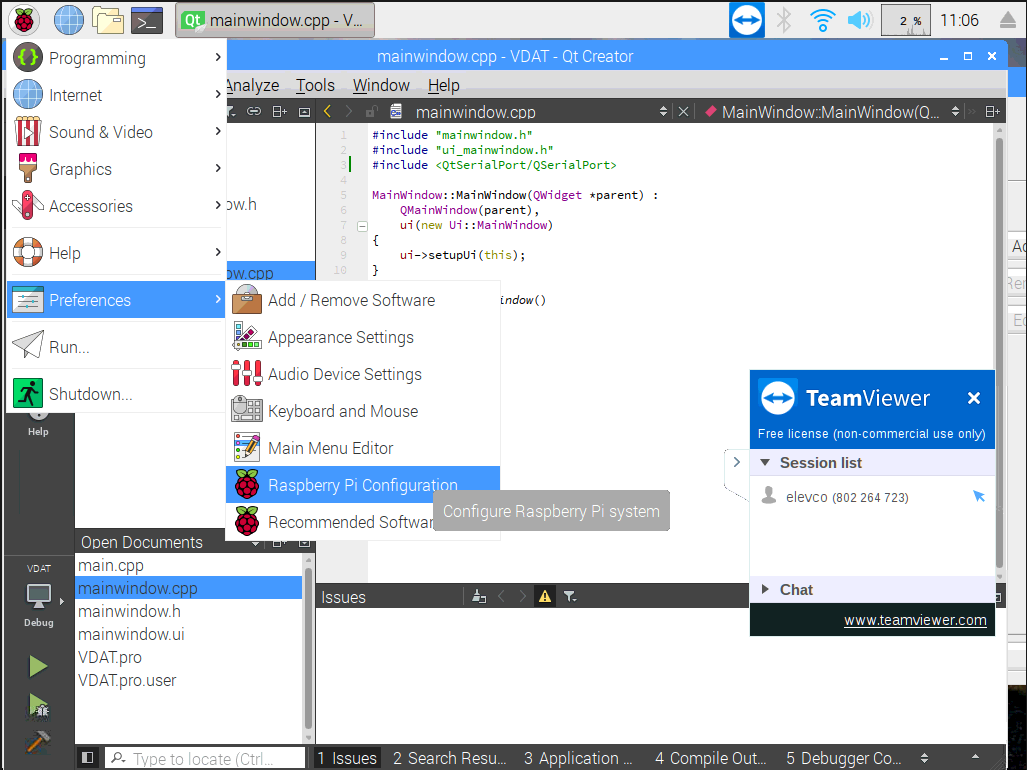
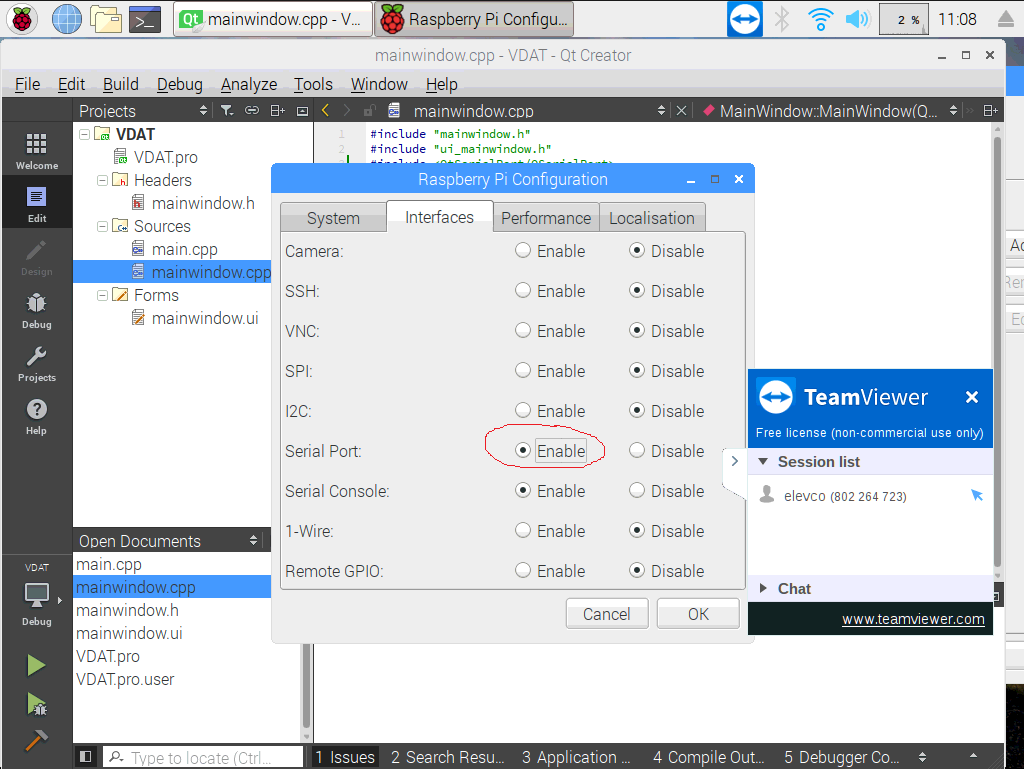
1. Tạo kết nối với RS232 của raspberry bằng QTCreater

