Arduino: une carte électronique aux multiples possibilités

8 Mars 2017

K. Sartor (Laboratoire de Thermodynamique ULg)

Licence CC-BY-NC-SA

Présentation

- Carte électronique programmable
  - µC (micro-controleur) Atmel AVR
    - Entrées/sorties digitales/analogiques
    - 5V (3.3V fonction des versions)
    - « temps réel » travaille à la µs
- Large gamme

Licence CC-**BY-NC-ND**

Source: toulouse.labo-robotique.com

Présentation (suite)

- Logiciel de programmation
  - open-source
  - Multi plateforme

- Prix
  - ~ 30 € pour Arduino R3 UNO officiel
  - Clones moins cher de -5 à -90 % (qualité?)

Licence CC-**BY-NC-ND**

Présentation (suite)

- Nombreux « shields » pour étendre les possibilités
  - Ex: Ethernet + SD Card

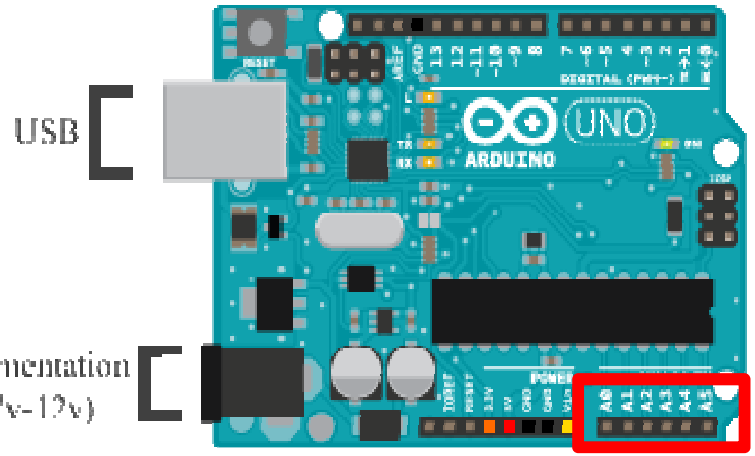
- Breakout
  - Ex: Radio fréquence

Source: mysensors

Licence CC-**BY-NC-ND**

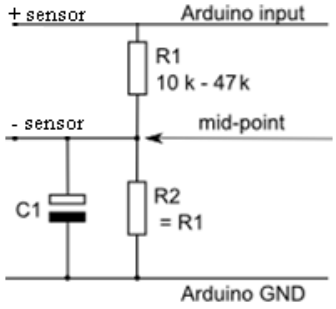
Mesures analogiques

- 6 entrées
  - (jusqu'à 16)



- Résolution ~ 5 mV (10 bits)

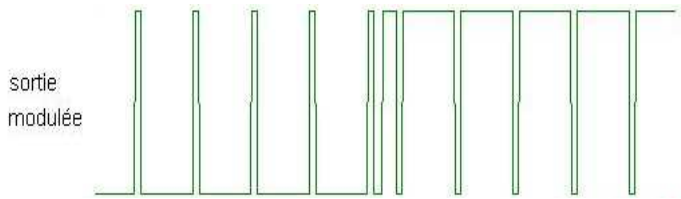
- Plage de mesures:
  - 0-5VDC nativement
  - 0-X VDC via un pont diviseur (perte de sensibilité)
  - -X à +X VDC via « biasing voltage divider »



Licence CC-**BY-NC-ND**

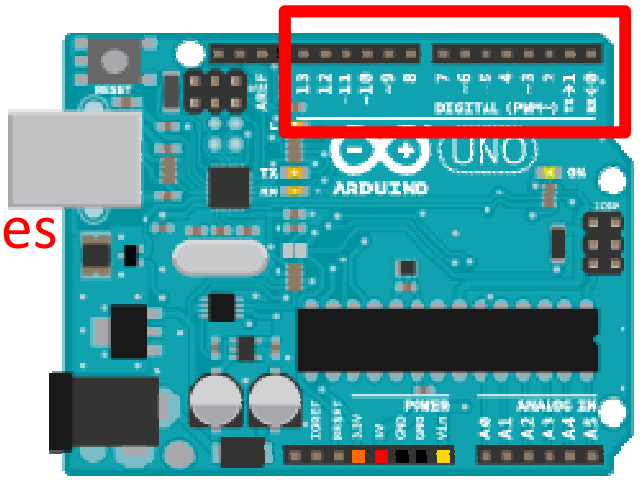
I/O digitales

- 14 à 54
  - soit entrée soit sortie
  - PWM de 4 à 15 (~)



