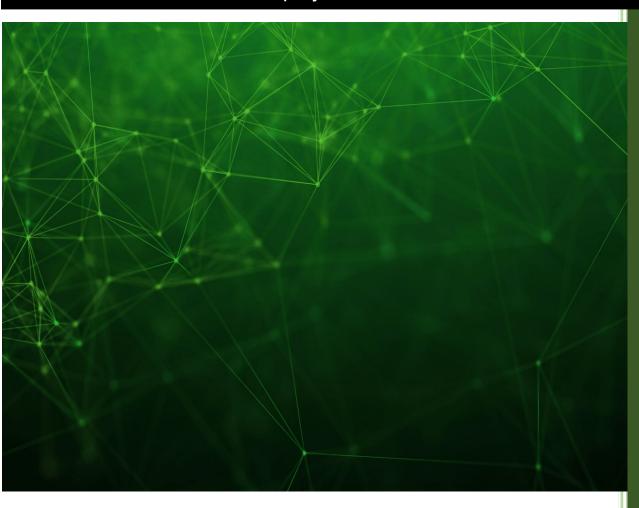


2023

# **SCRAPY Coder User Manual**

Numéro de projet: 2021-1-FR01-KA220-SCH-000031617





Le soutien de la Commission européenne à la production de cette publication ne constitue pas une approbation du contenu, qui reflète uniquement les points de vue des auteurs, et la Commission ne peut être tenue responsable de toute utilisation qui pourrait être faite des informations contenues dans ce document.





# Table des matières

1.	Introduction		2
	Connexion		
	Page d'accueil		
		Sélection de la langue	
	3.2.	Progrès	7
	3.3.	Tous les projets	7
	3.4.	Barre de navigation	11
4.	Environnement de programmation Blockly1		12
5.	Insignes		13
6.	Solutions de projets		





### 1. Introduction

Le<u>Codeur SCRAPY</u>est une application de programmation par glisser-déposer, conçue spécifiquement pour héberger une série de projets de bricolage basés sur le kit SCRAPY. L'environnement de programmation est développé à l'aide de<u>Google Blockly</u>, avec plusieurs blocs de code sur mesure pour répondre aux exigences de l'électronique, des capteurs et des composants du kit.

À l'aide du Coder, un utilisateur intéressé peut programmer plusieurs appareils électroniques DIY, apprenant en même temps les concepts informatiques physiques et les paradigmes de l'électricité et des circuits. Le Coder offre également la possibilité à l'utilisateur de créer ses propres projets et de les programmer à l'aide de l'environnement de codage.

Ce manuel d'utilisation aidera les utilisateurs et les éducateurs à démarrer avec le SCRAPY Coder, à comprendre ses différentes fonctionnalités et caractéristiques, et enfin à se familiariser avec l'environnement de programmation Blockly afin de mettre en œuvre les projets du Coder ainsi que les leurs.

### 2. Connexion

L'accès au SCRAPY Coder peut se faire en visitant le site Web du projet (<u>www.scrapykit.eu</u>) et cliquez simplement sur le bouton « Codeur » sur la page d'accueil.

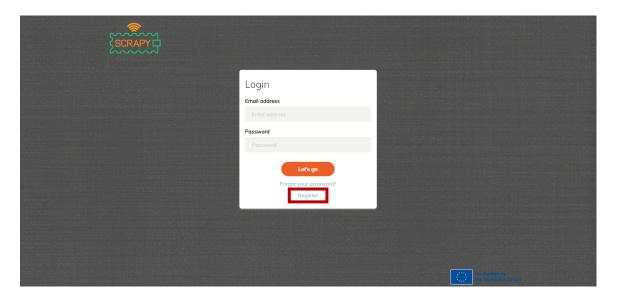


Alternativement, les utilisateurs peuvent utiliser l'URL suivante :www.coder.scrapykit.eu.





Les nouveaux utilisateurs doivent créer un compte. Sur la page de connexion, cliquez simplement sur le bouton « S'inscrire ».



