2023

Trust M chip & Raspberry Pi

INSTRUCTIONS

AKIYAMA,YUKA; COURTEMARCHE, JACOB; HASENFUS, HUNTER M; RAI, ANIKETH

Introduction

Below is the setup guide for the OPTIGA Trust M Explorer Application, these instructions have been taken from there github page.

Interface Setup

Virtual Monitor setup (Remote Access using VNC)

This step guides you on how to set up the required interface needed to communicate with Trust M. Start up the Raspberry Pi with HDMI cable to monitor and select Preferences->Raspberry Pi Configuration. Select the Interface tab. Enable I2C, SSH and VNC as follows.

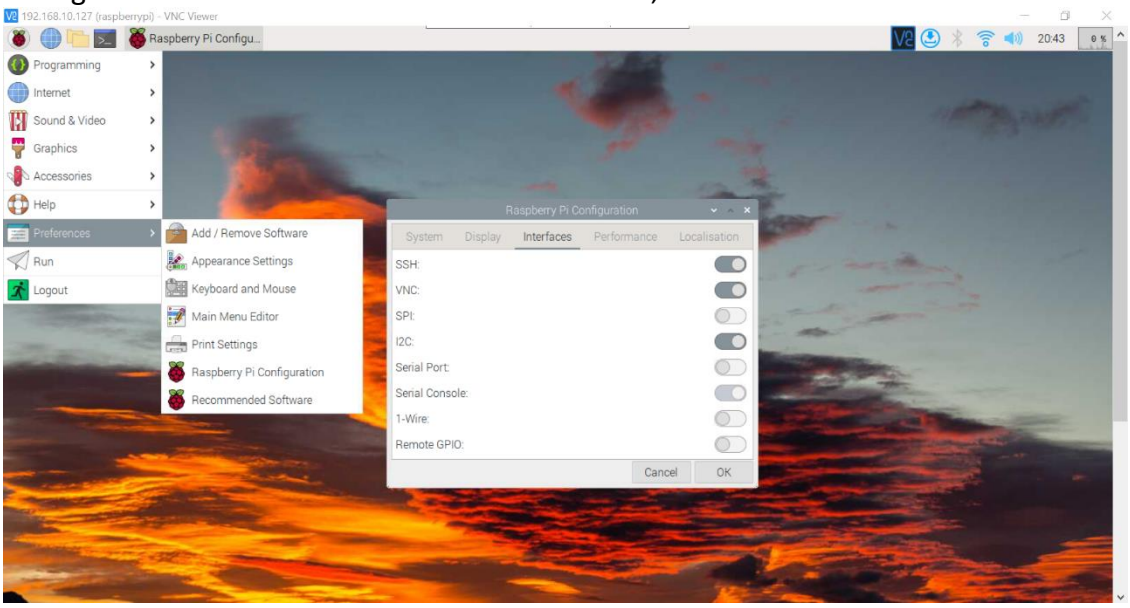


Figure 1: RPI Home Screen on monitor

Enter "hostname -I" into the Terminal and copy the IP address
hostname -I
192.168.###.###
Paste the IP Address of RPI3 into VNC Viewer on the host PC to connect to the RPI.

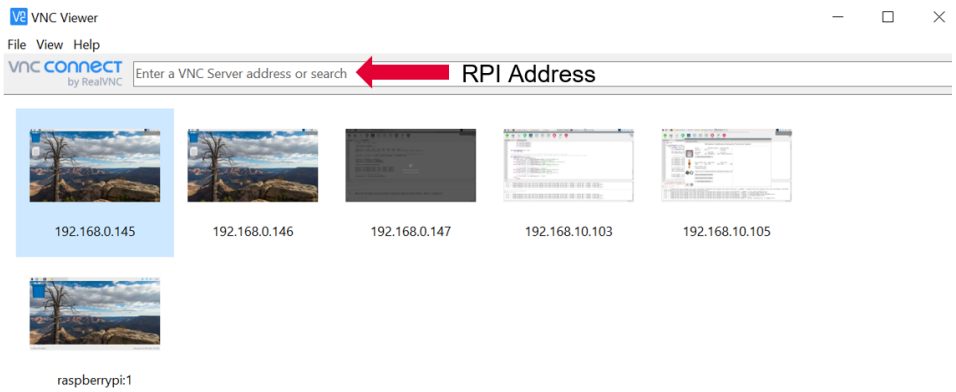


Figure 2: VNC Viewer Connection Screen

Enter the Username and the Password.
Username: pi
The password is the same as the password entered when setting up raspberry pi

