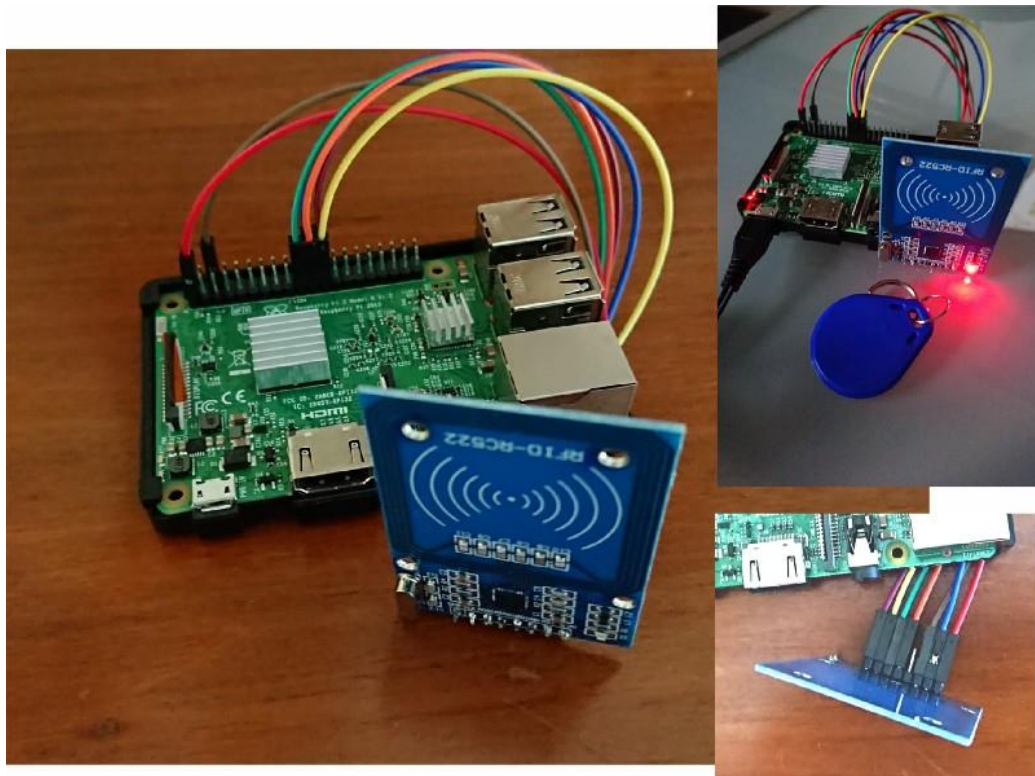


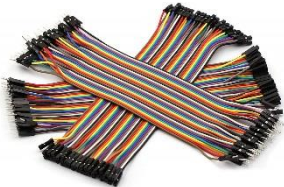
# 設置 RFID 讀寫器並記錄數據在 TANGLE 上

本章節目的是解釋如何使用 RFID 感應器在 IOTA Tangle 上記錄數據的過程。本章只是一個例子說明使用 Tangle 記錄數據是多麼容易的一件事。



## 準備材料

材料名稱	規格	數量	圖片
樹莓派 (Raspberry Pi 3 B+)		1	

Micro SD 卡	SanDisk 16GB Ultra microSDXC UHS-I Memory Card with Adapter - 98MB/s, C10, U1, Full HD, A1	1	
RC522 RFID 感應器/標籤 (tag)	Power Voltage : 3.3V, Operating frequency: 13.56MHz Read Range: 0 ~ 60mm (mifare1 card), Module Interface SPI Parameters The MF522-AN module design the circuit of card read by using the original Philips MFRC522 chip.	1	
電源	電源 5V 2.5A Micro USB 充電器	1	
跳線	彩色杜邦雙頭線 (母/母)/20 cm		
蜂鳴器	5V 電磁式有源蜂鳴器長音	1	

## ■ 樹莓派 (Raspberry Pi 3 B+)

Raspberry Pi 配備的通用的輸入/輸出接腳(General Purpose Input/Output，簡稱 GPIO), 主要用來連接周邊的電子零件與設備, 透過 GPIO 介面與程式, Raspberry Pi 可以輕鬆控制周邊的電子設備。目前 Raspberry Pi 2 或 3 Model B 型號的 GPIO 共有 40 個針腳, 其編號與名稱如下圖所示。主要是依據 Broadcom(BCM)CPU 的針腳而定義, 例如實際針腳 7(pin) 對應 GPIO 4。

