

1r. Connexions a Internet provades

Per connectar la Raspberry Pi 2W a l'entorn de desenvolupament i accedir de forma remota al sistema, es van provar diferents mètodes de connexió, cadascun amb les seves particularitats i desafiaments.

1.1 Connexió Serial (UART) – Intent fallit

Inicialment, es va intentar establir una connexió a través del port serial utilitzant UART (Universal Asynchronous Receiver-Transmitter). La Raspberry Pi 2W compta amb pins GPIO que permeten comunicació serial mitjançant un adaptador USB a TTL (com un conversor basat en el xip CP2102).

Diferències entre els ports de la Raspberry Pi 2W

La Raspberry Pi 2W disposa de diferents ports amb funcions específiques:

- **Port USB On-The-Go (OTG):** Permet connectar dispositius com teclats, ratolins i adaptadors de xarxa, a més de servir per a comunicació de dades si es configura adequadament.
- **UART en GPIO (pins 14 i 15 – TX i RX):** Aquest port permet establir comunicació serial, però requereix habilitar-se en la configuració i connectar un adaptador compatible.

Problemes trobats en la connexió UART

- **Sense resposta al terminal:** Tot i connectar correctament els pins TX, RX i GND de l'adaptador a la Raspberry Pi, la terminal no mostrava sortida de dades.
- **Possible deshabilitació del port en firmware:** Es va verificar que en moltes versions recents de Raspberry Pi OS, el port UART està deshabilitat per defecte en /boot/config.txt. Es va intentar habilitar agregant enable_uart=1, però la connexió seguia sense funcionar.
- **Baudrate incorrecte:** Es van realitzar proves amb diferents baudrats (9600, 115200), sent el 115200 el correcte. Sense obtenir una resposta satisfactòria.

A causa d'aquests problemes i a la falta d'un senyal estable, es va decidir canviar l'enfocament i provar amb altres mètodes.

1.2 Connexió mitjançant PuTTY – Intent fallit

Es va intentar accedir a la Raspberry Pi a través de PuTTY, configurant la connexió serial amb els paràmetres estàndard (115200 8N1 i sense control de flux). No

obstant això, al no haver-hi una resposta adequada des del port serial, aquest mètode tampoc va funcionar.

1.3 Connexió SSH (sense USB i via putty) – Èxit

Finalment, es va optar per la connexió mitjançant SSH a través de la xarxa, evitant l'ús del port USB o UART. Es va configurar la Raspberry Pi 2W per habilitar l'accés remot a través del port 22.

Procés i verificació:

- Es va comprovar l'assignació d'IP a la xarxa local (ifconfig a la Raspberry Pi o escanejant dispositius al router).
- Es va realitzar una prova de connectivitat amb ping des d'un altre equip a la xarxa.
- Es va intentar accedir amb `ssh bruno@bruno.local`, aconseguint establir connexió.
- Es va verificar que el port 22 estava obert i accessible amb `telnet IP-RaspberryPi 22`, confirmant que el servei SSH estava operatiu.

2. Configuracions realitzades a la Raspberry Pi

Per a la implementació del lector RFID RC522 a la Raspberry Pi 2W, es van provar diferents biblioteques de Ruby abans d'optar per una solució funcional.

2.1 Biblioteques de Ruby provades

Durant el desenvolupament, es van analitzar diverses biblioteques compatibles amb la Raspberry Pi i el lector RC522:

- **mfr522 (utilitzada en la implementació final):** Biblioteca específica per al control del lector RFID RC522 a Ruby. Simplifica la comunicació SPI i facilita l'obtenció de l'UID de les targetes.
- **pi_piper:** Biblioteca per a manipulació de GPIOs a Ruby. Es va considerar com a alternativa per gestionar el bus SPI manualment, però requeria una implementació més complexa.
- **wiringpi:** Adaptació de la popular llibreria WiringPi a Ruby. Tot i que permet control avançat de pins i comunicació SPI, el seu manteniment està descontinuado i es va descartar per possibles incompatibilitats.

2.2 Configuració final a la Raspberry Pi

Després d'avaluar les opcions, es va optar per **522**, ja que facilitava la integració amb Ruby i permetia la lectura de targetes RFID de manera eficient.

Perquè la Raspberry Pi reconegués el lector RC522 correctament, es van realitzar els següents ajustos:

- **Habilitació del SPI** a través de raspi-config.
- **Instal·lació de gemmes i dependències** necessàries per a la biblioteca.
- **Verificació de la comunicació SPI** amb `ls /dev/spidev*` per assegurar-se que la Raspberry Pi detectava el bus correctament.

