Function description for RS485 communication Raspberry Pi 3 B to Teensy4.0, V1.00
2022-05-22, Armin Rehberger

Notes:
Raspberry Pi 3 B, config, bloototh must be disabled, see below
Raspberry Pi 3 B, settings serial port with Linux11 (bullseye), see below
Teensy, with simultaneous LiquidChristal, performance drops drastically

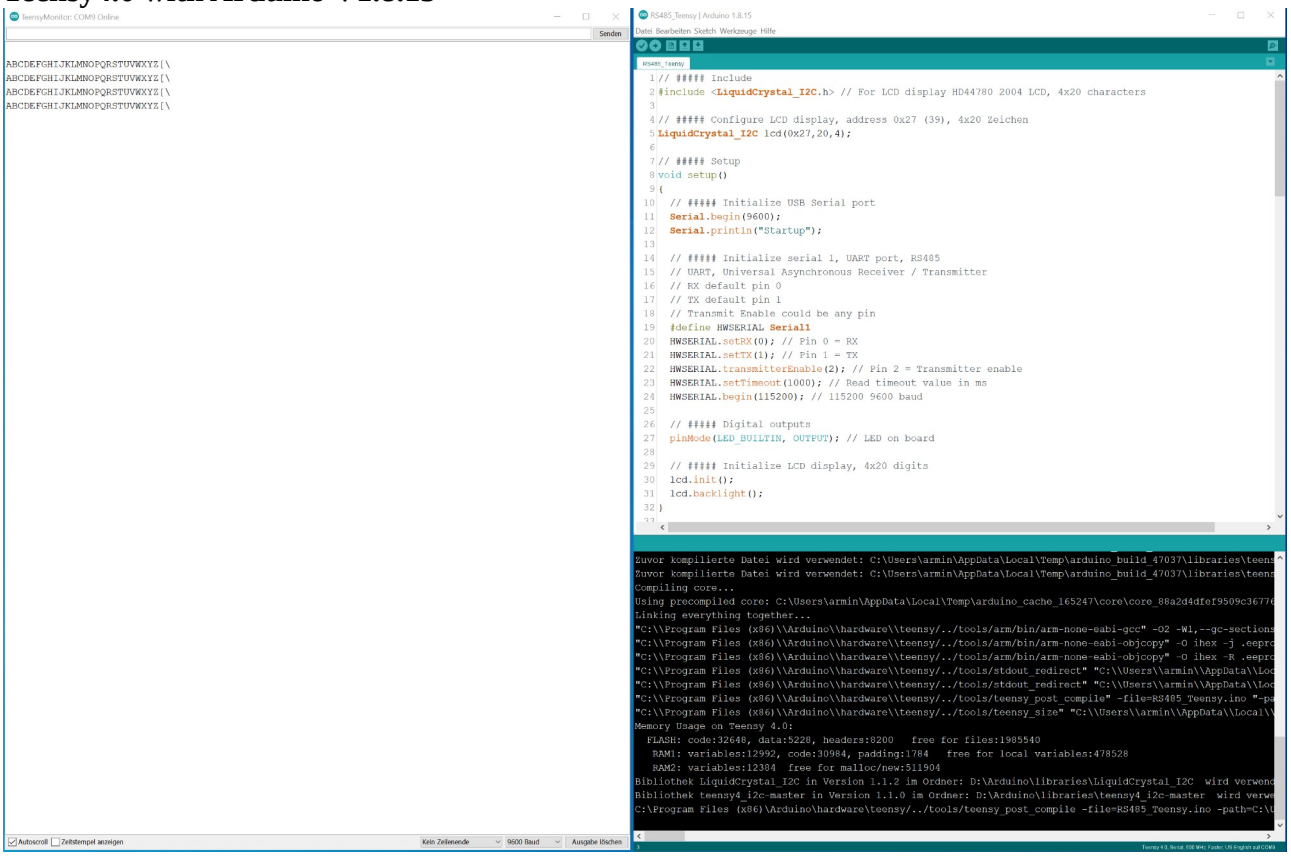
Example programs for RS485 communication between Raspberry Pi 3 B and Teensy4.0
ASCII Example with hardware flow control:
Communication is started by typing a character in the serial monitor of the Arduino software (Setting with no line end).
End character = LineFeed \n
Raspberry: **A_RS485.py**
Teensy: **A_RS485_Teensy.ino**

Decimal Example with hardware flow control:
Communication is started by typing a character in the serial monitor of the Arduino software (Setting with no line end).
Communicates a fixed amount of Bytes.
Last two Bytes are usede for CRC16 checksum.
Raspberry: **B_RS485.py**
Teensy: **B_RS485_Teensy.ino**

