

EPFL



ROBOPOLY

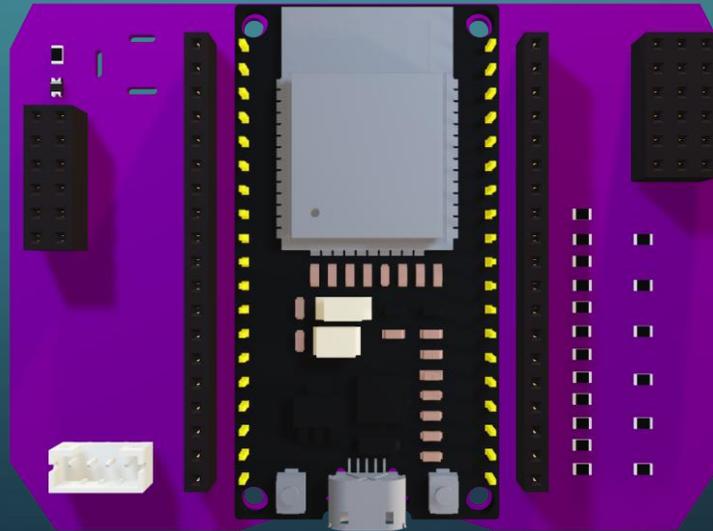


agepoly

<INTRODUCTION À L'ÉLECTRONIQUE PRATIQUE>

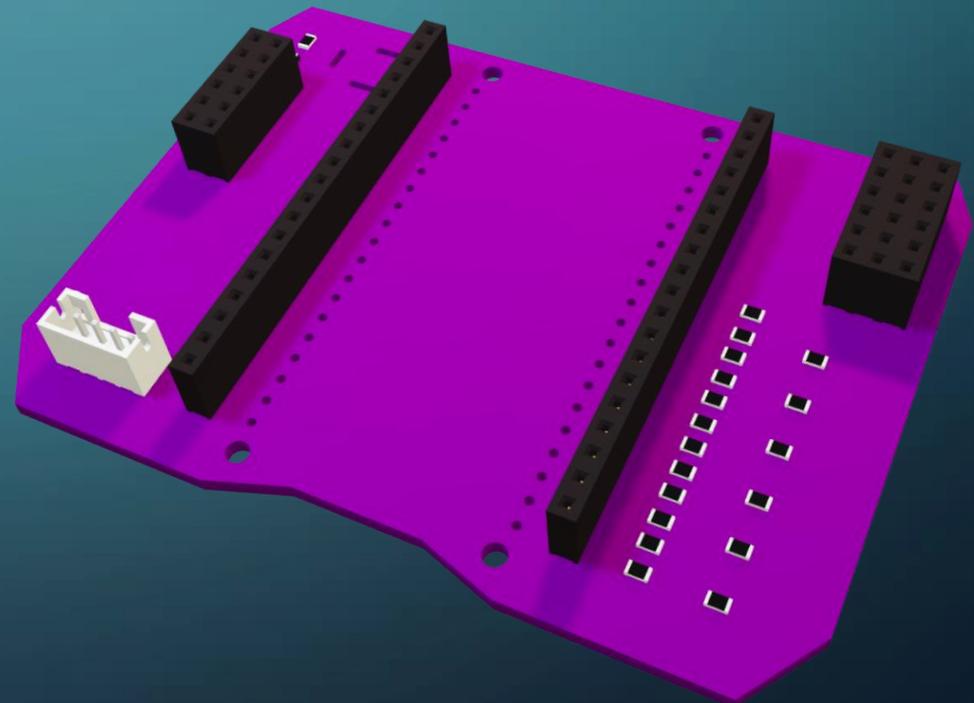
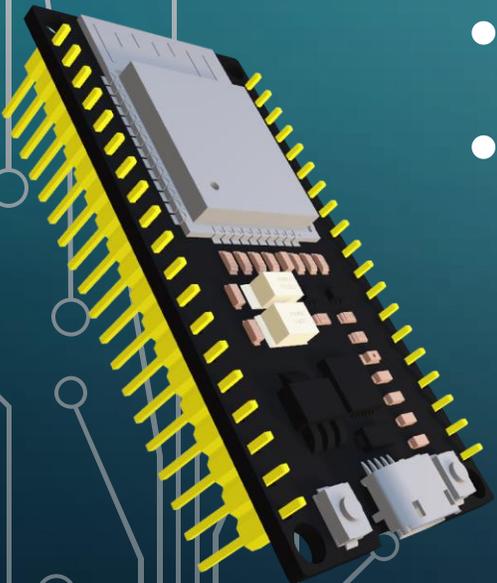
FLORIAN KLEIN

SONNY BASSO



PLAN D'AUJOURD'HUI

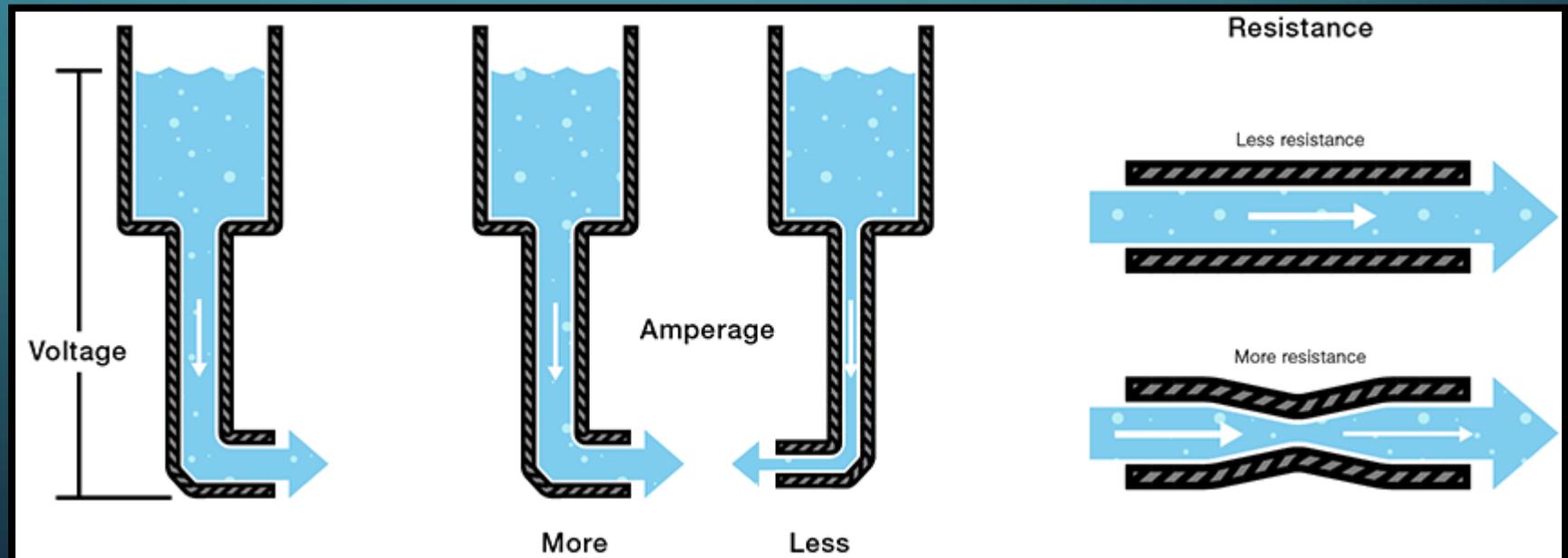
- Les bases de l'électronique
- Comment commencer un projet ?
- Microcontrôleurs
- Capteurs
- Actuateurs