- Lire un état haut (5V) ou un état bas (0V)
  - Plages de valeurs

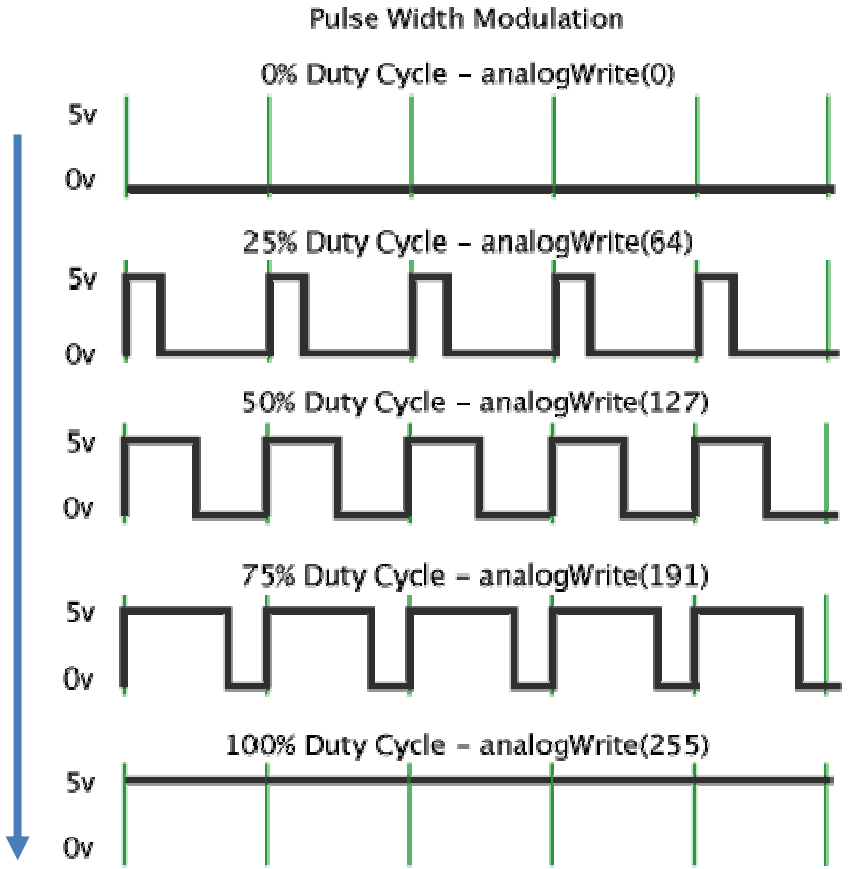
- Imposer un état haut ou bas
  - /!\ à la puissance de sortie (40 mA)
    - R ≥ 470 Ω => ~10 mA
  - Max 200 mA pour l'ensemble des sorties



Licence CC-**BY-NC-SA**

PWM

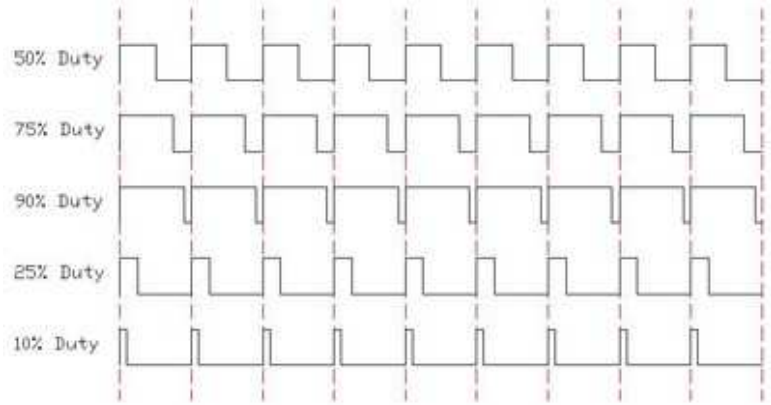
- Hacher la tension
  - Led, moteur,…
  - Via Transistor,MOSFET



