 **Warning:** Trying to access array offset on value of type null in **/home/clients/ee3f943e731fc0a12d3400116692186f/html/wp-content/plugins/unyson/framework/includes/option-types/typography-v2/class-fw-option-type-typography-v2.php** on line **148**

Warning: foreach() argument must be of type array|object, null given in **/home/clients/ee3f943e731fc0a12d3400116692186f/html/wp-content/plugins/unyson/framework/includes/option-types/typography-v2/class-fw-option-type-typography-v2.php** on line **148**

Warning: Trying to access array offset on value of type null in **/home/clients/ee3f943e731fc0a12d3400116692186f/html/wp-content/plugins/unyson/framework/helpers/general.php** on line **1275**

Warning: foreach() argument must be of type array|object, null given in **/home/clients/ee3f943e731fc0a12d3400116692186f/html/wp-content/plugins/unyson/framework/helpers/general.php** on line **1275**



SMART BUD

Demo (<https://bud.eco-sensors.ch>) | Tutoriels (<https://eco-sensors.ch/tutoriels/>)



NOUS SOUTENIR

Shop (<https://eco-sensors.ch/shop/>) & Sponsoring (<https://eco-sensors.ch/product-category/dons/>)



SOCIAL

Facebook (<https://www.facebook.com/ecosensors>) - Contact (<https://eco-sensors.ch/contact/>)

Un Joystick pour orienter un petit robot avec LoRa

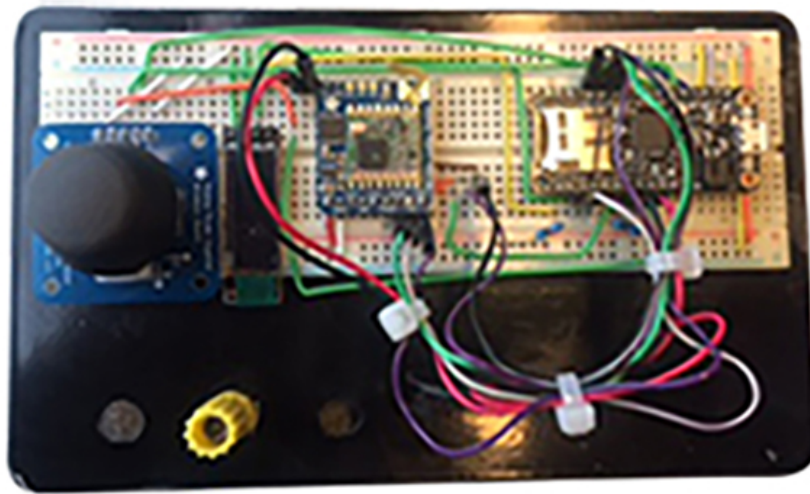
HOME (HTTPS://ECO-SENSORS.CH/)
/ LORAWAN / LORA (HTTPS://ECO-SENSORS.CH/CATEGORY/LORAWAN/)
/ UN JOYSTICK POUR ORIENTER UN PETIT ROBOT AVEC LORA

BY ECOSSENSORS (HTTPS://ECO-SENSORS.CH/AUTHOR/ECOSSENSORS/)

/

18 DÉCEMBRE 2017 (HTTPS://ECO-SENSORS.CH/UN-JOYSTICK-POUR-ORIENTER-UN-PETIT-ROBOT-AVEC-LORA/)

 0  2,945  0  



Dans cet article, je vais vous résumer comment préparer un Joystick et un mini display pour orienter un petit robot avec la technologie LoRa. Même, si je vous donnerai tout mon code, cet exercice n'a qu'un but expérimentale sur les modules et fonctionnalités proposé. L'acquis sera reporté sur un autre projet. Les suggestions d'experts sont les bienvenues pour améliorer le mode de fonctionnement.