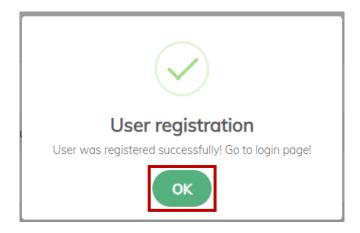
Remplissez vos informations et cliquez sur « S'inscrire ».



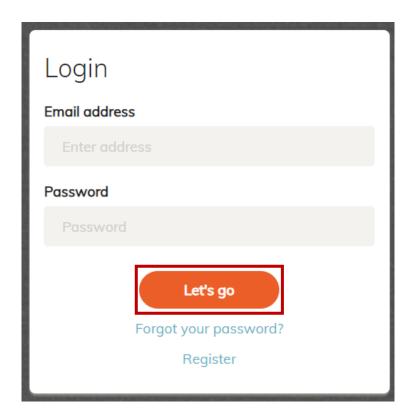




Un message contextuel apparaîtra après une inscription réussie. Cliquez sur « OK » et accédez à la page de connexion en cliquant sur « Vous cherchez à vous connecter ? ». De plus, un e-mail de confirmation arrivera dans votre boîte de réception, vérifiant votre inscription.



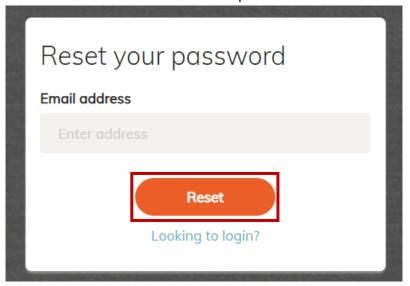
Sur la page de connexion, insérez votre adresse e-mail et votre mot de passe, puis cliquez sur « C'est parti ».





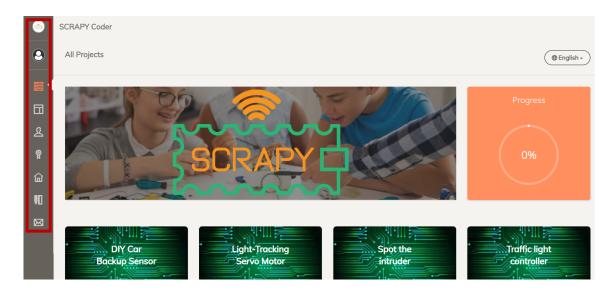


Si vous avez oublié votre mot de passe, cliquez simplement sur « Mot de passe oublié ? » et vous serez redirigé vers la page de réinitialisation du mot de passe. Là, insérez votre adresse e-mail et cliquez sur « Réinitialiser ». Un e-mail contenant des instructions supplémentaires arrivera dans votre boîte de réception.



# 3. Page d'accueil

Après vous être connecté avec succès à votre compte, vous serez redirigé vers la page d'accueil du Codeur. Là, la liste des projets disponibles sera présentée, ainsi que les options de sélection de langue, l'état d'avancement et quelques autres informations. La plupart des fonctionnalités de Coder sont accessibles via le panneau de navigation sur le côté gauche de la page d'accueil.









## 3.1. Sélection de la langue

Dans le coin supérieur droit se trouve un sélecteur de langue. Cliquez simplement dessus et choisissez votre langue préférée. Le SCRAPY Coder est fourni en six langues : anglais, grec, français, portugais, croate et néerlandais.

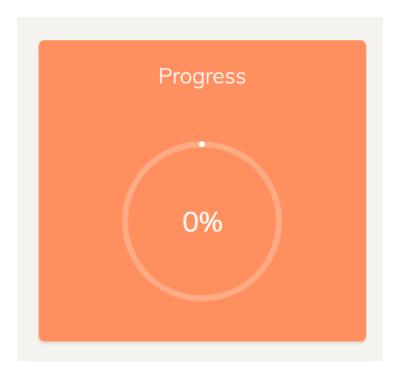






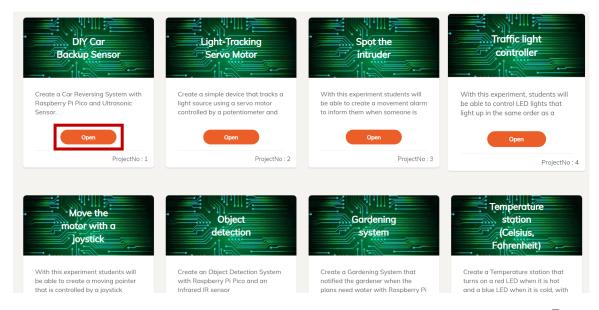
### 3.2. Progrès

Directement sous le sélecteur de langue se trouve une roue de progression. Chaque fois que vous terminez un projet, le pourcentage augmentera jusqu'à ce que vous atteigniez 100 % d'achèvement.



