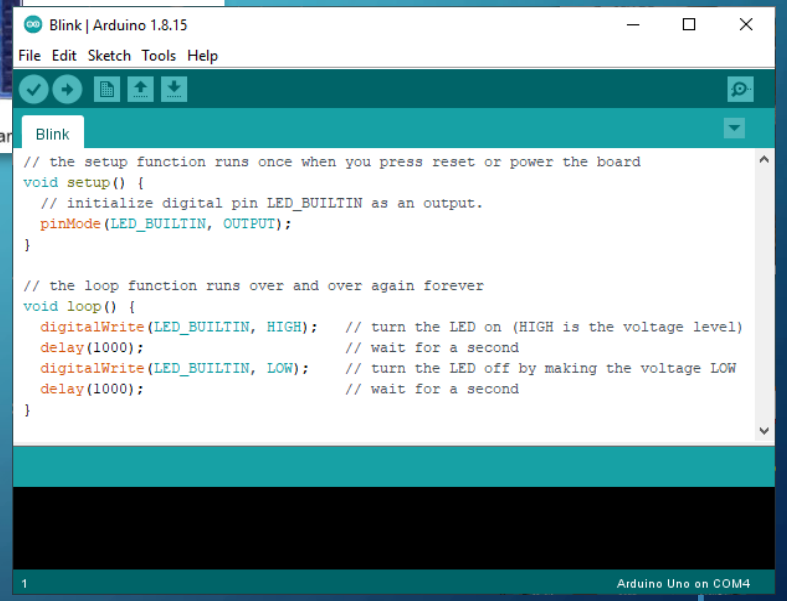
- Arduino connects to electronics through its ”pins”
- Pin signals are controlled through programs (turn on/off, etc.)
- Often connected through a ”breadboard”



Arduino og dens pins



Arduino forbundet til elektronik gennem breadboard



Det officielle Blink program i Arduino IDE

HVORFOR ARDUINO?

- Compare the following two programs
  - Which makes the most sense upon reading?

```
//Physical LED pin
#define LEDWRITEPORT PORTB
#define LEDPIN 5
//Physical button pin
#define BTNREADPORT PIND
#define BTNPIN 3

int main() {

  //Initialize pins
  DDRB |= (1 << LEDPIN); //ledPin set to output
  DDRD &= ~(1 << BTNPIN); //btnPin set to input

  while(1){
    //If button is pressed LED on
    if((PIND & (1 << BTNPIN)) >> BTNPIN == 1){
      PORTB |= (1 << LEDPIN);
    }
    //Else, LED off
    else{
      PORTB &= ~(1 << LEDPIN);
    }
  }
}
```

```
const int ledPin = 13; //Physical LED pin
const int btnPin = 3; //Physical button pin

void setup() {
  //Intialize pins
  pinMode(ledPin, OUTPUT); //ledPin set to output
  pinMode(btnPin, INPUT); //btnPin set to input
}

void loop() {
  //If button is pressed LED on
  if(digitalRead(btnPin) == HIGH){
    digitalWrite(ledPin, HIGH);
  }
  //Else, LED off
  else{
    digitalWrite(ledPin, LOW);
  }
}
```

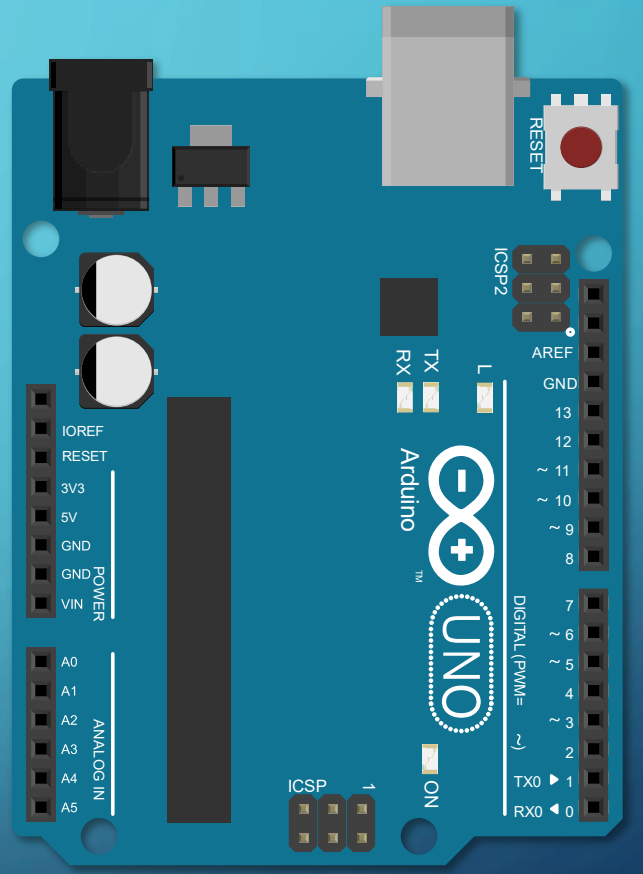
GETTING STARTED:

- Download and install Arduino IDE
- Get the version for you platform
  - For your own sake, do NOT get the Microsoft Store version!!!
- Examples from this workshop are on Github: https://github.com/iakop/ArduinoCrashcourseIoT

https://www.arduino.cc/en/software

EXAMPLE 0: BLINK

- To test if the Arduino works
- We'll compile and upload the program located in the IDE:
  "File > Examples > 01.Basics > Blink"

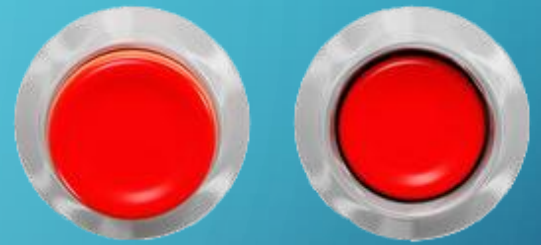


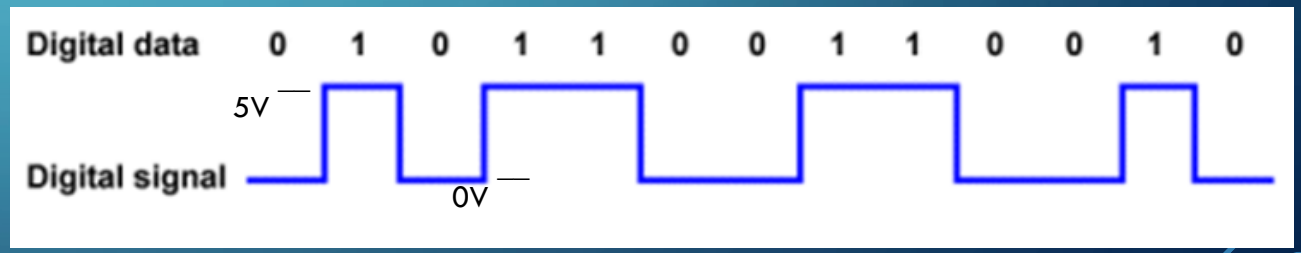
ABOUT DIGITAL COMPONENTS

- Blink example used a built-in LED
  - As a digital output
- Digitale outputs are for example:
  - LED's
  - Motors
  - Relays
- Digital when they are solely operating as on/off
- Digitale inputs are f.eks.
  - Buttons
  - PIR sensors
  - Tilt sensors

DIGITAL ON ARDUINO

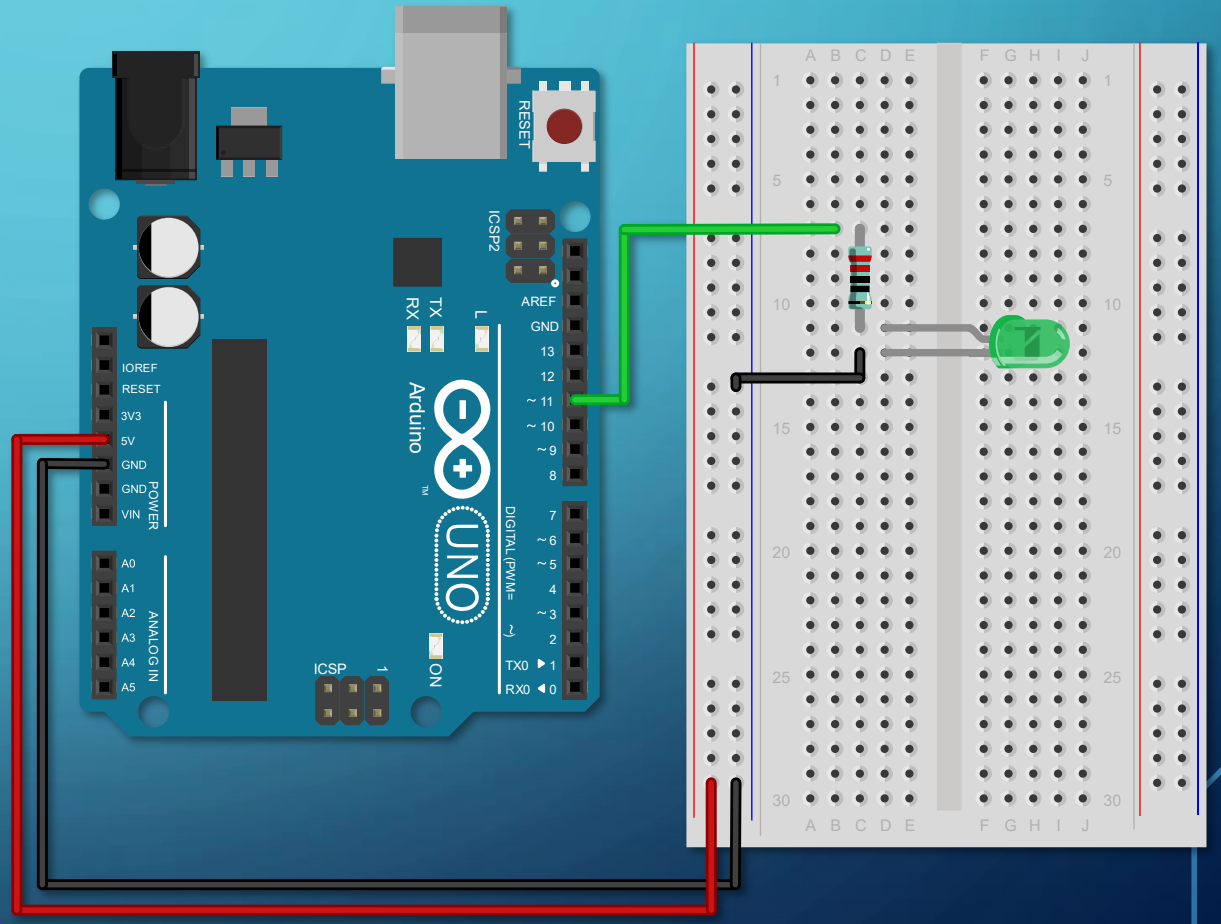
- Digital data are ”boolean” values;
  - Either ”True” or ”False”
  - Examples:
    - ”Is the door open”
    - ”Is the button pushed”
- Arduino gets the data from real voltages:
  - 5V for True
  - 0V for False
- Represented in code as HIGH or LOW