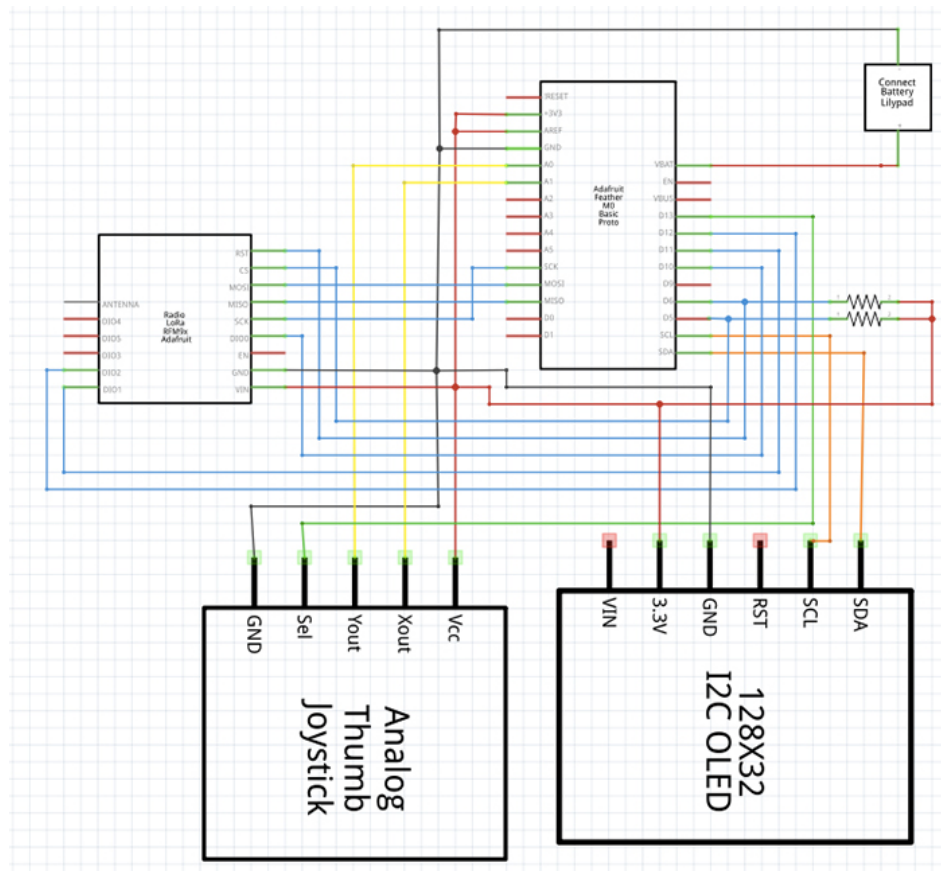
Cet article part du principe que vous connaissez l'IDE Arduino (<https://www.arduino.cc/en/Main/Software>) , C++ et comment installer des librairies (<https://www.arduino.cc/en/Guide/Libraries>)

❶ LES FONCTIONS `SI.SPRINTLN(F(« COUCOU »),2)` OU `SI.SPRINT(F(« COUCOU »),2)` PEUVENT ÊTRE REMPLACÉES PAR `SERIAL.PRINTLN(F(« COUCOU »))` OU `SERIAL.PRINT(F(« COUCOU »))`

Matériel

- Adafruit MO adalogger (<https://www.adafruit.com/product/2796>) (La carte SD n'est pas utilisée dans cet article)
- Joystick (<https://www.adafruit.com/product/512>)
- RFM95 Radio 868Mhz (<https://www.adafruit.com/product/3072>)
- Oled I2C 128×32 0.96 LCD Display

