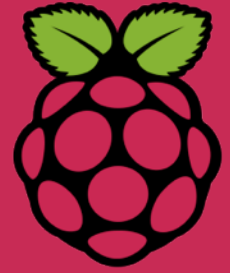
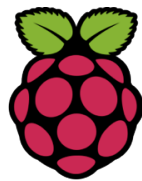


# RASPBERRY



Vous avez peu, voir pas de compétences en informatique et vous souhaitez faire de la programmation sans vous ruiner ? La **Raspberry** et le logiciel **Scratch** sont donc faits pour vous !



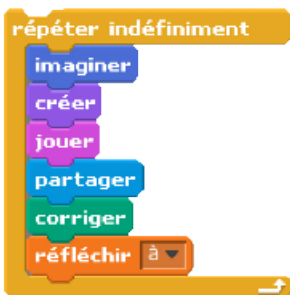
+



=

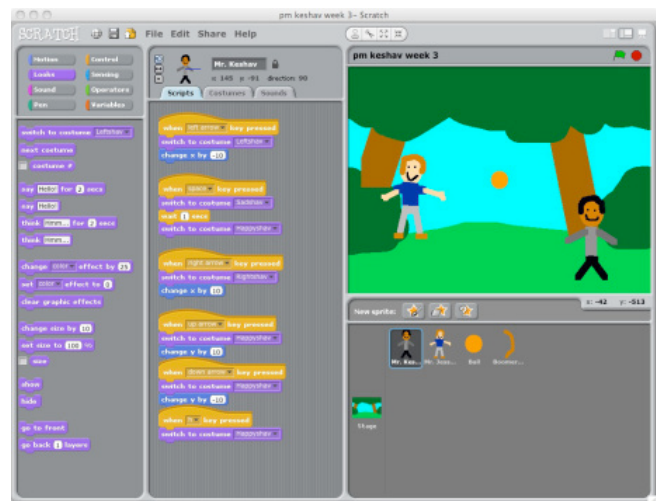


## SCRATCH C'EST QUOI EXACTEMENT ?



Il est facile de programmer une animation, un jeu, un programme de résolution de problème mathématique à l'aide du logiciel Scratch. Pour réaliser votre programme, il suffit d'imbriquer des blocs de code.

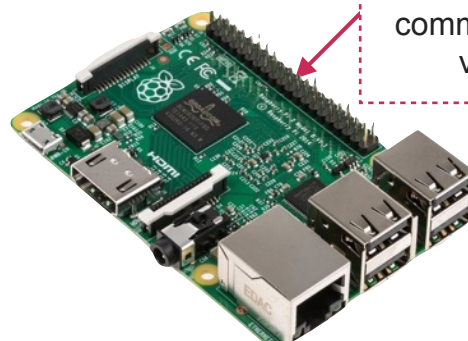
<https://scratch.mit.edu/>



Mais, il est possible d'aller plus loin dans la programmation à l'aide du port GPIO et ainsi communiquer les informations avec l'environnement extérieur : piloter des del, des moteurs, en fonction de l'état d'interrupteurs ou capteurs. C'est l'objet de ce document.

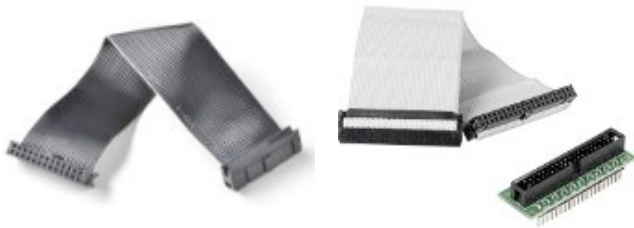
Ce document est en correspondance avec les informations du site :