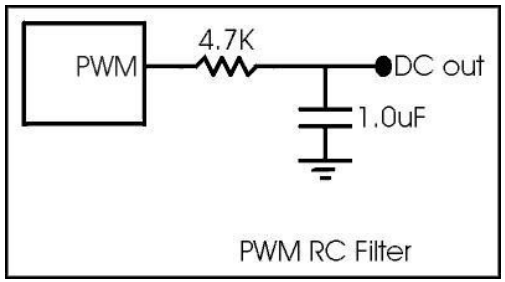
Licence CC-**BY-NC-SA**

« Sortie Analogique »

- 0 à 5V via PWM
  - 256 niveaux



  - Ajout d'un filtre RC au besoin

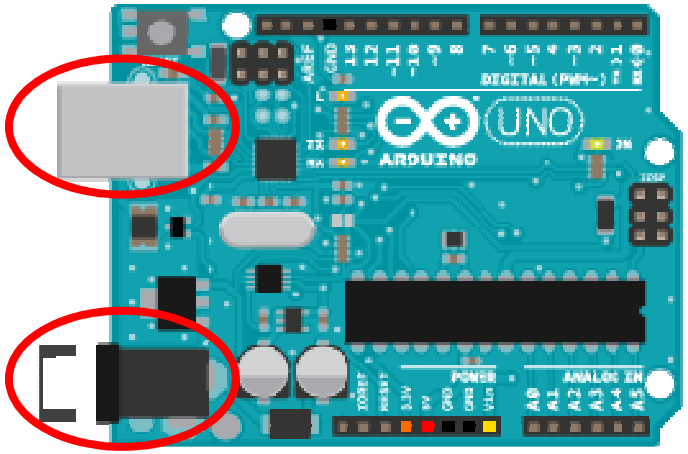


Source: https://arduino-info.wikispaces.com/Analog-Output

Licence CC-**BY-NC-ND**

Alimentation

- Directement depuis un câble USB
  - Si USB (PC) max 500 mA
  - Si externe, voir information sur la batterie / transformateur avec un maximum de 1000 mA
- Alimentation externe 7-12V



- Pin 5V Imax – 50 mA pour le µC
- Pin 3.3V – 150 mA

=> 450 mA max (USB) pour les sorties, les rails 5 et 3.3V

Licence CC-**BY-NC-ND**

Communication (limitée)

- Port Série (digital)
  - Suite de 1 et 0 logique
  - Via l'USB ou les pins RX/TX
- Sorties digitales
- Extension possibles
  - WiFi, Ethernet, Bluetooth, GSM,RF
  - Ecran LCD
    - I2C: 4 pins au lieu de 14

Licence CC-**BY-NC-ND**

Idées de projets

- Mesures de température, humidité, pression, luminosité, gaz,…
- Fréquencemètre, compteur impulsion,… (compteur énergie, vitesse,….)
- Pilotage de sorties digitales (relais, servo, jeux de lumières et son, …)
- N'importe quel moteur de recherche… vous en proposera
  - « Useless machine »
- Limites: la puissance et la mémoire de la carte

Licence CC-**BY-NC-ND**

Programmation

- Séquentielle (A puis B puis C) et évènementielle
- Via un logiciel de programmation
  - Windows, Linux, Mac
- Langage typé C/C++
- Nombreuses librairies disponibles
  - Capteur, (servo-)moteur,…

Licence CC-**BY-NC-ND**

Guide et références

– https://www.arduino.cc/en/Reference/HomePage

Structure

- setup()
- loop()