### 3.3. Tous les projets

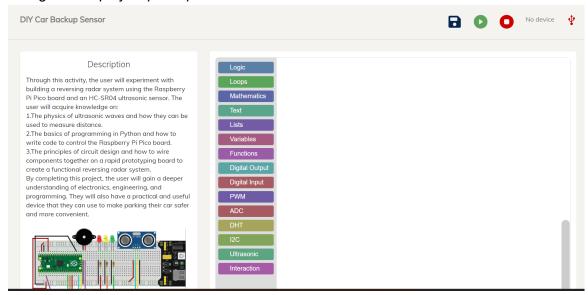
Le Coder donne accès à 12 projets de bricolage, utilisant l'électronique, les capteurs et d'autres composants du kit SCRAPY. Tous les projets sont en accès libre en cliquant simplement sur le bouton « Ouvrir ».



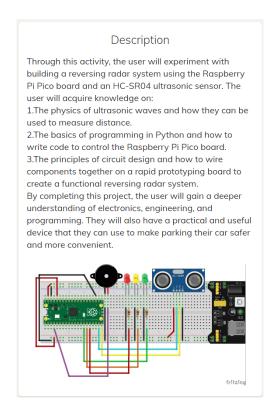




En cliquant sur le bouton « OUVRIR », l'utilisateur sera dirigé vers l'environnement de codage de ce projet spécifique.

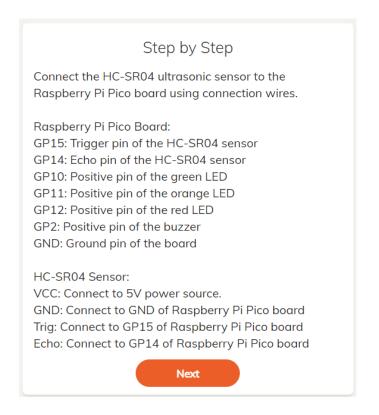


Là, l'utilisateur peut voir l'environnement de codage Blockly, une description du projet et un guide étape par étape pour la mise en œuvre du projet.











De plus, il y a quelques options dans le coin supérieur droit liées à la connectivité du Raspberry Pi Pico.



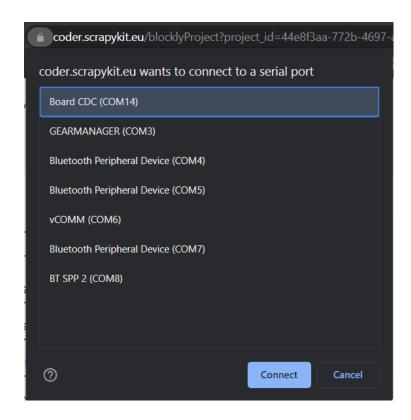




Pour faire interagir le Coder avec le monde physique, vous devez connecter votre Raspberry Pi Pico. La première étape consiste à vous assurer que le micrologiciel de votre Pico est installé. Parcourez les pages 12 à 19 du manuel du kit SCRAPY si vous ne savez pas comment procéder. Assurez-vous de fermer l'application Thonny Python. Ensuite, allez sur le projet de votre choix et cliquez sur l'icône d'option Appareil.



Une liste de tous les appareils disponibles, connectés à votre ordinateur apparaîtra. Choisissez celui appelé Board CDC. Cliquez ensuite sur « Connecter ».