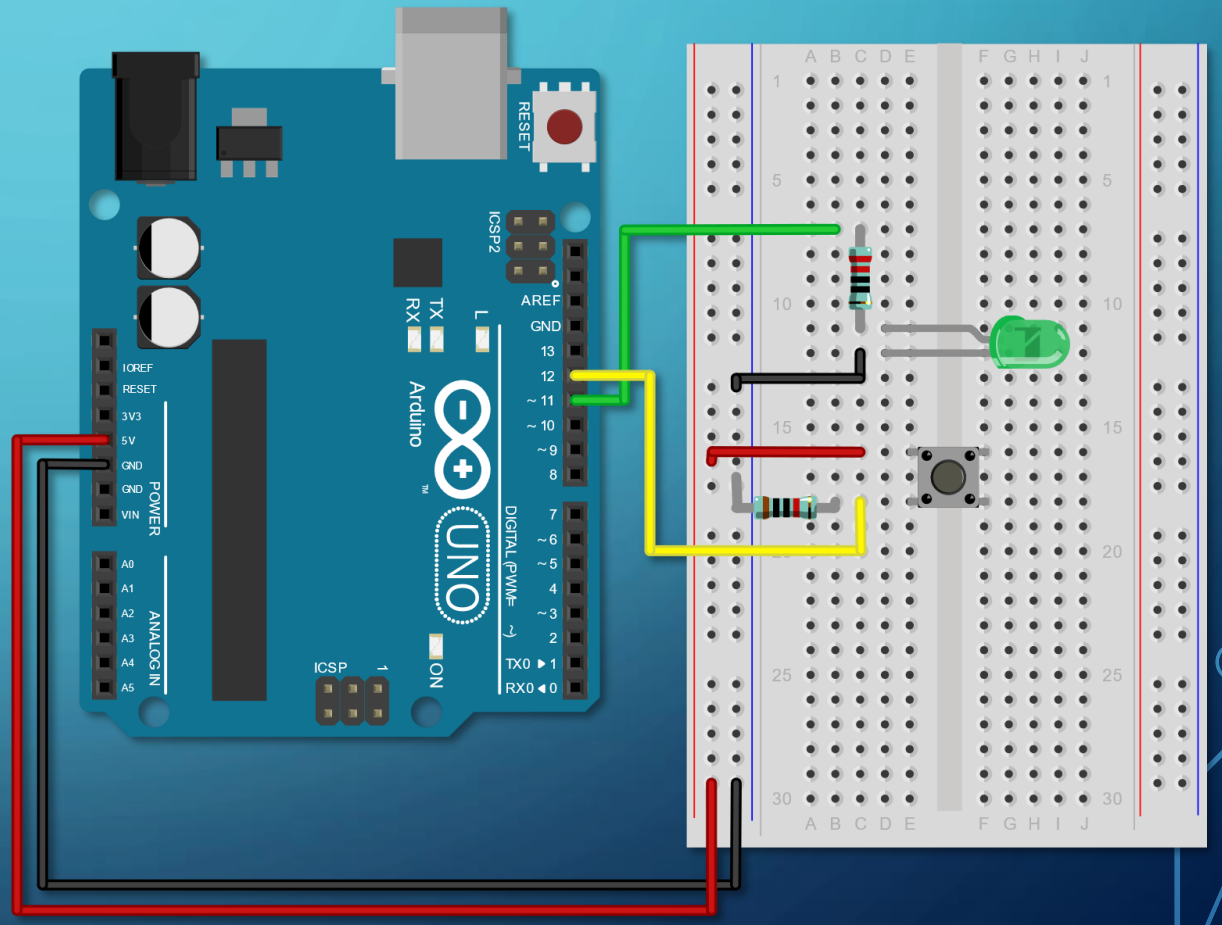
EXAMPLE 1: BLINKY

- We'll make our own Blink, "Blinky"
- Done with external circuit
- And writing the code ourselves



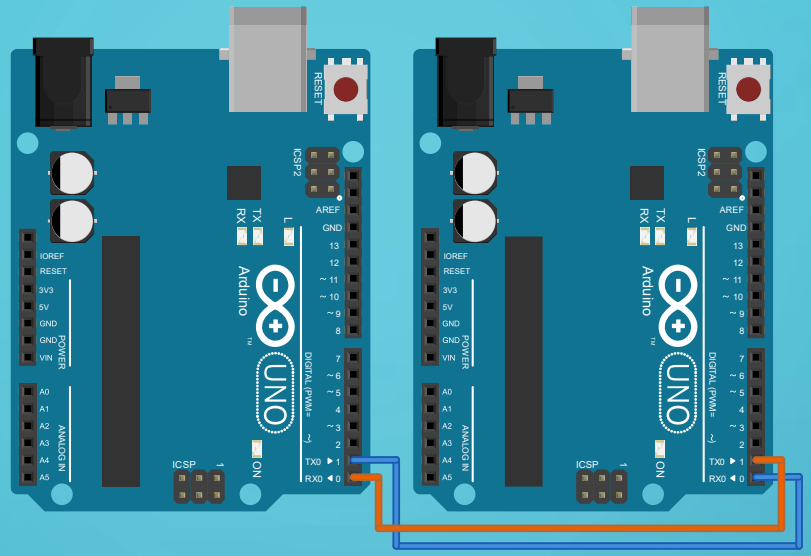
EXAMPLE 2: BUTTON DEBOUNCE

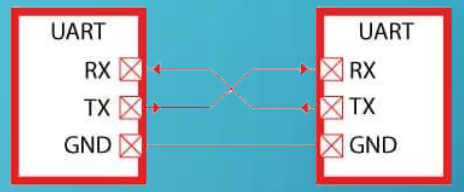
- We attach a button to the Blinky example
- Will make the LED toggle on or off
- Needs som mechanical phenomena to be handled in code

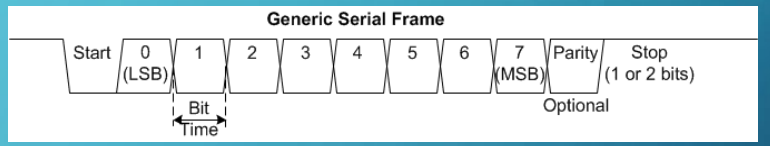


SERIAL - UART

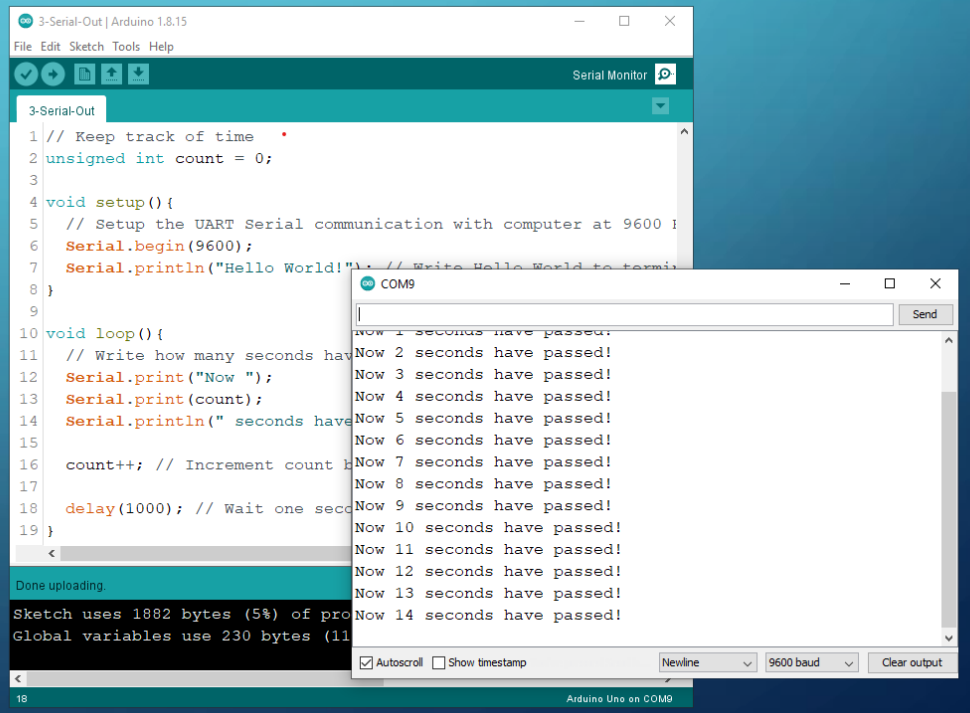
- Serial is communication through a ”series” of characters
  - Opposite to parallel
- Arduino can talk to the computer through Serial
  - Specifically UART
    - Universal Asynchronous Receiver/Transmitter
- Data sent through TX/RX pins through converter to USB