Connexion



Préparatifs

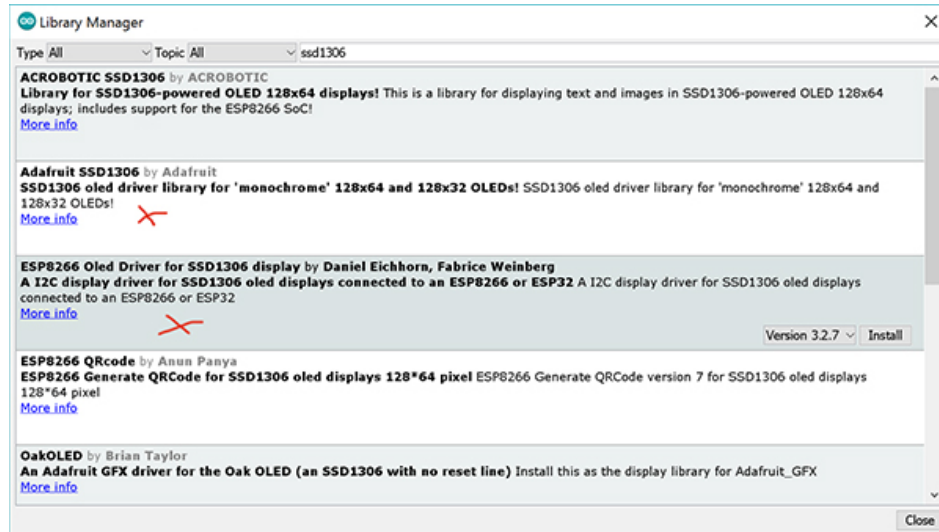
OLED LCD Display

Référence : <https://projetsdiy.fr/ssd1306-mini-ecran-oled-i2c-128x64-arduino/> (<https://projetsdiy.fr/ssd1306-mini-ecran-oled-i2c-128x64-arduino/>)

D'abord, il faut préparer votre IDE. Adafruit (<https://www.adafruit.com/>) a développé une librairie très puissante qui

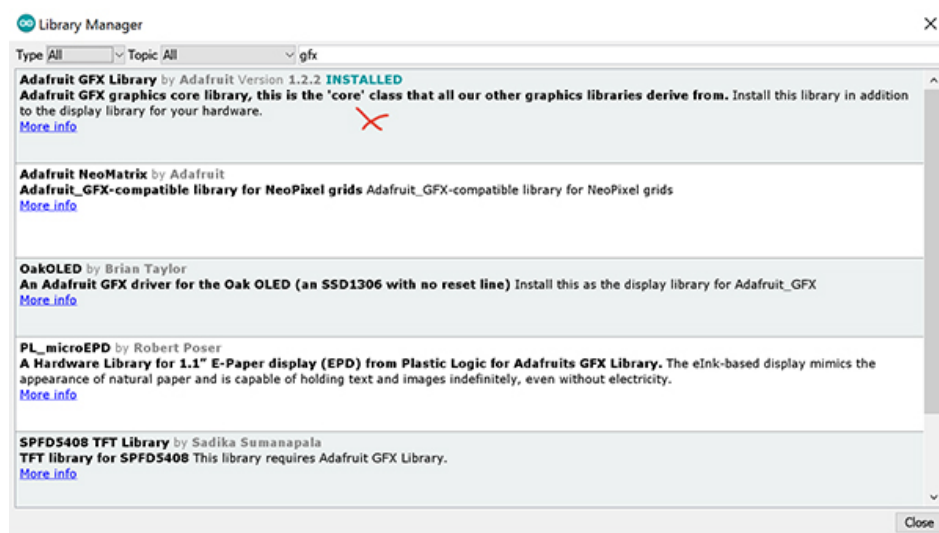
va nous permettre de gérer l'affichage de notre mini écran mais aussi de tracer plein de chose très facilement grâce à la librairie dédiée, GFX Library

Ouvrer votre IDE puis le gestionnaire des librairies et installez les deux librairies comme indiqué sur l'image



Adafruit SSD1306 & ESP8266 Oled Drivers

Rechercher maintenant la librairie Adafruit GFX et installez-là encore.



Adafruit GFX Library

Fermez et ré-ouvrez votre IDE.