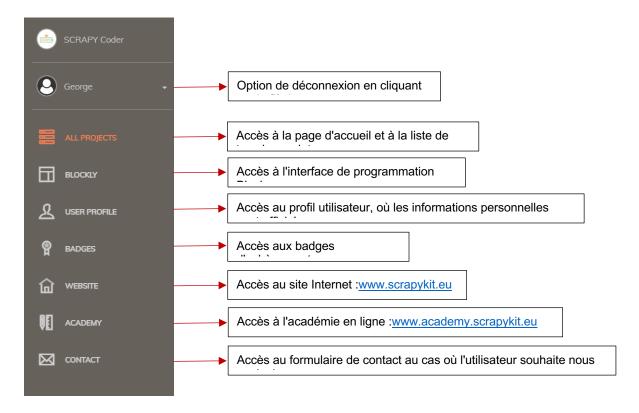
Vous recevrez un message indiquant que l'appareil est connecté, qui est également visible dans le coin supérieur droit.



Vous êtes maintenant prêt à commencer à créer votre code pour le projet de votre choix. Une fois prêt, cliquez sur le bouton Jouer et voyez ce qui se passe dans le monde physique. Vous recevrez également un message indiquant que votre programme est en cours d'exécution. Si vous devez arrêter votre programme, cliquez simplement sur le bouton Arrêter.

### 3.4. Barre de navigation

La barre de navigation de gauche offre un accès rapide à plusieurs fonctionnalités et informations liées au Coder, ainsi qu'au projet SCRAPY.





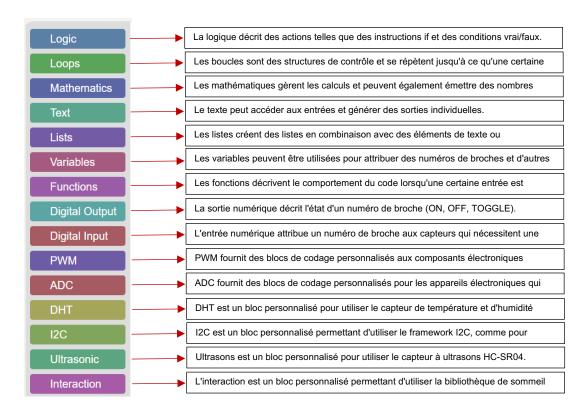


# 4. Environnement de programmation Blockly

L'accès à l'environnement de programmation Blockly peut se faire par la barre de navigation de gauche en cliquant simplement sur « BLOCKLY ». L'application vous redirigera vers l'interface suivante :



Blockly propose une variété de blocs afin de créer différents programmes en fonction de vos besoins. Tous les blocs sont accessibles via la barre « Blocs » sur la gauche où toutes les catégories de codage sont affichées. Les blocs de codage peuvent être glissés et déposés sur l'espace de codage blanc. Chaque catégorie propose plusieurs options de codage.

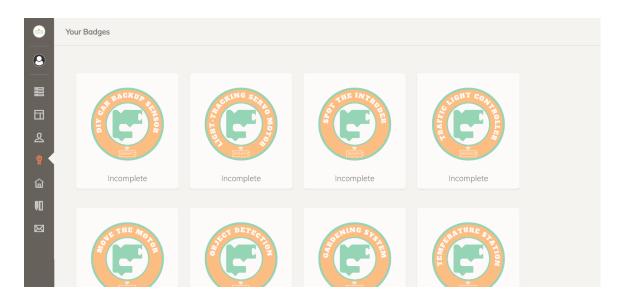


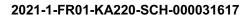




# 5. Insignes

SCRAPY Coder attribue des badges d'achèvement pour chaque projet terminé. Ces badges sont accessibles via le panneau de navigation de gauche, en cliquant simplement sur « BADGES ». Sur la page Badges, les utilisateurs peuvent voir les badges attribués qui sont marqués comme « complets ».