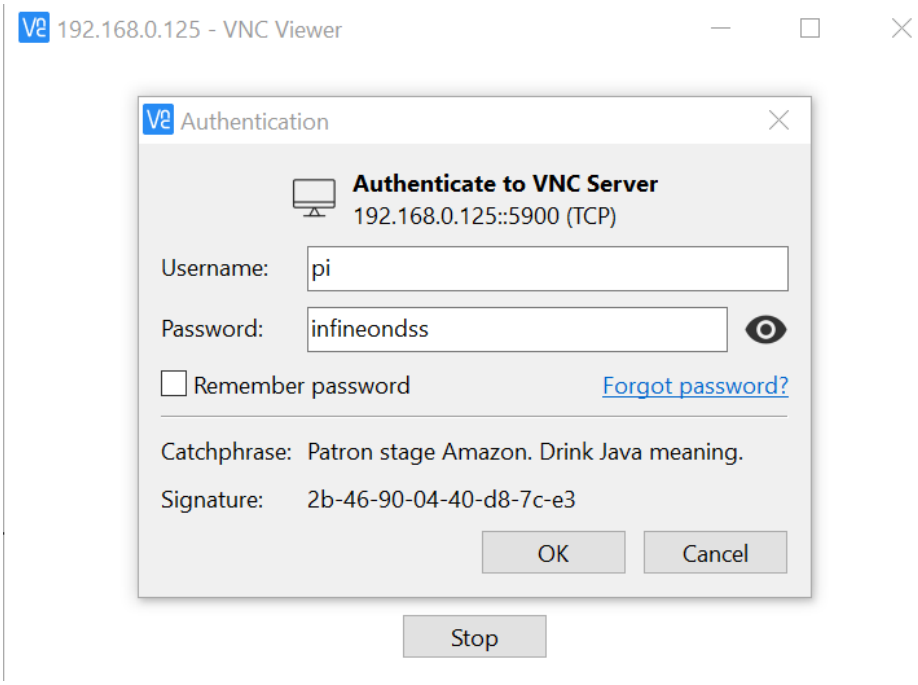


Figure 3: VNC Viewer Authentication Menu

You should be successfully connected and able to view the RPI through VNC connection on your device.

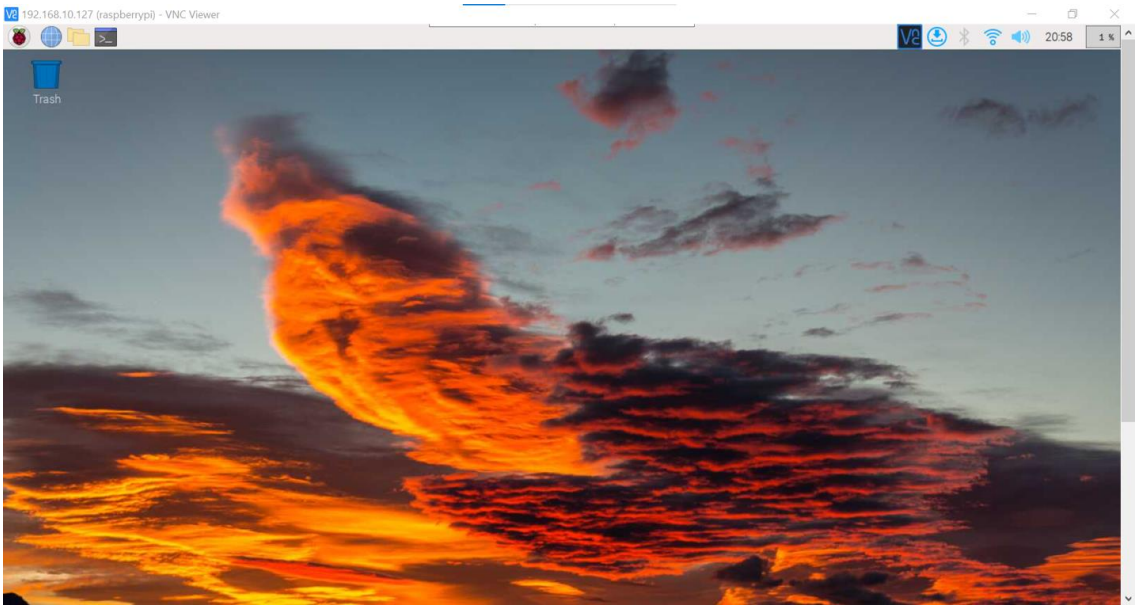


Figure 4: RPI Home Screen on VNC Viewer

Physical Monitor Setup

We have to connect the raspberry pi 4B as we connect any other CPU to the monitor.
A better understanding on how to connect Raspberry pi 4B to the monitor is given in the documents of projects.raspberrypi.org. Here is the link [Setup_RaspberryPI.]
You can follow through the steps and complete the process.

