

Faire Fonctionner FreeRTOS sur Arduino.



- ❖ **Étape n°1** : Installer logiciel permettant d'envoyer les programmes sur l'Arduino via ce lien.

- <https://www.arduino.cc/en/Main/Software>.

- ❖ **Étape n°2** : Programme Test

Crée un nouveau projet, ensuite la première chose à faire c'est de Compiler notre programme.

- Assurez-vous d'avoir un message similaire pour montrer que la compilation est terminée.

```
Le croquis utilise 444 octets (1%) de l'espace de stockage de programmes. Le maximum est de 32256 octets.  
Les variables globales utilisent 9 octets (0%) de mémoire dynamique, ce qui laisse 2039 octets pour les variables locales. Le maximum
```

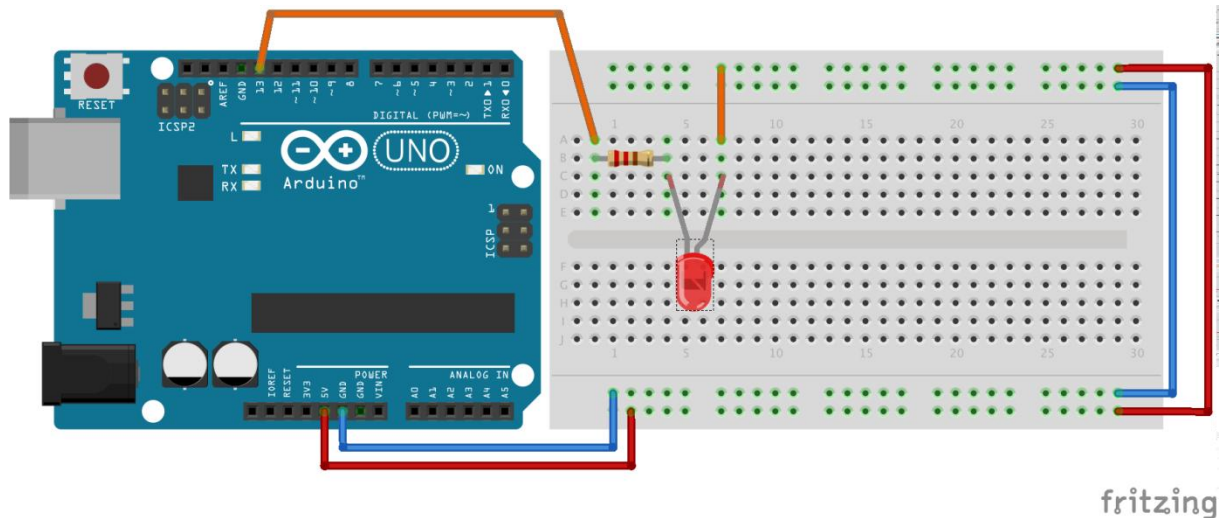
- Ensuite une fois cette sous tâche réaliser, exécuter un programme test pour s'assurer que votre matériel est bien fonctionnel ou détecter divers problèmes.

Ex de programme test : (faire clignoter une LED sur le port 13)

- ✓ Ne pas oublier de compiler.

```
//declaration des sorties  
int Led_1=13;  
      //Configuration de la carte  
void setup() {  
    pinMode(Led_1,OUTPUT);  
}  
//Debut de notre Programme  
void loop() {  
    digitalWrite(Led_1,LOW);  
    delay(1000);  
    digitalWrite(Led_1,HIGH);  
    delay(1000);  
}
```

- Réaliser le montage suivant ci-dessous. Et alimenter votre carte grâce au câble USB fournit avec la carte Arduino.



(Source de l'image : <https://sdz-upload.s3.amazonaws.com/prod/upload/LED5VPin13.png>)

- Une fois le montage réalisé, ne pas oublier de sélectionner la bonne carte Arduino sous rubrique Outils/Type de carte.

(NB: pour les personnes qui ne savent pas quel est le nom de leur matériel il est possible de récupérer les informations sur la carte sous la rubrique:

Outils/Récupérer les informations de la carte.)

- Assurer le bon fonctionnement du programme (vérifier que la LED clignote, si la LED ne clignote pas veuillez revoir le montage).

❖ **Étape n° 3:** FreeRTOS sur Arduino

- Télécharger FreeRTOS via le lien suivant.
(https://github.com/feilipu/Arduino_FreeRTOS_Library).
Ne pas oublier de l'inclure dans l'IDE: Croquis/Inclure une bibliothèque/Ajouter la bibliothèque .Zip (sélectionner le fichier téléchargé).
- S'assurez que vous avez la dernière version de FreeRTOS (v9.0.0-1).
Croquis/ Inclure une bibliothèque/ Gérer les bibliothèques/ Rechercher FreeRTOS. (Vérifier la version indiquée).
- Revenez au nouveau projet vide placer vous avant le void setup(). Faites par la suite croquis/ inclure une bibliothèque/ sélectionner Free RTOS.