Dans votre nouveau fichier.ino, vous devrez inclure ces librairies:

1 #include < SPI.h > // I2C
2 #include < Wire.h > // I2C
3 #include < Adafruit_GFX.h >
4 #include < Adafruit_SSD1306.h >
5 Adafruit_SSD1306 display;

⚠ SUPPRIMER LES ESPACES APRÈS LES < ET AVANT LES >

Puis dans fonction setup() (<https://playground.arduino.cc/French/Setup>), vous devez ajouter ceci pour afficher du texte:

```
1 /*
2 * Init the display
3 */
4 display.begin(SSD1306_SWITCHCAPVCC, 0x3C); //
5 // init done
6 display.display();
7 delay(2000);
8 display.clearDisplay(); // Clear the buffer.
9 display.setTextColor(WHITE); // Text color
10 display.setCursor(0,0); // Move the pointer/cu
11 display.println("Welcome aboard SmartIdea!");
12 display.display(); // Print to LCD
13 display.clearDisplay(); // empty the buffer b
```

Le Joystick

Référence: <https://learn.adafruit.com/joy-controller-feather/software> (<https://learn.adafruit.com/joy-controller-feather/software>)

La pin Xout vous retournera une valeur 0 à 1023 (ADC). Idem pour l'axe Y, une valeur pourra être entre 0 et 1023 pour être mesurée à sortie Yout.

Cependant, pour le Joystick que j'utilise, les valeurs minimales et maximales mesurées sont de 13 et 1012 alors que le Joystick se trouvait en butée en haut, en bas, à gauche et à droite. Si vous utilisez un autre Joystick, il faudrait mesurer ces valeurs.

Xout affichera 12 quand le joystick poussé en butée vers l'avant. Il affichera 1012 quand il est poussé en butée vers le bas (vers vous).

Yout affichera 12 quand le joystick poussé en butée la gauche. Il affichera 1012 quand il est poussé en butée vers la droite.

SIG est LOW quand il est pressé, si non il est HIGH

Avant la fonction `setup()` (<https://playground.arduino.cc/French/Setup>), vous devez définir une variable de type struct (<https://playground.arduino.cc/Code/Struct>) qui contiendra les valeurs liées à l'état du Joystick

```
1 struct { // Joystick axis structure (2 axes per joystick)
2   int8_t pin; // Analog pin when joystick axis is pressed
```



```

3   int lower; // Typical value in left/upper position
4   int upper; // Typical value in right/lower position
5   uint8_t key1; // Key code to send when left, right, up, down
6   uint8_t key2; // Key code to send when down, left, right, up
7   int value; // Last-read-and-mapped value (0-255)
8   int8_t state;
9 } axis[] = {
10  {A0, MAX_PULSE, MIN_PULSE, KEY_UP_ARROW, KEY_DOWN_ARROW},
11  {A1, MIN_PULSE, MAX_PULSE, KEY_LEFT_ARROW, KEY_RIGHT_ARROW},
12 }; #define N_AXES (sizeof(axis) / sizeof(axis_t))

```

des pins analogiques, avec la fonction `analogReadResolution()` (<https://www.arduino.cc/en/Reference/AnalogReadResolution>). J'ai aussi choisi d'avoir un niveau tension de référence de 3.3V, à la sortie des pins analogiques, en la connectant au 3.3V. Vous devez donc utiliser la fonction `analogReference(EXTERNAL)` (<https://www.arduino.cc/reference/en/language/functions/analog-io/analogreference/>). Etant donné que la carte Feather MO Adalogger (<https://learn.adafruit.com/adafruit-feather-m0-adalogger/overview>) a un processeur ATSAM21G18 ARM Cortex M0 (48 MHz), je dois utiliser la même fonction avec le paramètre `AR_EXTERNAL` `analogReference(AV_EXTERNAL)` (<https://learn.adafruit.com/adafruit-feather-m0-adalogger/adapting-sketches-to-m0#analog-references>). Le `INPUT_PULLUP` (<https://learn.adafruit.com/adafruit-feather-m0-adalogger/adapting-sketches-to-m0#pin-outputs-and-pullups>) est aussi une particularité de la carte. Avec la fonction `pinMode` (<https://playground.arduino.cc/French/PinMode>), vous configurez la pin (broche) spécifiée, pour qu'elle se comporte soit en entrée, soit en sortie.

```
1 analogReadResolution(10); // Vous pouvez remp
2 analogReference(AR_EXTERNAL);
3 pinMode(JX, INPUT_PULLUP);
4 pinMode(JY, INPUT_PULLUP);
5 pinMode(JS, INPUT_PULLUP);
```

Toujours dans la fonction `setup()`, il vous faudra encore initier votre Joystick. Voir ce code (<https://github.com/pierrot10/joystick/blob/master/sender.ino#L147>) sur Github

Ensuite dans la boucle `loop()` (<https://playground.arduino.cc/French/loop>), vous devez ajouter ce code (<https://github.com/pierrot10/joystick/blob/master/sender.ino#L267>) qui va définit la position du Joystick et envoyer son état au module (Reciever) qui va recevoir le paquet LoRa, et orienter son objet en fonction de la position du Joystick.