How to connect the Infineon chip to Raspberry pi 4

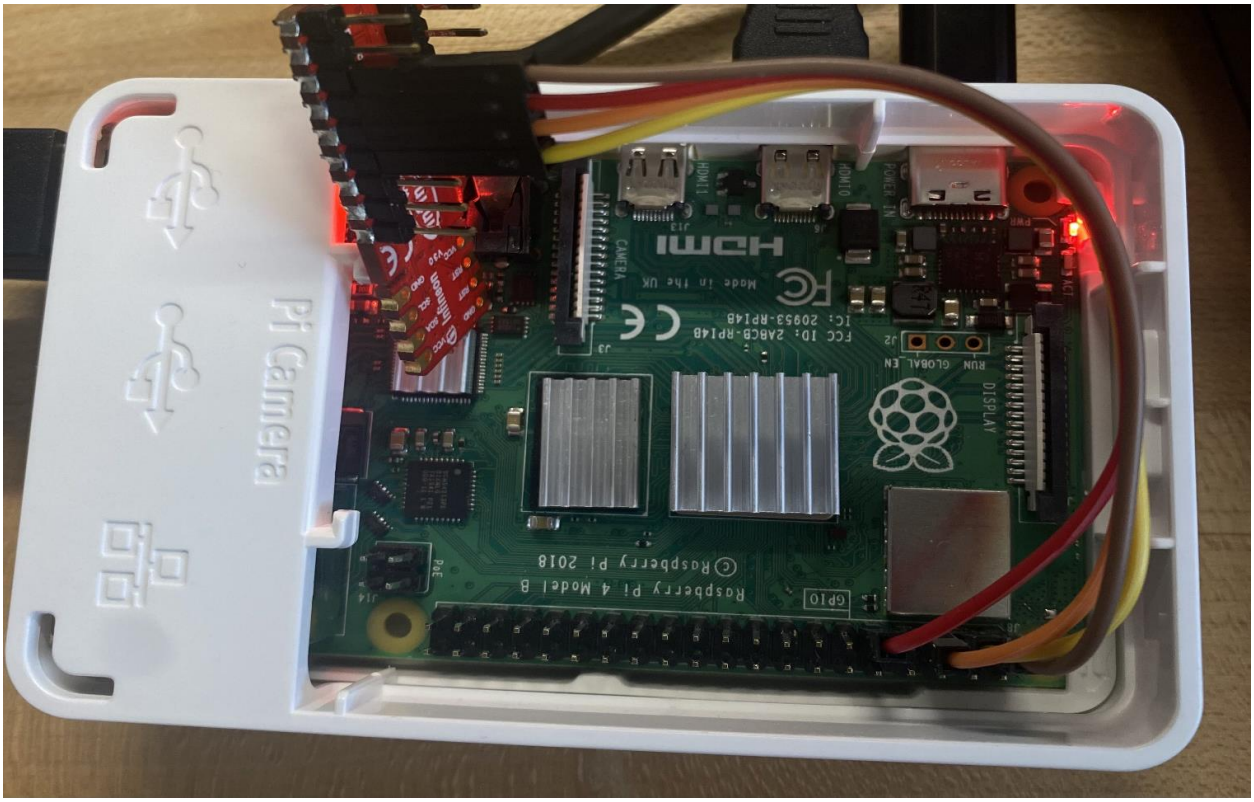


Figure 5: The connection between the Infineon optiga trust m chip and the Raspberry pi 4B

Main Components to connect Infineon and raspberry:

1. 3v3 – 3 volts power supply connection
2. GND – supply voltage Ground
3. SDA (Serial Data line)
4. SCL (Serial Clock line)

SDA and SCL are two communication lines used in the I2C protocol for Infineon OPTIGA Trust M. SDA stands for Serial Data Line, while SCL stands for Serial Clock Line. These lines are used to transmit data between devices connected via I2C. The state of these lines is set to high when VCC is powered up.