# 6. Solutions de projets

Projet n°1 – Capteur de recul de voiture DIY

```
HC-SR04 ultrasonic sensor with trigger 14 and echo 15
set ultrasonic v to
set red led to
                    Output pin number 12
                      Output pin number 11
set yellow led to
                      Output pin number 10
set green led to
                   Output pin number 2
set buzzer * to
repeat while *
                 true 🔻
do
     set distance v to
                          Distance in cm
                                           ultrasonic *
     distance *
                                      20
     do
           Pin - State ON
                            green led *
            Pin - State OFF
                             red led ▼
            Pin - State OFF
                             yellow led ▼
            Pin - State OFF
                             buzzer 🔻
     else if
                  distance *
                                      5
     do
           Pin - State ON
                            yellow led ▼
           Pin - State OFF
                             red led 🔻
            Pin - State OFF
                             green led 🔻
            Pin - State OFF
                             buzzer 🔻
     else
           Pin - State ON
                            red led 🔻
            toggle buzzer 🔻
            Pin - State OFF
                             yellow led ▼
            Pin - State OFF
                             red led ▼
     Sleep 0.1 seconds
```





### Projet n°2 – Servomoteur de suivi de lumière

```
PWM for pin number 0
set servo ▼ to
set potentiometer pin ▼ to
                             ADC for GPIO26 ▼
                   ADC for GPIO27 ▼
set Idr_pin ▼ to
repeat while V
                 true ▼
    set potentiometer value ▼ to
                                    Read
                                             potentiometer pin V
    set Idr value ▼ to
                          Read
                                  [ldr pin ▼
     set angle ▼ to
                       Angle from potentiometer value
                                                      potentiometer_value V
     set speed ▼ to
                        Speed from LDR value
                                               ldr_value ▼
     Frequency 50
                     servo ▼
     set duty ▼ to
                        angle ▼
                                          180
     set duty ▼ to
                     round ▼
                                    duty ▼ ]
                                                    65025
     Duty u16
                 servo ▼
     with cycle
                 duty ▼
```

### Projet n°3 – Repérer l'intrus

```
Input pin number 28 with PULL UP resistance
set motion_sensor *
                   Output pin number 14
set led v to
                Output pin number 15
repeat while
                 true 🔻
do
     Get value
                            motion_sensor *
     do
          toggle buzzer 🔻
                   led ▼
          toggle
          Sleep 0.5 seconds
```





### Projet n°4 – Contrôleur de feux tricolores

```
set button v to
                  Input pin number 7 with PULL DOWN resistance
                  Output pin number 16
set buzzer v to
set green led to
                     Output pin number 11
set vellow led v to
                      Output pin number 12
                   Output pin number 13
set red led to
               I2C 0 for SCL pin 21 and SDA pin 20
set [i2c 🔻 to ]
set oled to
                SSD1306_I2C with width (128) and height (64)
                                                            i2c
repeat while *
                true 🔻
do
    Pin - State ON
                    red led 🔻
    Pin - State OFF
                     yellow led
    Pin - State OFF
                     green_led 🔻
    Pin - State OFF
                     [buzzer ▼
    Store message PLEASE WAIT in the screen buffer 0 and 10
     Show messages
                      oled 🔻
     Get value
                            button 🔻
    do
         Pin - State OFF
                           red led 🔻
          Pin - State ON yellow led v
          Store message CROSSING: in the screen buffer 0 and 20
                                                                    oled
          Store message ALLOWED in the screen buffer 0 and 30
          Sleep 1.5 seconds
         Pin - State ON green_led •
          Sleep 10 seconds
          Show messages
                           oled 🔻
          toggle buzzer 🔻
          Sleep 0.5 seconds
```





### Projet n°5 – Déplacer le moteur avec un joystick

```
to map_value with: value, in_min, in_max, out_min, out_max
  set value_in_min * to
                            value - - -
                                           in_min •
  set out_max_out_min - to
                                            - v Out_min v
                                out_max *
  set value_in_min_out_max_out_min - to
                                            value_in_min • × •
                                                                   out_max_out_min •
  set in_max_in_min * to [
                             in_max •
                                              in_min *
  set (in_max_in_min_out_min + ) to
                                     in_max_in_min + Out_min +
  set map_value_return * to
                                value_in_min_out_max_out_min 🔻 🗦 🕶
                                                                      in_max_in_min_out_min -
                                                                         map_value_return *
                                                      return
                                                              round *
set x_axis_pin - to ( 26
set servo_pin * to
set servo_max_angle * to [
                          45
set servo_min_angle * to
                          0
set x_axis_adc = to
                     ADC for pin GPIO26 *
                      PWM for pin number 13
set servo_pwm * to [
Frequency 30 servo_pwm •
Duty_u16 with cycle 0 servo_pwm •
repeat while *
                true -
                                 x_axis_adc *
    set x_axis_val - to
                          Read |
    set angle_x * to
                        map_value with:
                                  value
                                         x_axis_val *
                                         0
                                 in_min
                                         65535
                                in_max
                                         servo_min_angle *
                               out_min
                               out_max
                                         servo_max_angle
    Duty_u16
                servo_pwm *
    with cycle
                map_value_with:
                                  angle_x •
                          value
                         in_min
                                  servo_min_angle *
                                  servo_max_angle
                         in_max
                        out_min
                                  50
                        out_max
                                  5000
    Sleep 0.5 seconds
```