<https://cymplecy.wordpress.com>



1 Port nommé **GPIO** pour communiquer avec des interfaces via (**Scratch**, Python, ...)

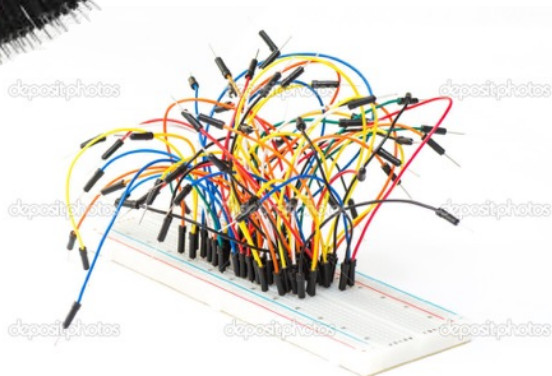
## CE QU'IL VOUS FAUDRA



**Fils d'extension GPIO ou Kit**  
Env. 5€ à 15€



**Fils à connecter**  
Env. 4€ les 30

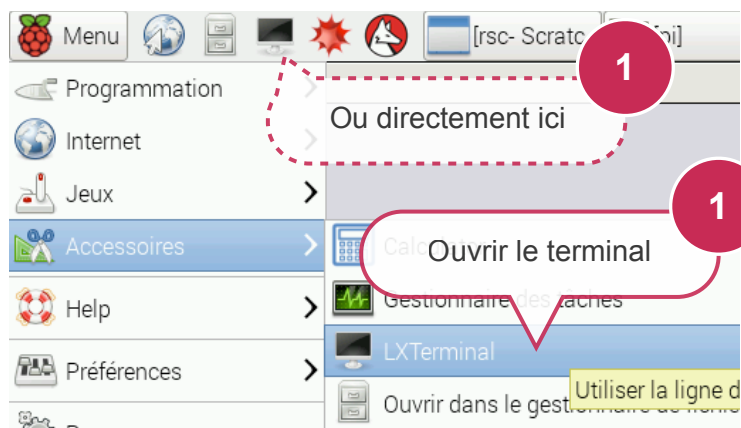


**Composants électroniques**  
Type bouton poussoir, capteur fin de course, del, buzzer, résistances 10k et 330 Ohms

**Platine d'essais**  
Env. 15€

## INSTALLATION DU LOGICIEL.

Une fois votre raspberry connecté à internet via un câble réseau ou la clé Wifi, vous allez pouvoir télécharger et installer le logiciel Scratch spécifique pour le port GPIO.



3  
Taper la commande suivante afin de télécharger le logiciel :  
`wget http://bit.ly/1wxrqp -O isgh7.sh`



4  
Taper la commande suivante afin d'installer le logiciel :  
`sudo bash isgh7.sh`

# LE PORT GPIO



	3,3 V	■	■	5 V
Entrée - Pin3	●	■	■	5 V
Entrée - Pin5	●	■	■	0 V
Entrée - Pin7	●	●	●	Pin8 - Entrée
0 V	■	●	●	Pin10 - Entrée
Sortie - Pin11	●	●	●	Pin12 - Sortie
Sortie - Pin13	●	■	■	0 V
Sortie - Pin15	●	●	●	Pin16 - Sortie
	3,3 V	■	●	Pin18 - Sortie
Entrée - Pin19	●	■	■	0 V
Entrée - Pin21	●	●	●	Pin22 - Entrée
Entrée - Pin23	●	●	●	Pin24 - Entrée
0 V	■	●	●	Pin26 - Entrée
	●	●	●	
Entrée - Pin29	●	■	■	0 V
Entrée - Pin31	●	●	●	Pin32 - Entrée
Entrée - Pin33	●	■	■	0 V
Entrée - Pin35	●	●	●	Pin36 - Entrée
Entrée - Pin37	●	●	●	Pin38 - Entrée
0 V	■	●	●	Pin40 - Entrée

	3,3 V	■	■	5 V
Entrée - Pin3	●	■	■	5 V
Entrée - Pin5	●	■	■	0 V
Entrée - Pin7	●	●	●	Pin8 - Entrée
0 V	■	●	●	Pin10 - Entrée
Sortie - Pin11	●	●	●	Pin12 - Sortie
Sortie - Pin13	●	■	■	0 V
Sortie - Pin15	●	●	●	Pin16 - Sortie
	3,3 V	■	●	Pin18 - Sortie
Entrée - Pin19	●	■	■	0 V
Entrée - Pin21	●	●	●	Pin22 - Entrée
Entrée - Pin23	●	●	●	Pin24 - Entrée
0 V	■	●	●	Pin26 - Entrée



## Des sorties supplémentaires ?

Si vous avez besoin de sorties supplémentaires, vous pouvez utiliser toutes les entrées en indiquant le code adéquate aux sorties :