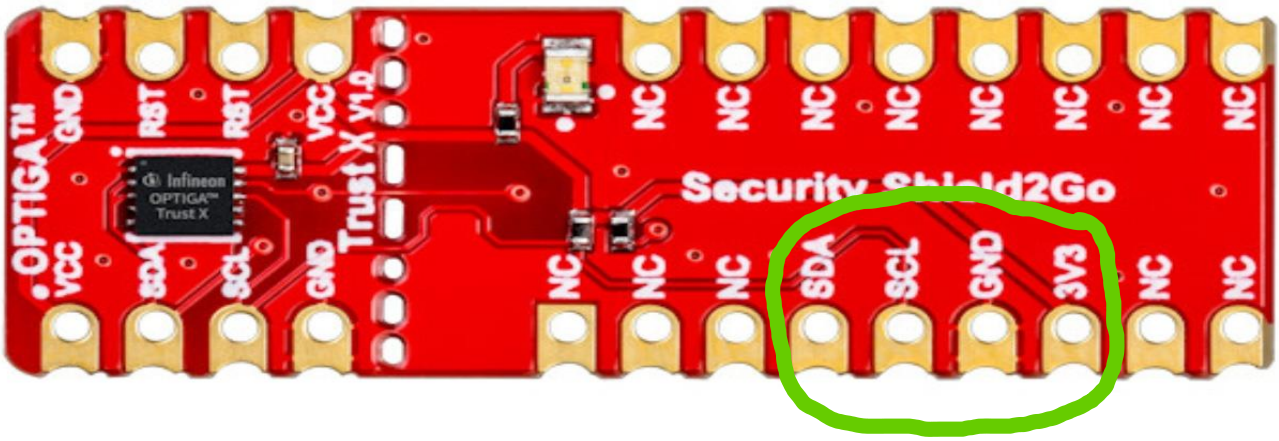


Figure 6: the components used to connect with the raspberry pi 4B

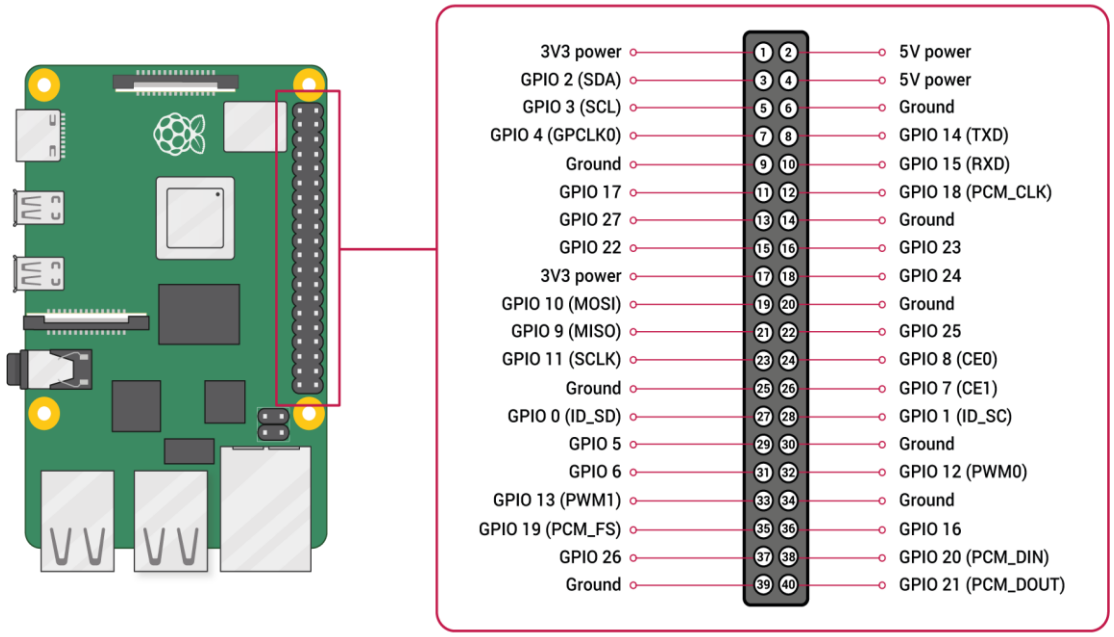


Figure 7: the pin architecture of the Raspberry pi 4B

We utilize the pin number (1,3,5 & 9) for connecting with the optiga trust m chip.
The connection is as follows:

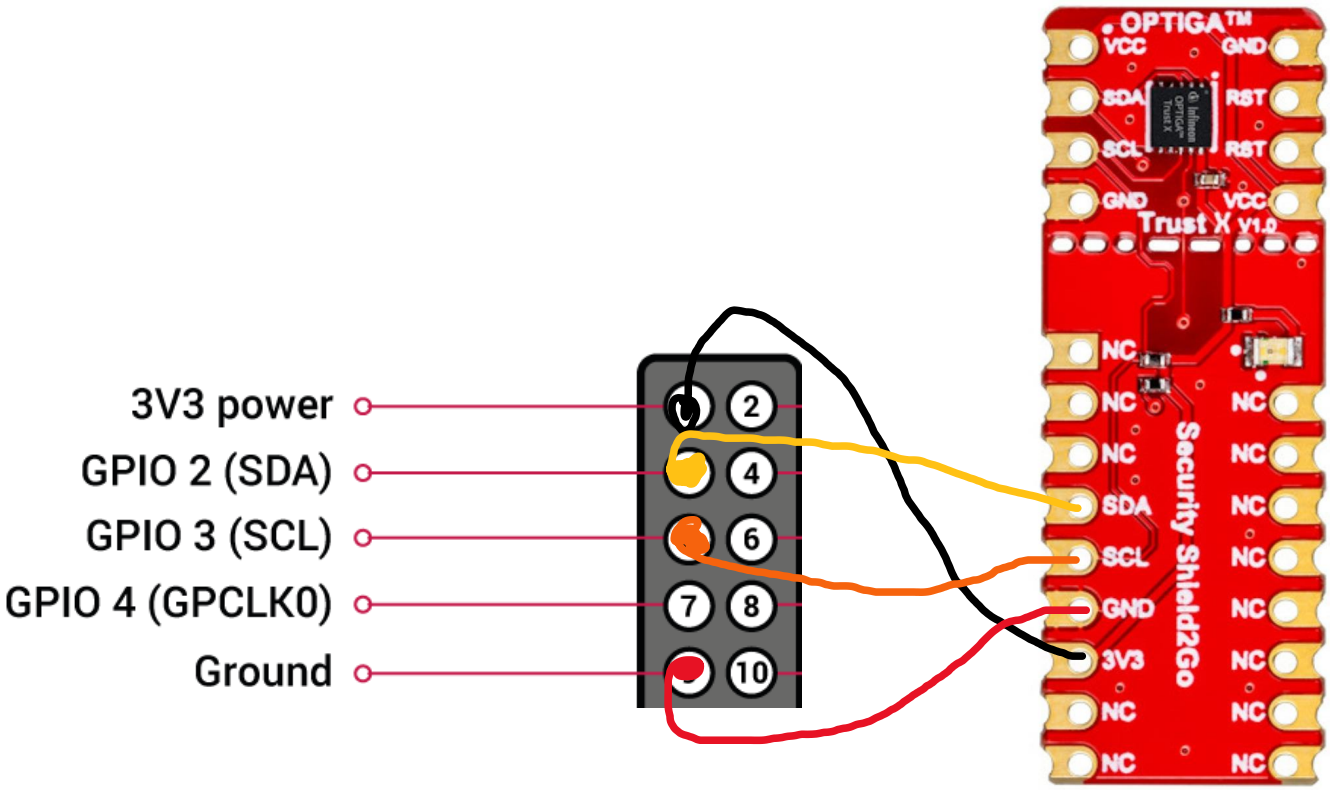


Figure 8: The connection between pins

Install Trust M Explorer

Trust M Explorer Installation Guide

Download Trust M_Explorer Source Code:

git clone --recurse-submodules https://github.com/Infineon/optiga-trust-m-explorer

Go to the following directory

cd optiga-trust-m-explorer

Execute Installation script:

./trust_m_installation_script.sh

To start the Trust M Explorer Application

Go to directory "optiga-trust-m-explorer/Python_TrustM_GUI"

./start_gui.sh

The installation script installs the following dependencies required and compiles the source code for the OPTIGA™ Trust M Explorer Application.

- python-wxtools
- OpenSSL development library (libssl-dev)
- OpenSSL 1.1.1d
- OPTIGA Trust M1/M3 library (source code)
- pthread

- rt
- PyPubSub

This process should take up to 15 minutes.
Once complete, go to your home directory and access the folder called optiga-trust-m-explorer.

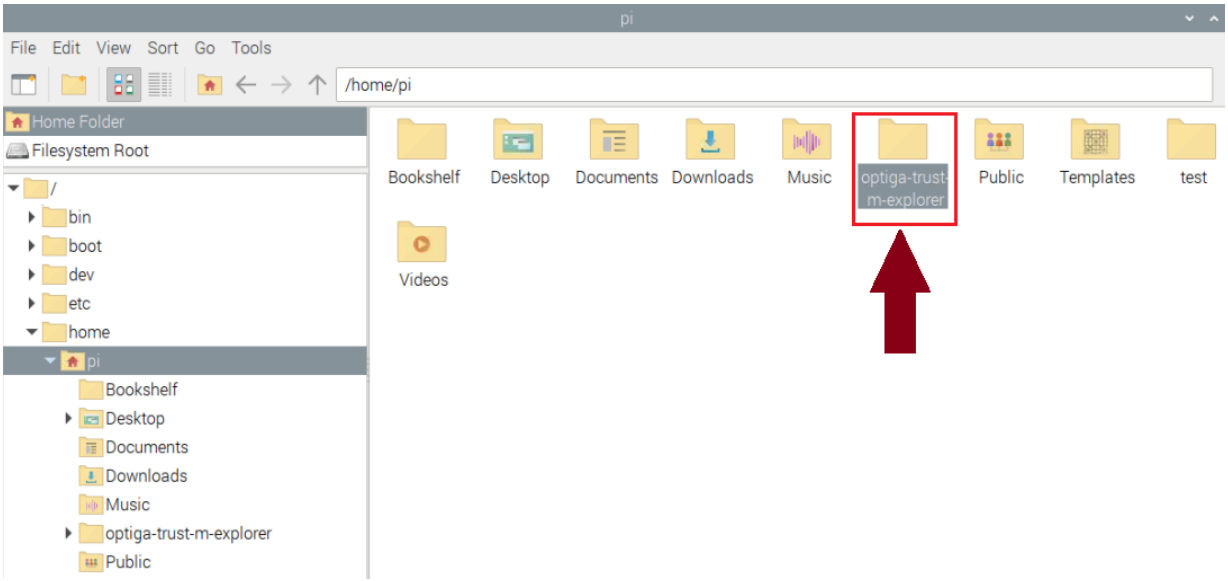


Figure 9: Trust M Explorer File Directory
Next, access the file called Python TrustM_GUI.