La radio LoRa

Pour cela, vous aurez besoin d'inclure a librairie RadioHead (<https://github.com/adafruit/RadioHead/archive/master.zip>)

et de déclarer les pin en fonction du schéma, ci-dessus

```
1 // Define pin for feather m0
2 #define RFM95_CS 5
3 #define RFM95_RST 6
4 #define RFM95_INT 10
```

et de déclarer un objet

```
1 RH_RF95 rf95(RFM95_CS, RFM95_INT)
```

Puis dans votre fonction setup(), vous initialisez le module Radio et vous affichez un message sur votre écran LCD.

```

1  /*
2  * Init LoRa / Radio
3  */
4  pinMode(RFM95_RST, OUTPUT);
5  digitalWrite(RFM95_RST, HIGH);
6  while (!rf95.init()) {
7    Si.sprintln(F("LoRa radio init failed"),2);
8    display.print("LoRa radio init failed");
9    display.display();
10   //display.clearDisplay();
11   while (1);
12 }
13
14 Si.sprintln(F("LoRa radio init OK!"),2);
15 //display.setCursor(0,0);
16 display.print("Radio ready"); // Prepare text
17 // Defaults after init are 434.0MHz, modulation
18 // We need 868.0Mhz for Europe
19 if (!rf95.setFrequency(RF95_FREQ))
20 {
21   Si.sprintln(F("setFrequency failed"),2);
22   while (1);
23 }
24
25 Si.sprint(F("Freq at "),2); Si.sprintln(RF95_
26 display.print(" "); // Prepare text to be pin
27 display.print(RF95_FREQ); // Prepare text to
28 display.println("Mhz"); // Prepare text to be
29 display.display(); // Dipslay on LCD

```

```

30 display.clearDisplay(); // Clear cache
31 delay(2000); // Wait 2sec

```

Vous enverrez vos données au module distant (reciever) grâce à ce code

```

1  memset(srfm, '\0', SRFMLEN); // Clean the buffer
2  sprintf(srfm, "snap"); // Copy into srfm
3  r = send_rf95(srfm); // Send the vlaue of srfm
4
5  if(r == 1)
6  {
7      display.println(SENT); // Prepare the text
8  }
9  else if(r == 0)
10 {
11     display.println(NOLISTNER);
12 }
13 else if(r == -1)
14 {
15     display.println(FAILED);
16 }
17
18 display.display();
19 display.clearDisplay();

```


qui va faire appel à la fonction `send_rf95()` (<https://github.com/pierrot10/joystick/blob/master/sender.ino#L548>).

Cet article peut être aussi combiner avec celui-ci [Un raspberry Zero, un LCD et LoRa pour recevoir des données rfm95](https://eco-sensors.ch/un-raspberry-zero-un-lcd-et-lora-pour-recevoir-des-donnees-rfm95/) (<https://eco-sensors.ch/un-raspberry-zero-un-lcd-et-lora-pour-recevoir-des-donnees-rfm95/>)

Conclusion et code complet

Cette page ne vous permettra pas de mettre en place ce Joystick sans réflexion, sans chercher à comprendre et sans analyses. De tout évidence, il vous faudra chercher à comprendre le code en entier (<https://github.com/pierrot10/joystick/blob/master/sender.ino>) ainsi que les diverses variables que je n'ai pas publié sur cette page. Mais j'espère qu'il vous aura aidé, du moins une des partie, à démarrer ou à vous orienter.