exemple pour pin7 : pin7on ou pin7off

envoyer à tous

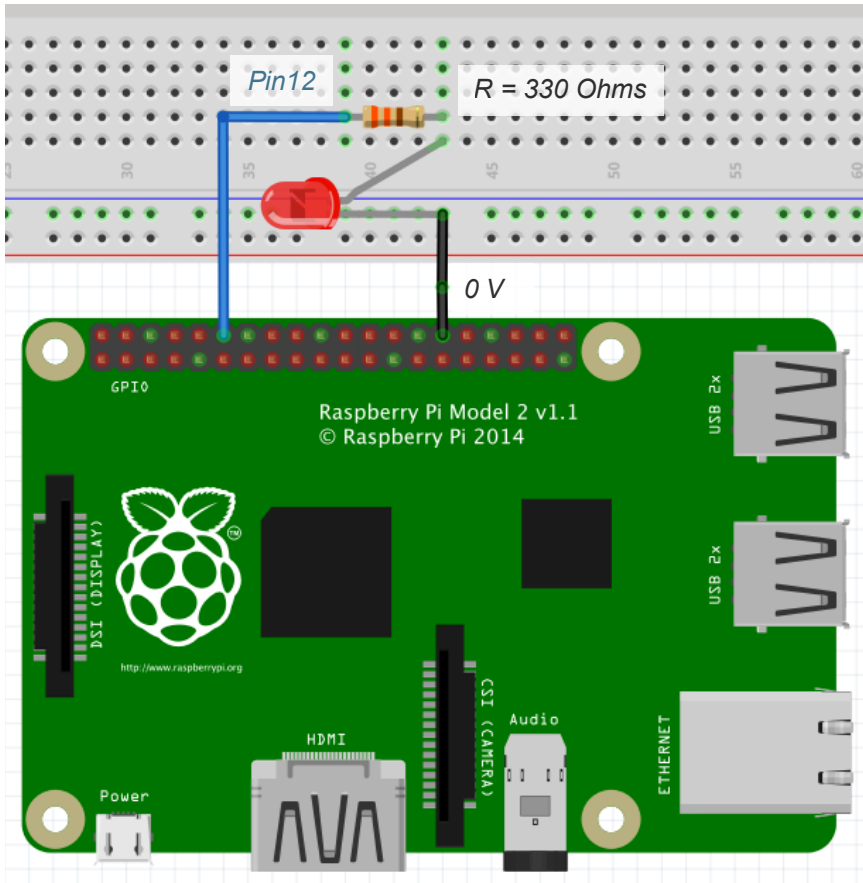
envoyer à tous

## 1 entrée supplémentaire ?

Si vous avez besoin d'une entrée supplémentaire, vous pouvez utiliser uniquement la pin11 en indiquant le code Config11In.

envoyer à tous

# MON PREMIER PROGRAMME



```

quand [drapeau] pressé
  répéter indéfiniment
    envoyer à tous pin12on
    attendre 1 secondes
    envoyer à tous pin12off
    attendre 1 secondes
  
```

Ce programme permet d'allumer une del toutes les secondes depuis la sortie pin12.

*La résistance de 330 Ohms est importante, elle permet de limiter le courant dans la del. Sans cette résistance, la del serait « grillée » très rapidement (quelques secondes)!*



- quand [drapeau] pressé
- quand [espace] est pressé
- quand [Objet1] pressé
- attendre 1 secondes
- répéter indéfiniment
- répéter 10 fois
- envoyer à tous
- envoyer à tous et attendre

```

quand [drapeau] pressé
  répéter indéfiniment
    envoyer à tous nouveau/edit...
  
```

Il faut indiquer l'état des sorties à l'aide du clavier



Il est aussi possible d'indiquer l'état logique (en anglais) pour piloter les sorties :

High et Low

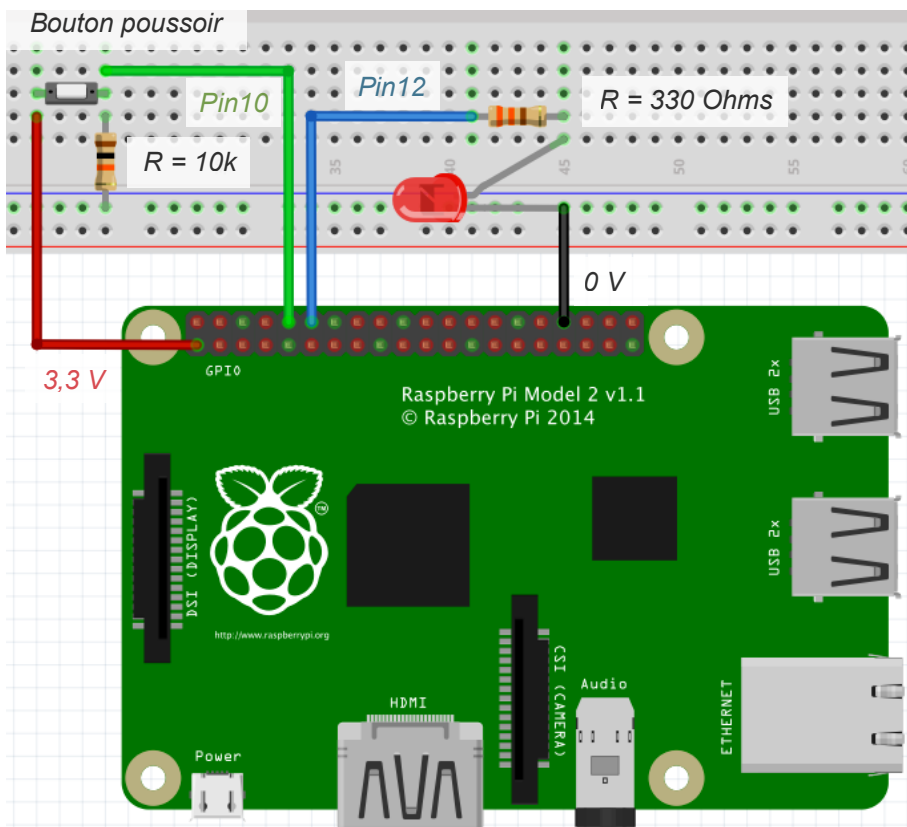
```

quand [drapeau] pressé
  répéter indéfiniment
    envoyer à tous pin12high
    attendre 1 secondes
    envoyer à tous pin12low
    attendre 1 secondes
  