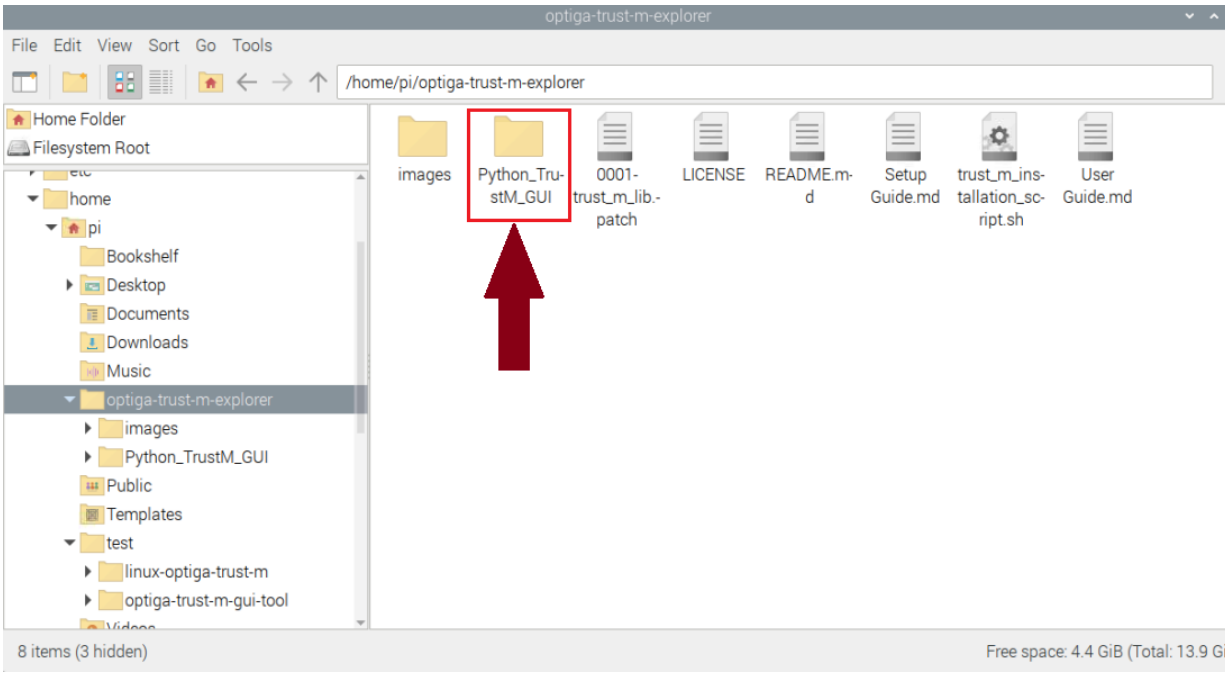


Figure 9: Python Trust M GUI File Directory
Execute "start_gui.sh" and select execute in terminal.