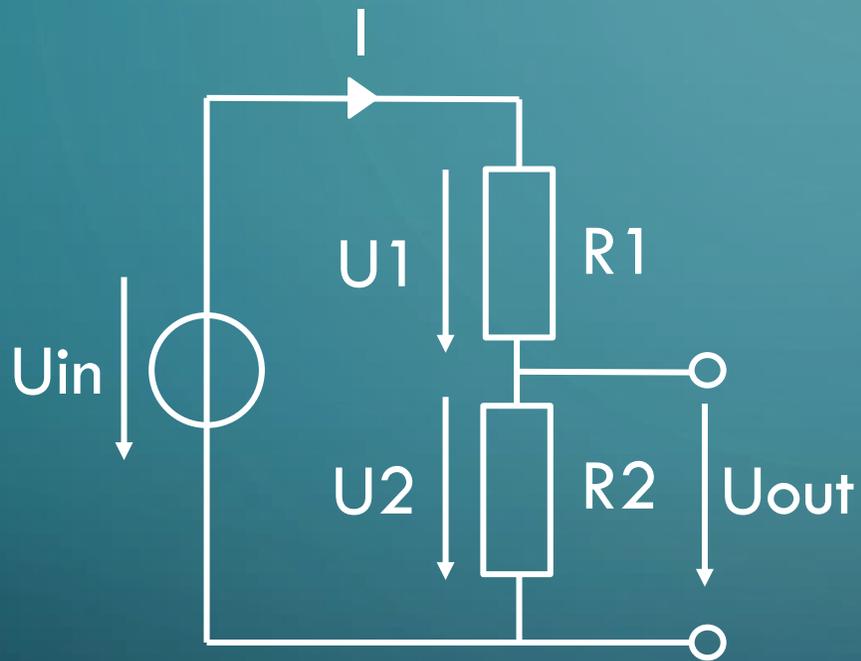
LES BASES



Loi d'Ohm : $U = R \cdot I$
 $P = U \cdot I$



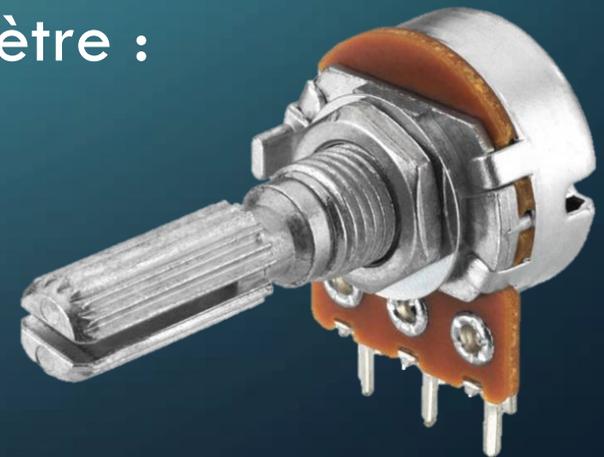
DIVISEUR DE TENSION



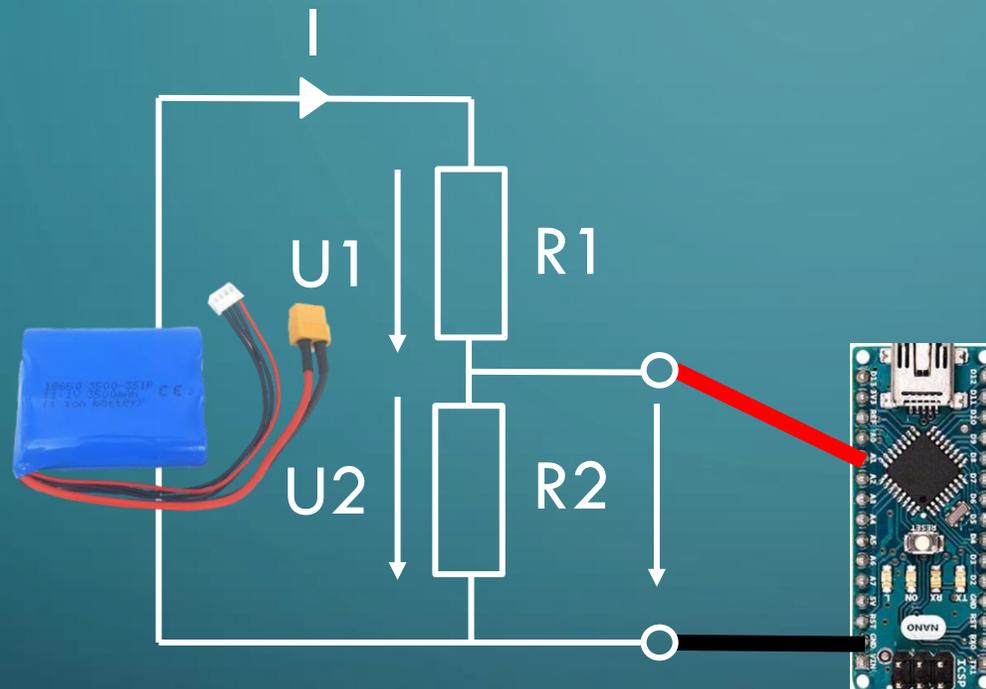
$$I = U_{in} / (R1 + R2)$$

$$U_{out} = I \cdot R2 = U_{in} \cdot R2 / (R1 + R2)$$

Potentiomètre :



MESURE DE LA TENSION D'UNE BATTERIE LI-ION 3S AVEC UN ARDUINO



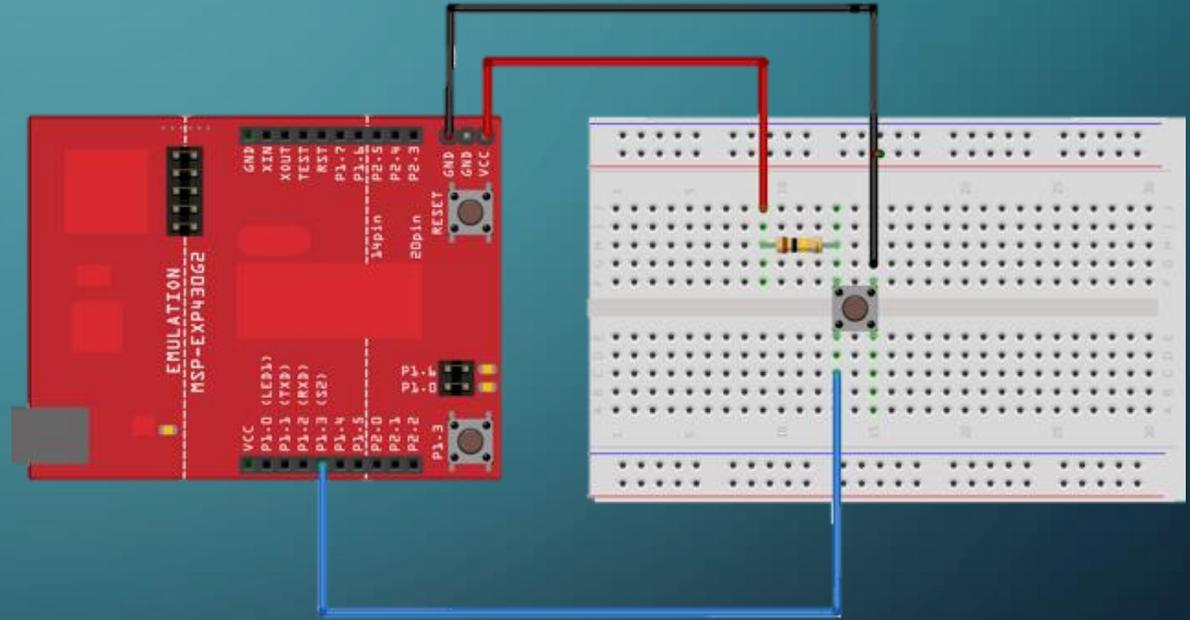
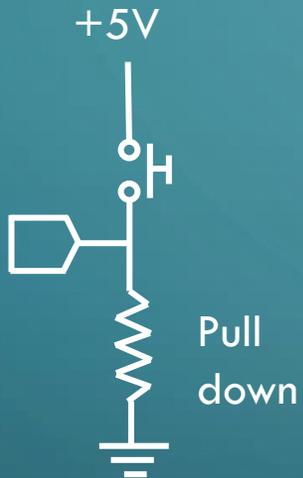
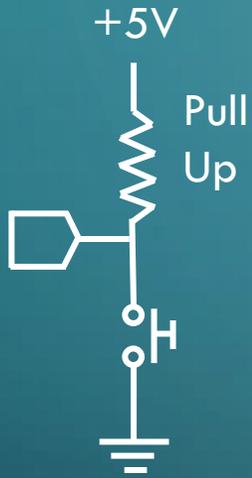
LI-ION 3S : 9.6V-12.6V