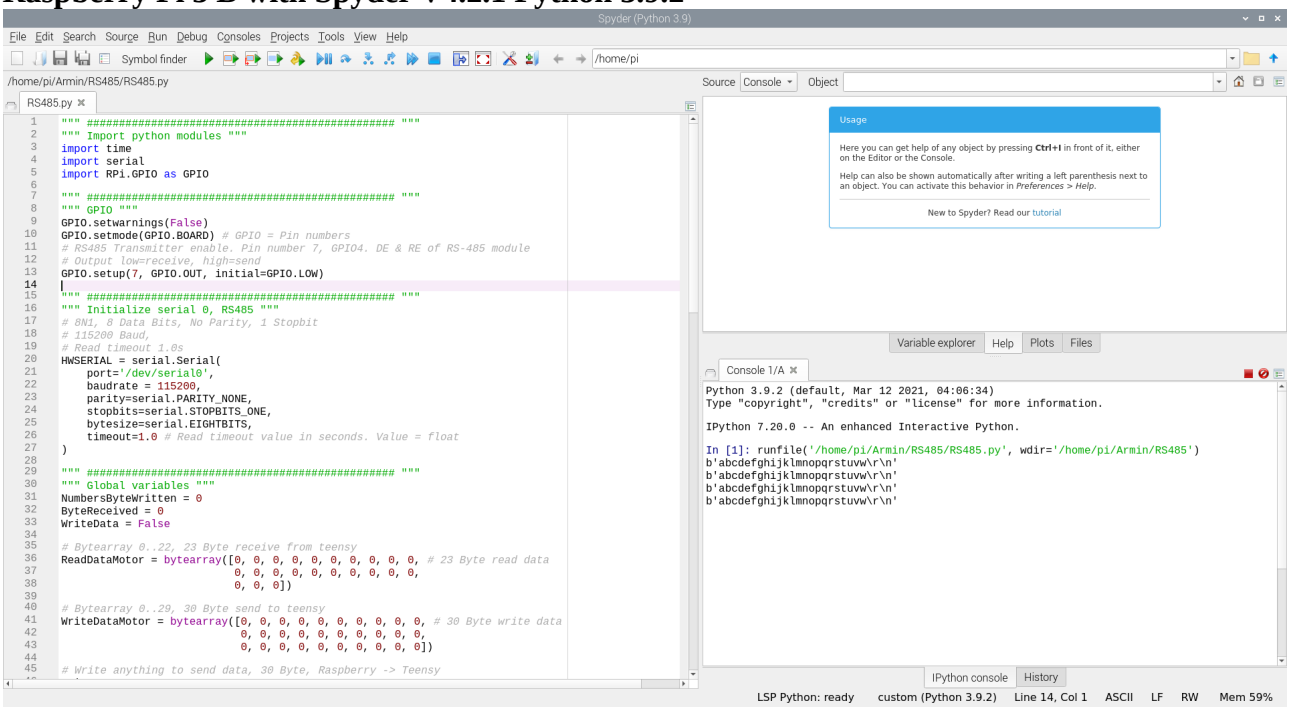
Decimal Example with hardware flow control:
Communication is started automatically.
Communicates a fixed amount of Bytes.
Last two Bytes are usede for CRC16 checksum.
Raspberry: **C_RS485.py**
Teensy: **C_RS485_Teensy.ino**

Decimal Example with automatic flow control:
Communication is started automatically.
Communicates a fixed amount of Bytes.
Last two Bytes are usede for CRC16 checksum.
Raspberry: **D_RS485.py**
Teensy: **D_RS485_Teensy.ino**
Example D: Reached baudrate 500000 bits/s, 0.0002% error