```

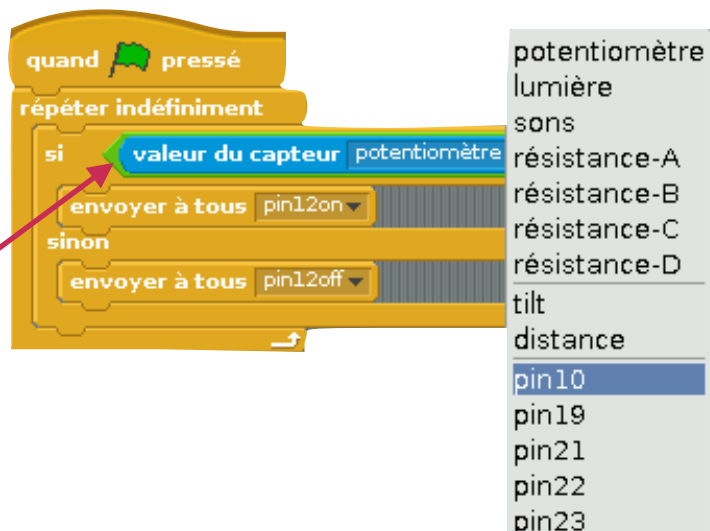
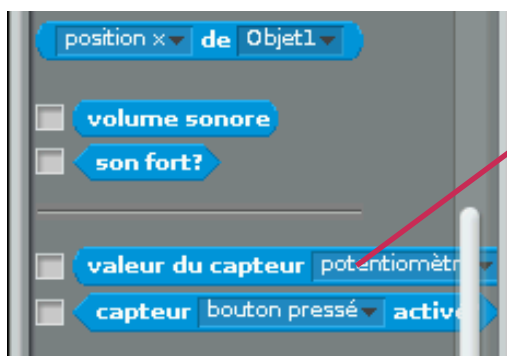
# MON DEUXIÈME PROGRAMME

Ce programme permet d'allumer une del depuis la pin12 si le bouton poussoir est activé (1 logique) depuis la pin10.

Dans le cas contraire la del reste éteinte.

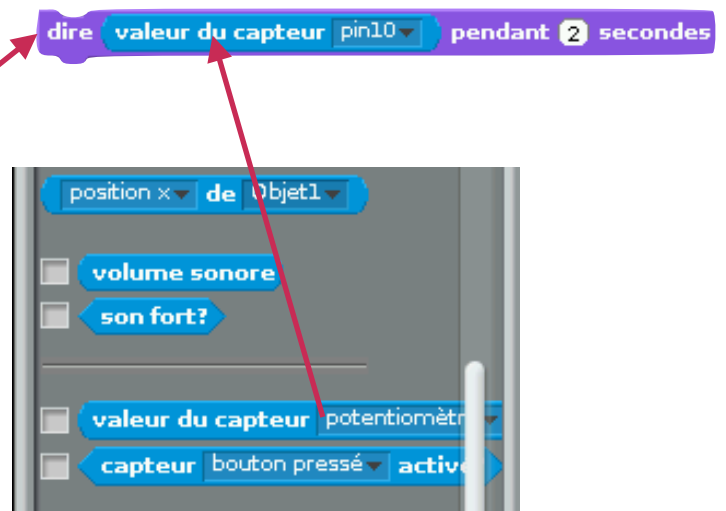
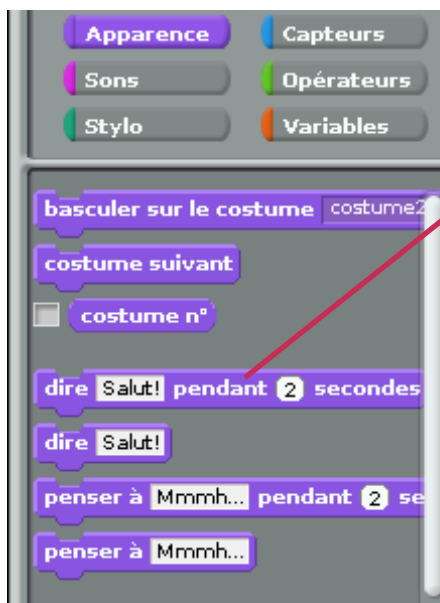


La résistance de 10k est importante, elle permet d'indiquer l'état 0 logique quand le bouton n'est pas activé. Sans cette résistance, l'entrée Pin10 serait à un état aléatoire. On dit que c'est une résistance en pulldown.



# DIALOGUER AVEC L'INTERFACE GRAPHIQUE

Ce programme est une simple amélioration du programme précédent. Il permet juste d'indiquer l'état de la pin10 dans une bulle.



# MON PREMIER JEU

Un bouton poussoir branché sur l'entrée Pin10 permet de faire avancer l'objet 1 : un tapis

Un autre bouton poussoir branché sur l'entrée Pin8 permet de faire reculer le tapis.