TENSION ANALOGIQUE D'ENTRÉE
ARDUINO NANO : 0V-5V

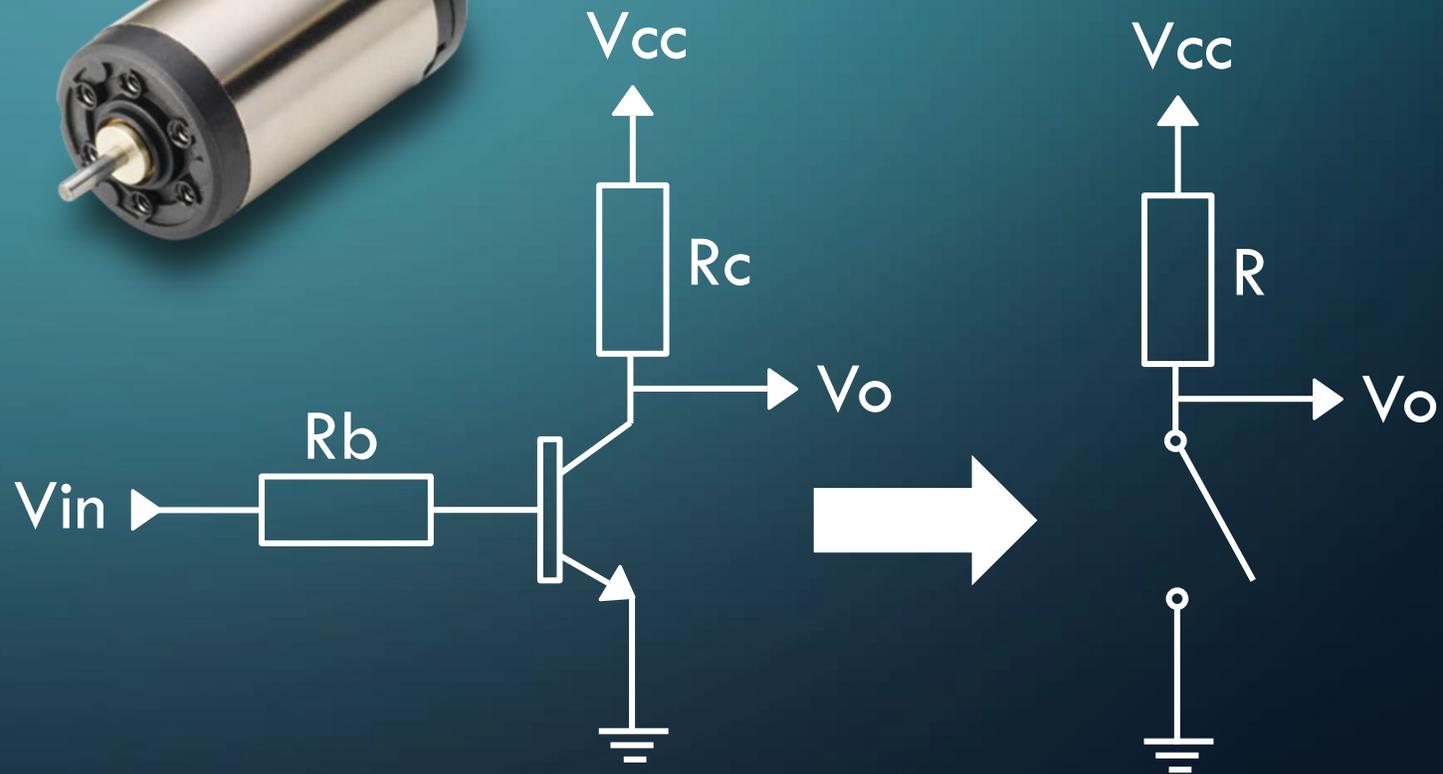
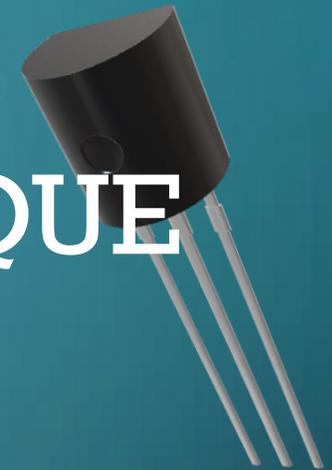
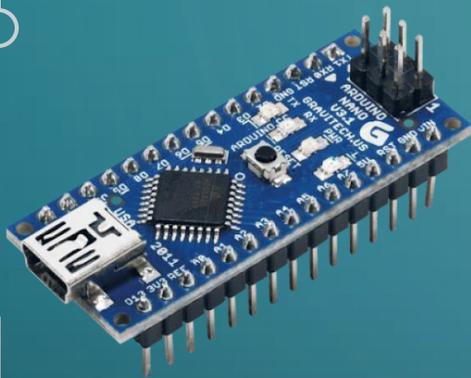
→ RESCALE LA TENSION POUR QU'ELLE
SOIT ENTRE 0V ET 5V → $R1=20\text{KOHM}$ ET
 $R2=10\text{KOHM}$

- 12.6V → 4.2V
- 9.6V → 3.2V

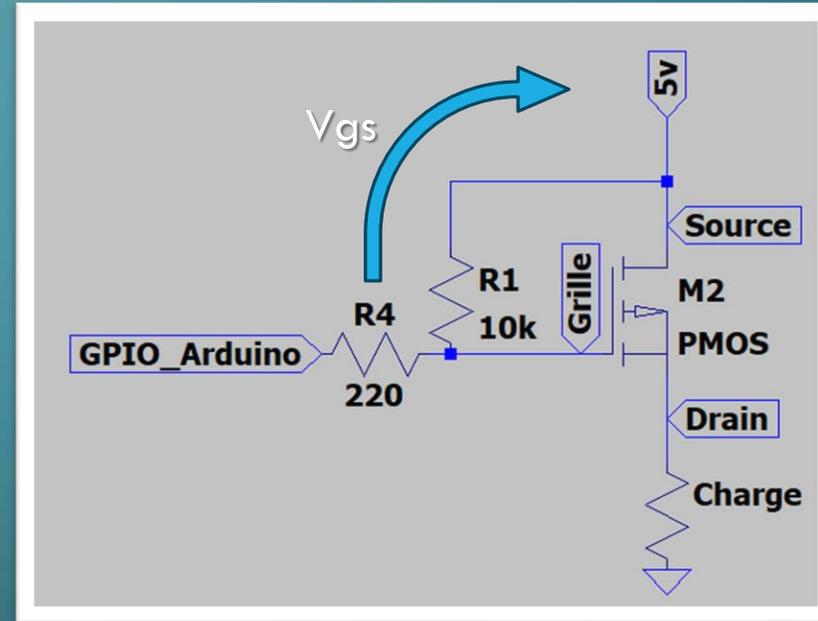
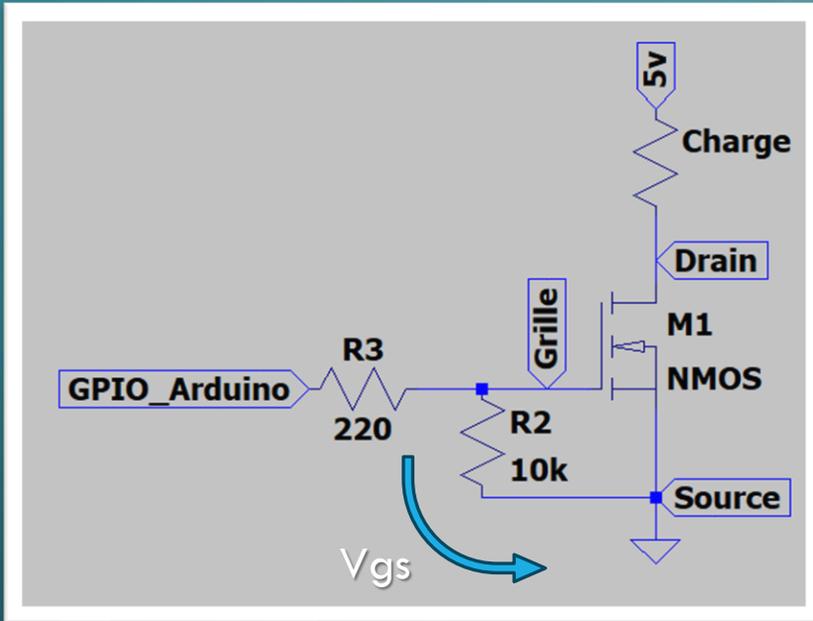
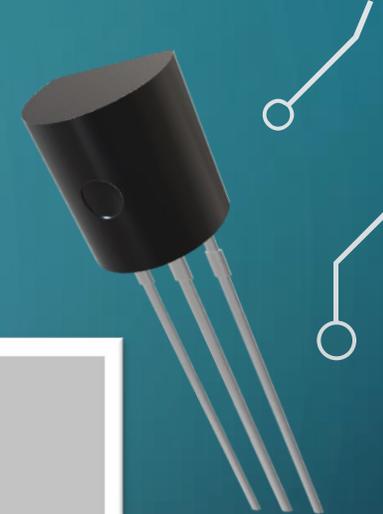
PULL UP/PULL DOWN ET BUTTONS/SWITCHES



LE TRANSISTOR : L'INTERRUPTEUR ÉLECTRONIQUE



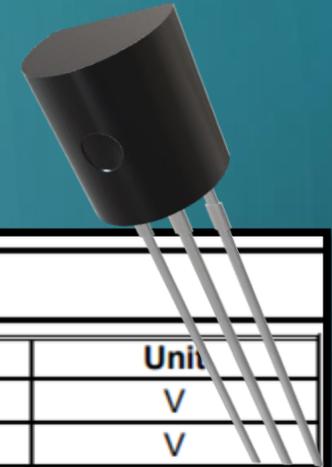
NMOS VS PMOS



GPIO Arduino	Charge
LOW	OFF
HIGH	ON

GPIO Arduino	Charge
LOW	ON
HIGH	OFF

Comment bien choisir un transistor



Maximum Ratings (@T_A = +25°C, unless otherwise specified.)

Characteristic	Symbol	Value	Unit
Drain-Source Voltage	V _{DSS}	130	V
Gate-Source Voltage	V _{GSS}	±20	V
Continuous Drain Current (Note 6) V _{GS} = 10V	I _D	T _A = +25°C	1.0
		T _A = +70°C	0.8
	I _D	t < 10s, T _A = +25°C	1.2
		T _A = +70°C	1.0
Pulsed Drain Current (10µs Pulse, Duty Cycle ≤ 1%)	I _{DM}	3.3	A
Maximum Body Diode Continuous Current (Note 6)	I _S	1.0	A

Electrical Characteristics (@T_A = +25°C, unless otherwise specified.)