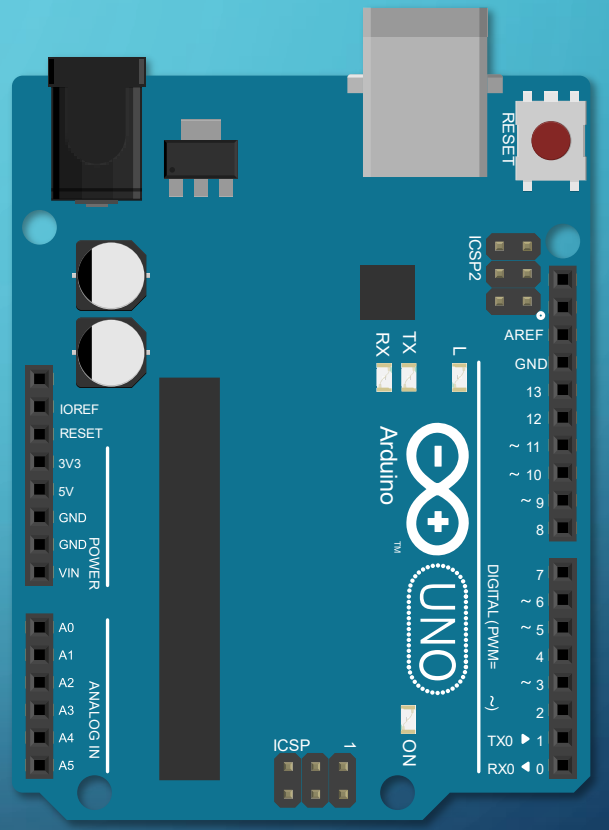


Billede: UART forbindelse og dataframe
URL: https://microcontrollerslab.com/uart-communication-working-applications/



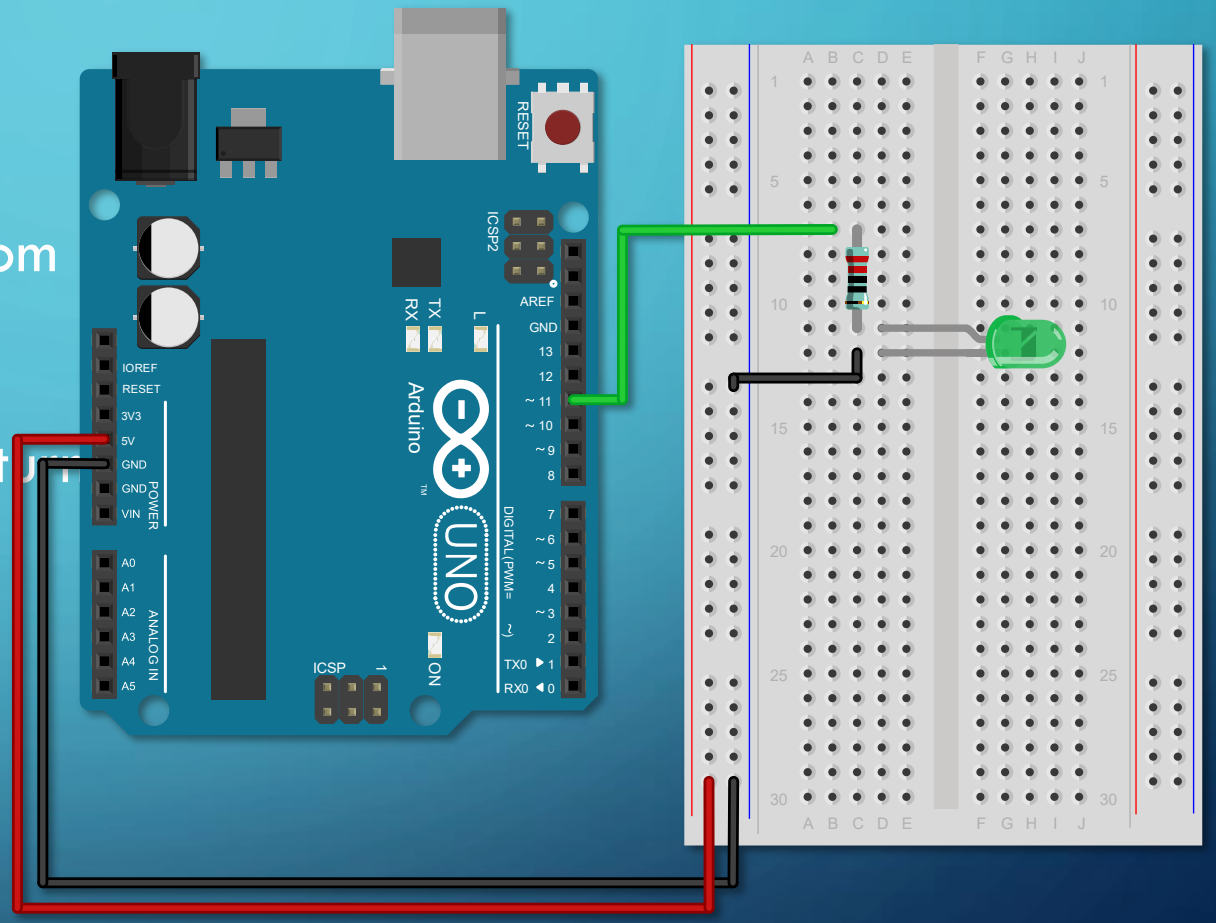
EXAMPLE 3: SERIAL OUT

- Arduino will write a message to the computer
- Serial Monitor can be found in ”Tools>Serial Monitor”



EXAMPLE 4: SERIAL IN

- Arduino will receive a message from the computer
- In Serial Monitor we'll be able to turn on/off an LED

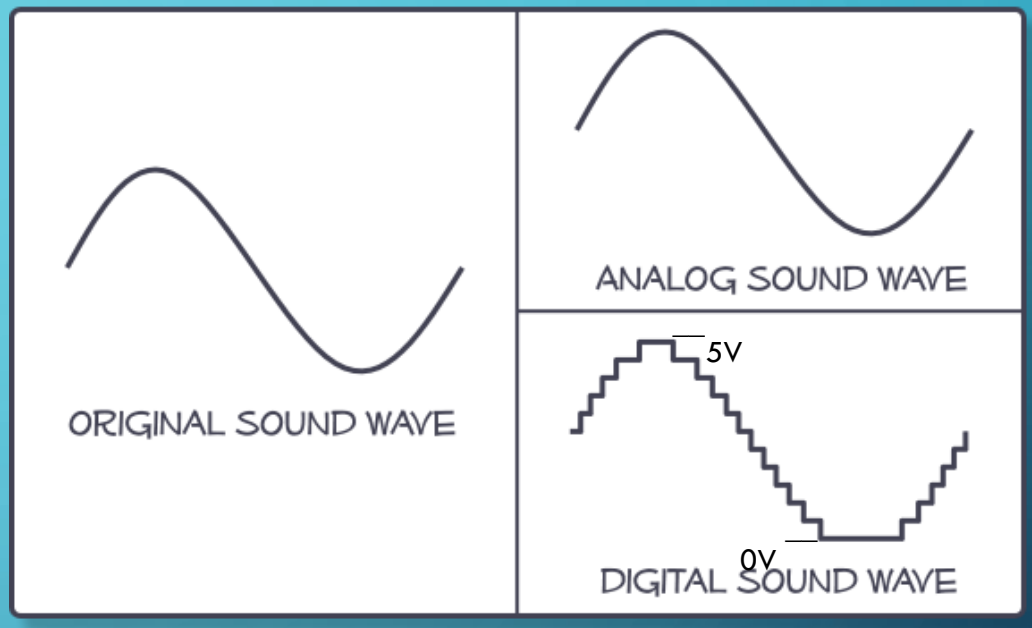


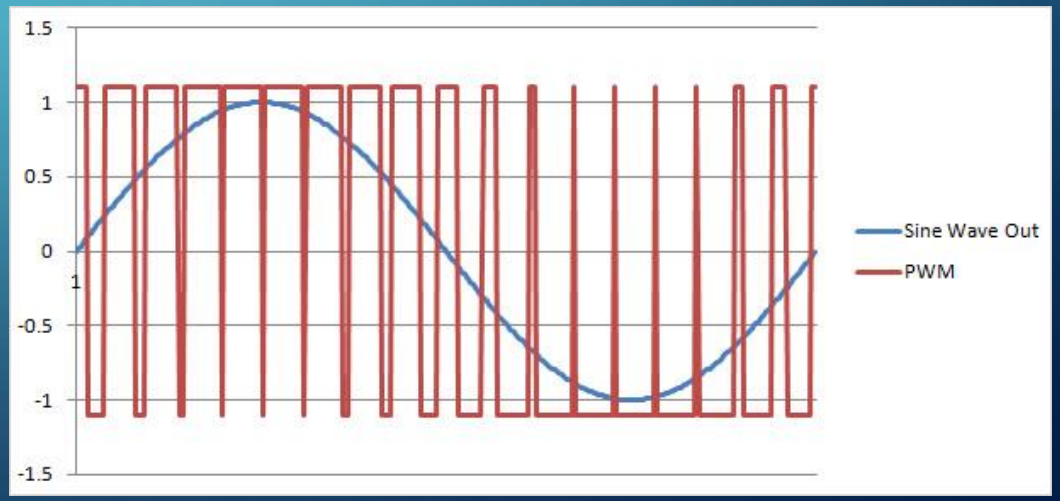
ABOUT ANALOG COMPONENTS

- Analog outputs and inputs are not just True or False
  - Values ”in-between”
- Analoge outputs are for example:
  - LED (Variable brightness)
  - Motor (Variabel speed)
  - Servomotor
- Analoge inputs er f.eks.:
  - Potentiometer
  - Joystick
  - Light Dependent Resistor