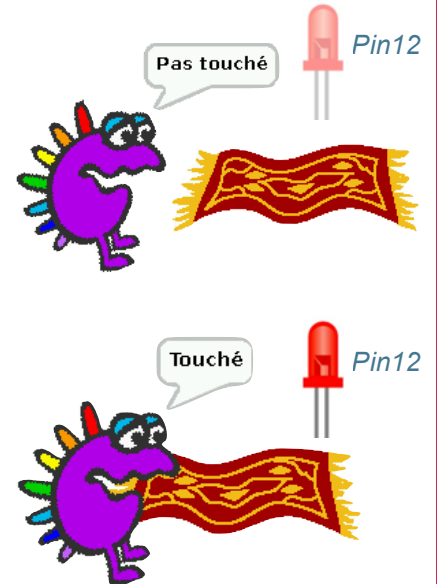
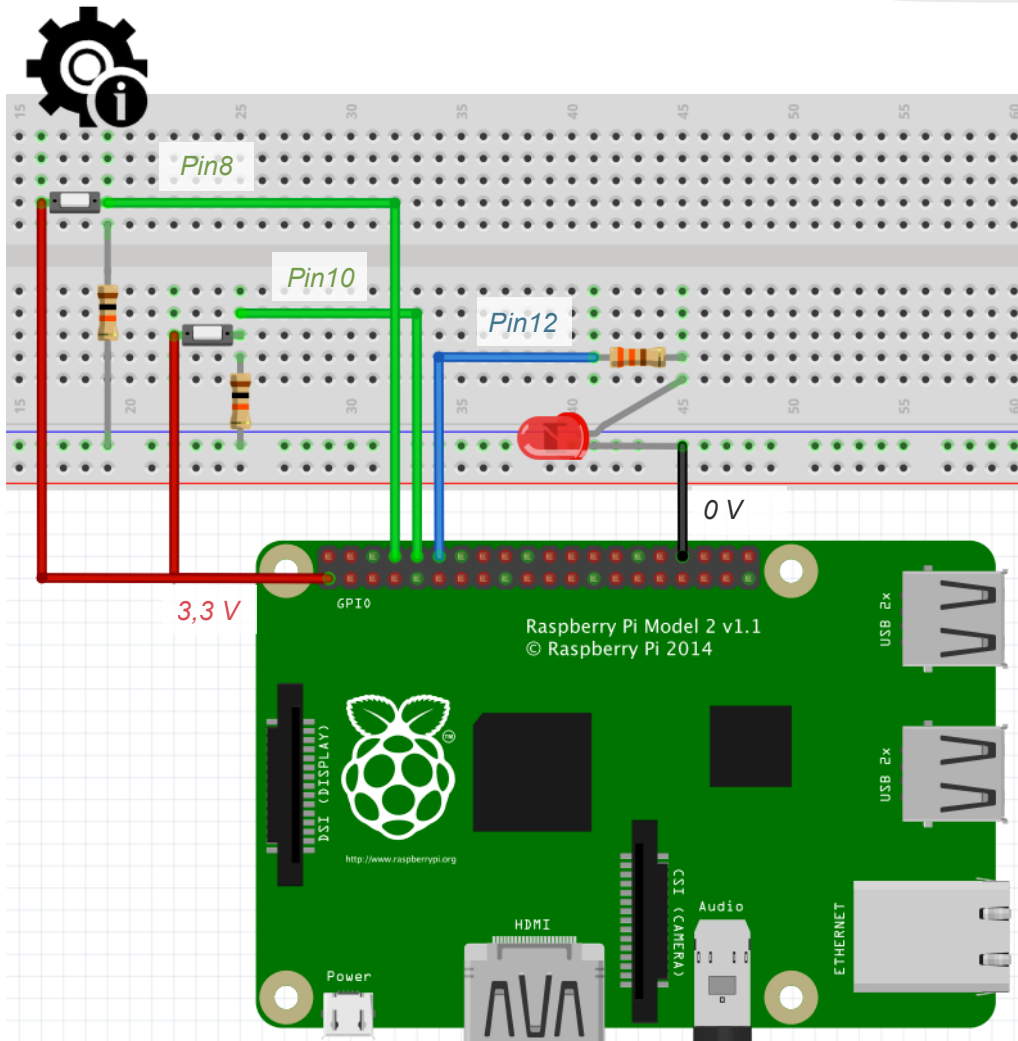
Le programme lié à l'objet 2 permet d'indiquer si le tapis (objet1) à touché le monstre (objet2).