Characteristic	Symbol	Min	Typ	Max	Unit	Test Condition
OFF CHARACTERISTICS (Note 7)						
Drain-Source Breakdown Voltage	BV _{DSS}	130	—	—	V	V _{GS} = 0V, I _D = 250µA
Zero Gate Voltage Drain Current	I _{DSS}	—	—	100	nA	V _{DS} = 120V, V _{GS} = 0V
Gate-Body Leakage	I _{GSS}	—	—	±100	nA	V _{GS} = ±20V, V _{DS} = 0V
ON CHARACTERISTICS (Note 7)						
Gate Threshold Voltage	V _{GS(TH)}	2.0	2.7	4.0	V	V _{DS} = V _{GS} , I _D = 250µA
Static Drain-Source On-Resistance	R _{DS(ON)}	—	0.41	0.75	Ω	V _{GS} = 10V, I _D = 2.0A
		—	0.43	0.85		V _{GS} = 6.0V, I _D = 2.0A
Diode Forward Voltage	V _{SD}	—	0.8	1.2	V	V _{GS} = 0V, I _S = 1.0A

COMMENCER UN PROJET

Idée géniale

Quel est l'objectif du projet

Capteur

Mesure de l'environnement

Processeur

Décide l'action

Actuateur

Agit sur l'environnement

EXEMPLE : LE KHNUM (NOTRE KIT BOTANIQUE)

