

RFID avec Raspberry

Objectif :

Lire des cartes (puces) RFID avec le Raspberry Pi et un lecteur RFID RC522.



Montage :

Le montage doit être réalisé le Raspberry éteint.

Brancher le lecteur RFID RC522 sur la Raspberry :

Remarque :

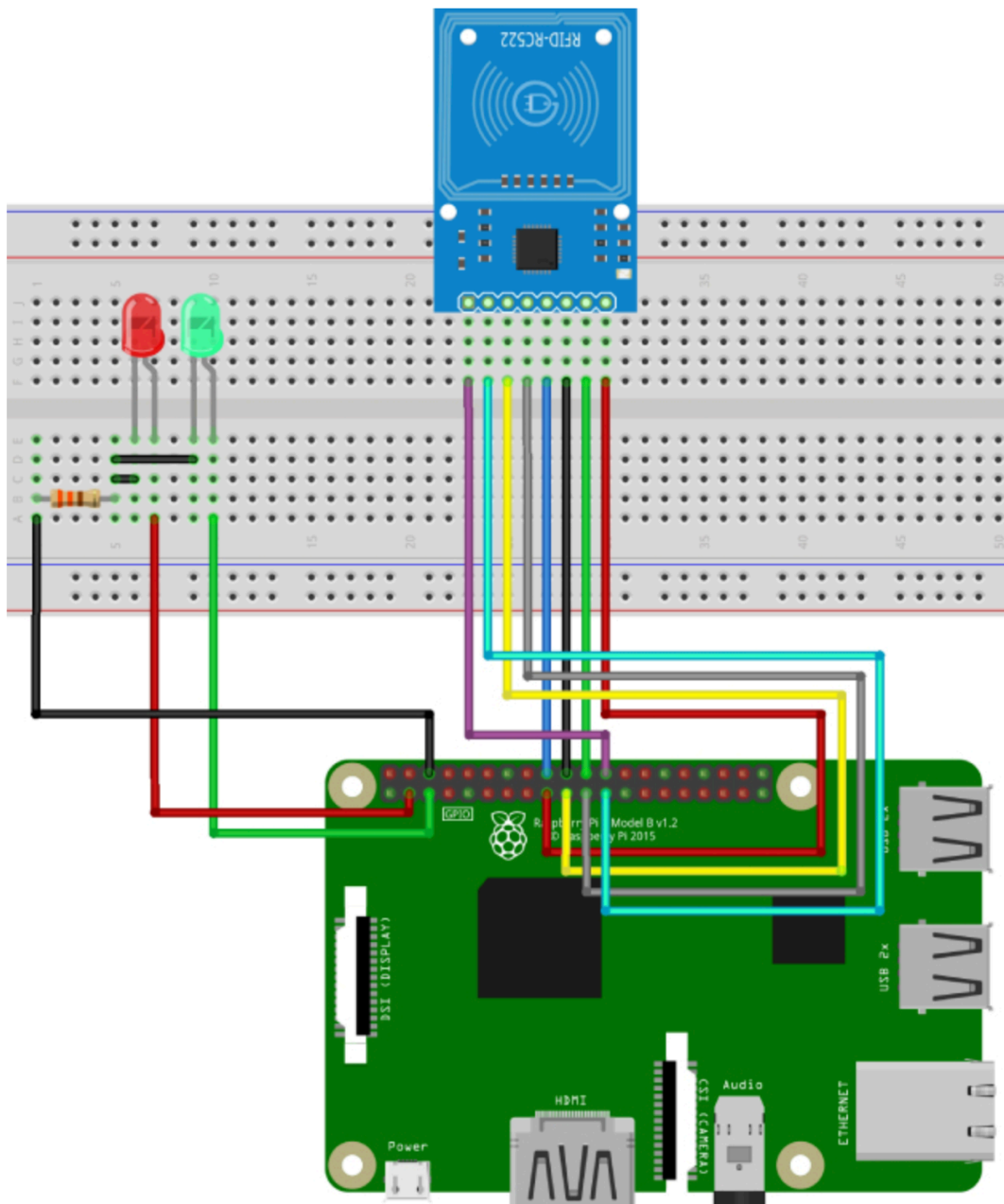
Deux modèles de lecteurs RFID sont disponibles à l'école. Le tableau ci-dessous donne le branchement du module RC522 avec le Raspberry Pi. Si le modèle fourni est différent du RC522, il est conseillé de consulter sa documentation sur internet.

Pour contrôler le lecteur RFID, on va utiliser une bibliothèque écrite avec les ports GPIO directement renseignés dans le code. Cela signifie qu'on ne peut pas les configurer différemment et qu'on est obligé d'utiliser exactement les mêmes ports que ceux prévus dans la librairie.