Egalement une del branchée sur la sortie Pin12 s'allume dès que le tapis touche le monstre.



```
quand [drapeau] pressé
  répéter indéfiniment
    si [valeur du capteur pin10] = 1
      avancer de 2 pas
    si [valeur du capteur pin8] = 1
      avancer de -2 pas
```

```
quand [drapeau] pressé
  répéter indéfiniment
    si [couleur touche] ?
      dire Touché
      envoyer à tous pin12on
    sinon
      dire Pas touché
      envoyer à tous pin12off
```

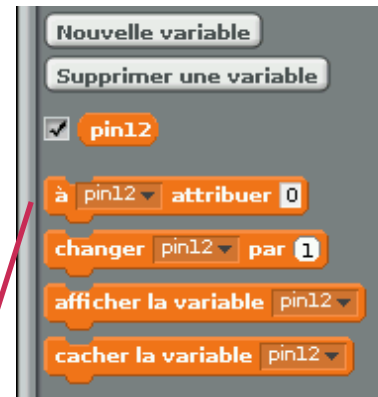
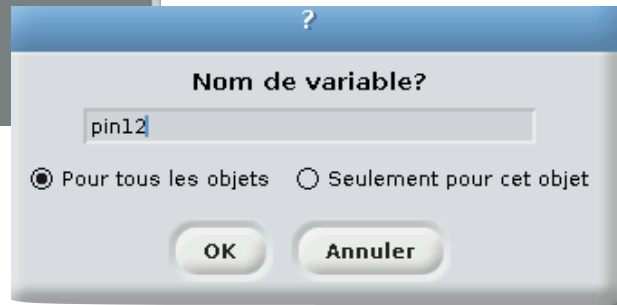


# UTILISER DES VARIABLES



Il est possible de créer des variables dans le menu Variables > Nouvelle variable afin d'enregistrer un score à afficher ou des valeurs à sauvegarder.

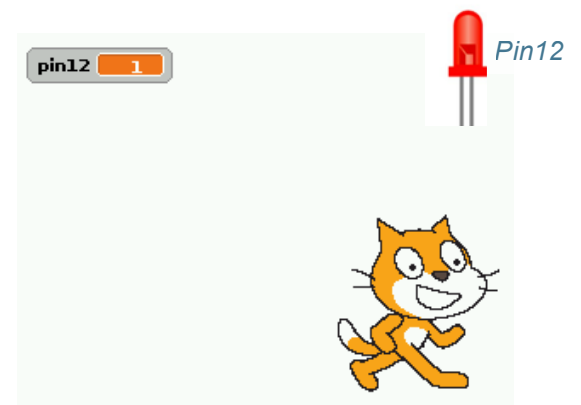
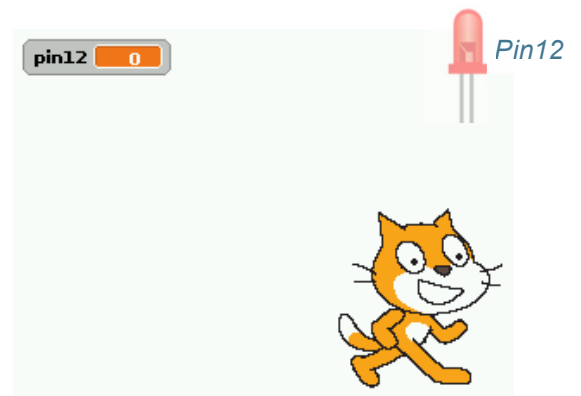
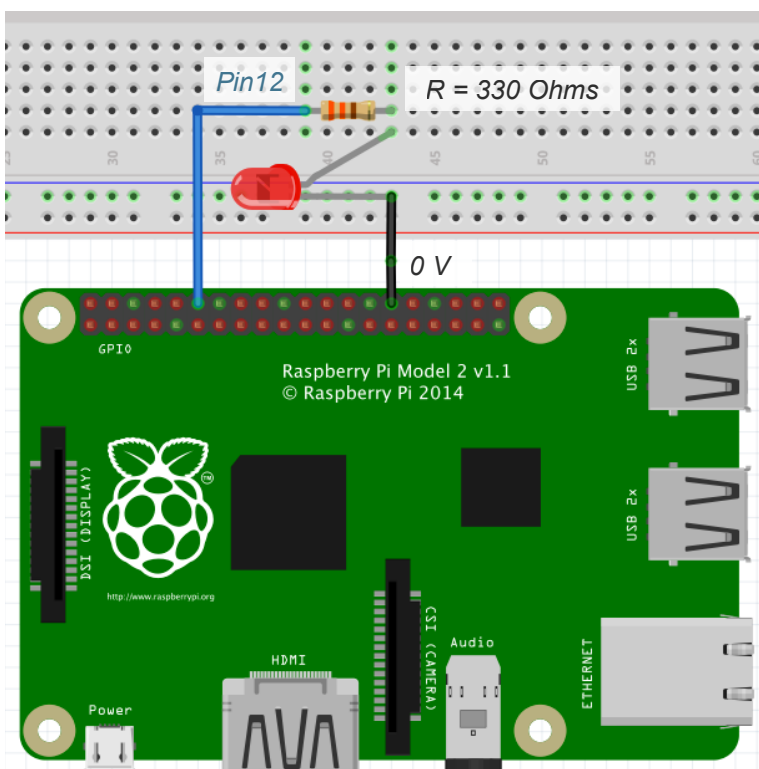
Mais il est aussi possible d'utiliser les variables pour piloter les sorties. Exemple ici avec la sortie Pin12.