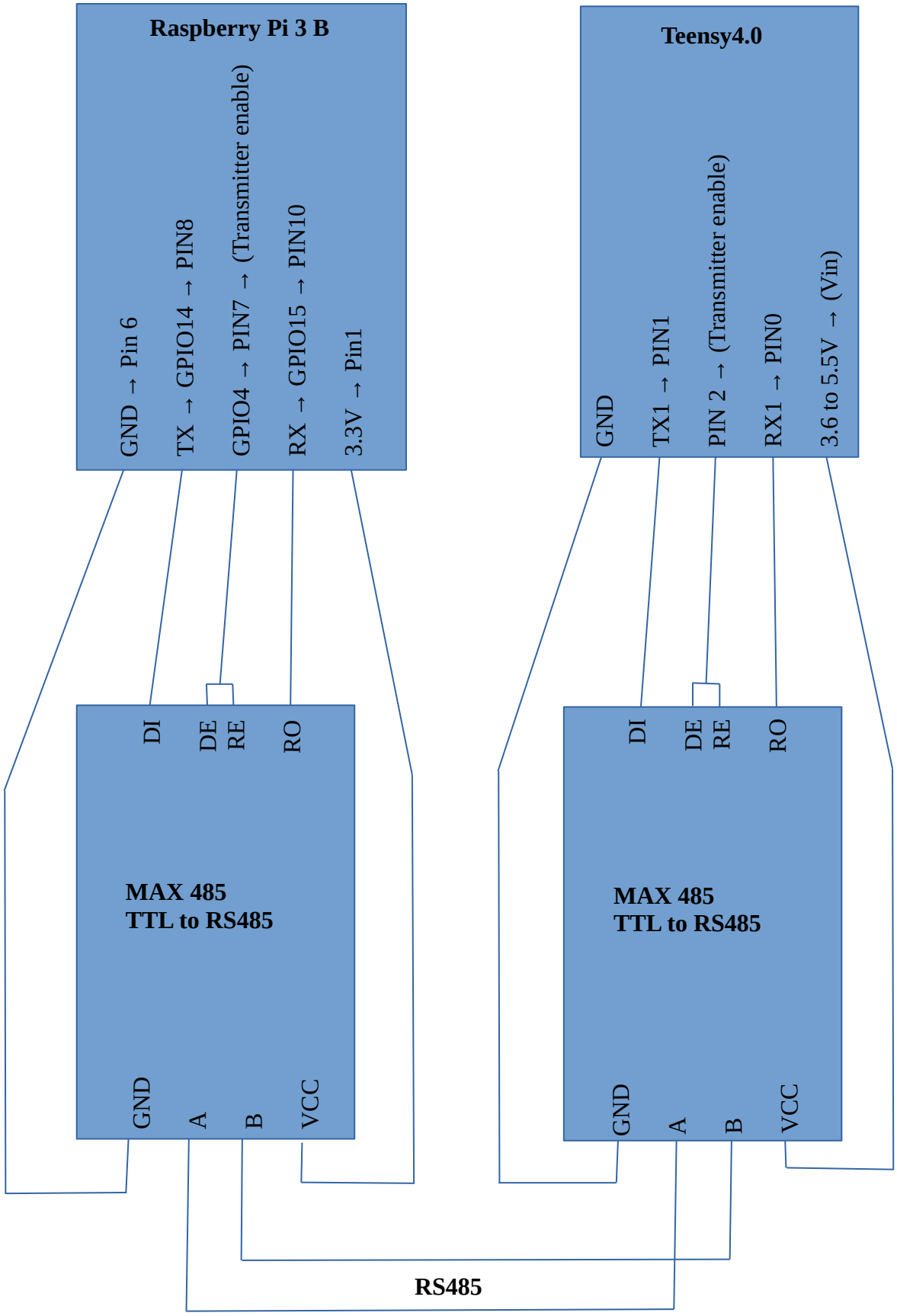
Teensy4.0 with Arduino V1.8.15



Raspberry Pi 3 B with Spyder V4.2.1 Python 3.9.2

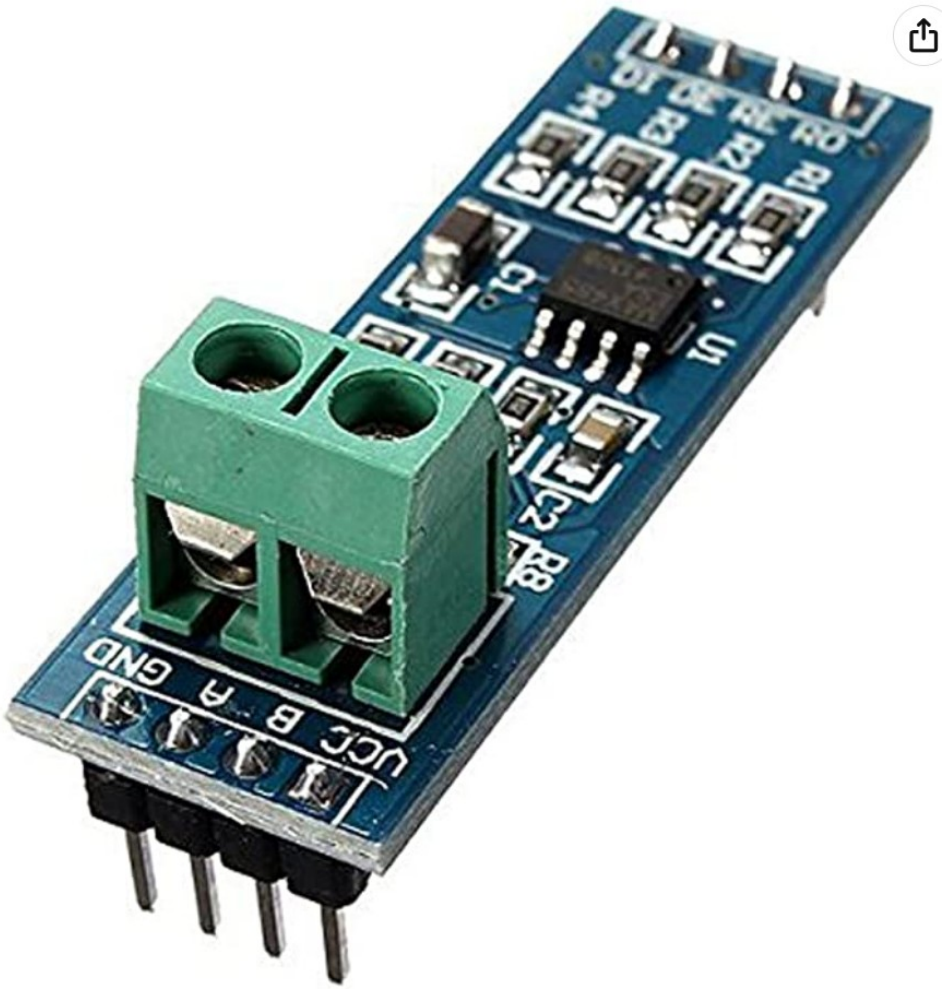


Wiring Raspberry Pi 3 B → MAX 485 → MAX 485 → Teensy4.0
With hardware flow control (Transmitter enable)