ANALOG ON ARDUINO

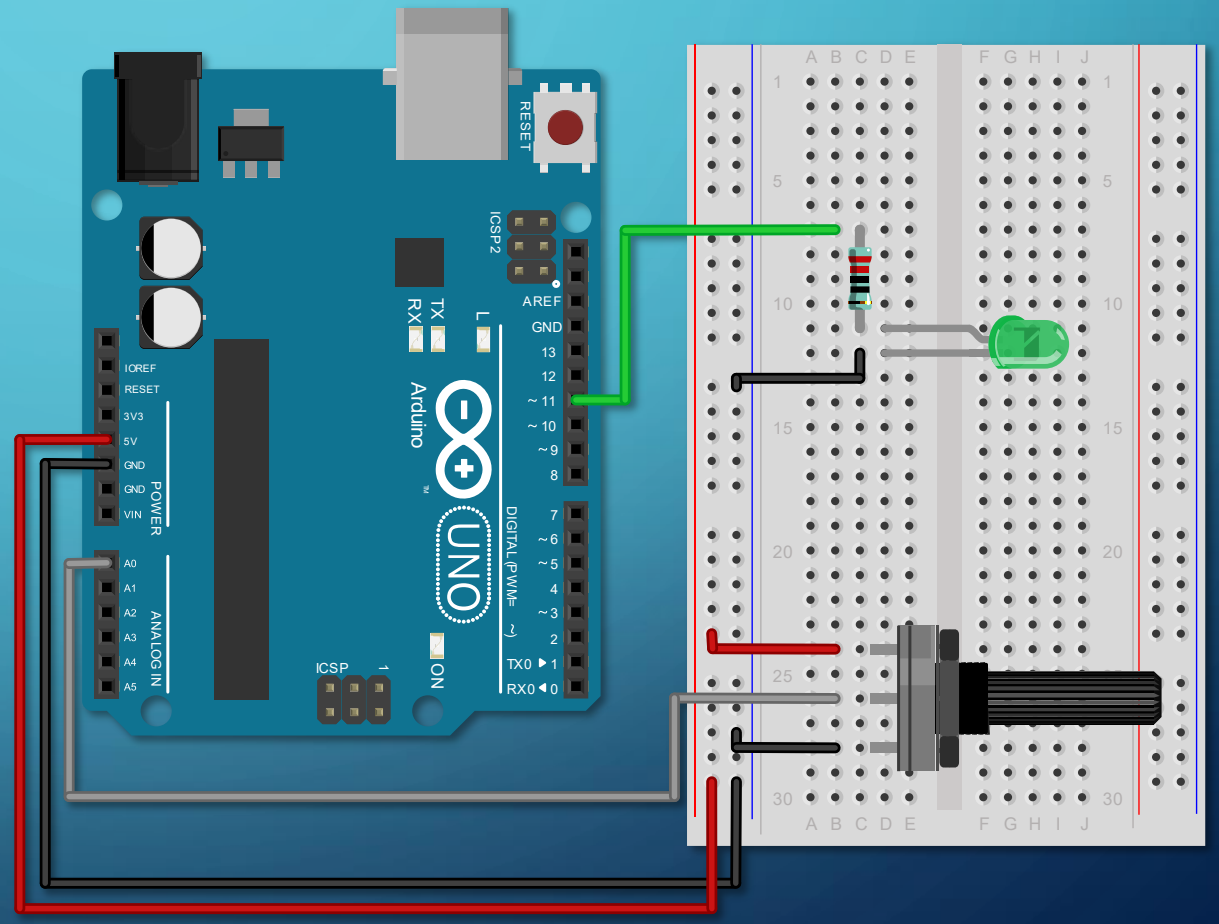
- Arduino can also measure analog data
  - Done on the analog input pins
- It can produce a ”semi” analog signal
  - Done via PWM (Pulse Width Modulation)





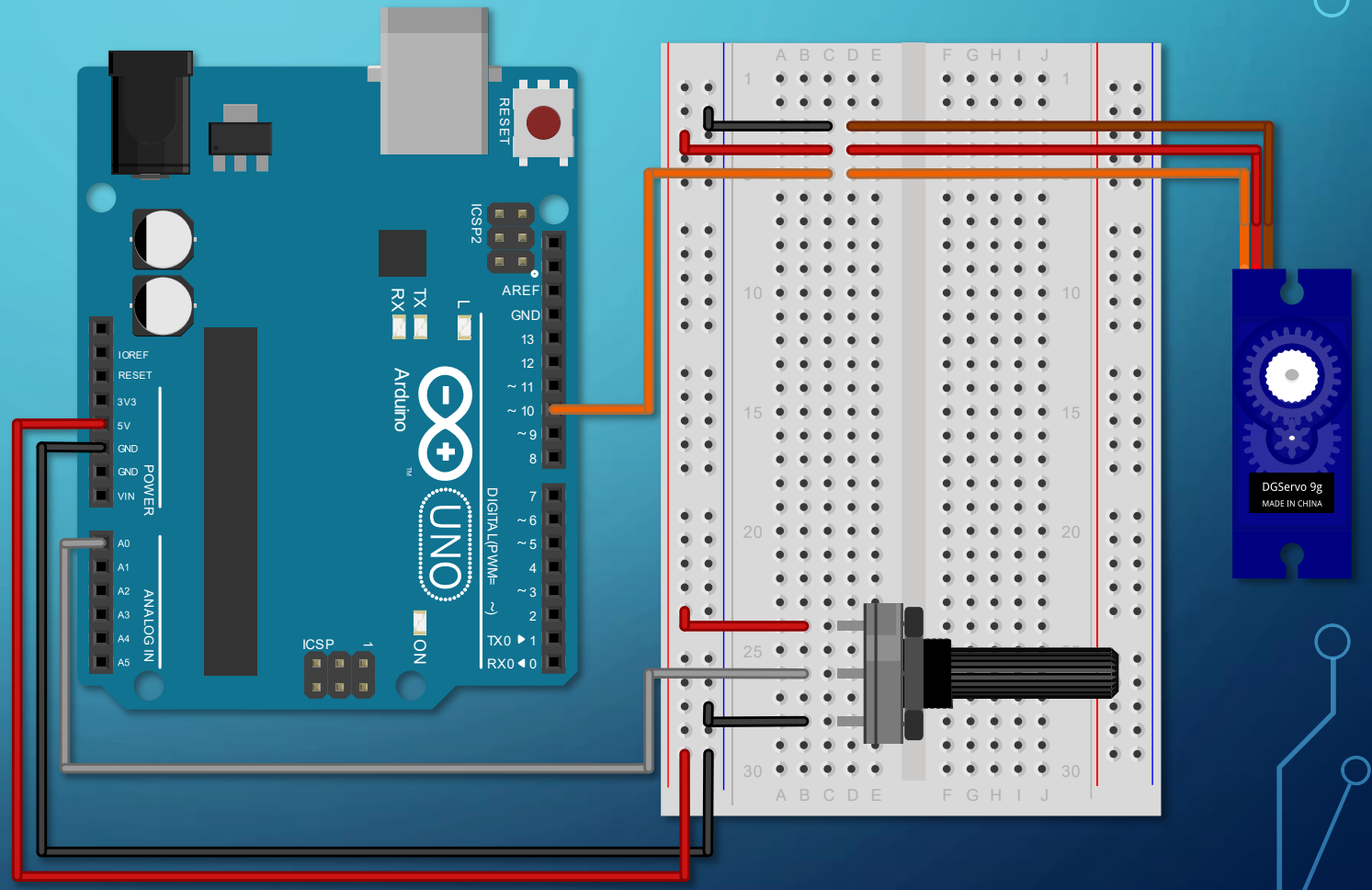
EXAMPLE 5: ANALOG LED

- We'll read the value of a potentiometer
- Use this data to dim an LED
- Status is written over Serial



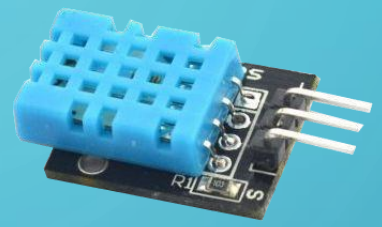
EXAMPLE 6: SERVO

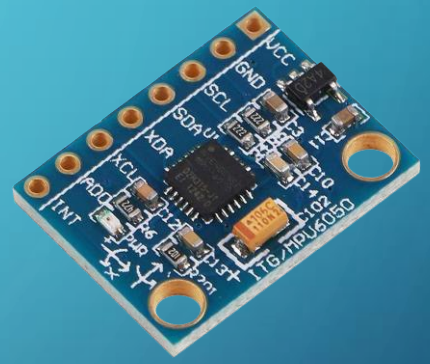
- We'll include a "library" to control the Servo motor
- Controlled from potentiometer
- Status written over Serial