Idée géniale

Arrosage automatique d'une plante

Capteur

Humidité de sol

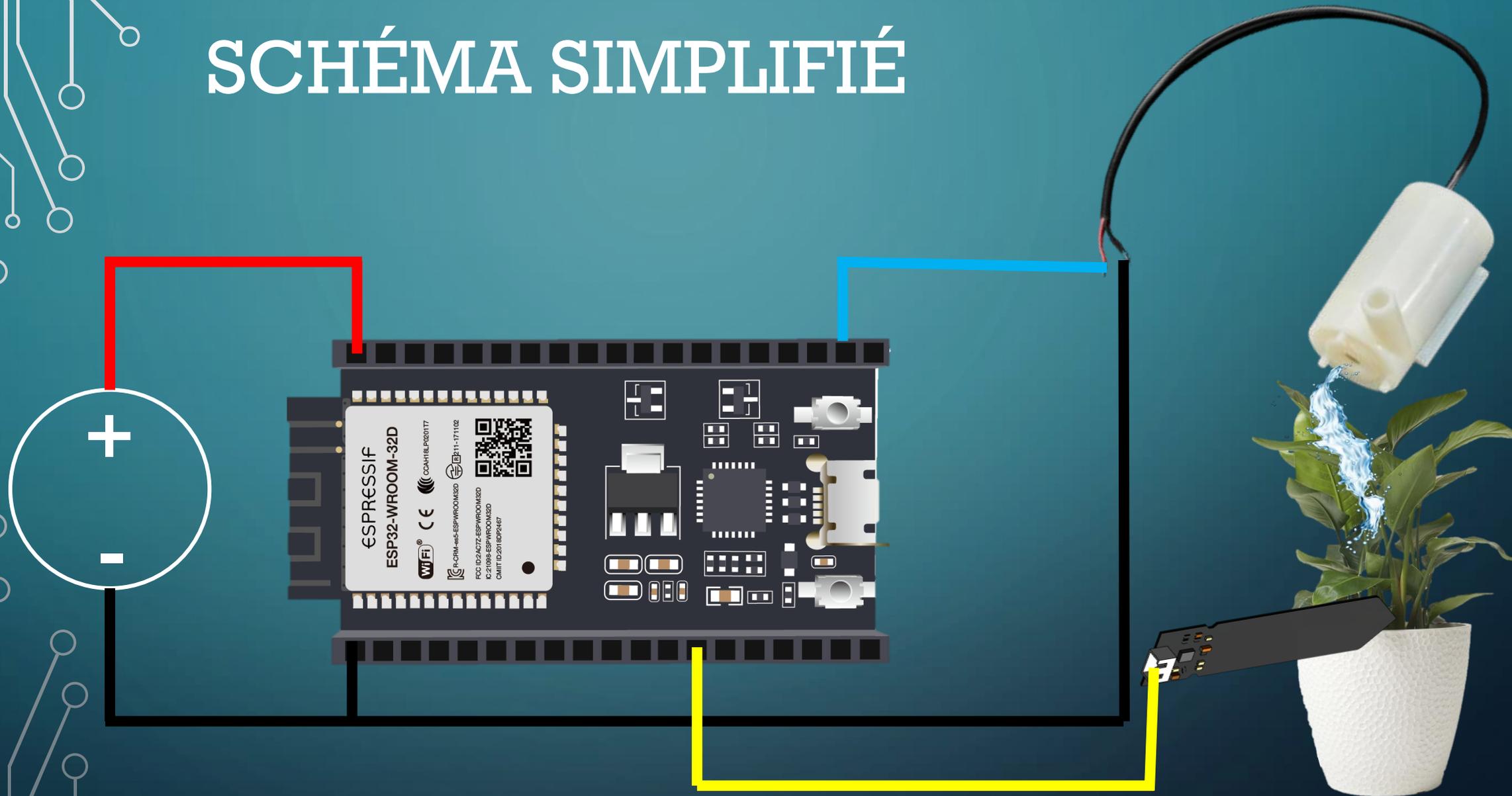
Processeur

ESP32 | Raison : Wifi/Bluetooth

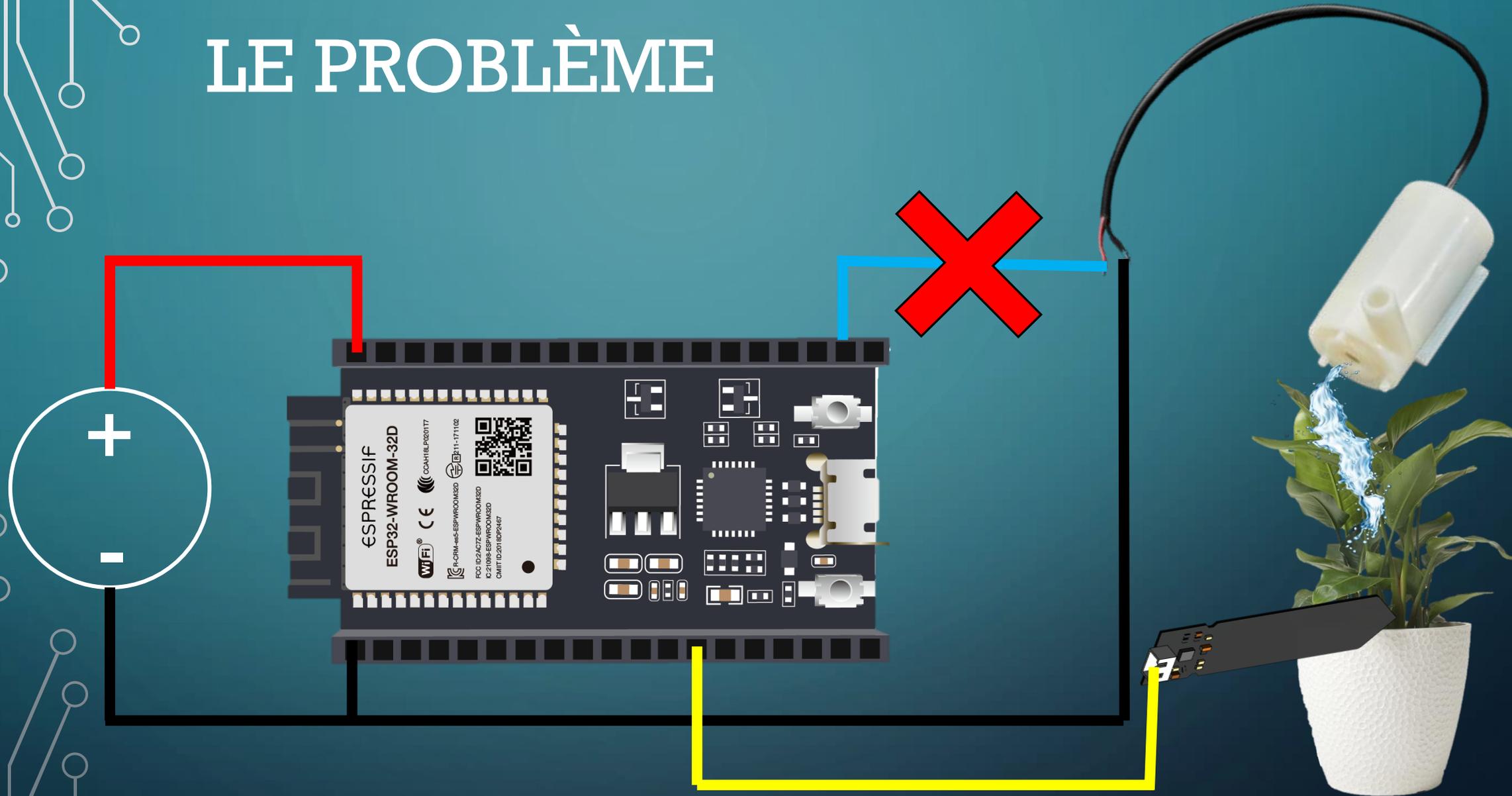
Actuateur

Pompe péristaltique | Pomper l'eau

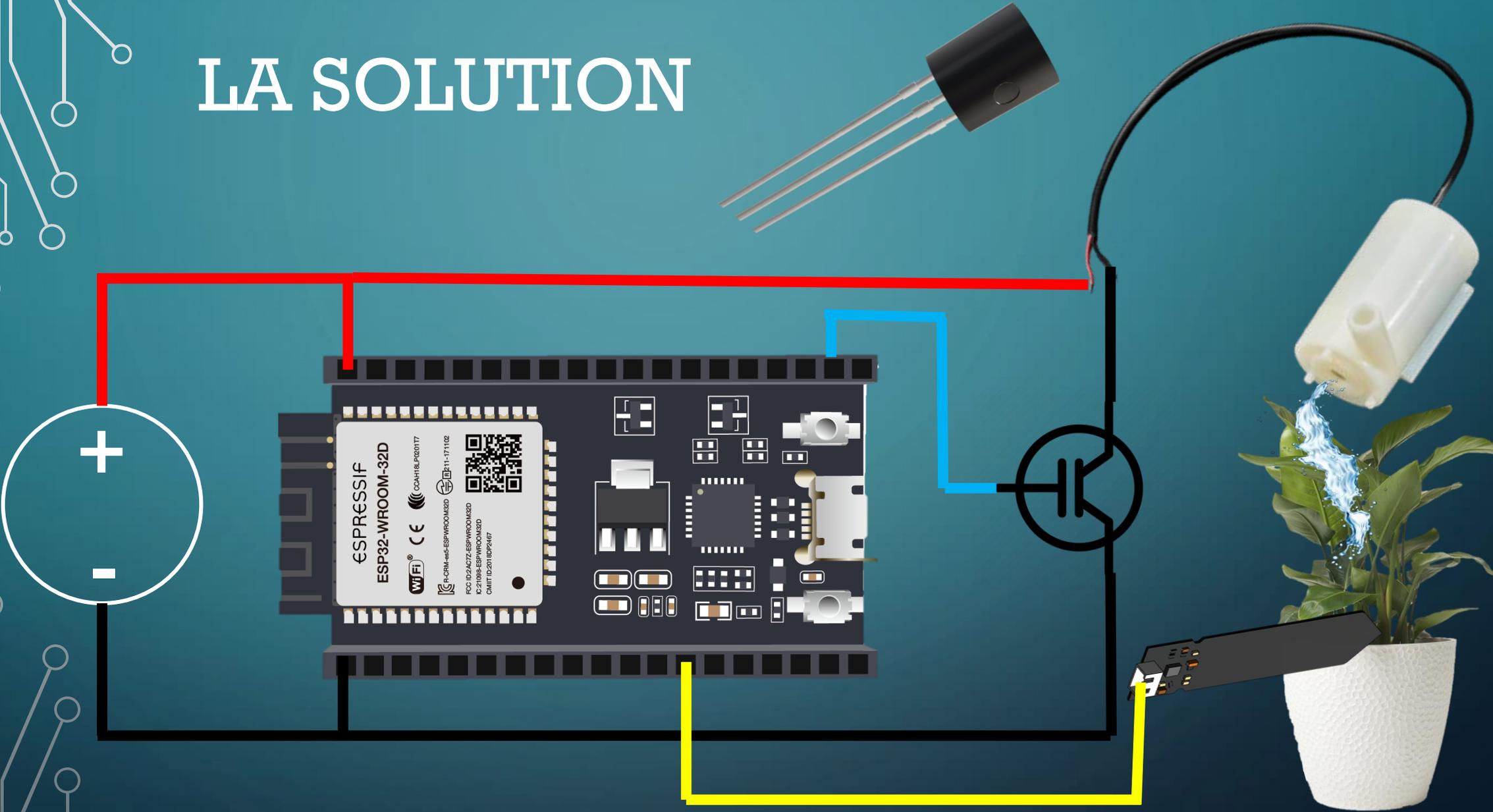
SCHÉMA SIMPLIFIÉ



LE PROBLÈME

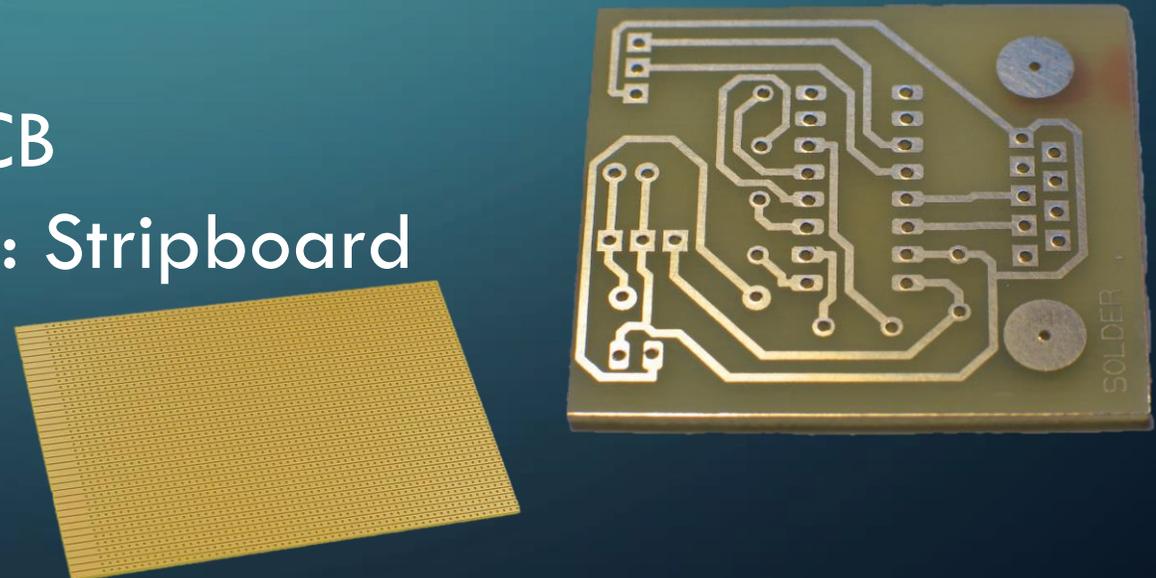
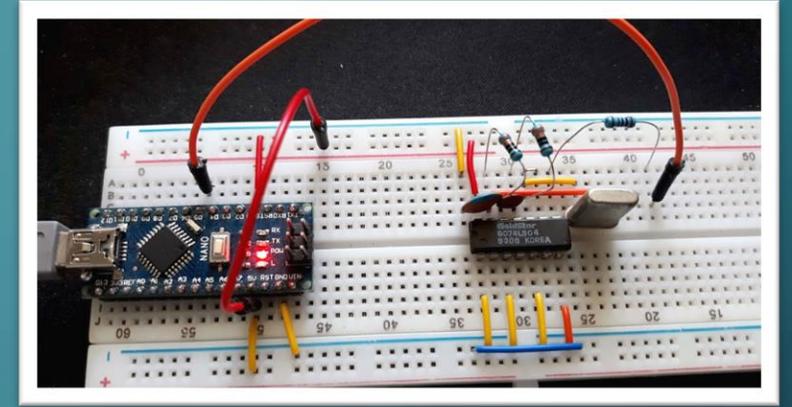