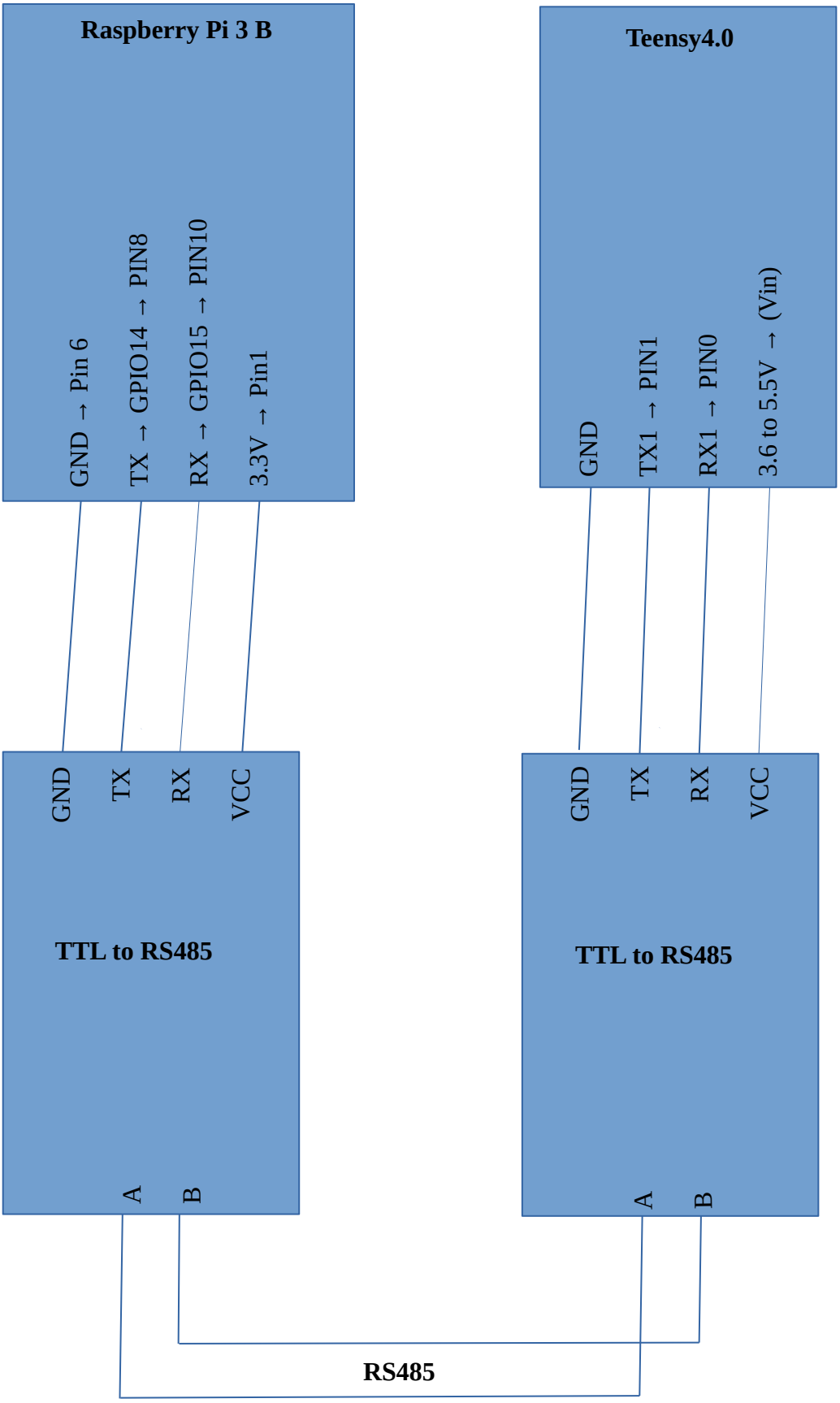


MAX485 TTL RS485 Adapter. UART Serial 3.3V or 5.0V

With hardware Flow control

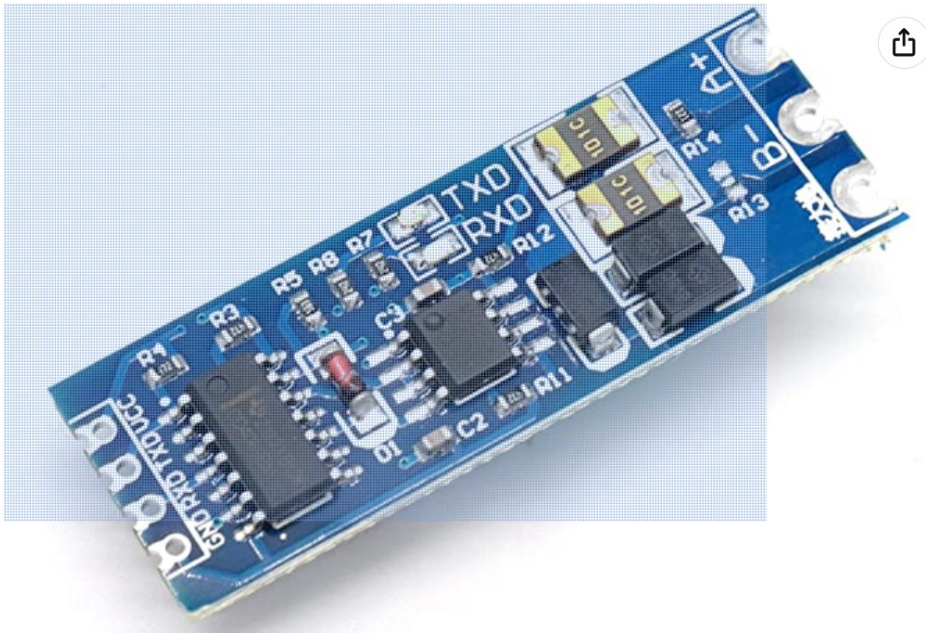


Wiring Raspberry Pi 3 B → TTL-RS485 → TTL-RS485 → Teensy4.0
With automatic flow control