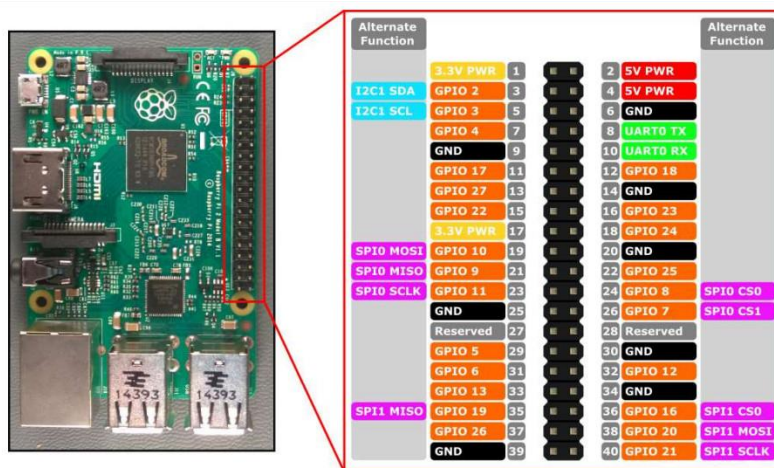


圖 1 樹莓派 3 B+ 針腳圖

注意：要驗證此針腳圖，打開一個終端並輸入指令：`gpio readall`

```
pi@sensor1:~/dht11-raspib3 $ gpio readall
```

BCM	wPi	Name	Mode	V	Physical	V	Mode	Name	wPi	BCM
2	8	3.3v			1	2		5v		
3	9	SDA.1	ALT0	1	3	4		5v		
4	7	SCL.1	ALT0	1	5	6		0v		
		GPIO. 7	IN	1	7	8	0	IN	15	14
		0v			9	10	1	IN	16	15
17	0	GPIO. 0	IN	0	11	12	0	IN	1	18
27	2	GPIO. 2	IN	0	13	14		0v		
22	3	GPIO. 3	IN	0	15	16	0	IN	4	23
		3.3v			17	18	0	IN	5	24
10	12	MOSI	ALT0	0	19	20		0v		
9	13	MISO	ALT0	0	21	22	0	IN	6	25
11	14	SCLK	ALT0	0	23	24	1	OUT	10	8
		0v			25	26	1	OUT	11	7
0	30	SDA.0	IN	1	27	28	1	IN	31	1
5	21	GPIO. 21	IN	1	29	30		0v		
6	22	GPIO. 22	IN	1	31	32	0	IN	26	12
13	23	GPIO. 23	IN	0	33	34		0v		
19	24	GPIO. 24	IN	0	35	36	0	IN	27	16
26	25	GPIO. 25	IN	0	37	38	0	IN	28	20
		0v			39	40	0	IN	29	21

## ■ RFID RC522

RFID RC522 是一款基於 MFRC522 微控制器的低成本 RFID（射頻識別）感應器。該微控制器透過 SPI 協定提供數據，主要工作頻率範圍在 13.56MHz 射頻場域(RF-field)，去感應範圍內的 RFID 標籤。這裡將展示如何連接 RC522 以及如何編寫 Python 腳本讓你可以輕鬆讀取和寫入你的 RFID 標籤。

購買 RFID RC522 感應器時，會注意到的一件事是 9 成以上的感應器接腳是沒有事先焊接好的,你必須自己完成焊接;幸好焊接接腳工作,對一般初學者而言還算相當簡單的任務。如果你完全不想焊接,則可請朋友幫忙或請賣家是否可幫忙焊接好等。