- Vérifier que vous obtenez ceci.

```
#include <Arduino_FreeRTOS.h>
void setup() {
    // put your setup code here, to run once:
}
void loop() {
    // put your main code here, to run repeatedly:
}
```

- ❖ **Étape n°4 :** Réaliser le programme.

Compléter le projet avec le code suivant;

```
#include <Arduino_FreeRTOS.h>
void TaskBlink( void *pvParameters );
void TaskAnalogRead( void *pvParameters );
void setup(){
    xTaskCreate(
        TaskBlink
        , (const portCHAR *)"Blink"
        , 128
        , NULL
        , 2
        , NULL );
    xTaskCreate(
        TaskAnalogRead
        , (const portCHAR *) "AnalogRead"
        , 128 // Stack size
        , NULL
        , 1 // Priority
        , NULL );
}
```

(Ne pas oublier de compiler après chaque ligne pour ne pas faire d'erreur et s'assurer que le code se construit correctement).

Maintenant la fin du programme.

```

void loop()
{

}

/*-----*/
/*----- Taches -----*/
/*-----*/
void TaskBlink(void *pvParameters){
    (void) pvParameters;
    pinMode(13, OUTPUT);
    while(1){
        digitalWrite(13, HIGH);
        vTaskDelay( 2000 / portTICK_PERIOD_MS );
        digitalWrite(13, LOW);
        vTaskDelay( 2000 / portTICK_PERIOD_MS );
    }
}

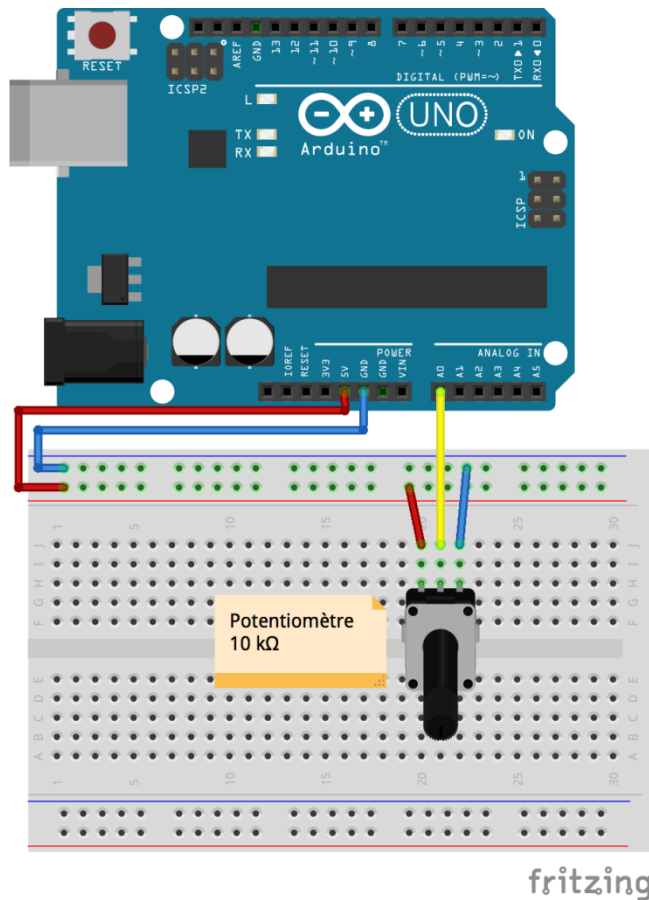
void TaskAnalogRead(void *pvParameters){
    (void) pvParameters;
    Serial.begin(9600);
    while(1){
        int sensorValue = analogRead(A0);
        Serial.println(sensorValue);
        vTaskDelay(1);
    }
}

```

(Ne pas oublier de compiler après chaque ligne pour ne pas faire d'erreur et s'assurer que le code se construit correctement).

❖ **Étape n°5 :** faire le montage suivant.

Tout en complément avec le montage de départ (celui avec la LED sur le port 13).
Réaliser maintenant ce schéma sans intervenir sur l'autre schéma.



❖ **Étape n°6:** Compiler (Compilation sans erreur), téléverser le programme sur votre carte Arduino.

Et vérifier le bon fonctionnement de votre programme que la LED clignote, ainsi que changer la valeur du potentiomètre. (Ne pas oublier d'ouvrir le terminal série qui vous indiquera que la valeur de changement du potentiomètre).



Fin de la manipulation