Ce programme permet d'allumer une del toutes les secondes depuis la sortie pin12 à l'aide de la variable créée et nommée pin12.



Seules les valeurs logiques fonctionnent :  
**1 ou 0**





# QUELQUES BLOCS DE CODE UTILES

envoyer à tous AllOff

Permet de mettre toutes les sorties (pin11, pin12, pin13, pin15, pin16 et pin18) à l'état logique 0 ou 1.

envoyer à tous AllOn

envoyer à tous pinpattern000101

Permet de piloter en un seul bloc toutes les sorties (pin11, pin12, pin13, pin15, pin16 et pin18). Dans cet exemple seul pin11 et pin13 sont activés (état logique 1)

Pin18 Pin16 Pin15 Pin13 Pin12 Pin11

```
quand [drapeau] pressé
  répéter indéfiniment
    envoyer à tous GetTime
    dire valeur du capteur fulldatetime pendant 2 secondes
    dire valeur du capteur hours pendant 2 secondes
    dire valeur du capteur minutes pendant 2 secondes
```

Programme qui permet d'indiquer la date, l'heure et les minutes.  
Exemple ici avec 15 pour 15 heure.



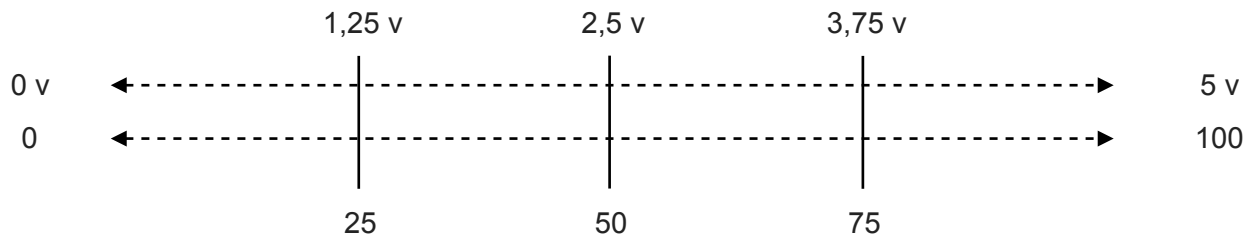
```
quand [drapeau] pressé
  répéter indéfiniment
    envoyer à tous GetTime
    dire valeur du capteur fulldatetime
    dire valeur du capteur hours
    dire valeur du capteur minutes
```

# UTILISER LES SORTIES ANALOGIQUES

Toutes les sorties (pin11, pin12, pin13, pin15, pin16 et pin18) peuvent prendre une valeur analogique.



Intéressant pour piloter par exemple la puissance d'une del ou la vitesse d'un moteur.



Pour créer une sortie analogique, il faut obligatoirement passer par les variables et créer une nouvelle variable au nom de :

Power + le numéro de la Pin

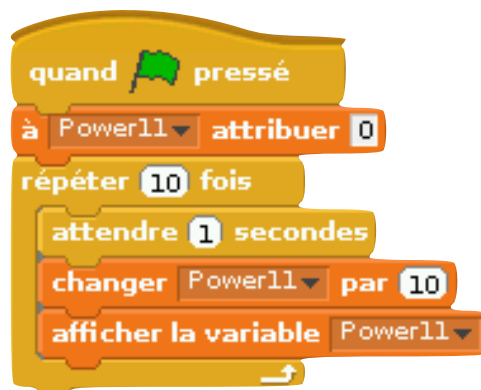
Exemple : **Power11** pour la sortie Pin11



Soit Pin11 à 2,5 volts

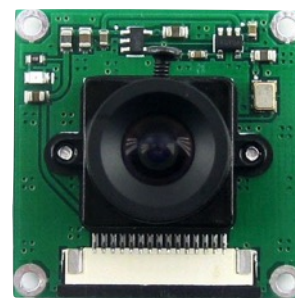
Ce programme permet d'allumer de façon progressive la puissance lumineuse d'une del sur la sortie Pin11.

Toutes les secondes et à 10 reprises, la valeur de la puissance de la del augmente de 10.