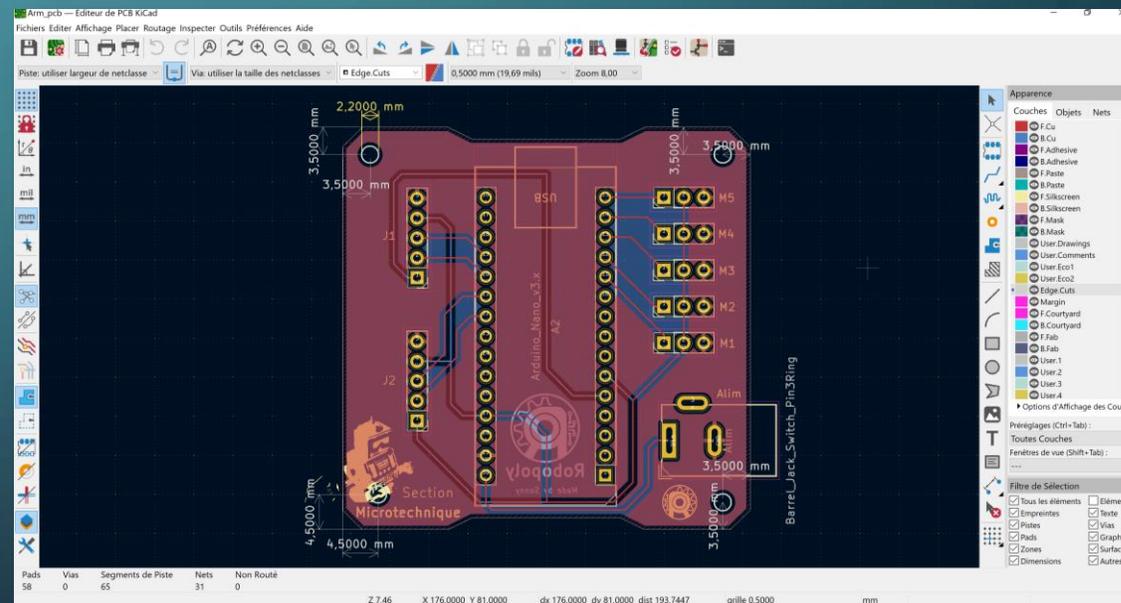
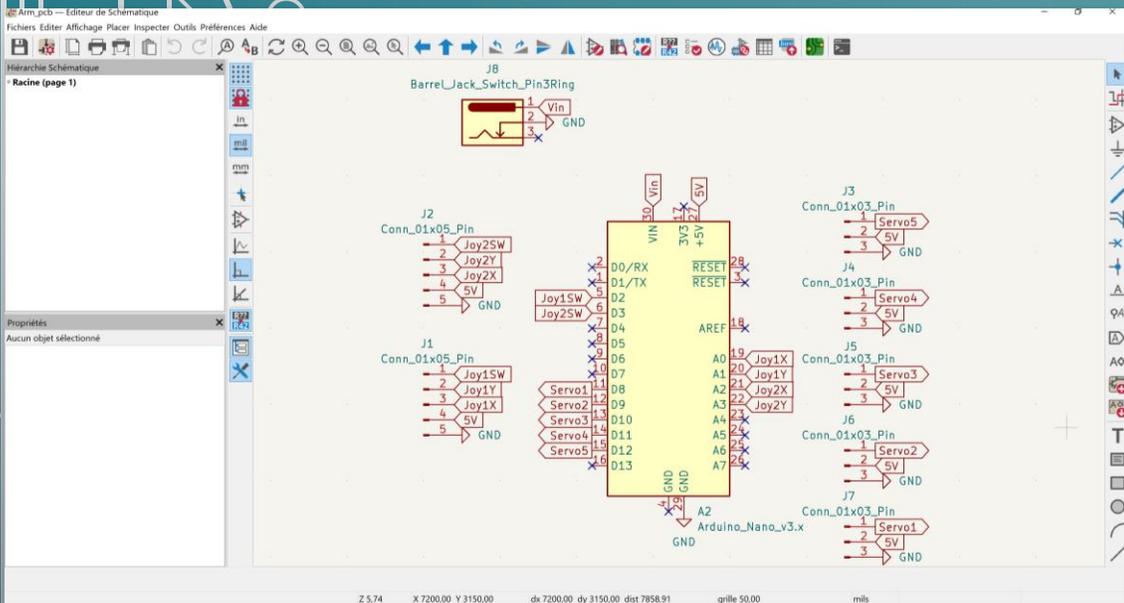
LA SOLUTION



BREADBOARDS/PCB

- Pas de soudure
 - Prototypage rapide
 - Simple à modifier, mesurer et déverminer
-
- Version robuste : PCB
 - Version petit circuit : Stripboard

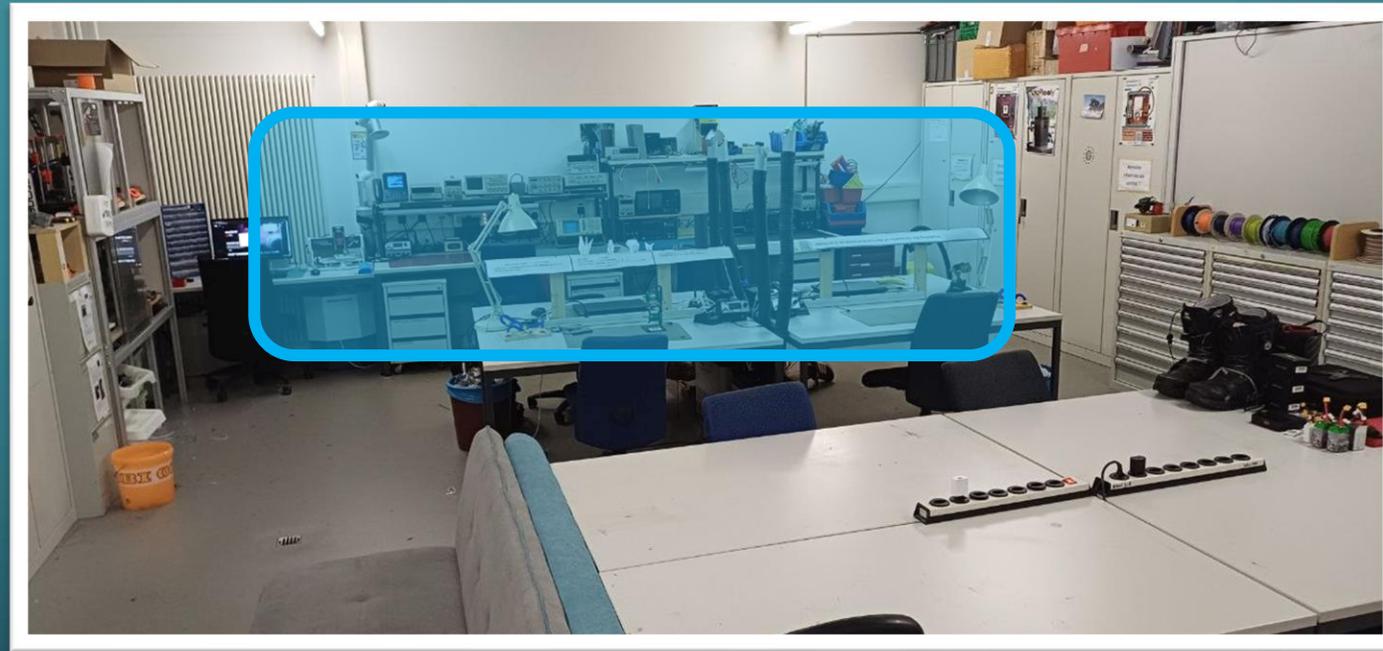




LES OUTILS



LES OUTILS



LES OUTILS



LES OUTILS



LES OUTILS



LES OUTILS



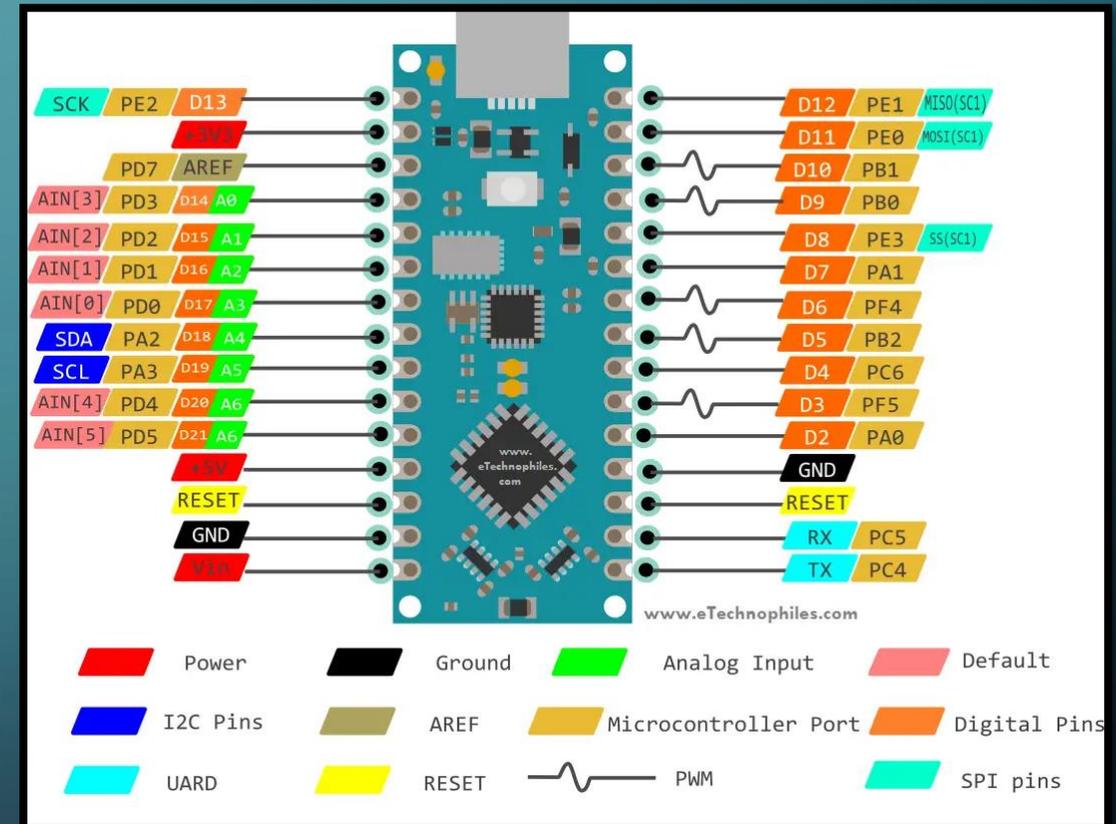
LES OUTILS



LE CONTRÔLE : MICROCONTRÔLEUR

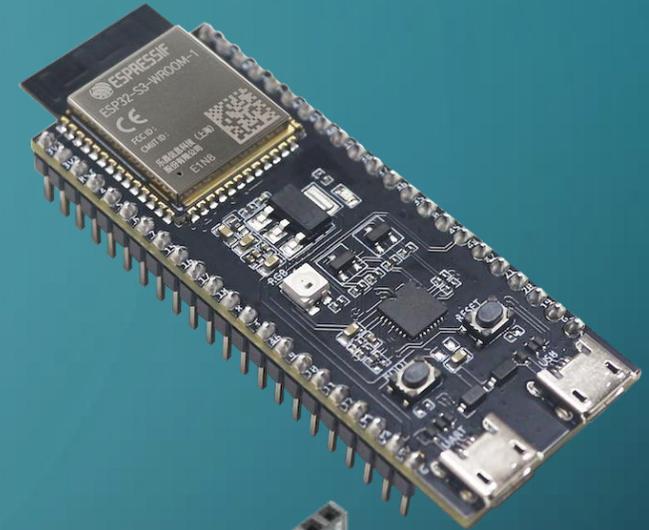
Programmé en C/C++

- Digital I/O:
0 **ou** 5 V entrée et sortie
- Analog:
0 **à** 5V entrée traduits en valeurs de 0 à 1024
- Protocols (I2C, UART, SPI)