Board pin name	Board pin	Physical RPi pin	RPi pin name	Beaglebone Black pin name
SDA	1	24	GPIO8, CE0	P9_17, SPI0_CS0
SCK	2	23	GPIO11, SCKL	P9_22, SPI0_SCLK
MOSI	3	19	GPIO10, MOSI	P9_18, SPI0_D1
MISO	4	21	GPIO9, MISO	P9_21, SPI0_D0
IRQ	5	18	GPIO24	P9_15, GPIO_48
GND	6	20	Ground	Ground
RST	7	22	GPIO25	P9_23, GPIO_49
3.3V	8	17	3V3	VDD_3V3

Brancher les LEDs

On utilise une seule résistance pour deux LEDs (pour limiter le nombre de câbles).



Programme (lecture du badge RFID) :

Activer le protocole SPI de la Raspberry Pi, s'il n'a pas été activé auparavant.

Commande `sudo raspi-config`, sur Raspberry, puis choisir Interfacing Options, SPI, Yes et enfin Finish.

Installer la librairie pi-rc522

```
sudo pip3 install pi-rc522
```

Programme pour lire l'identifiant d'un badge RFID

Chaque badge RFID possède un identifiant unique. C'est en lisant cet identifiant et en le comparant à une liste d'identifiants autorisés que l'on peut savoir si un badge est autorisé.

Ecrire le programme dans le dossier /home/pi/TP_RFID (Créer le dossier TP_RFID).

Dans ce dossier, créer un fichier `read_rfid_uid.py` et écrire le code suivant :

```
#!/usr/bin/env python3.5
#-- coding: utf-8 --

import RPi.GPIO as GPIO # Bibliothèque GPIOs
from pirc522 import RFID
import time

GPIO.setmode(GPIO.BOARD) # Mode de numérotation (Board)
GPIO.setwarnings(False) # Désactive les messages d'alerte

rc522 = RFID() # On instancie la librairie

print('En attente d\'un badge (pour quitter, Ctrl + c): ') # Affiche un
message demandant à l'utilisateur de passer son badge

# Boucle infinie pour lire en boucle
while True :
    rc522.wait_for_tag() # On attend qu'une puce RFID passe à portée
    (error, tag_type) = rc522.request() # Quand une puce a été lue, on
    récupère ses infos

    if not error : # Si pas d'erreur
        (error, uid) = rc522.anticoll() # On nettoie les possibles
        collisions (si plusieurs cartes passent en même temps)

        if not error : # Si on a réussi à nettoyer
            print('Vous avez passé le badge avec l\'id : {}'.format(uid))
# Affiche l'identifiant unique du badge RFID
            time.sleep(1) # Attend 1 seconde pour ne pas lire le tag des
            centaines de fois en quelques ms
```

Programme pour vérifier la validité d'un badge :

On va créer un nouveau programme (dans dossier TP_RFID) en copiant et modifiant le programme précédent pour qu'il compare l'uid lu à celui écrit dans le programme, et allumer la LED rouge ou verte selon le résultat (valide ou non).

```
#!/usr/bin/env python3.5
#-- coding: utf-8 --
import RPi.GPIO as GPIO
from pirc522 import RFID
import time

GPIO.setmode(GPIO.BOARD)
GPIO.setwarnings(False)

LED_RED = 3 # Numéro du port GPIO LED rouge
LED_GREEN = 5 # Numéro du port GPIO LED verte
RFID_UID = [07, 03, 20, 22, 12] #Définit l'UID du badge RFID par exemple
# Modifier la ligne RFID_UID pour mettre l'identifiant du tag RFID

# Fonction permettant d'allumer une led
def turn_led_on (led) :
    GPIO.setup(led, GPIO.OUT) # Active le contrôle du GPIO
    GPIO.output(led, GPIO.HIGH) # Allume la LED

# Fonction permettant d'éteindre une LED
def turn_led_off (led) :
    GPIO.setup(led, GPIO.OUT) # Active le contrôle du GPIO
    GPIO.output(led, GPIO.LOW) # Eteint la LED

# Fonction permettant d'allumer la rouge et éteindre la verte
def turn_red_on () :
    turn_led_off(LED_GREEN) # Eteint la LED verte
    turn_led_on(LED_RED) # Allume la LED rouge

# Fonction permettant d'allumer la verte et éteindre la rouge
def turn_green_on () :
    turn_led_off(LED_RED) # Eteint la LED rouge
    turn_led_on(LED_GREEN) # Allume la LED verte

rc522 = RFID()
print('En attente d\'un badge (pour quitter, Ctrl + c): ')

while True :
    rc522.wait_for_tag()
    (error, tag_type) = rc522.request()
    if not error :
        (error, uid) = rc522.anticoll()
        if not error :
            if RFID_UID == uid :
                print('Badge {} autorisé !'.format(uid))
                turn_green_on()
            else :
                print('Badge {} interdit !'.format(uid))
                turn_red_on()
            time.sleep(1)
```

Ajout d'un servomoteur :

- Ajouter un servomoteur au montage
- Modifier le programme pour qu'il puisse gérer le servomoteur :
 - Le servomoteur tournera de 90° si le badge est valide puis retournera à sa position initiale au bout de 5 s.