TTL RS485 Adapter. UART Serial 3.3V or 5.0V

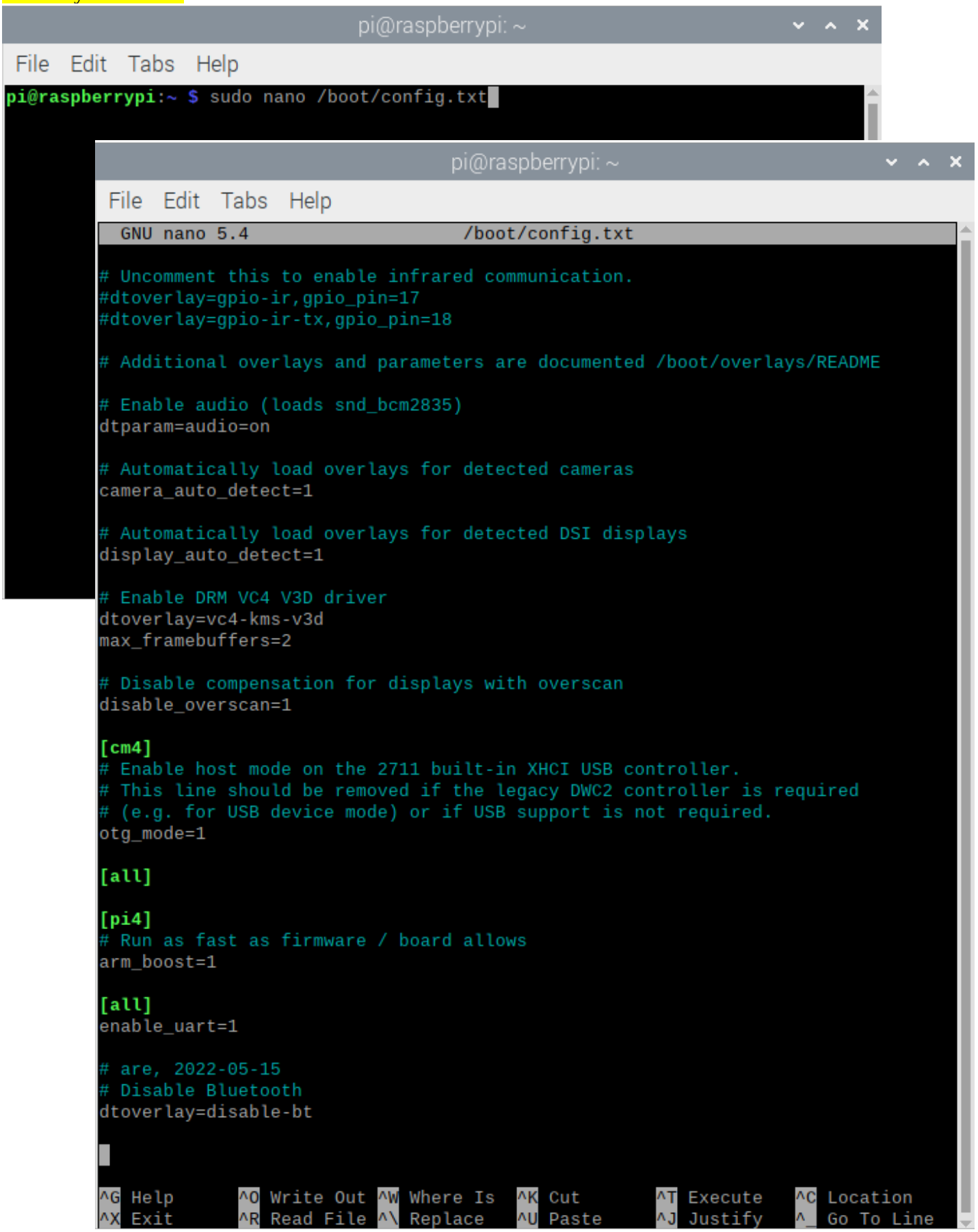
Automatic Flow control