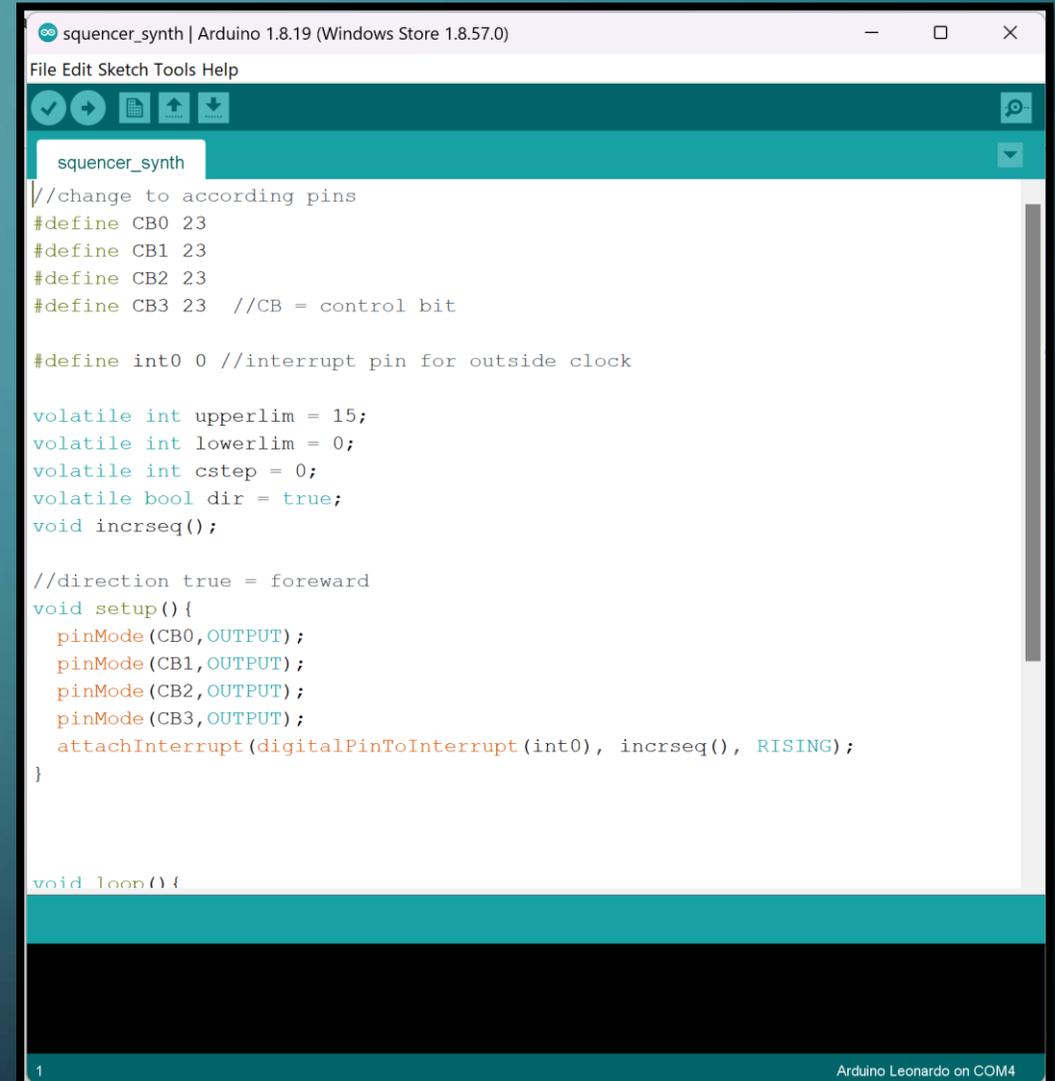
LEQUEL CHOISIR ?

- Arduino → Facile à programmer
- ESP → Bluetooth/WiFi



ARDUINO IDE

- Programme en C
- Connecte l'Arduino/ESP à ton ordi en USB et upload le programme
- Powersupplies: for now with USB,
- Later course via Vin (external)



```
sequencer_synth | Arduino 1.8.19 (Windows Store 1.8.57.0)
File Edit Sketch Tools Help
sequencer_synth
//change to according pins
#define CB0 23
#define CB1 23
#define CB2 23
#define CB3 23 //CB = control bit

#define int0 0 //interrupt pin for outside clock

volatile int upperlim = 15;
volatile int lowerlim = 0;
volatile int cstep = 0;
volatile bool dir = true;
void incrseq();

//direction true = forward
void setup() {
  pinMode(CB0, OUTPUT);
  pinMode(CB1, OUTPUT);
  pinMode(CB2, OUTPUT);
  pinMode(CB3, OUTPUT);
  attachInterrupt(digitalPinToInterrupt(int0), incrseq(), RISING);
}

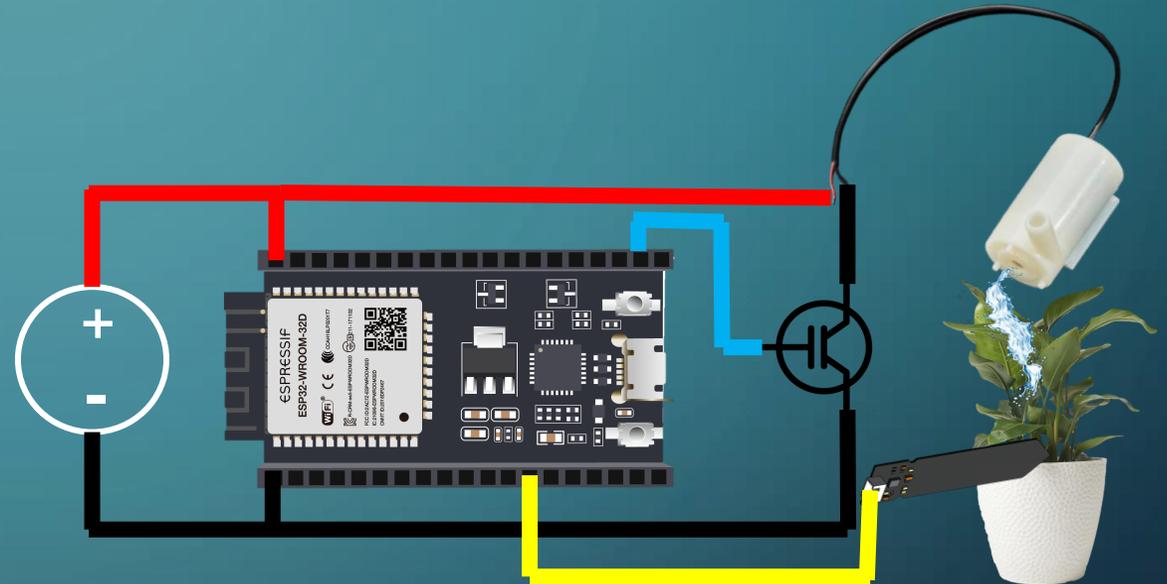
void loop() {
```

1 Arduino Leonardo on COM4

EXAMPLE CODE

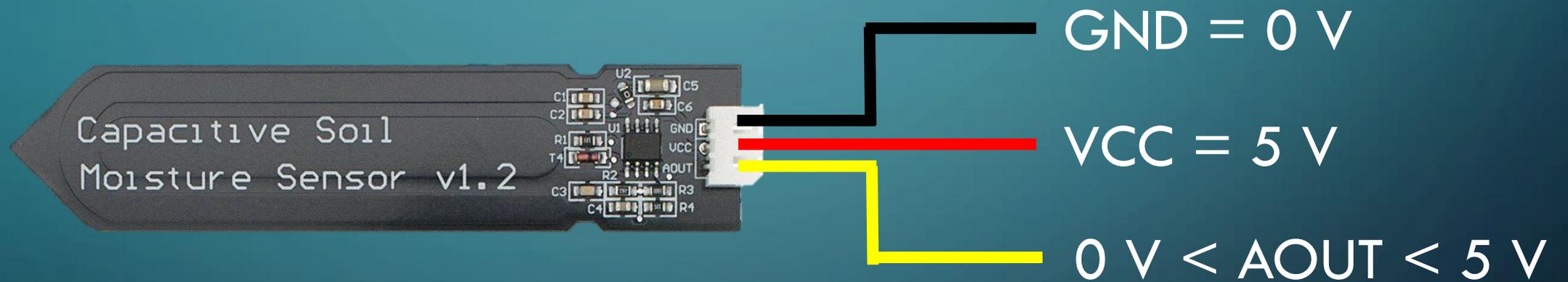
```
#include <HX711.h>
void setup(){
  Serial.begin( 9600 );
  Serial.println( "Hello world" );
  pinMode( PIN1, OUTPUT );
  digitalWrite( PIN1, LOW );
}

void loop(){
  int humidity = analogRead( A0 );
  if ( humidity < 200 )
  {
    digitalWrite( PIN1, HIGH );
    delay( 1000 );
    digitalWrite( PIN1, LOW );
  }
  delay( 1000 );
}
```