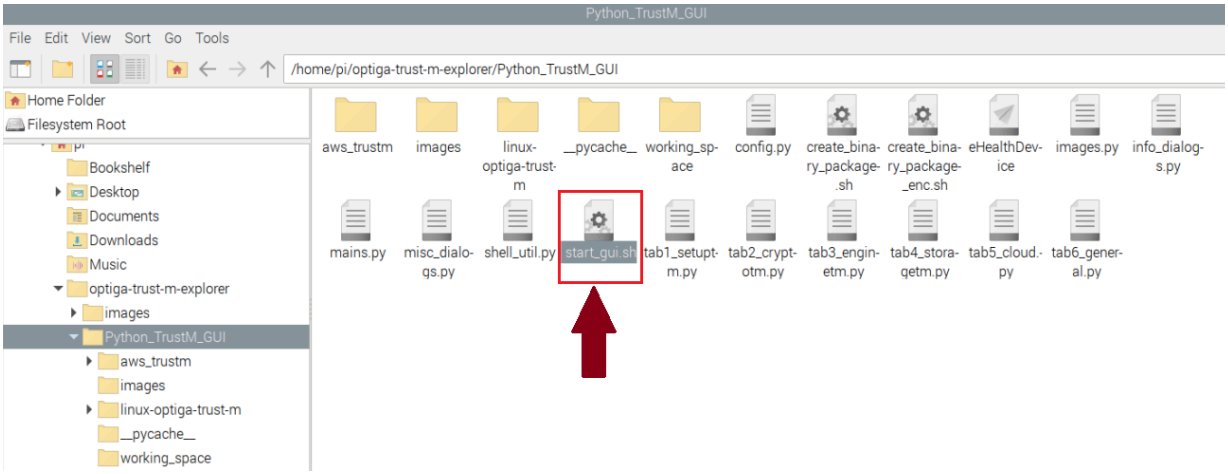


Figure 10: Selecting start_gui.sh

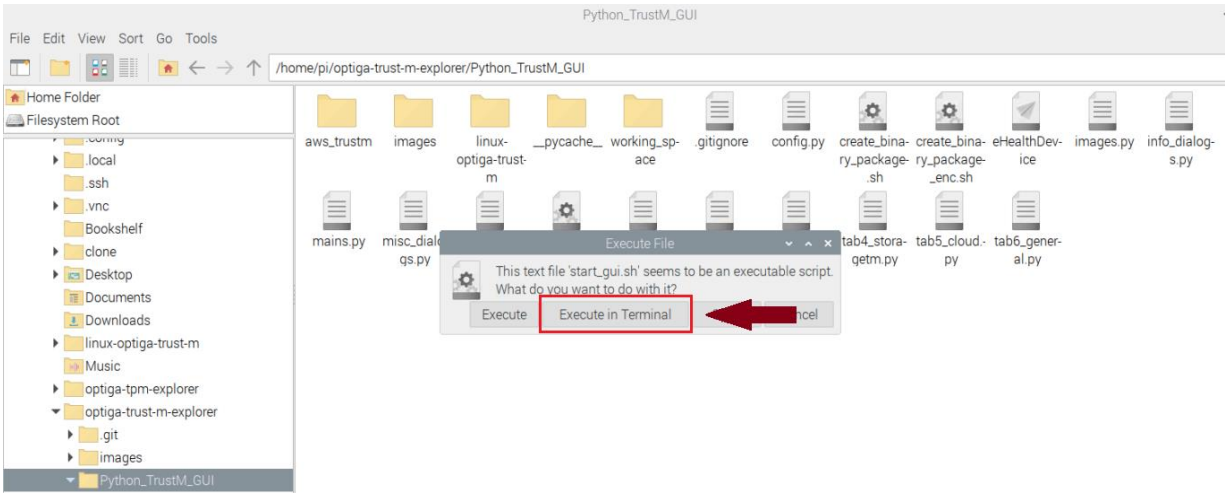
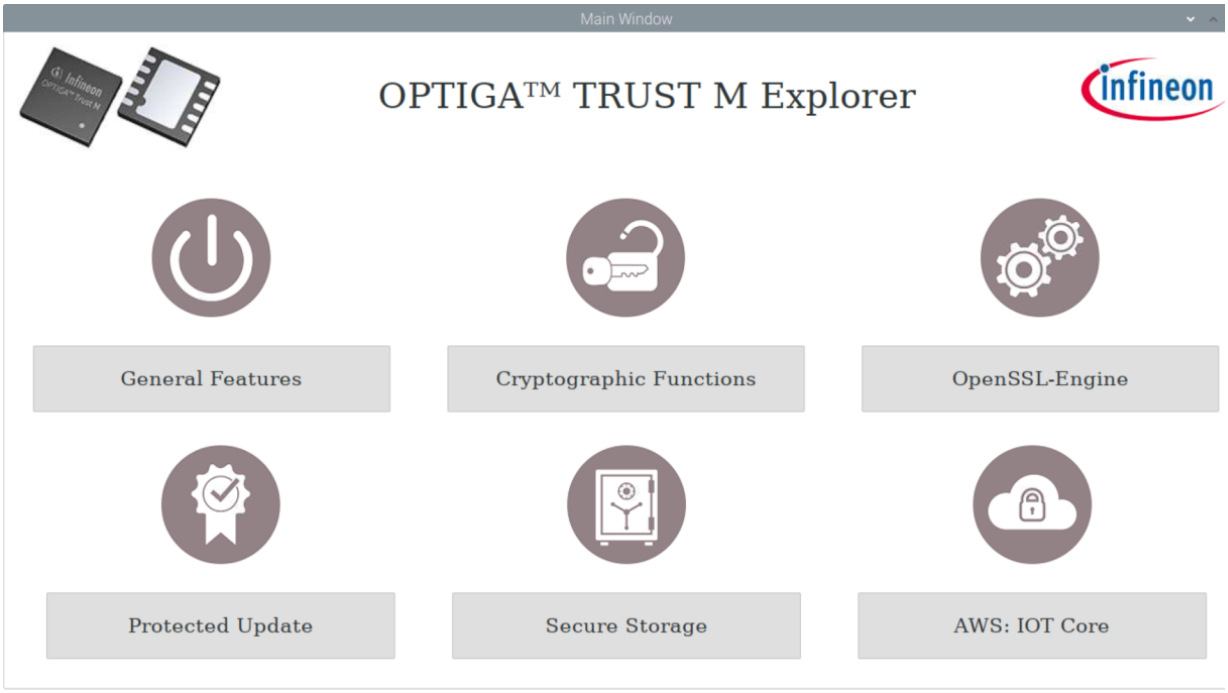


Figure 11: Executing start_gui.sh in terminal

A terminal will pop up and the OPTIGA Trust M Explorer interface will be open.