Raspberry Pi 3 B, config, disable bluetooth

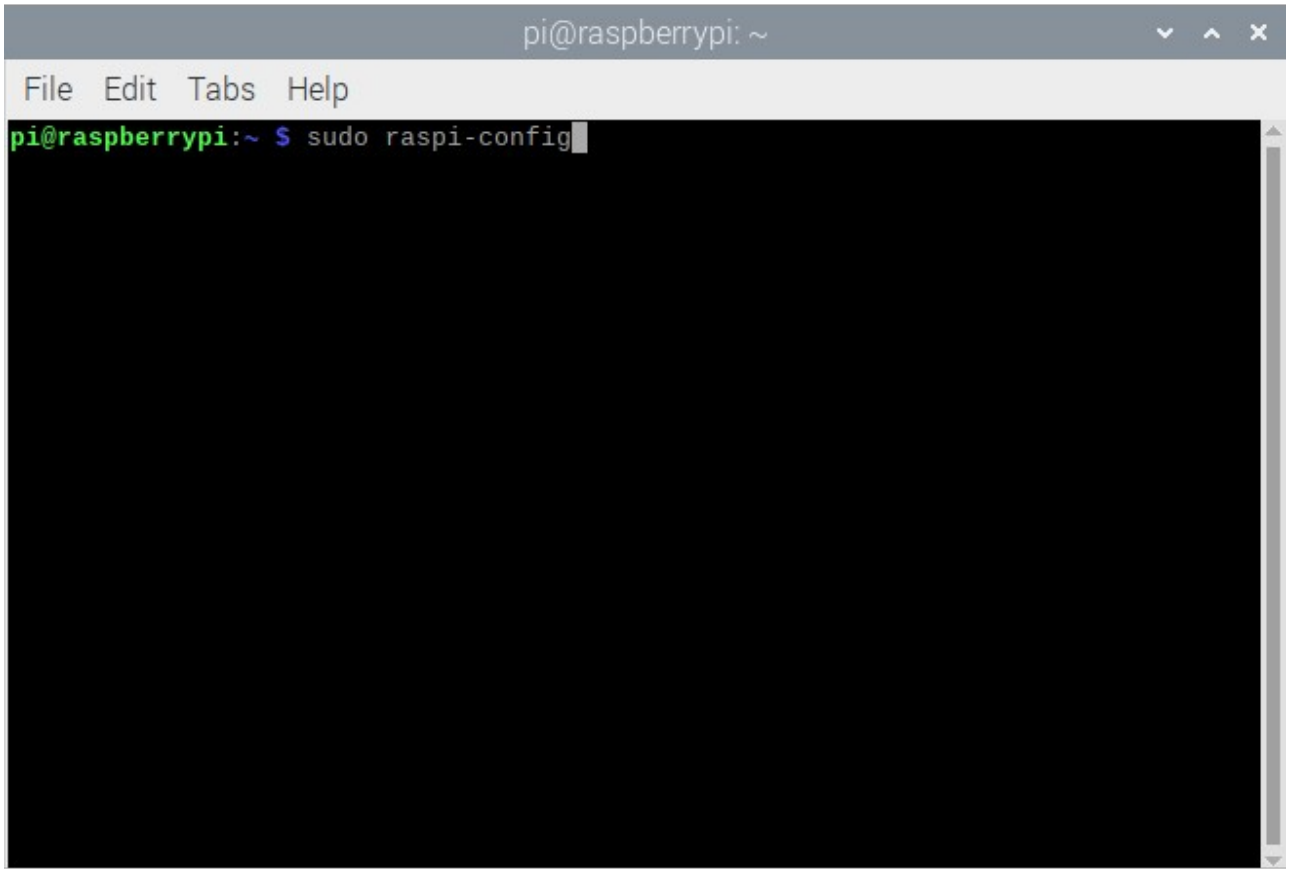
Open a terminal and type sudo nano /*boot*/config.txt