# UTILISER LA CAMERA PI

Ou toutes autres cameras en USB



envoyer à tous photo ▼

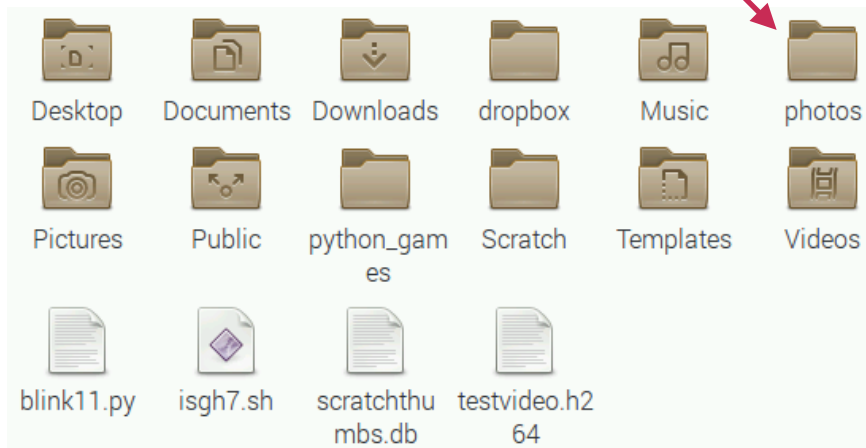
Le bloc « **Envoyer à tous photo** » permet de prendre automatique une photo via la camera.



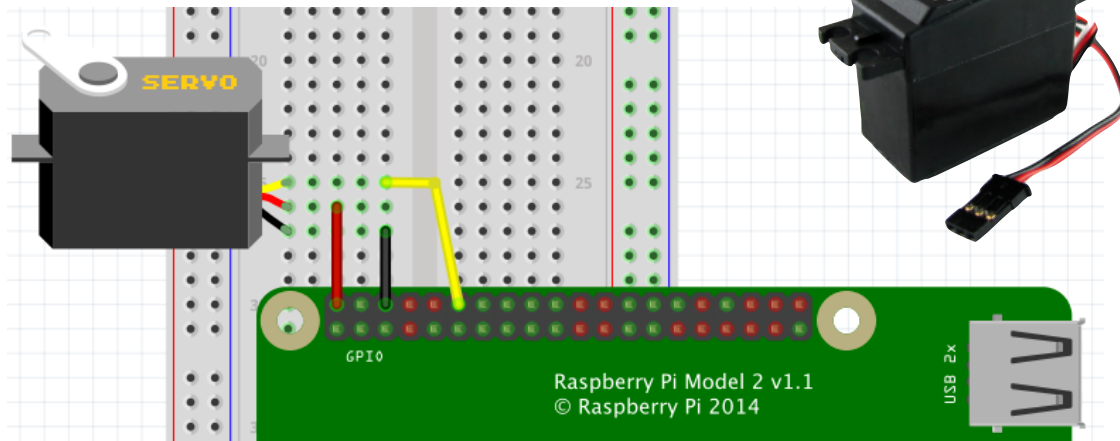
Ce programme permet de prendre 10 photos a raison de 1 photo toute les secondes. Pour visualiser si la photo est en train d'être prise, une del en sortie Pin8 s'allume également.

Les photos sont stockées dans / home / PI / photos et le nom des fichiers s'incrémentes automatiquement.

Attention à bien gérer les fichiers car vous risquez de manquer d'espace assez rapidement !



# UTILISER UN SERVOMOTEUR



Nouvelle variable

Supprimer une variable

Servo12

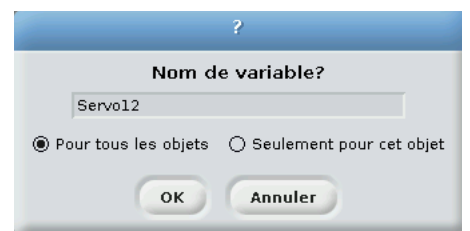
à Servo12 attribuer 0

changer Servo12 par 1

afficher la variable Servo12

cacher la variable Servo12

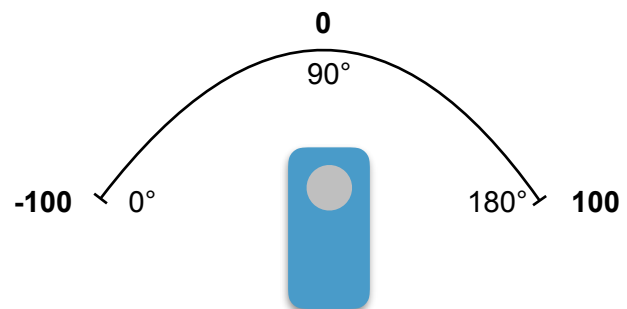
Pour utiliser un servomoteur, il faut obligatoirement passer par les variables et créer une nouvelle variable au nom de :



Servo + le numéro de la Pin

90°

Exemple : **Servo12** pour la sortie Pin12



quand pressé

répéter indéfiniment

si **touche espace** pressée?

à Servo12 attribuer 0

si **touche flèche bas** pressée?

à Servo12 attribuer -100

si **touche flèche haut** pressée?

à Servo12 attribuer 100

Ce programme permet de positionner le servomoteur en position milieu avec la touche « Espace » du clavier.

Ou en position toute à gauche avec la touche « flèche de bas ».

Ou en position toute à droite avec la touche « flèche du haut ».

**attention**

S'il vous semble que le servomoteur force pour atteindre les limites droite (100) et gauche (-100), utilisez une autre valeur tel que 90, seuls les tests permettront de déterminer les limites de votre servomoteur.