Control Structures

- if
- if...else
- for
- switch case
- while
- do... while
- break
- continue
- return
- goto

Further Syntax

- ; (semicolon)
- {} (curly braces)
- // (single line comment)
- /* */ (multi-line comment)

Variables

Constants

- HIGH | LOW
- INPUT | OUTPUT | INPUT_PULLUP
- LED_BUILTIN
- true | false
- integer constants
- floating point constants

Data Types

- void
- boolean
- char
- unsigned char
- byte
- int
- unsigned int
- word
- long
- unsigned long

Functions

Digital I/O

- pinMode()
- digitalWrite()
- digitalRead()

Analog I/O

- analogReference()
- analogRead()
- analogWrite() - *PWM*

Due & Zero only

- analogReadResolution()
- analogWriteResolution()

Advanced I/O

- tone()
- noTone()
- shiftOut()
- shiftIn()
- pulseIn()

Lire une tension et allumer une led en fonction de la valeur

```
// Variables
const int ledPin =  13;
const int analogPin = 0;
float voltage = 0;

// Configuration
void setup() {
  Serial.begin(9600);
  digitalWrite(ledPin, LOW);
}

//Boucle principale
void loop() {
  int sensorValue = analogRead(analogPin);
  voltage = sensorValue * (5.0 / 1023.0);
  Serial.println(voltage);
  if (voltage >0.99){
    digitalWrite(ledPin, HIGH);
  }else{
    digitalWrite(ledPin, LOW);
  }
  delay(1000);
}
```

Licence CC-**BY-NC-SA**

Les fonctions

- Lisibilité du code
- Création de fonctions génériques utilisables ailleurs
- Évolution du code plus facile

Licence CC-**BY-NC-SA**

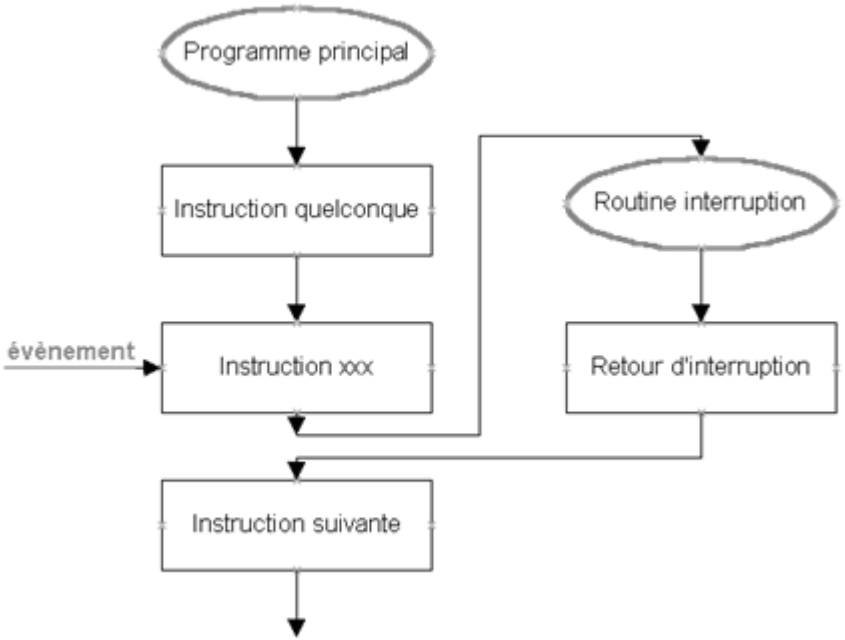
Les fonctions: exemple

```
// Variables
const int ledPin =  13;
const int analogPin = 0;
float voltage = 0;

// Configuration
void setup() {
  Serial.begin(9600);
  digitalWrite(ledPin, LOW);
}

void check_voltage(float voltage, int ledPin){
  if (voltage >0.99){
    digitalWrite(ledPin, HIGH);
  }else{
    digitalWrite(ledPin, LOW);
  }
}
//Boucle principale
void loop() {
  int sensorValue = analogRead(analogPin);
  voltage = sensorValue * (5.0 / 1023.0);
  Serial.println(voltage);
  check_voltage(voltage, ledPin);
  delay(1000);
}
```