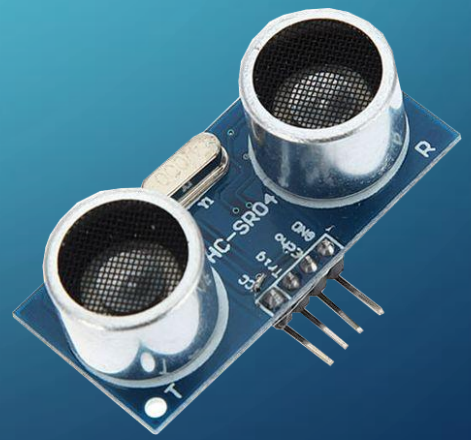


OTHER COMPONENTS - SENSORS

- Temperature-/humidity sensor
- Accelerometer/gyroscope
- Ultrasound distance sensor
- Often communicate through protocols protokoller
  - Handled easily through libraries





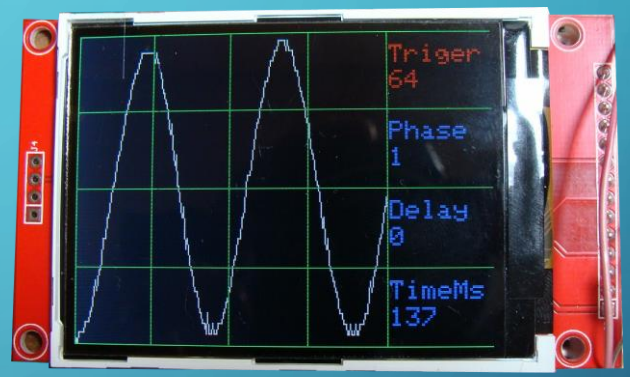


OTHER COMPONENTS - ACTUATORS

- Stepper Motors
- Speakers
- Linear actuators
- Pistons
- Most use no, or simple, protocols
  - Steppers and speakers have libraries

OTHER COMPONENTS - DISPLAYS

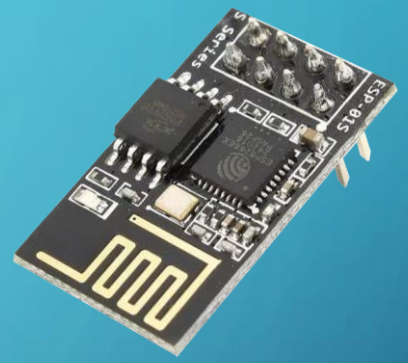
- Character displays (LCD)
- OLED
- TFT LCD
- Many different libraries each
  - Depends on use-case

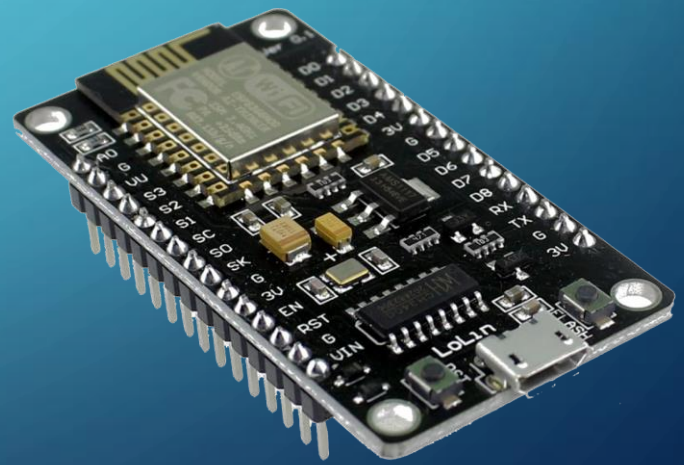


SETUP OF ESP32

ESP32

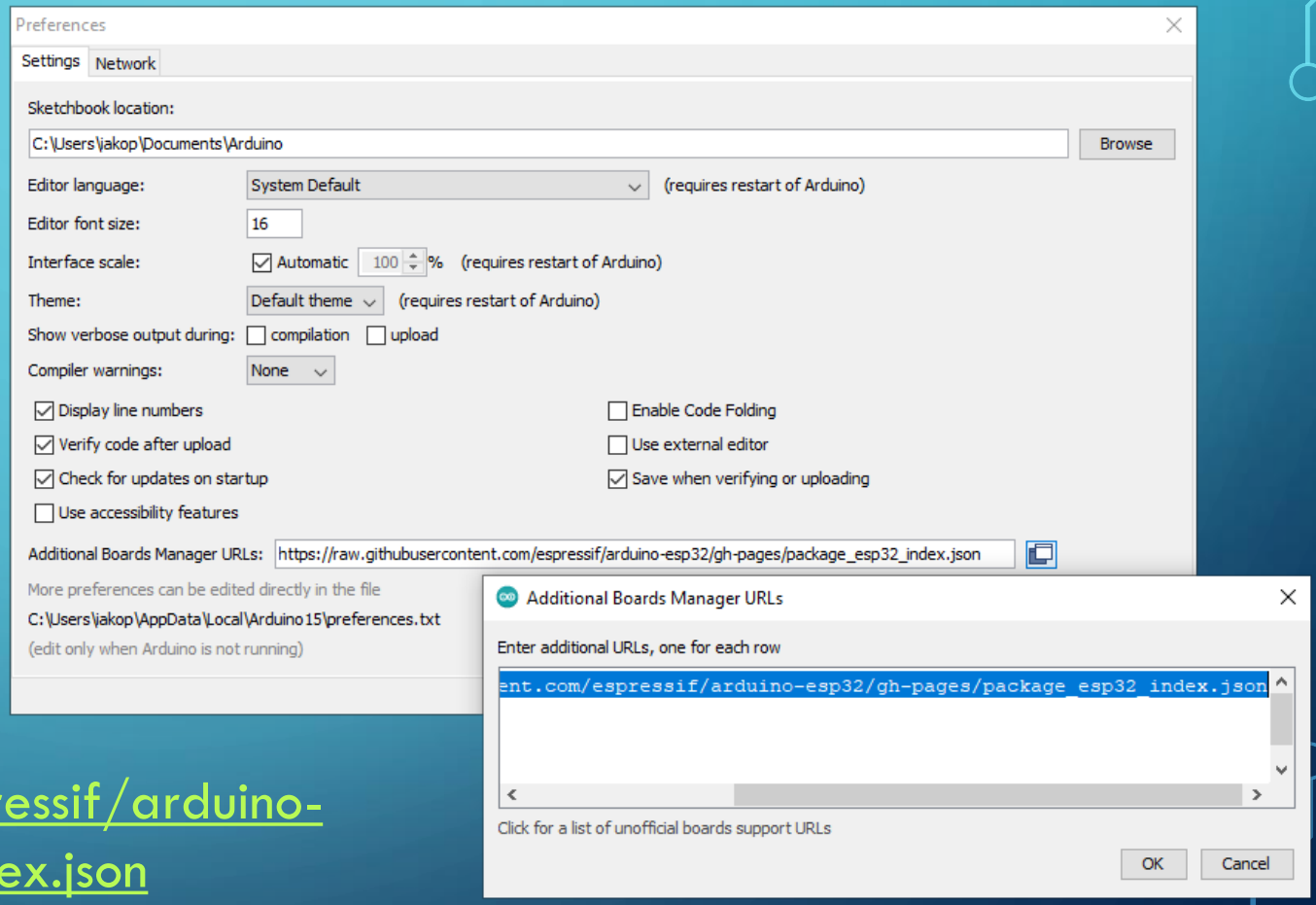
- WiFi SoC
  - ”System on Chip”
  - Read: MCU with built-in WiFi
- Heir to ESP8266
- Can be programmed through Arduino IDE
- Requires a little setup





BOARD DEFINITION

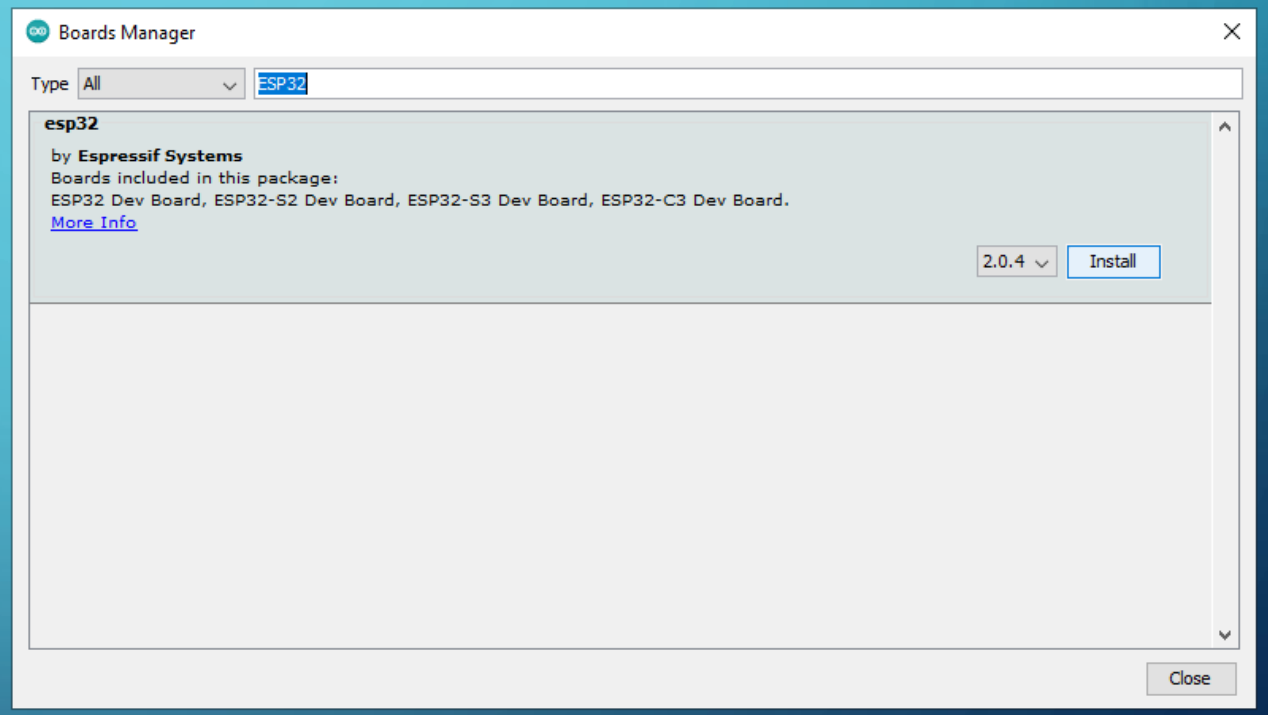
- In Arduino IDE:
  - ”File > Preferences”
- Additional board manager URLs:



- Insert: https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json

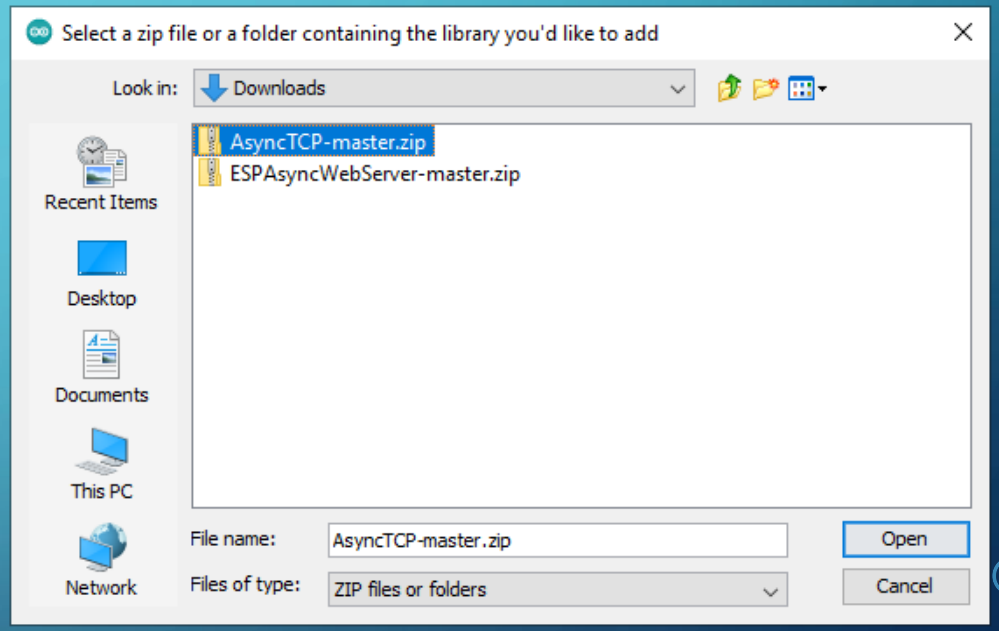
BOARD DEFINITION

- In ”Tools > Board > Board Manager”
- Search for ESP32 and press Install
  - Takes a while



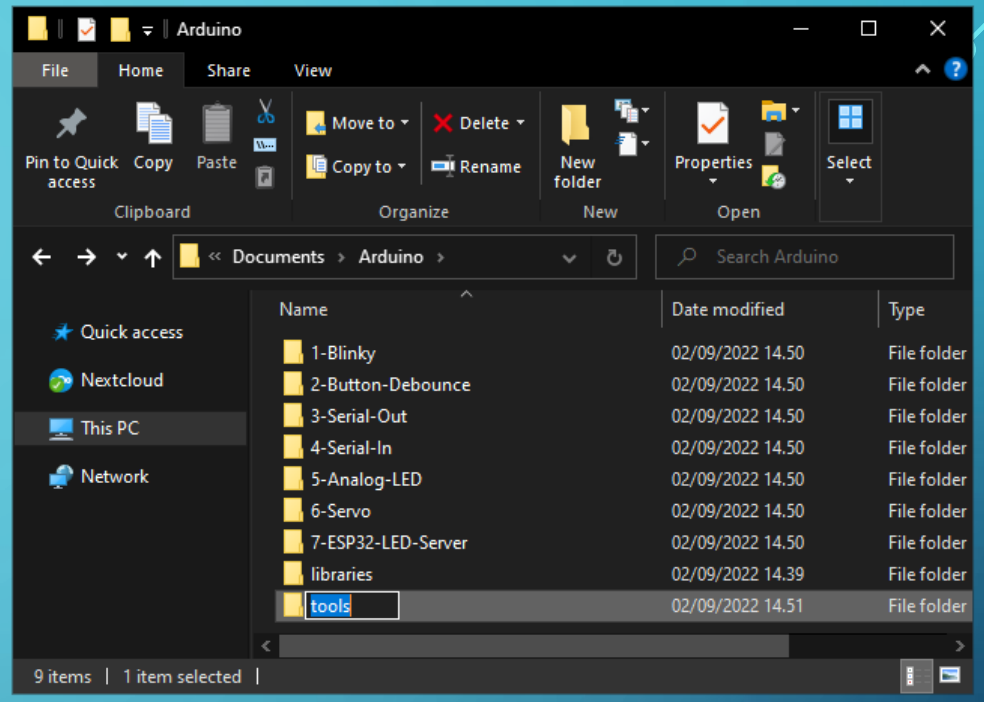
EXTERNAL LIBRARIES

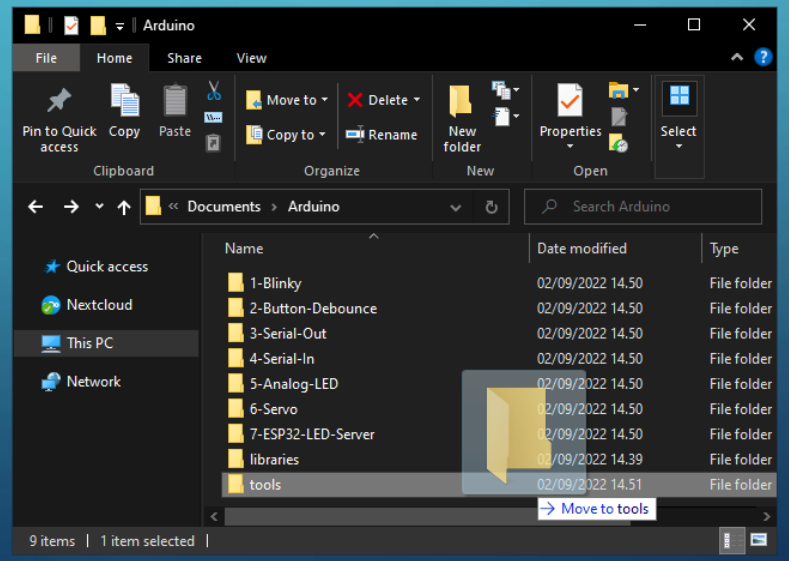
- Download the following two links:
- https://github.com/me-no-dev/ESPAsyncWebServer/archive/refs/heads/master.zip
- https://github.com/me-no-dev/AsyncTCP/archive/refs/heads/master.zip
- In Arduino IDE ”Sketch > Include Library > Add .ZIP Library…”
  - For every library, select and install it

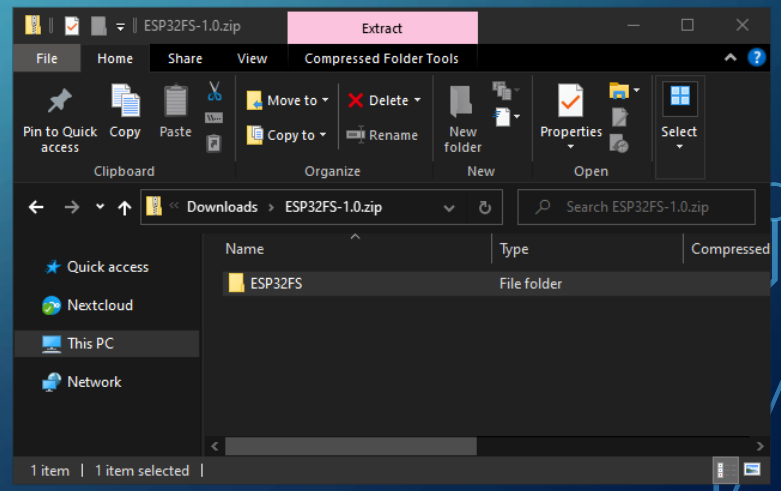


EXTERNAL TOOLS

- Download this tool plugin:
- https://github.com/me-no-dev/arduino-esp32fs-plugin/releases/download/1.0/ESP32FS-1.0.zip
- In your PC's local "Documents/Arduino" folder create a "tools" folder
- Unpack ESP32FS into the tools folder
- Restart Arduino IDE