3. Problemes trobats

Durant la implementació del lector RFID RC522 a la Raspberry Pi 2W, van sorgir diversos desafiaments tècnics relacionats amb l'alimentació, la comunicació i la configuració dels pins.

- **Manca d'alimentació suficient:** Inicialment, la Raspberry Pi 2W s'alimentava des del port USB d'un portàtil, cosa que no proporcionava prou corrent per alimentar tant la placa com el lector RFID RC522. Es va solucionar utilitzant una font d'alimentació amb més amperatge.
- **Problemes amb la connexió serial:** La comunicació via UART va presentar dificultats, la qual cosa va portar a optar per la connexió SSH per configurar el sistema de forma remota.
- **Absència de pantalla:** La falta d'un monitor connectat a la Raspberry Pi va dificultar la depuració inicial, obligant a fer servir accés remot i registres de logs per detectar errors.
- **Dificultat en la soldadura de pins:** La Raspberry Pi 2W no compta amb pins GPIO preinstal·lats, per la qual cosa va ser necessari soldar-los manualment abans de connectar el lector RFID RC522. Es va requerir precisió per evitar errors de connexió o ponts no desitjats.

3.1 Assignació de pins i la seva funció

Per connectar el lector RFID RC522 a la Raspberry Pi 2W, es va establir la següent assignació de pins, segons la funcionalitat de cadascú en el protocol SPI:

Pin RC522 Pin Raspberry Pi (GPIO) Funció

VCC	3.3V (Pin 1)	Alimentació
GND	GND (Pin 6)	Terra
RST	GPIO 25 (Pin 22)	Reset del mòdul
SDA (SS)	GPIO 8 (Pin 24)	Selecció de xip (CS)

Pin RC522 Pin Raspberry Pi (GPIO) Funció

SCK	GPIO 11 (Pin 23)	Relloige SPI (SCLK)
MOSI	GPIO 10 (Pin 19)	Dades de sortida (MOSI)
MISO	GPIO 9 (Pin 21)	Dades d'entrada (MISO)

3.2 Configuració en Ruby

Per interactuar amb aquests pins, es va utilitzar la biblioteca `mfr522`, que simplifica la comunicació SPI a Ruby. En inicialitzar el mòdul, la biblioteca s'encarrega de gestionar la transferència de dades amb la Raspberry Pi.

Aquí, `mfr.picc_request` activa la comunicació amb la targeta RFID, i `picc_select` obté l'UID de la targeta detectada.

Malgrat els problemes inicials, un cop configurada l'alimentació, la soldadura de pins i l'assignació correcta en el codi, el sistema va quedar completament funcional.

4. Codi implementat i ajustos realitzats

Durant la implementació del codi a Ruby amb la biblioteca `mfr522`, es van realitzar algunes modificacions per millorar la sortida i evitar informació innecessària en la detecció de targetes RFID.

4.1 Eliminació de p uid i ajust del SAK

Inicialment, el mètode `picc_select` retornava un arrelament amb informació detallada sobre la targeta detectada. Això incloïa el **SAK (Select Acknowledge)**, un byte addicional que sol indicar el tipus de targeta RFID. No obstant això, aquesta dada no era necessària per al propòsit del projecte i a més generava soroll en la sortida del programa.

Per aquesta raó, es va eliminar la línia `p uid`, que imprimia la resposta completa, i es va ajustar el codi per extreure únicament l'UID rellevant.

4.2 Implementació final del codi a Ruby

```
require 'mfr522'

mfr522 = MFRC522.new

if mfr522.picc_request(MFRC522::PICC_REQA)
  uid = mfr522.picc_select
  p uid
  clean_uid = uid[0]
  if uid
    puts "Tarjeta detectada: " + clean_uid.map { |b| "%02X" % b }.join(":")
  end
end
```

4.3 Raó de la retallada del SAK

El SAK és un identificador addicional que algunes targetes RFID envien automàticament com a part del procés de selecció. Tot i que en certs escenaris pot ser útil (com en sistemes on es diferencien targetes MIFARE Classic d'altres variants), en aquesta implementació només es necessitava l'UID per a identificació bàsica.

En remoure el SAK:

- S'obté una sortida més neta i precisa.
- S'eviten problemes d'interpretació de dades innecessàries.
- Es facilita la integració del codi amb altres sistemes que només requereixen l'UID.

Amb aquest ajust, el codi va quedar optimitzat per identificar targetes de manera eficient sense incloure informació redundant.