Internship Notes

Giuseppe Cervone
giuseppe.cervone@edu.unife.it

1 Devices

2 Project details

The aim of the project is to develop what in cybersecurity it's called a data diode, a unidirectional communication device for data exchange. The importance behind this project is that industrial data diodes are expensive devices, and implementing a high amount of them like for this specific use case is a very costly move, but with software implementations (firewalls) or hardware implementations (serial or optical communications) it is possible to have cheaper data diodes that are almost as functional for most use case scenario.

2.1 Ver 0.0.1

The first implementation of a data diode I will try out is a project based on a github repo that is not mantained anymore. https://github.com/thephez/data-diode.

Here below is an image of what I will try to implement in this version 0.0.1.

The machines are configured as follows:

- Raspberry Pi 3 Model B Ver1.2:
  - Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
  - 1GB RAM
  - BCM43438 wireless LAN and Bluetooth Low Energy (BLE) on board
  - 100 Base Ethernet
  - 40-pin extended GPIO
  - 4 USB 2 ports
  - 4 Pole stereo output and composite video port
  - Full size HDMI
  - CSI camera port for connecting a Raspberry Pi camera
  - DSI display port for connecting a Raspberry Pi touchscreen display
  - Micro SD port for loading your operating system and storing data
  - Upgraded switched Micro USB power source up to 2.5A
  - Raspberry Pi OS Lite
  - 16GB microSD card

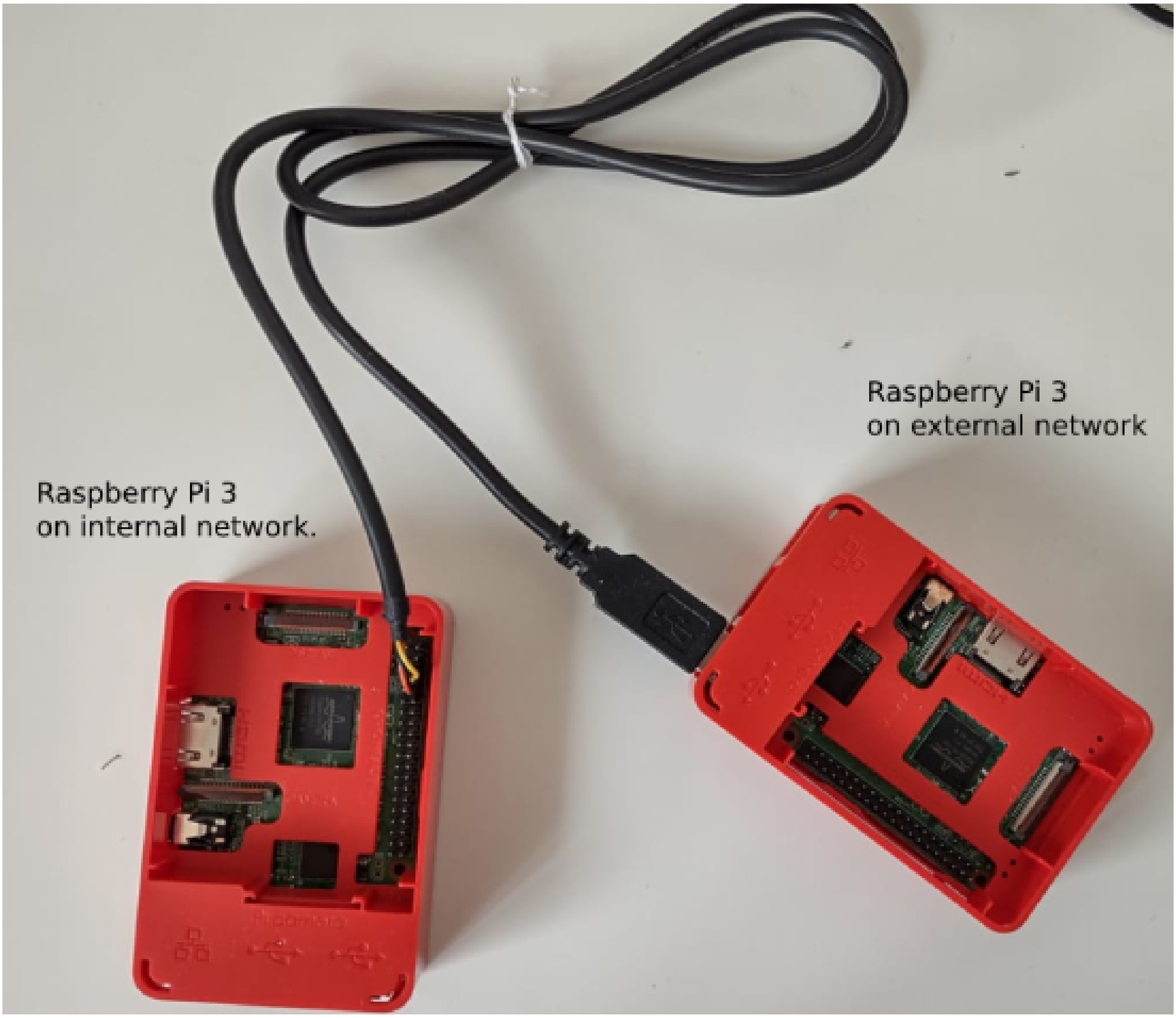

Figure 1: Versione 0.0.1

Using the RPI-Imager tool I installed the latest Raspberry PI OS image. Upon first boot up, remember that raspberry pi login is username: pi, password: raspberry. As Raspberry Pi OS Lite is an OS without a desktop environment, so we can use raspi-config to enable a series of features useful for testing software:

- In the system options:
  - Enable WIFI
  - Change username and password if needed
  - Change hostname to whatever it's preferred. This is useful in the future to enable SSH
- In the interface options:
  - Enable SSH
  - Enable Serial interface, remembering to disable login shell but enable serial interface.
- Run sudo apt-get upgrade to make sure all packages are up to date before using raspi-config
- In my config the SSH internal ip's should be:
  - ssh pi@192.168.1.157 RX
  - ssh pi@192.168.1.198 TX
  - password raspberry

Having the machines setup, I moved on to the first test, sending an "Hello, World!" from Raspberry 1 to Raspberry 2. I will be starting with a serial communication via a TTL-232R-3V3 cable, which is USB to UART, with +3.3V TTL levels UART signals. The cable has 3-pins on one end, and USB on the other. Using GPIO14 pin on the RPi we can trasmit data, using GPIO15 we receive data. Below we can see from the documentation of the cable the correct way to plug in the various cables in the correct pins.

With the cable plugged in we run the command ls /dev/tty* on both raspberry machines so we have an idea of what port we are using on each raspberry in the serial communication (For the port that's sending data it should be /dev/ttyS0, while for the other port it should be USB0).

4.2 TTL-232R-5V and TTL-232R-3V3 Cable Signal Descriptions

Header Pin Number	Name	Type	Colour	Description
1	GND	GND	Black	Device ground supply pin.
2	CTS#	Input	Brown	Clear to Send Control input / Handshake signal.
3	VCC	Output	Red	+5V output,

Header Pin Number	Name	Type	Colour	Description
4	TXD	Output	Orange	Transmit Asynchronous Data output.
5	RXD	Input	Yellow	Receive Asynchronous Data input.
6	RTS#	Output	Green	Request To Send Control Output / Handshake signal.

Table 4.1 TTL-232R-5V and TTL-232R-3V3 Cable Signal Descriptions

Figure 2: Pinout del cavo seriale