**步驟 1:** 將接腳穿過 RC522 的孔,將接腳緊緊固定,使其更容易焊接到 RFID RC522 電路。

**步驟 2:** 現在使用熱烙鐵和一些焊料，慢慢焊接每個接腳。請記住，最好在對焊點施加焊接之前稍微加熱接頭，這樣可以確保焊料更多地粘附在接頭上，從而減少產生冷接頭的可能性。還建議你小心使用的焊料量。

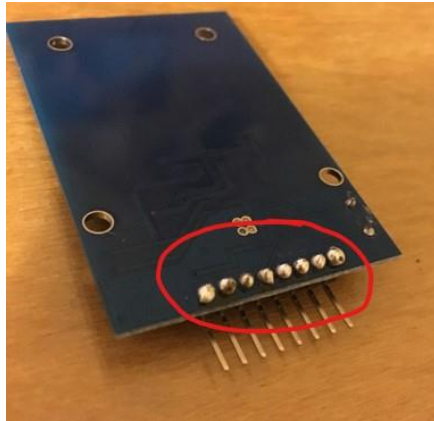


圖 2 焊接好的接腳

## 安裝 RFID 感應器

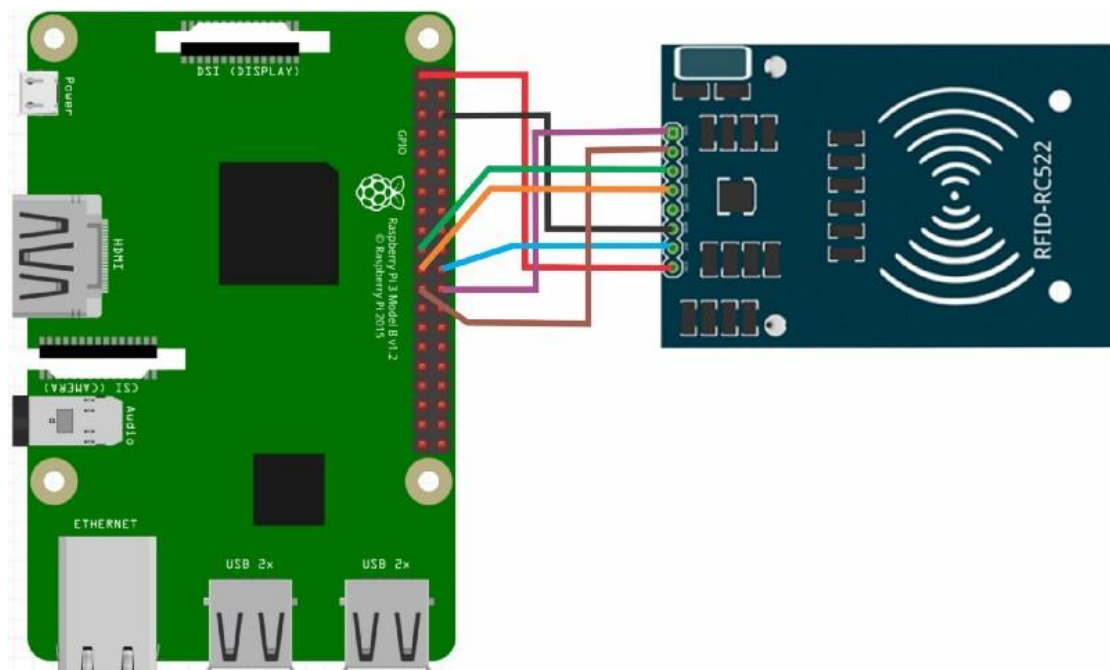


圖 3 RFID-RC52 感應器與樹莓派接線完成圖

在 RFID RC522 上,你會發現到它上面有 8 個可能的連接,分別是 SDA (串列數據信號), SCK (串列時鐘), MOSI (主機輸出,從機輸入), MISO (主機輸入,從機輸出), IRQ (中斷請求), GND (接地), RST (復位電路) 和 3.3v (3.3v 電源輸入)。

我們需要將 RFID RC522 這些連接到樹莓派的 GPIO 接腳上。連接非常簡單,只需將 7 個 GPIO 接腳直接連接到 RFID 感應器即可。請按照下表查看樹莓派接腳與連接的 RC522 接腳的位置。

RC522 接腳	樹莓派接腳 (PCB 板編號)
SDA	Pin 24
SCK	Pin 23
MOSI	Pin 19
MISO	Pin 21
GND	Pin 6
RST	Pin 22
3.3V	Pin 1