Performance Testing

After the application has been installed, we will test the commands.

It is recommended for the command line that you use the repository supplied by Infineon here: https://github.com/Infineon/linux-optiga-trust-m

Follow the instructions to build then install the command line tools in the README. To execute the following commands, you must navigate to the bin folder after compilation.

OpenSSL Engine

We are using Public and Private Keys for RSA Encryption and Decryption:
RSA 2048 Key-pair generation:
time (openssl genrsa -out private_key.pem 2048
openssl rsa -in private_key.pem -out -pubout public_key.pem)

Create plain text file for encryption and decryption:
echo "Hello World" > encrypt.txt

RSA 2048 Encryption:

time openssl rsautl -encrypt -inkey public_key.pem -pubin -in encrypt.txt -out encrypt.dat

RSA 2048 Decryption:
time openssl rsautl -decrypt -inkey private_key.pem -in encrypt.dat -out new_encrypt.txt

Random Number gen:
time openssl rand -base64 1024

TLS Handshake:
Client : time ./client <ipaddr> <portnum>
Server : time ./server

...

OPTIGA Trust M V3

RSA 2048 Key generation:
time ./trustm_rsa_keygen -g 0xE0FC -t 0x13 -k 0x42 -o rsa_E0FC_pub.pem -s

RSA 2048 Encryption:
time ./trustm_rsa_enc -p rsa_E0FC_pub.pem -o datain.enc -i datain.txt

RSA 2048 Decryption:
time ./trustm_rsa_dec -k 0xE0FC -o datain.dec -i datain.enc

Random Number generation:
time openssl rand -engine trustm_engine -base64 1024

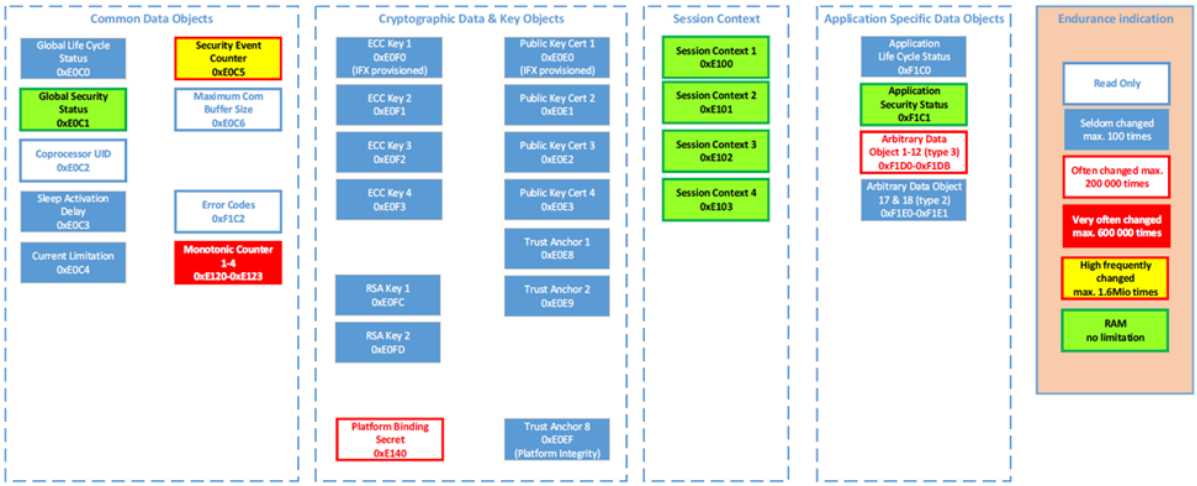
TLS Handshake:

TLS handshake between a simple client and server requires setup which is detailed here in the linux-optiga-trust-m repository.

Once you have the simpleTest_Client and simpleTest_Server built, you can simply run each as an executable wrapped in the time command.

If you want to connect across two separate devices, you must edit the .c file for simpleTest_Client, set the other device's IP address in the DEFAULT_IP macro, and recompile the linux-optiga-trust-m project with make.

If further help needed, here is the link to the contents of the linux-optiga-trust-m: linux-optiga-trust-m



Optional:
How to profile a performance using perf

Command for profiling:
To perf correctly for firefox hotspot:
perf_5.10 record -g -F 999 <Your_Command>
To convert perf report to txt:
perf_5.10 script -F +pid > filename.txt