Travaux pratiques Informatique Embarquée

Université de Toulon
IUT GEII Première année

UNIVERSITÉ DE TOULON

GEII IUT DE TOULON

TPs Informatique Embarquée

frank.buloup@univ-amu.fr

Séances de TPs

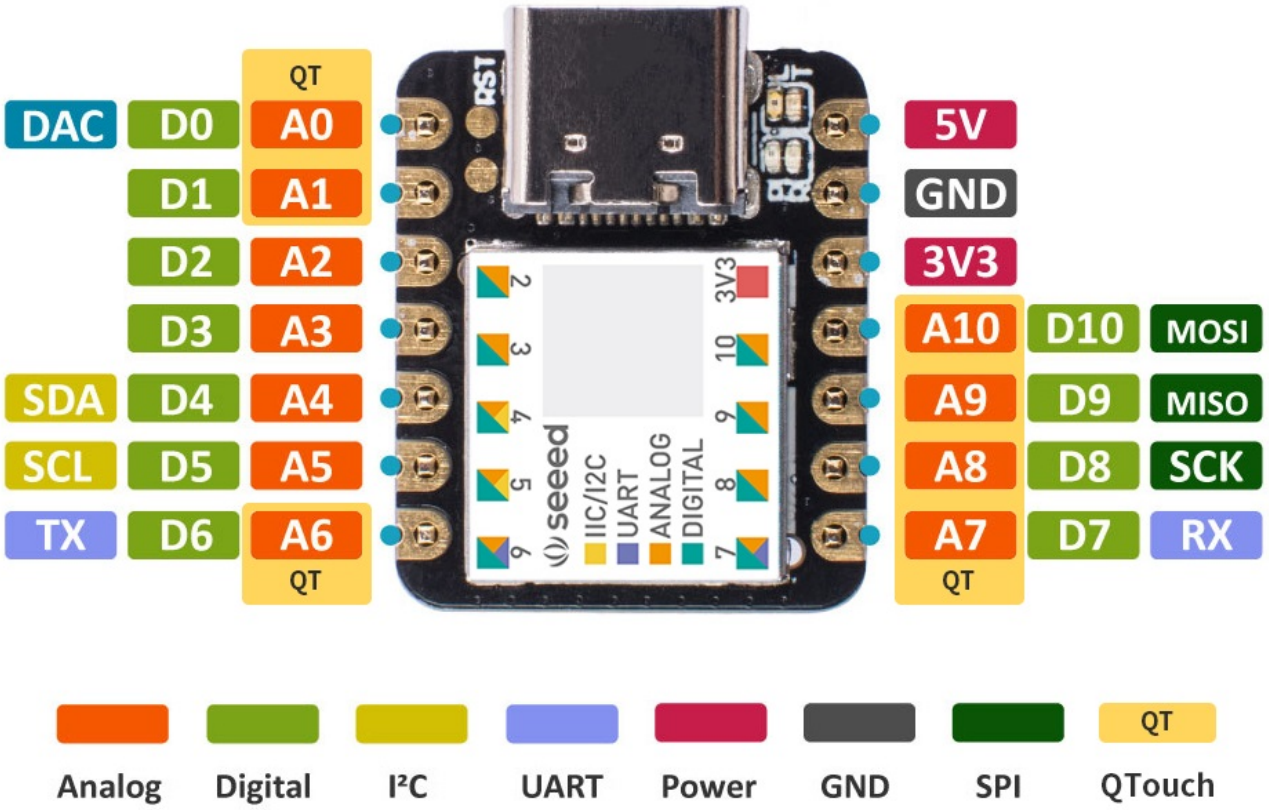
- Mardi aprèm
- 9 de 3h (13h30-16h30) et 2 de 1h30 (13h30-15h)
- Travail en monômes
- Vous pouvez vous aider et demander de l'aide !

Il y a des règles durant ces séances !

- Si pas de carte alors pas de TP !
- Pas de téléphone : éteint dans le sac
- Pas d'IA durant les séances
- Lors de l'évaluation, vous n'aurez pas accès aux outils d'IA

2 La carte XIAO SAMD21 et sa programmation

- Vue d'ensemble de la carte
- Le Main
- Les entrées/sorties numériques
- Les entrées/sorties analogiques
- Système temps réel



Arduino language Ref.

Le Main

```
#include <Arduino.h>

// Declared weak in Arduino.h to allow user redefinitions.
int atexit(void (* /*func*/ )()) { return 0; }

// Weak empty variant initialization function.
// May be redefined by variant files.
void initVariant() __attribute__((weak));
void initVariant() { }

void setupUSB() __attribute__((weak));
void setupUSB() { }

int main(void)
{
    init();

    initVariant();

#if defined(USBCON)
    USBDevice.attach();
#endif

    setup();

    for (;;) {
        loop();
        if (serialEventRun) serialEventRun();
    }

    return 0;
}
```

La fonction **loop()**

- Le nom **loop()** est mal choisi !
- C'est un "faux ami"
- C'est une fonction pas une boucle !
- Mais cette fonction est bien appelée dans une boucle **for(;;)**
- Et cette boucle n'a pas de fin d'exécution : boucle infinie !

digitalWrite()

Last revision • 23/04/2025

Description

Write a `HIGH` or a `LOW` value to a digital pin.