🔗 ADAFRUIT (HTTPS://ECO-SENSORS.CH/TAG/ADAFRUIT/)
🔗 ARDUINO (HTTPS://ECO-SENSORS.CH/TAG/ARDUINO/) DISPLAY
(HTTPS://ECO-SENSORS.CH/TAG/DISPLAY/) DRIVERS (HTTPS://
ECO-SENSORS.CH/TAG/DRIVERS/) ESP8266 (HTTPS://ECO-
SENSORS.CH/TAG/ESP8266/) GFX (HTTPS://ECO-SENSORS.CH/
TAG/GFX/) I2C (HTTPS://ECO-SENSORS.CH/TAG/I2C/) IDE
(HTTPS://ECO-SENSORS.CH/TAG/IDE/) JOYSTICK (HTTPS://ECO-
SENSORS.CH/TAG/JOYSTICK/) LCD (HTTPS://ECO-SENSORS.CH/
TAG/LCD/) LIBRARY (HTTPS://ECO-SENSORS.CH/TAG/LIBRARY/)
LORA (HTTPS://ECO-SENSORS.CH/TAG/LORA/) NEWS (HTTPS://
ECO-SENSORS.CH/TAG/NEWS/) OLED (HTTPS://ECO-SENSORS.CH/
TAG/OLED/) RFM95 (HTTPS://ECO-SENSORS.CH/TAG/RFM95/)
ROBOT (HTTPS://ECO-SENSORS.CH/TAG/ROBOT/) SSD1306
(HTTPS://ECO-SENSORS.CH/TAG/SSD1306/)

PREV POST

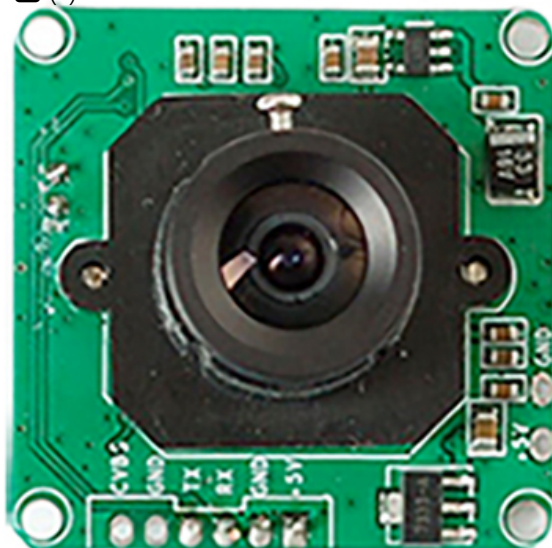
NEXT POST

(https://eco-sensors.ch/comment- (https://eco-sensors.ch/snap-si-tu-
accéder-a-mon-raspberry-depuis- bouges-je-timmortalise/)
nimporte-ou/)

Related **Posts:**

Mesure de la qualité de l'air (PM2.5/PM10) –

15 AVRIL 2020



Snap! Si tu bouges, je t'immortalise! (<https://>

23 DÉCEMBRE 2017

Leave **Comment:**

FULL NAME

EMAIL ADDRESS

PHONE NUMBER

YOUR COMMENT

SUBMIT NOW

Nous **soutenir**



STM32
LORA

DISCOVERY KIT

(<https://eco-sensors.ch/product/stm32-lora-discovery-kit/>)

CHF30.00



ST-
LINK

(<https://eco-sensors.ch/product/st-link/>) CHF59.00

Inscrivez-vous à notre newsletter

Prénom

Nom de famille

Email

Votre secteur

Je souhaite juste vous suivre



Votre fonction

Je souhaite juste vous suivre



Votre exploitation/association (facultatif)

Indiquez-nous pour qui vous euvrez

Vos cultures, seront-elle sensibles aux dommages causés par le gel (facultatif)

non



Sponsor/donateur potentiel

peut-être



Commentaire (facultatif)

Que pouvons-nous vous apporter?

S'ABONNER



 info@eco-sensors.ch (mailto:info@eco-sensors.ch)

© Copyright 2020 EcoSensors. - Tous droits réservés.