當 RFID 感應器接收到感應時蜂鳴器則會發出嗶聲,具提示作用。

蜂鳴器接腳	樹莓派接腳 (PCB 板編號)
GND	Pin 9
+	Pin 16

## 檢查是否啟用 SPI 介面：

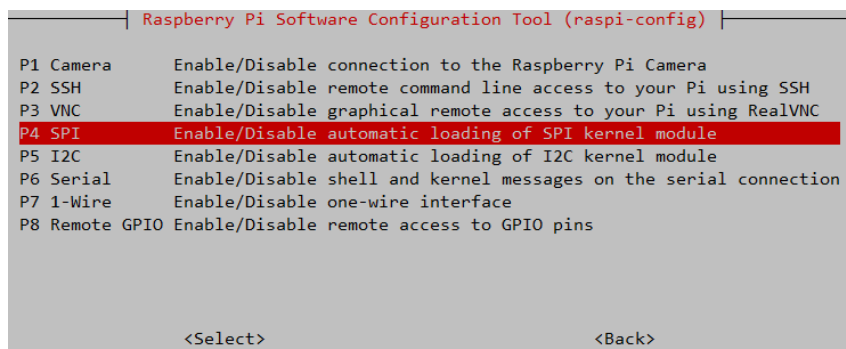
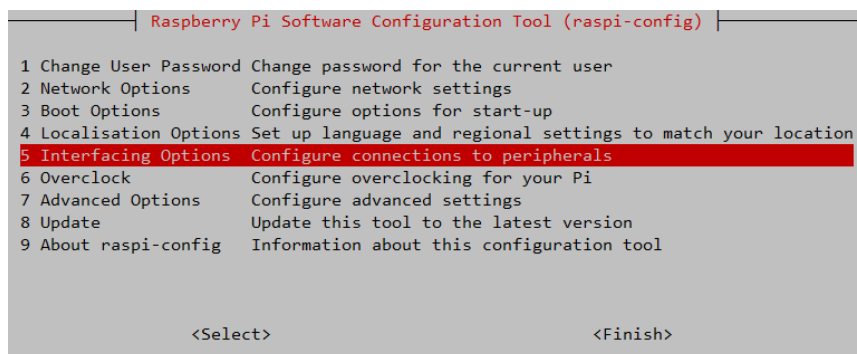
在使用 RFID RC522 之前，我們必須對樹莓派系統設定進行更改。預設情況下樹莓派是停用禁 SPI 功能的。只要啟用該介面即可。

啟用 SPI 介面非常簡單，只需按照以下步驟配置：

**步驟 1:** 首先執行 `raspi-config` 命令工具。

```
sudo raspi-config
```

**步驟 2:** 點選【5 Interfacing Options】→選擇【SPI】啟用 SPI 介面即可,如下所示。



在 SPI 介面完全啟用之前，我們首先必須重新啟動系統,執行以下指令。

```
sudo reboot
```

**步驟 3:** 一旦系統完成重啟，我們現在可以檢查以確保它實際上已啟用。最簡單的方法是執行以下指令以查看是否列出了 `spi_bcm2835`。

樹莓派重啟後，檢查是否列出了 `spi_bcm2835`：

```
pi@rasp-iota:~$ lsmod | grep spi
spidev                16384  0
spi_bcm2835           16384  0
```



## 更新系統：

登入樹莓派並打開終端機,使用以下指令更新

```
sudo apt-get update && sudo apt-get upgrade -y  
sudo reboot
```

現在我們已經將 RFID RC522 連接到樹莓派，可以啟動它並開始用 Python 編寫如何從 RFID 讀取數據以及將數據寫入到 RFID Tag 簡單程式並將數據永久記錄到 IOTA Tange 網路上。

## 安裝 python3 套件：

```
sudo apt-get install python3-dev python3-pip
```

我們必須首先使用我們在上一步中下載的 python **pip** 工具將 Python Library **spidev** 安裝到我們的樹莓派中。該 **spidev** 函式庫主要功能是與 SPI 介面相互溝通處理的。

執行以下指令,透過 pip 安裝 **spidev**。

```
sudo pip3 install spidev
```

接下來使用 pip 繼續安裝 MFRC522 函式庫。

```
sudo pip3 install mfrc522
```

## 編寫一個小程式讀取 RFID 標籤：

我們的第一個簡單 Python 程式,是教大家如何讀取 RFID 標籤數據。

新增 rfidreader.py 檔案

```
sudo nano rfidreader.py
```

內容如下所示

```
#!/usr/bin/env python
```

```
import RPi.GPIO as GPIO
```

```
import sys
```

```
from mfrc522 import SimpleMFRC522
```

```
from time import sleep
```

```
#Select GPIO mode
```

```
GPIO.setmode(GPIO.BOARD)

#Set buzzer - pin 16 as output
buzzer=16
GPIO.setup(buzzer,GPIO.OUT)

reader = SimpleMFRC522()

print("Hold a Tag near the Reader")

try:
    id, text = reader.read()
    print(id)
    print(text)

    GPIO.output(buzzer,GPIO.HIGH)
    sleep(0.2)    # Delay in seconds
    GPIO.output(buzzer,GPIO.LOW)
    sleep(0.2)

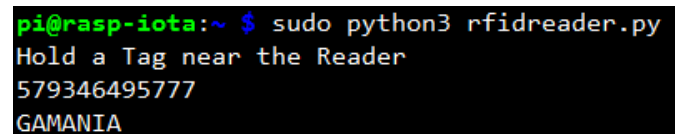
finally:
    GPIO.cleanup()
```