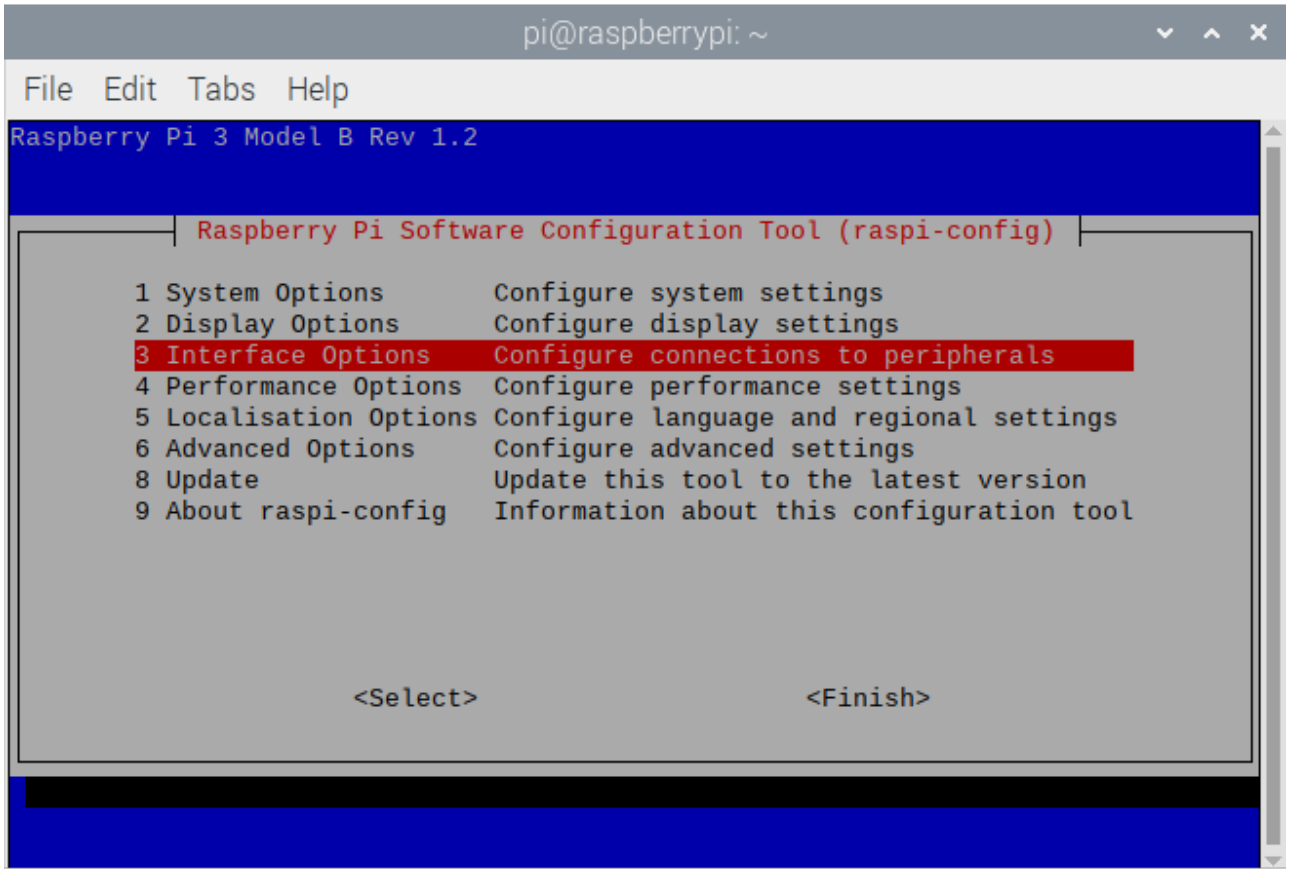
dtoverlay=disable-bt

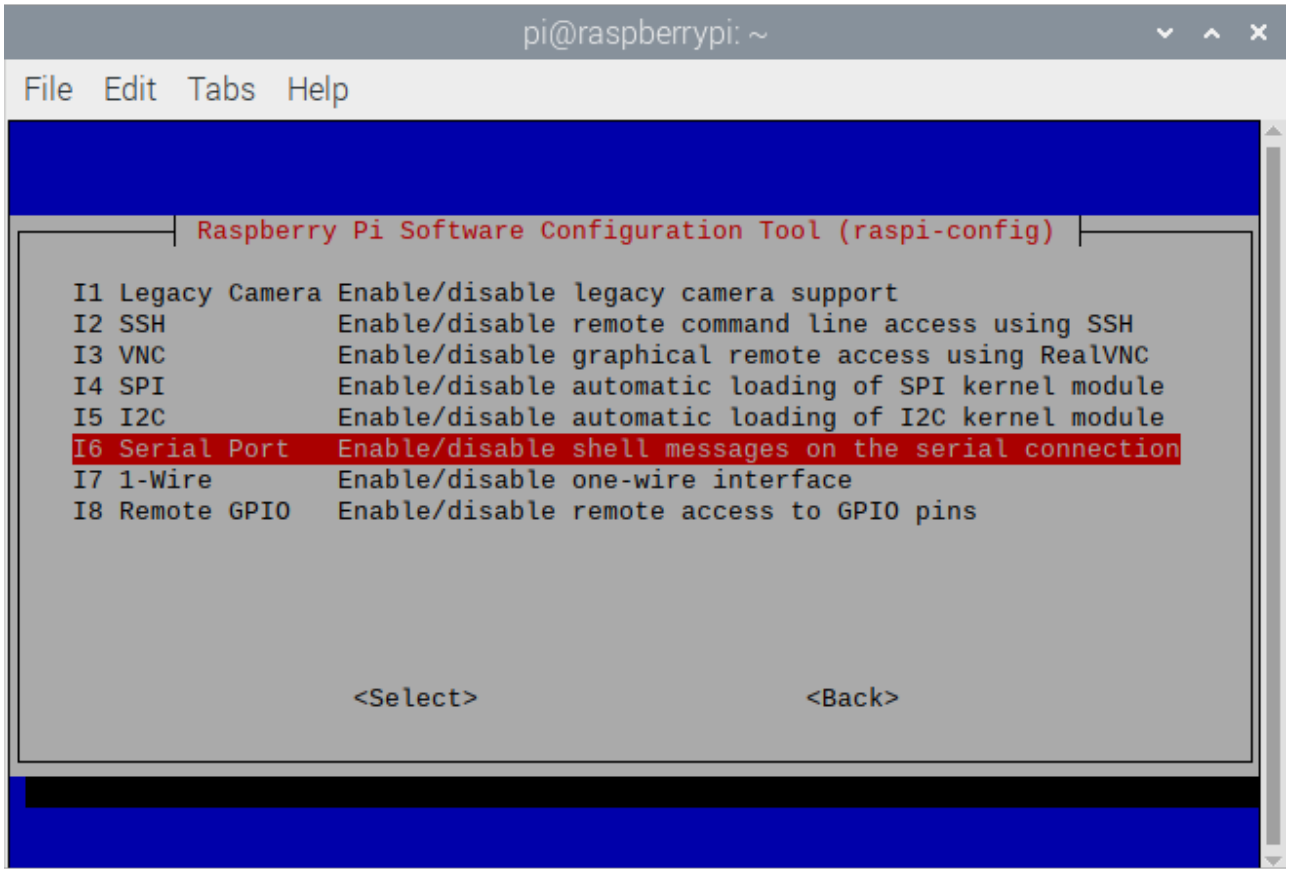