Interruption

```
//Boucle principale
void loop()
  int sensorValue = analogRead(analogPin);
  voltage = sensorValue * (5.0 / 1023.0);
  Serial.println(voltage);
  check_voltage(voltage, ledPin);
  delay(1000);
}
```

Evènement



Licence CC-**BY-NC-SA**

Interruption

- Lorsqu'une pin digitale passe d'un état à un autre*, on exécute une fonction avant tout autre

  *Types de déclenchement

  - **LOW** : le pin est à un état bas
    - (HIGH seulement sur Due, Zero, MKR1000)
  - **RISING** : le pin passe d'un état bas à haut
  - **FALLING** : le pin passe d'un état haut à bas
  - **CHANGE** : le pin change d'état (les deux précédents)

Licence CC-**BY-NC-ND**

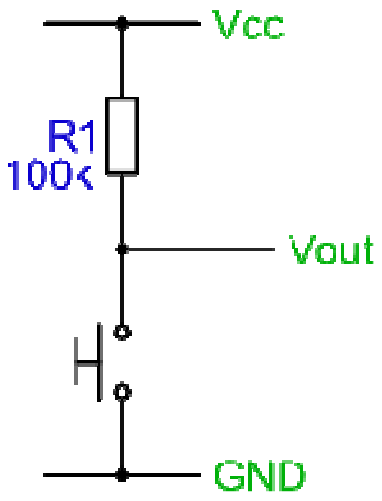
Interruption: limitations

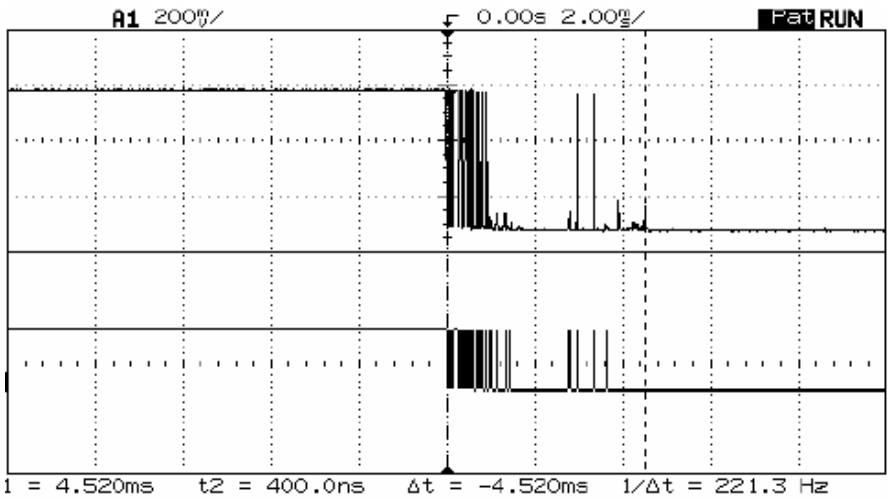
- Dans la fonction liée à l'interruption,
  - compteur millis() ne s'incrémentera pas
  - delay() ne fonctionnera pas
  - déconseillé d'utiliser des fonctions liée au temps
    - Transmission série, I2C

Licence CC-**BY-NC-SA**

Rebond (bounce)