完成後按【CTRL+X】離開存檔。

執行：

```
sudo python3 rfidreader.py
```

將 RFID 標籤去觸動 RFID 感應器,會顯示如下結果：



```
pi@rasp-iota:~ $ sudo python3 rfidreader.py
Hold a Tag near the Reader
579346495777
GAMANIA
```

## 再編寫一個寫入 RFID 標籤小程式：

現在我們已經編寫了使用 RC522 讀取 RFID 標籤的程式，現在我們再編寫一支 python 程式將數據從 RC522 寫入你的 RFID 標籤。



新增 rfidwrite.py 檔案

```
sudo nano rfidwrite.py
```

內容如下所示

```
#!/usr/bin/env python
```

```
import RPi.GPIO as GPIO
```

```
import sys
```

```
from mfrc522 import SimpleMFRC522
```

```
from time import sleep
```

```
#Select GPIO mode
```

```
GPIO.setmode(GPIO.BOARD)
```

```
#Set buzzer - pin 16 as output
```

```
buzzer=16
```

```
GPIO.setup(buzzer,GPIO.OUT)
```

```
reader = SimpleMFRC522()
```

```
try:
```

```
    while True:
```

```
        text = input('Your Name: ')
```

```
        print("Now place your Tag to write")
```

```
        id, text = reader.write(text)
```

```
        print("recorded")
```

```
        print(id)
```

```
        print(text)
```

```
        GPIO.output(buzzer,GPIO.HIGH)
```

```
        sleep(0.2)  # Delay in seconds
```

```
        GPIO.output(buzzer,GPIO.LOW)
```

```
        sleep(0.2)
```

```
        break
```

```
finally:
```

```
    GPIO.cleanup()
```

完成後按【CTRL+X】離開存檔。

執行：

```
sudo python3 rfidwrite.py
```

將 RFID 標籤去觸動 RFID 感應器,會顯示如下結果：

```
pi@rasp-iota:~ $ sudo python3 rfidwrite.py
Your Name: GAMANIA
Now place your Tag to write
recorded
579346495777
GAMANIA
```

## 將 RFID 標籤數據記錄在 Tangle 上

要與 IOTA Tangle 進行溝通，你需要安裝 PyOTA 函式庫  
(<https://github.com/iotaledger/iota.lib.py>)

```
sudo apt-get install libffi-dev
```

```
sudo pip3 install pyota[ccurl]
```

```
git clone https://github.com/iotaledger/iota.lib.py.git
```

記錄數據到 Tangle 對於此步驟，你將需要一個 IOTA 地址。為此，我建議在你的計算機上安裝 trinity 錢包 (<https://trinity.iota.org>) 並取得新地址。

## 使用 RFID 進行投票:

為了在 Tangle 上記錄數據，我們將編寫一個小程序，要求用戶利用 RFID 感應進行投票（是或否），然後將 RFID 標籤號碼和名稱一起記錄在 Tangle 上。

新增 vote.py 檔案

```
sudo nano vote.py
```

內容如下所示

```
### Import libraries
from datetime import datetime
import json
import iota
import sys
import RPi.GPIO as GPIO
from mfrc522 import SimpleMFRC522
```

```

from time import sleep

#Select GPIO mode
GPIO.setmode(GPIO.BOARD)

#Set buzzer - pin 16 as output
buzzer=16
GPIO.setup(buzzer,GPIO.OUT)

### IOTA address
IOTAAddr =
b"QBSETYBUTTJCI9SDVQNHWGXPXTRN9HPHXCQUFETVHJ9EAMK9TISAJOAVMKSYOJILCDLBZ
HMGQTXZVWFIGCYHWZYFEAY"

### IOTA full node
api = iota.lota("https://nodes.thetangle.org:443")

project = "Voting via RFID"
reader = SimpleMFRC522()

try:
    while True:
        print("\nIOTA Project Voting")

### only yes or no allowed
        while True:
            casted_vote = input("\nCast your vote (YES/NO) and hit Enter: ").lower()
            if casted_vote == "yes":
                print("You voted YES")
                break
            elif casted_vote == "no":
                print("You voted NO")
                break

        print("\nThank you, now hold your ID card near the reader")

        id, text = reader.read()
        data = {'tagID': str(id), 'tagText':str(text), 'project': project, 'casted_vote':