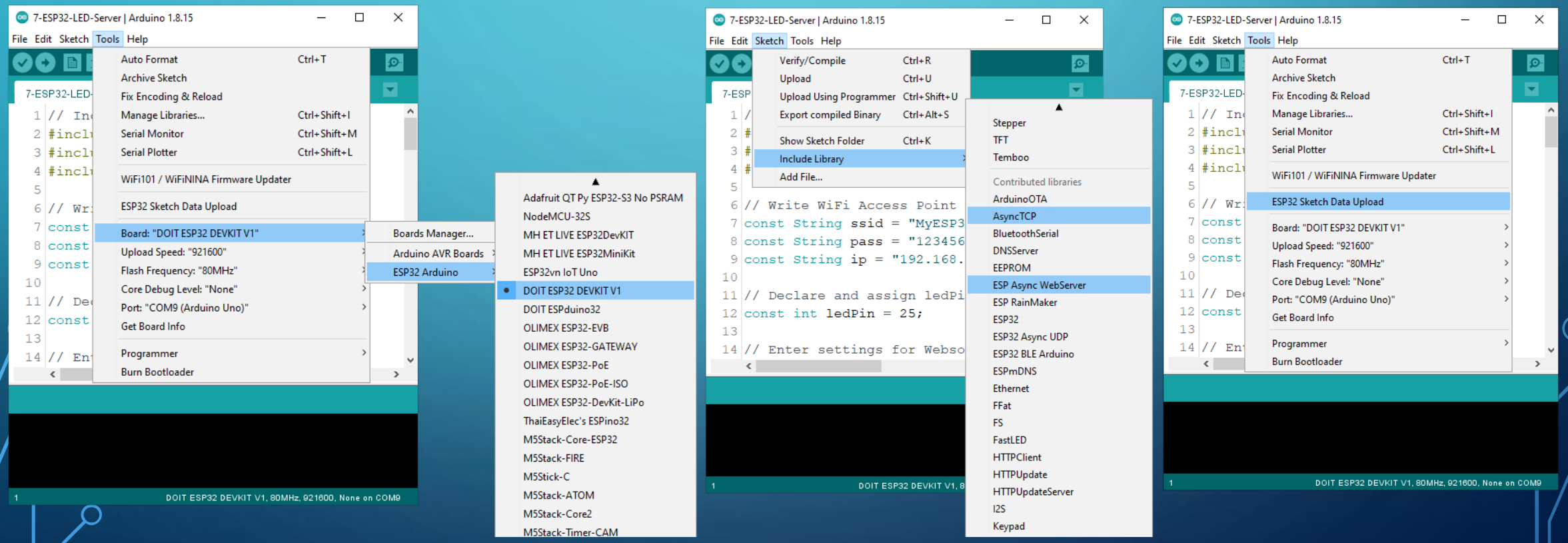






EXTERNAL TOOLS AND LIBRARIES

- We should now be able to pick the following in Arduino IDE



IOT ON ESP32

IOT IN GENERAL

- IoT (Internet of Things covers devices ”available” over a network
- Home network:
  - Home assistants
  - Philips Hue
- In the field:
  - WasteHero
  - TrapMe
  - MinkPolice

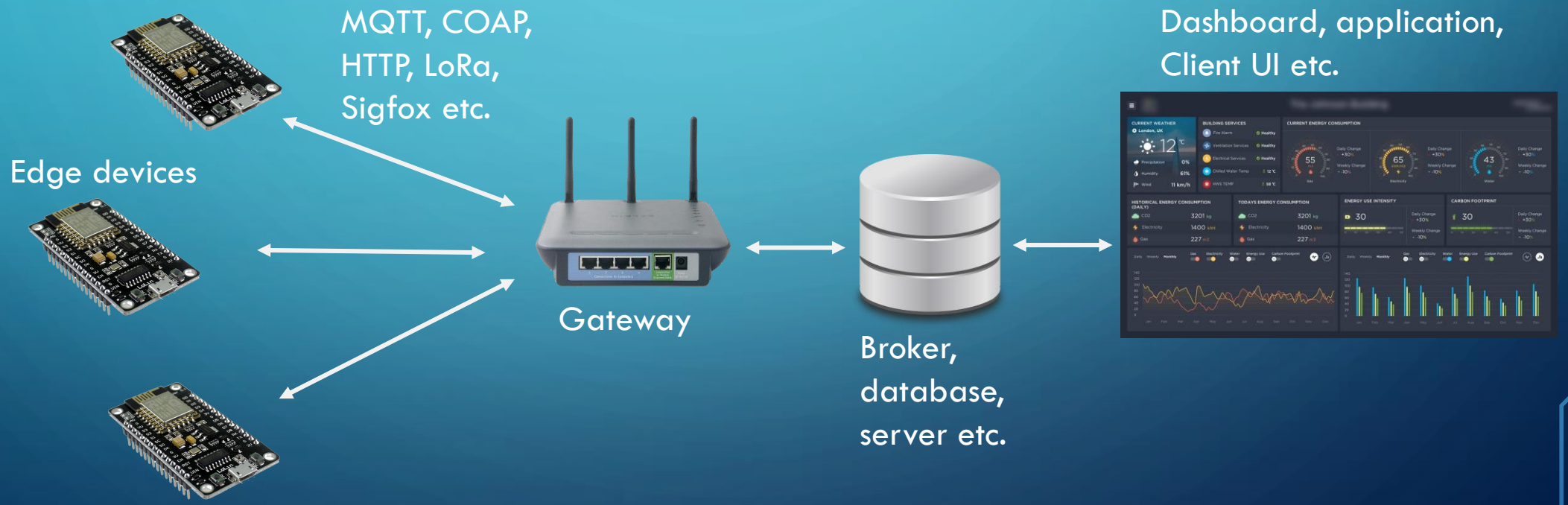
https://wastehero.io/

https://trapme.io/da

https://www.minkpolice.com/en/

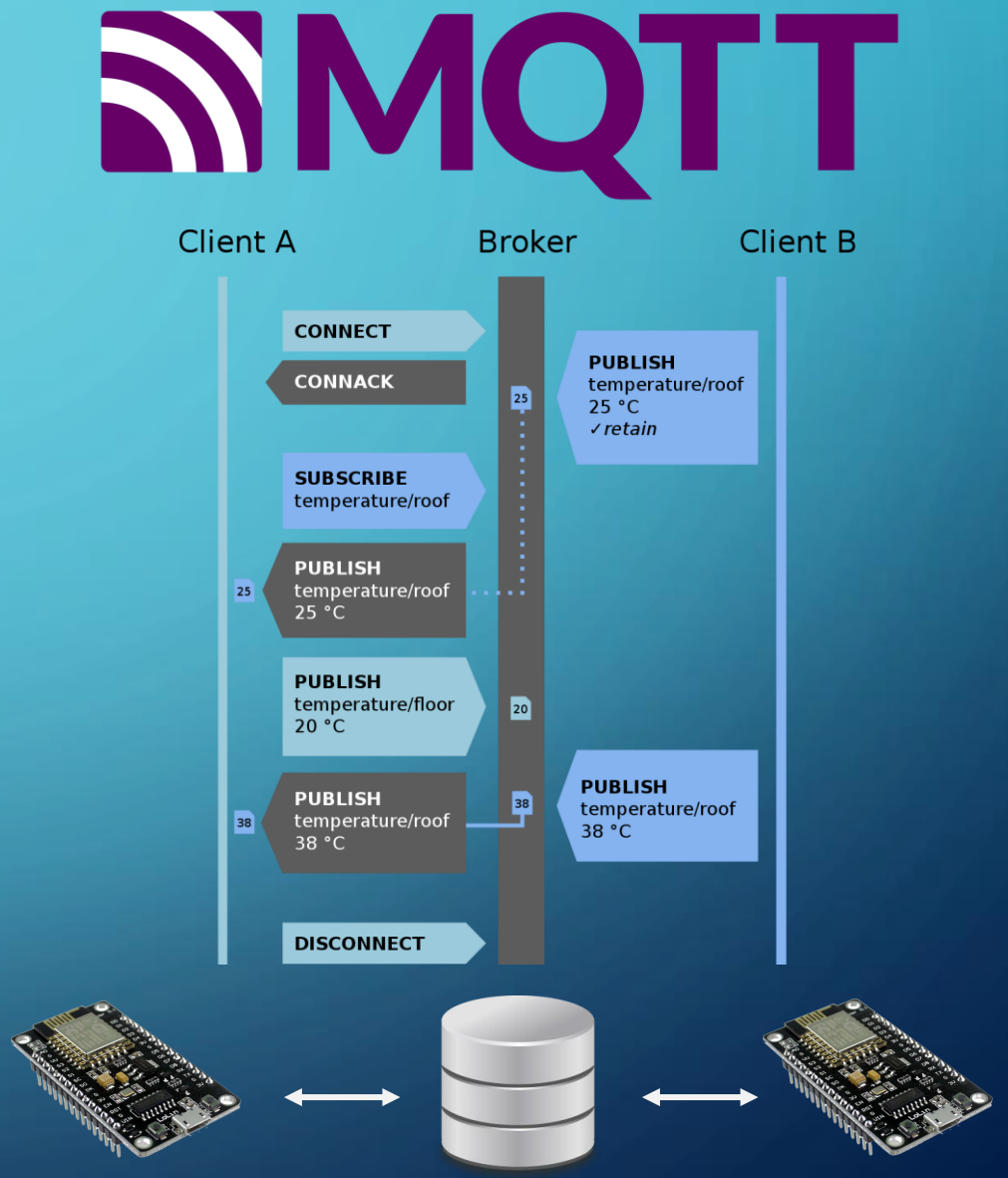
IOT IN GENERAL

- IoT deployment architectures usually look like:



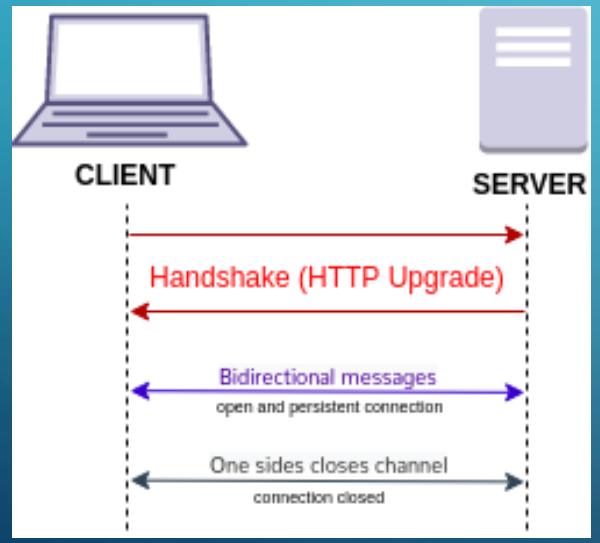
IOT IN GENERAL

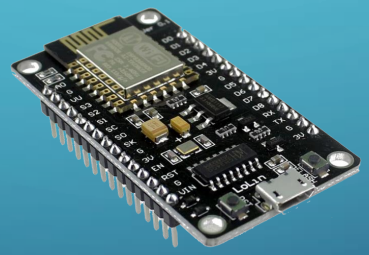
- One of the most popular protocols for IoT is MQTT
  - Simple and performant
- Requires clients and broker
  - Example from Wikipedia
    - Two thermostats on a network talk to the same broker
    - Exchange data asynchonously, the broker passes relevant data on



ESP32 WEBSERVER

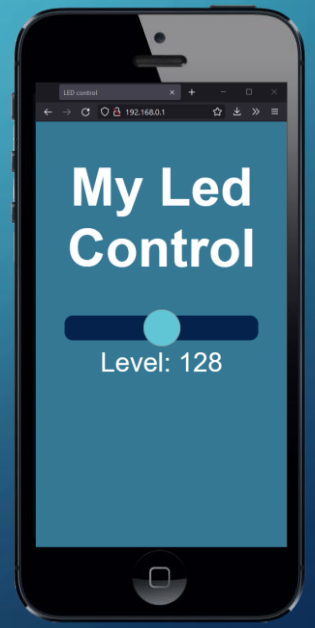
- To avoid setting up an entire network, we'll connect diretly to the ESP32
- Websocket is simply direct messages "a-la Serial"





Edge Device + Access Point + Server

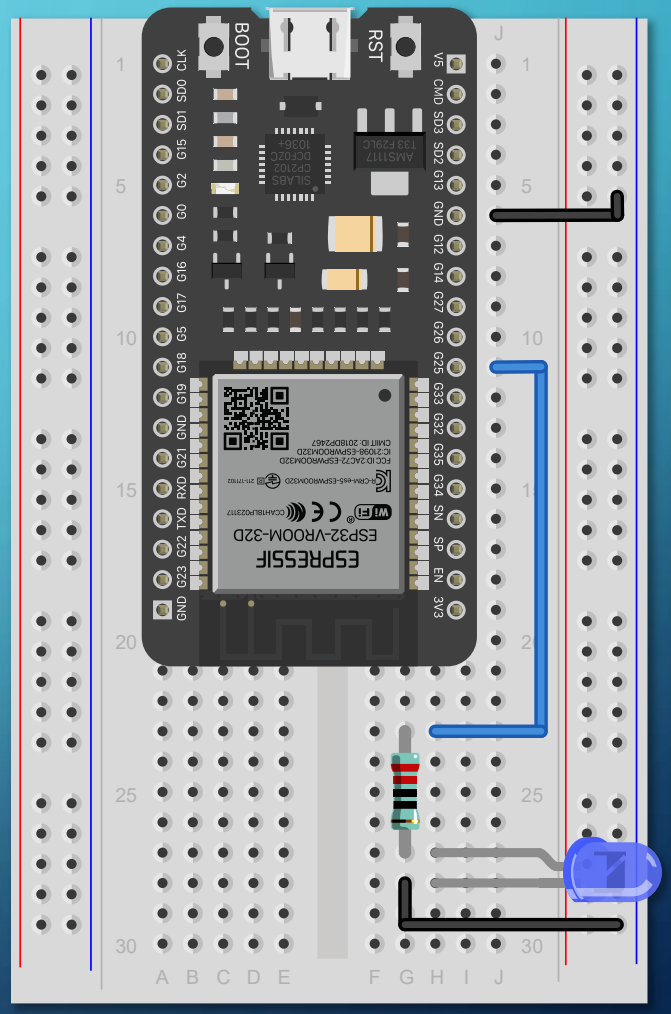
HTTP + WebSocket

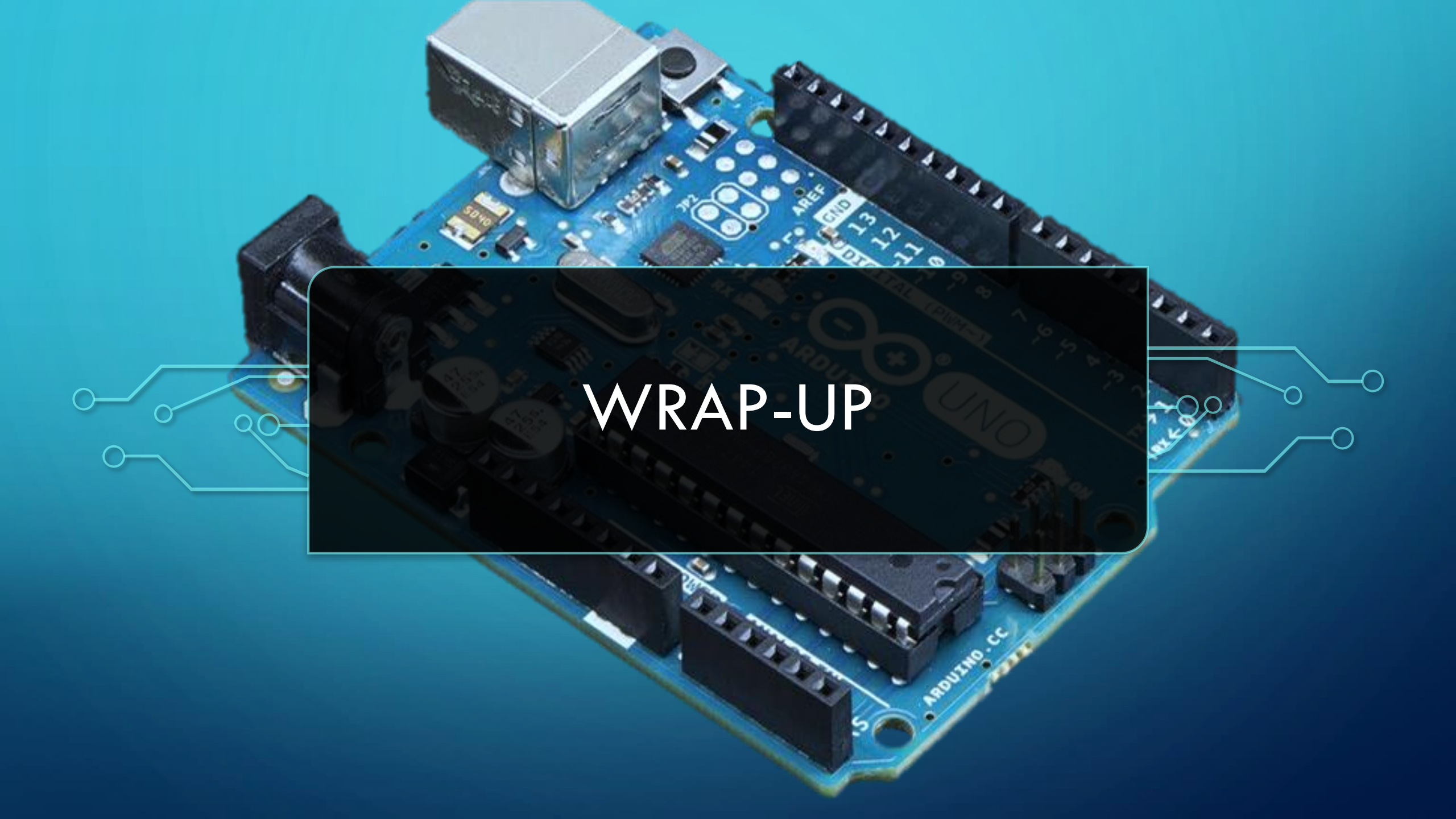


Client UI - Webbrowser

EXAMPLE 7: ESP32 LED SERVER

- We simply attach an LED to ESP32
- Most of the time will be used to code, decorate, and setup the webpage





WRAP-UP

OPSAMLING

- Now you know the basics of:
  - Digital I/O
  - Serial UART
  - Analog I/O
  - Introduced to miscellaneous components
- Have set up ESP32 for Arduino development
- Can build a simple web service
  - ESP32 as server and access point
  - HTML/Javascript for communication
- Introduced to general IoT
  - Architecture
  - MQTT

EVALUATION

- Will happily accept direct feedback
  - What did you like, what not?
  - Could something be formulated better?
  - How did the material fit the time available?
- Most of all, THANK YOU for participating!

RESOURCES

- Some online resources you can use:
  - Slides, examples, etc.
    - https://github.com/iakop/ArduinoCrashcourseIoT
  - Arduino IDE Download
    - https://www.arduino.cc/en/software
  - Arduino Language Reference
    - https://www.arduino.cc/reference/en/