If the pin has been configured as an `OUTPUT` with `pinMode()`, its voltage will be set to the corresponding value: 5V (or 3.3V on 3.3V boards) for `HIGH` and 0V (ground) for `LOW`.

If the pin is configured as an `INPUT`, `digitalWrite()` will enable (`HIGH`) or disable (`LOW`) the **internal pull-up** on the input pin. It is recommended to set the `pinMode()` to `INPUT_PULLUP` to enable the internal pull-up resistor. See the **Digital Pins** tutorial for more information.

If you do not set the pin as an `OUTPUT`, and connect an LED to it, when calling `digitalWrite(pin, HIGH)`, the LED may appear dim. Without explicitly setting `pinMode()`, `digitalWrite()` will have enabled the internal pull-up resistor, which acts like a large current-limiting resistor.

digitalRead()

Last revision • 23/04/2025

Description

Reads the value from a specified digital pin, either `HIGH` or `LOW`.

Syntax

Use the following function to read the value of a digital pin:

```
digitalRead(pin)
```

analogWrite()

Last revision • 09/05/2025

Description

Writes an analog value (**PWM wave**) to a pin. Can be used to light a LED at varying brightness or drive a motor at various speeds. After a call to `analogWrite()`, the pin will generate a steady rectangular wave of the specified duty cycle until the next call to `analogWrite()` (or a call to `digitalRead()` or `digitalWrite()`) on the same pin.

Check your board pinout to know which are the officially supported PWM pins. While some boards have additional pins capable of PWM, using them is recommended only for advanced users that can account for timer availability and potential conflicts with other uses of those pins.

In addition to PWM capabilities some boards have true analog output when using `analogWrite()` on the `DAC` marked pins. Check your board pinout to find out if the DAC is available.

analogRead()

Last revision • 09/05/2025

Description

Reads the value from a specified analog input pin.

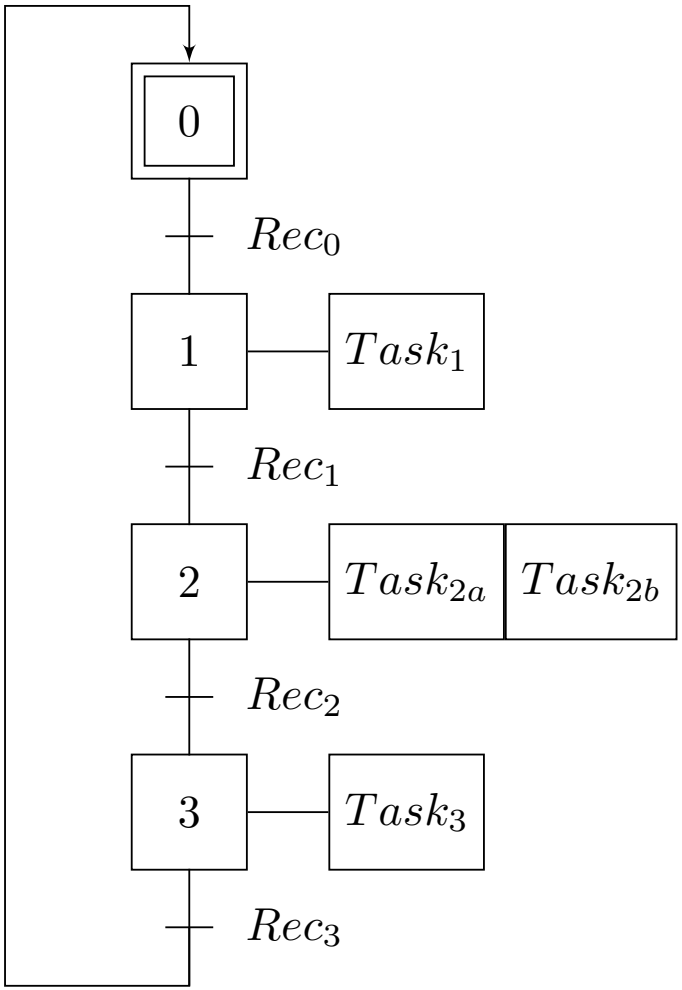
An Arduino UNO, for example, contains a multichannel, 10-bit analog to digital converter (ADC). This means that it will map input voltages between 0 and the operating voltage (+5 VDC) into integer values between 0 and 1023. This yields a resolution between readings of: 5 volts / 1024 units or 0.0049 volts (4.9 mV) per unit.

The voltage input range can be changed using **analogReference()**. The default `analogRead()` resolution on Arduino boards is set to 10 bits, for compatibility. You need to use **analogReadResolution()** to change it to a higher resolution.

3 Exemples d'implémentation d'un GRAFCET

- Exemple général
- Application TP1

Exemple général




```
// ...<-Acquisition des entrées
Rec0 = ...;
Rec1 = ...;
Rec2 = ...;
Rec3 = ...;

if(currentState == state0 && Rec0) currentState = state1;
if(currentState == state1 && Rec1) currentState = state2;
if(currentState == state2 && Rec2) currentState = state3;
if(currentState == state3 && Rec3) currentState = state0;

switch(currentState) {
    case state1:
        task1();
        break;
    case state2:
        task2A();
        task2B();
        break;
    case state3:
        task3();
        break;
    default:
        taskDefault();
        break;
// ...
}
// ...
void taskDefault() {
    // ...
}
void task1() {
    //...
}
//...
```

Application TP1