- Problème récurrent avec les relais et switch





Source: A Guide to Debouncing - Jack G. Ganssle

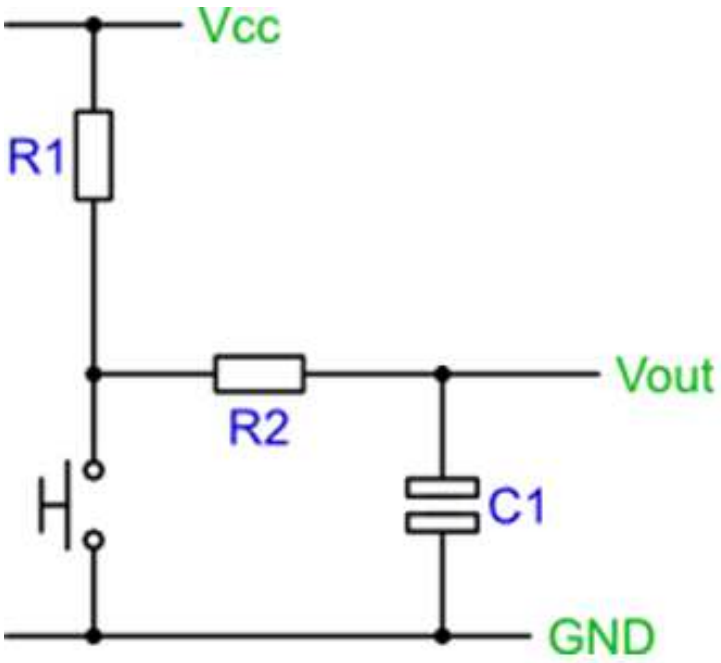
Licence CC-**BY-NC-ND**

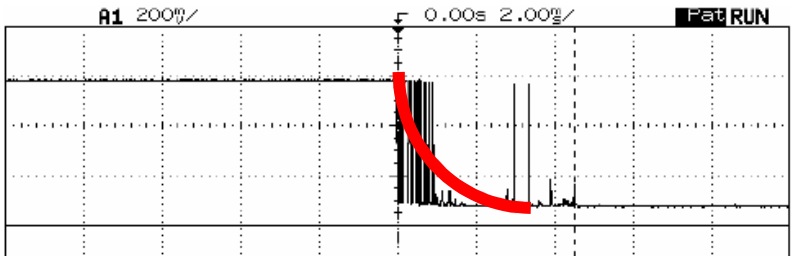
Rebond: approche logicielle

```
volatile unsigned long last_micros=10000;
long debouncing_time = 100; //Debouncing Time in Milliseconds
void debounce() {
  if((long)(micros() - last_micros) >= debouncing_time * 1000) {
    // votre code
    last_micros = micros();
  }
}
void setup() {
  attachInterrupt(digitalPinToInterrupt(interruptPin), debounce  , FALLING);
}
```