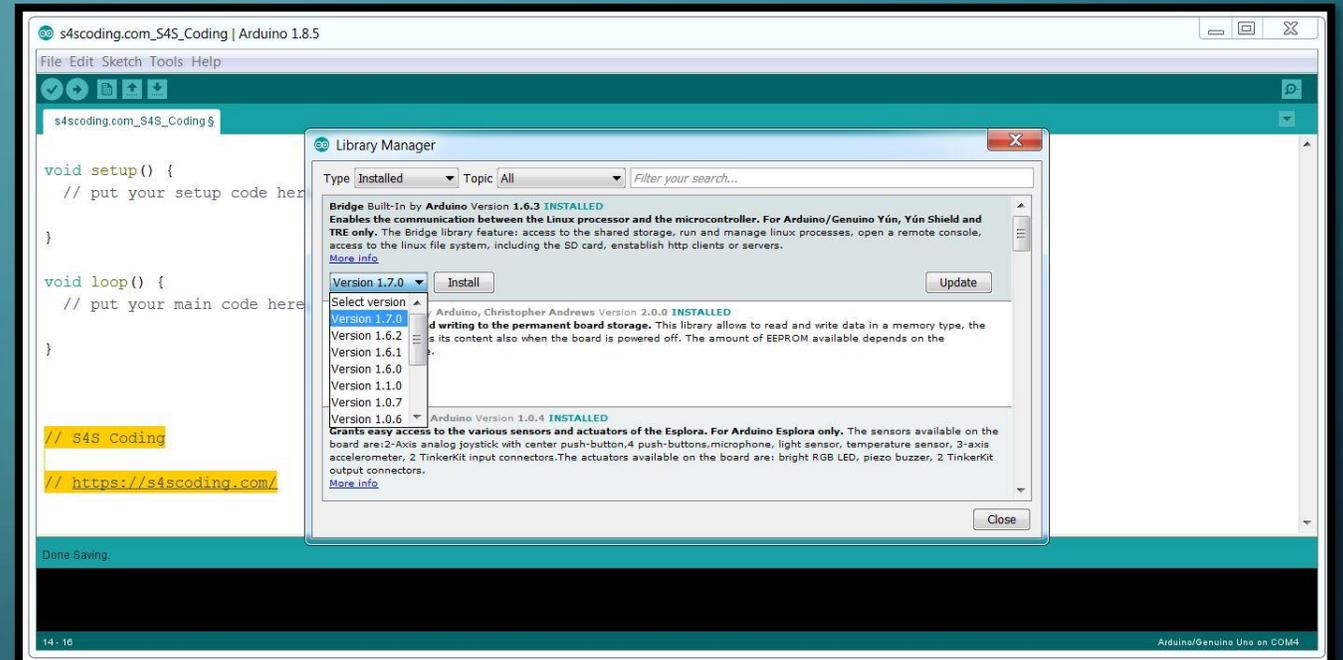
CAPTEURS - ANALOG

- Peut être lu avec la fonction : `analogRead(nom_du_pin)`



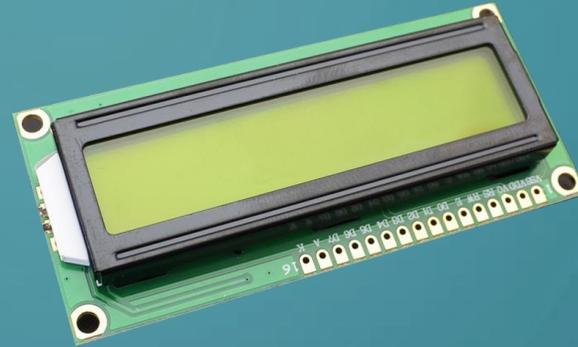
CAPTEURS - NUMÉRIQUE

- Digital I/O Pins
- Protocols:
 - Serial
 - I2C
 - SPI



DIFFERENT ACTUATORS

- Moteurs DC
- LCD → I2C
- Buzzer
- Led
- Pompe
- Actuateur linéaire
- etc



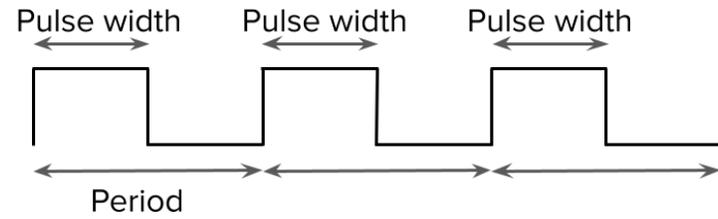
MOTEURS

	Moteur DC à balais (brushed)	Moteur DC sans balais (brushless)	Moteur pas-à-pas	Servo moteur
Image				
Avantages	Pas cher, facile à contrôler, bon couple à faible vitesse	Durable, efficace, silencieux	Positionnement précis, stable à faible vitesse	Précision de position, bon ratio couple/poids
Inconvénients	Usure des balais, bruit, entretien	Contrôle complexe (besoin d'ESC), coût élevé	Complexe à haute vitesse, consommation élevée, poids élevé	Rotation limitée (sauf pour les servos 360°) et lente
Application	Jouets, outils électriques, petits appareils	Drones, ventilateurs...	Imprimantes 3D, CNC, robots	Bras robotiques, articulations, roues de contrôle de direction

MOTEURS



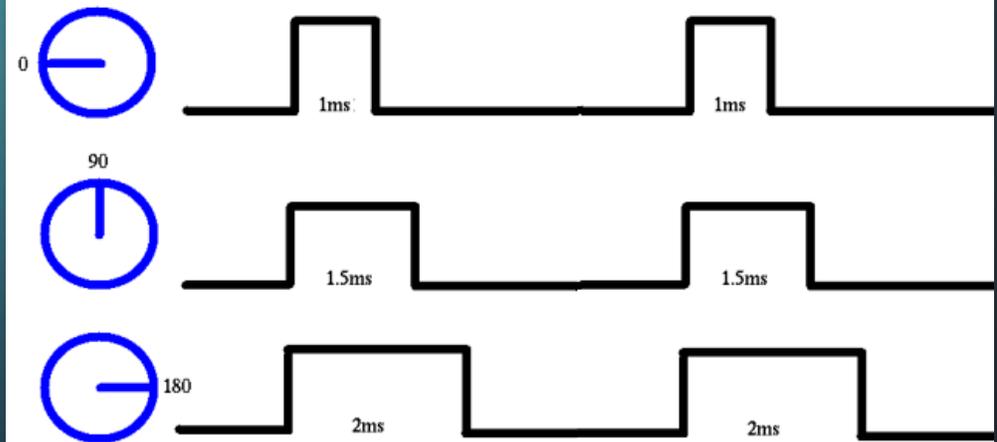
PWM



$$\text{Duty cycle (\%)} = 100 \times \frac{\text{Pulse width}}{\text{Period}}$$

1 ms → 1 kHz

1 us → 1 MHz





**MERCI DE VOTRE
ATTENTION**



CONTROL INPUT

- Joystick : 2 x potentiometer, 2 x HW-504
 - `analogRead();` x 2
- Rotary encoder
 - Library « `RotaryEncoder` »
- Bouton poussoir
 - `digitalRead();`
- Switch
 - `digitalRead();`



INTERACTION ENVIRONNEMENT -HUMAIN

- Motion sensor : HC-SR01
 - `digitalRead()`;
- Microphone analogique : MAX9814, FC04
 - `analogRead()`;
- RFID : RC552
 - I2C et SPI

Distance

- Ultrasonic : HC-SR04, range 2cm – 400cm
 - `PulseIn()`;
- Time-of-flight : GYUL53L0x, range 3cm – 15m
 - I2C

MESURER L'ENVIRONNEMENT

- Capteur capacitif humidité sol : Moisture Sensor v1.2 (super)
 - `analogRead()`
- Capteur température : ds18b20
 - `analogRead();`
- Capteur humidité d'air : DHT21 /22
 - Library DHT (pour utiliser le protocole)
- Capteur 9 axes : MPU9250 x, y, z, rot(x), rot(y), rot(z), champ magnétique 3D
 - I2C
- Capteur poids : HX711
 - Library HX711

ACTUATEUR/OUTPUT

- LCD
 - Soit I2C, soit protocole propre
- Servo : Moteur qui tourne à valeur fixe ; limité à 180° ou 360°
 - `analogWrite()`
- Stepper à utiliser avec pont-H (tb6612fng)
 - `analogWrite()` pour vitesse, `digitalWrite()` pour control
- Pompe, câble chauffant, actuateur linéaire etc...