1.1. Bật serialport trên raspberry





- Cố định tần số lõi của raspberry để cố định tần số cổng mini UART

b1. Mở file cấu hình /boot/config.txt

*cd /boot*

*sudo nano config.txt*

b2. Thay đổi hoặc thêm vào 2 dòng sau:

*core\_freq=250 (dòng này cần phải khai báo trước)*

*enable\_uart=1*

b3. Ctrl+X, Yes và khởi động lại PI (reboot)

1.2. Download thư viện về raspberry

*sudo apt-get install libqt5serialport5  
 sudo apt-get install libqt5serialport5-dev*

*sudo apt-get install wiringpi*

1.3. Khai báo thư viện trong QTCreater

*QT += serialport*

#include <QtSerialPort/QSerialPort>

1.4. Thử kết nối và test

2. Sử dụng thư viện MODBUS RS485  [RS485\_for\_Raspberry\_Pi](https://www.cooking-hacks.com/documentation/tutorials/modbus-module-shield-tutorial-for-arduino-raspberry-pi-intel-galileo/index.html)

2.1. Các hàm khởi tạo

ModbusMaster485(); // Creates class object using default Modbus slave ID 1

ModbusMaster485(**uint8\_t**); // Creates class object using specified Modbus slave ID

begin(); // Sets up the serial port using default 19200 baud rate

begin(**unsigned** **long** ); // Sets up the serial port using specified baud rate

2.2. Các hàm gửi và nhận dữ liệu

**uint8\_t** **readCoils**(**uint16\_t**, **uint16\_t**); // Modbus function 0x01 Read Coils

**uint8\_t** **readDiscreteInputs**(**uint16\_t**, **uint16\_t**); // Modbus function 0x02 Read Discrete Inputs

**uint8\_t** **readHoldingRegisters**(**uint16\_t**, **uint16\_t**); // Modbus function 0x03 Read Holding Registers

**uint8\_t** **readInputRegisters**(**uint16\_t**, **uint8\_t**); // Modbus function 0x04 Read Input Registers

**uint8\_t** **writeSingleCoil**(**uint16\_t**, **uint8\_t**); // Modbus function 0x05 Write Single Coil

**uint8\_t** **writeSingleRegister**(**uint16\_t**, **uint16\_t**); // Modbus function 0x06 Write Single Register

**uint8\_t** **writeMultipleCoils**(**uint16\_t**, **uint16\_t**); // Modbus function 0x0F Write Multiple Coils

**uint8\_t** **writeMultipleRegisters**(**uint16\_t**, **uint16\_t**);// Modbus function 0x10 Write Multiple Registers

2.3. Các hàm tiện ích

**void** **setSlaveAddress**(**uint8\_t**); // Changes the slave address

**uint8\_t** **available**(**void**); // Returns the availbale data read

**uint16\_t** **getResponseBuffer**(**uint8\_t**); // Retrieve data from response buffer

**void** **clearResponseBuffer**(); // Clear Modbus response buffer

**uint8\_t** **setTransmitBuffer**(**uint8\_t**, **uint16\_t**); // Place data in transmit buffer

**void** **clearTransmitBuffer**(); // Place data in transmit buffer