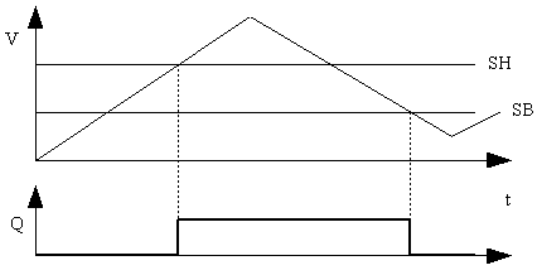
Licence CC-**BY-NC-SA**

Rebond: approche matérielle





R1 > 1 kΩ
R2 = 18 kΩ
C1 = 1 µF
Debounce time= 10ms

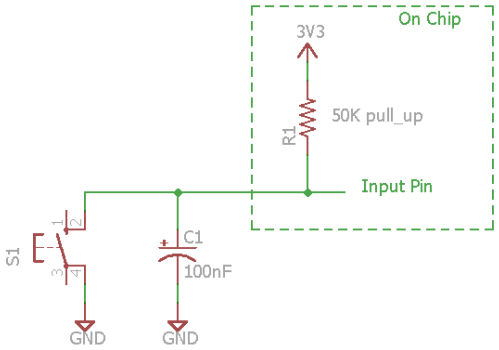


Bascule de Schmidt

Licence CC-**BY-NC-SA**

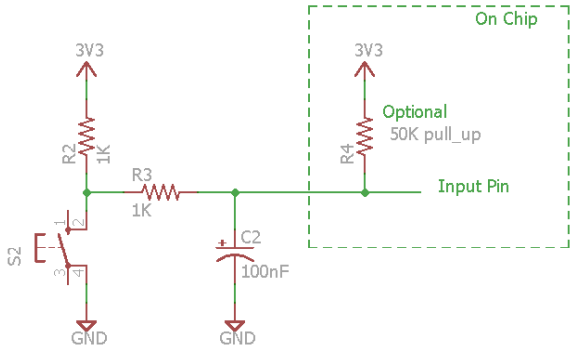
Autres circuits

Circuit 1



Active Low with Edge Triggering
Rising Edge Only : 2.3mSec @ 1.25V

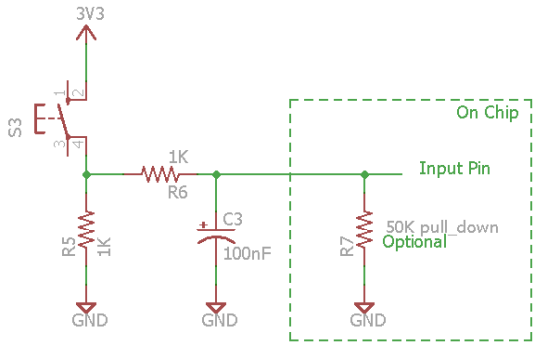
Circuit 2



Active Low filter with Edge Triggering
Falling Edge : 43nSec @ 1.16V
Rising Edge : 95nSec @ 1.25

With R3 = 10K, multiplies the R/C by
a factor of 10, to approx. 0.5mSec for both edges

Circuit 3



Active High filter with Edge Triggering
Rising Edge : 48nSec @ 1.25V
Falling Edge : 87nSec @ 1.16V

With R6 = 10K, multiplies the R/C by
a factor 10, to approx. 0.5 mSec for both edges

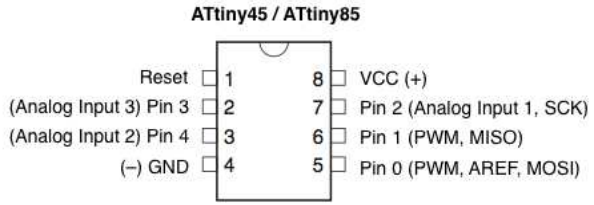
http://protological.com/debounce-calaculator/
http://www.ganssle.com/debouncing-pt2.htm

Source: https://www.raspberrypi.org/forums/viewtopic.php?t=134394
Complément d'information: http://www.labbookpages.co.uk/electronics/debounce.html

Licence CC-**BY-NC-SA**

Alternatives

- ESP8266
  - WiFi intégré
  - Nombreuses librairies compatibles
  - 1 entrée analogique 0 - 1V
    - Multiplexeur jusque 5V, hardware (diode) 1V
- ATtiny



- RPi
  - Ordinateur (nativement uniquement I/O digitales)
    - Modules complémentaires
  - N'est pas temps réel

Licence CC-**BY-NC-ND**

Ardublock

- Plugin pour programmer en blocs de function

[1]

- http://blog.ardublock.com (Anglais)
- [1] http://www.semageek.com/arduino-presentation-et-traduction-en-francais-de-ardublock/ (Français)

Licence CC-**BY-NC-ND**

Les petites astuces

- Empêcher la réinitialisation à chaque ouverture du port série:
  - http://playground.arduino.cc/Main/DisablingAutoResetOnSerialConnection

- Augmenter la fréquence I/O digitale (120 kHz à 2,4MHz), Compensation des mesures analogiques (dûe à la tension d'alimentation)
  https://www.codeproject.com/tips/987180/arduino-tips-tricks (anglais)

Licence CC-**BY-NC-ND**

Sources

- Sauf mention contraire,
  - Les images proviennent du site arduino.cc, de Wikipédia , des datasheets des composants ou ont été réalisée par l'auteur
  - Les informations sur la programmation proviennent du site arduino.cc, de l'aide de « l'IDE » ou des datasheets des composants.

Licence CC-**BY-NC-ND**

Merci pour votre attention

- Questions ?
- Remarques ?
- Commentaires ?

Licence CC-**BY-NC-SA**