### Projet n°6 – Détection d'objets

```
set ir_pin v to Input pin number 7

repeat while v true v

do set ir_state v to Get value ir_pin v

Sleep 0.5 seconds
```

### Projet n°7 – Système de jardinage

```
set [led_pin ▼ to [
                    Output pin number 15
set sensor pin ▼ to
                       ADC for pin GPIO26 ▼
                      40000
set threshlod ▼ to
repeat (while ▼
                  true ▼
do
     set humidity ▼ to
                           Read
                                   sensor pin 🔻
     threshlod V
                  humidity V
           Pin - State ON
     do
                            led pin ▼
           Pin - State OFF
                             [led pin ▼
     Sleep 0.5 seconds
```





Projet n°8 – Station de température (Celsius, Fahrenheit)

```
set red_led ▼ to C Output pin number 14
                     Output pin number 15
set blue led ▼ to
                   Sensor DHT_11 for pin number 21
set sensor ▼ to
repeat while ▼
                 true ▼
do
    Measure
                sensor ▼
                          Humidity
     set humidity ▼ to
                                     sensor
     set temp ▼ to
                       Temparature
                                     sensor
     ٥
                 temp ▼
                                   40
     do
           Pin - State ON
                            red led ▼
           Pin - State OFF
                            blue led ▼
     else
           Pin - State ON
                            blue led ▼
           Pin - State OFF
                             red led
     Sleep 0.5 seconds
```





### Projet No 9 - Alarme incendie

```
set green led v to
                      Output pin number 5
                    Output pin number 4
set red led v to
                   Output pin number 17
set buzzer v to
set flame sensor • to
                         Input pin number 28
repeat while
                 true
do
     Get value
                              flame_sensor 🔻
     do
           Pin - State ON
                            red led 🕶
           Pin - State ON
                            buzzer 🔻
           Pin - State OFF
                             green_led
           Sleep 1 seconds
     else
           Pin - State ON
                            green_led
           Pin - State OFF
                             red led ▼
           Pin - State OFF
                             buzzer 🔻
            Sleep 1 seconds
```

Projet n°10 – Sèche-linge intelligent

```
set buzzer v to
                    Output pin number 17
set rain_sensor •
                  l to
                        Input pin number 1
repeat while
                  true 🔻
do
     if
                 Get value
                             rain sensor *
     do
          Pin - State ON
                           [buzzer ▼
          Sleep 1 seconds
     else if
                 Get value
                             rain sensor •
     do
          Pin - State OFF | buzzer •
          Sleep 1 seconds
```





Projet n°11 – Lumière qui frappe

```
set red led v to
                    Output pin number 18
set vibration sensor v to
                             Input pin number 21
repeat while .
                  true 🔻
do
     📮 if
                 Get value
                             vibration sensor *
                                                         0
          Pin - State ON
     do
                           red led •
          Sleep 5 seconds
     else if
                 Get value
                             vibration_sensor *
     do
          Pin - State OFF
                            red led 🔻
          Sleep 5 seconds
```

Projet n°12 – Alarme d'avertissement de détection sonore

```
set buzzer to C Output pin number 18
set sound sensor v to
                         Input pin number 21
repeat while
                 true 🔻
do
     Get value
                            sound sensor *
     do
         toggle 🏢
                  [buzzer ▼
          Sleep 5 seconds
     else if
                Get value
                            sound sensor *
     do
         Pin - State OFF
                           buzzer 🔻
          Sleep 5 seconds
```