```

```
casted_vote}
```

```
pt = iota.ProposedTransaction(address = iota.Address(IOTAAddr),
                               message =
iota.TryteString.from_unicode(json.dumps(data)),
                               tag      = iota.Tag(b'IOTA999RFID999VOTE'),
                               value    = 0)

GPIO.output(buzzer,GPIO.HIGH)
sleep(0.2) # Delay in seconds
GPIO.output(buzzer,GPIO.LOW)
sleep(0.2)

print("\nID card detected...Sending transaction...Please wait...")

FinalBundle = api.send_transfer(depth=3, transfers=[pt],
min_weight_magnitude=14)['bundle']

print("\nTransaction sucessfully recorded")

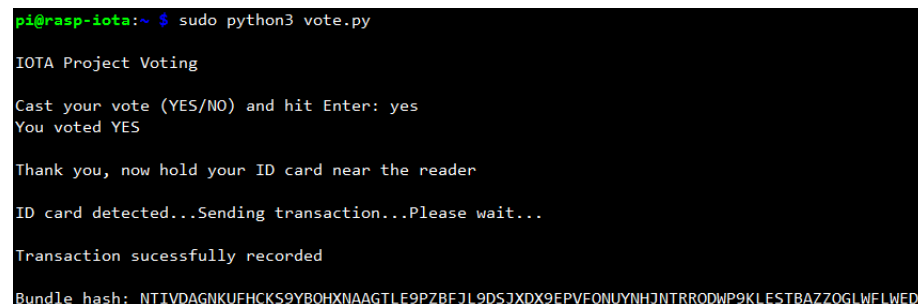
#bundle is broadcasted, let's print it
print("\nBundle hash: %s" % (FinalBundle.hash))
break

finally:
    GPIO.cleanup()
```

完成後按【CTRL+X】離開存檔。

現在讓我們執行投票：

```
sudo python3 vote.py
```



```
pi@rasp-iota:~$ sudo python3 vote.py
IOTA Project Voting
Cast your vote (YES/NO) and hit Enter: yes
You voted YES

Thank you, now hold your ID card near the reader

ID card detected...Sending transaction...Please wait...

Transaction sucessfully recorded

Bundle hash: NTIVDAGNKUFHCKS9YBOHXNAAGTLE9PZBFJL9DSJDX9EPVFONUYNHJNTRRODWP9KLESTBAZZOGLWFLWED
```

你現在可以使用 IOTA Explorer（例如 [thetangle.org](https://thetangle.org)）並檢查數據是否已成功記錄。  
可以用 Address 或 Bundle hash 輸入查詢，如下圖以 Bundle hash 所示。

### Bundle

NTIVDAGNKUFHCKS9YBOHXNAAGTLE9PZBFJL9DSJDX9EPVFONUYNHJNTRRODWP9KLESTBAZZOGLWFLWED

May 19, 2019 21:13:48 - 5 minutes and 39 seconds ago

Confirmed

0 inputs

1 outputs

→ QBSEYBUTTJCI9SDVQNHWGXPXTRN9HPHXQCQFETVHJ9EAMK9TISAJOAVMKSJOJILC  
DLBZHMGTZXVWFIGC  
NROULRAVV9CBUXOVIFENJDKJAOWZXSWFWMACWIAFDPUJXITBAJWC  
MTWNRERZWSDTUFIHPIR9CV9HA9999

### Transaction

NROULRAVV9CBUXOVIFENJDKJAOWZXSWFWMACWIAFDPUJXITBAJWCMTWNRERZWSDTUFIHPIR9CV9HA9999

May 19, 2019 21:13:48 - 6 minutes and 15 seconds ago

Confirmed on 2019-05-19 at 21:17:43

Value 0 I

Conversion 0 USD

Index in bundle 0 / 0

Tag IOTA999RFID999VOTE999999999

Weight magnitude 14

Address QBSEYBUTTJCI9SDVQNHWGXPXTRN9HPHXQCQFETVHJ9EAMK9TISAJOAVMKSJOJILC  
DLBZHMGTZXVWFIGCYHWZYFEAY

Bundle NTIVDAGNKUFHCKS9YBOHXNAAGTLE9PZBFJL9DSJDX9EPVFONUYNHJNTRRODWP9KLESTBAZZOGLWFLWED

Nonce POWSRVIO9FREE9AUEVMTMPPNNNN

Message

Trytes "tagText": "GAMANIA "

Text "casted\_vote": "yes"

Text "project": "Voting via RFID"

JSON "tagID": "579346495777"