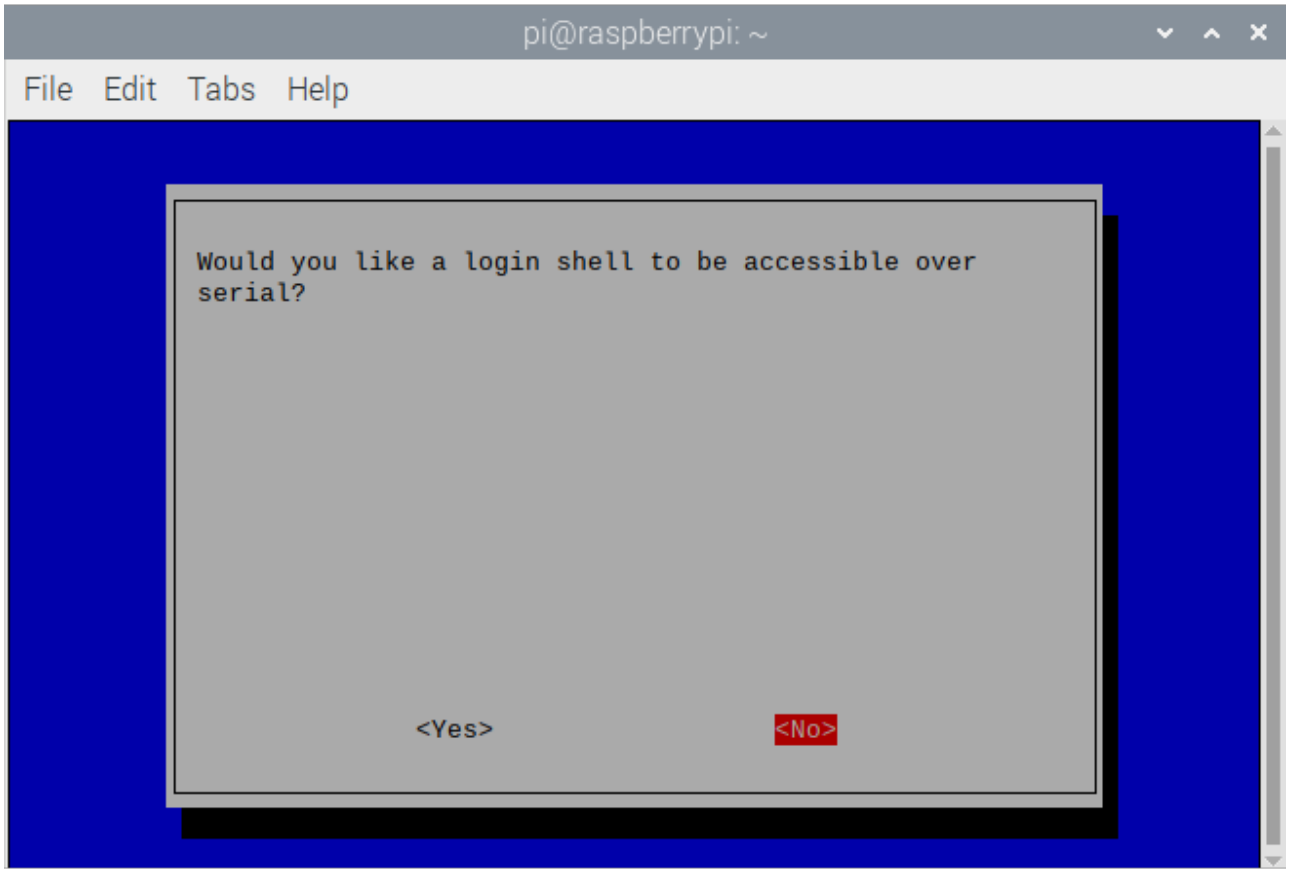
Raspberry Pi 3 B, settings serial port with Linux11 (bullseye)

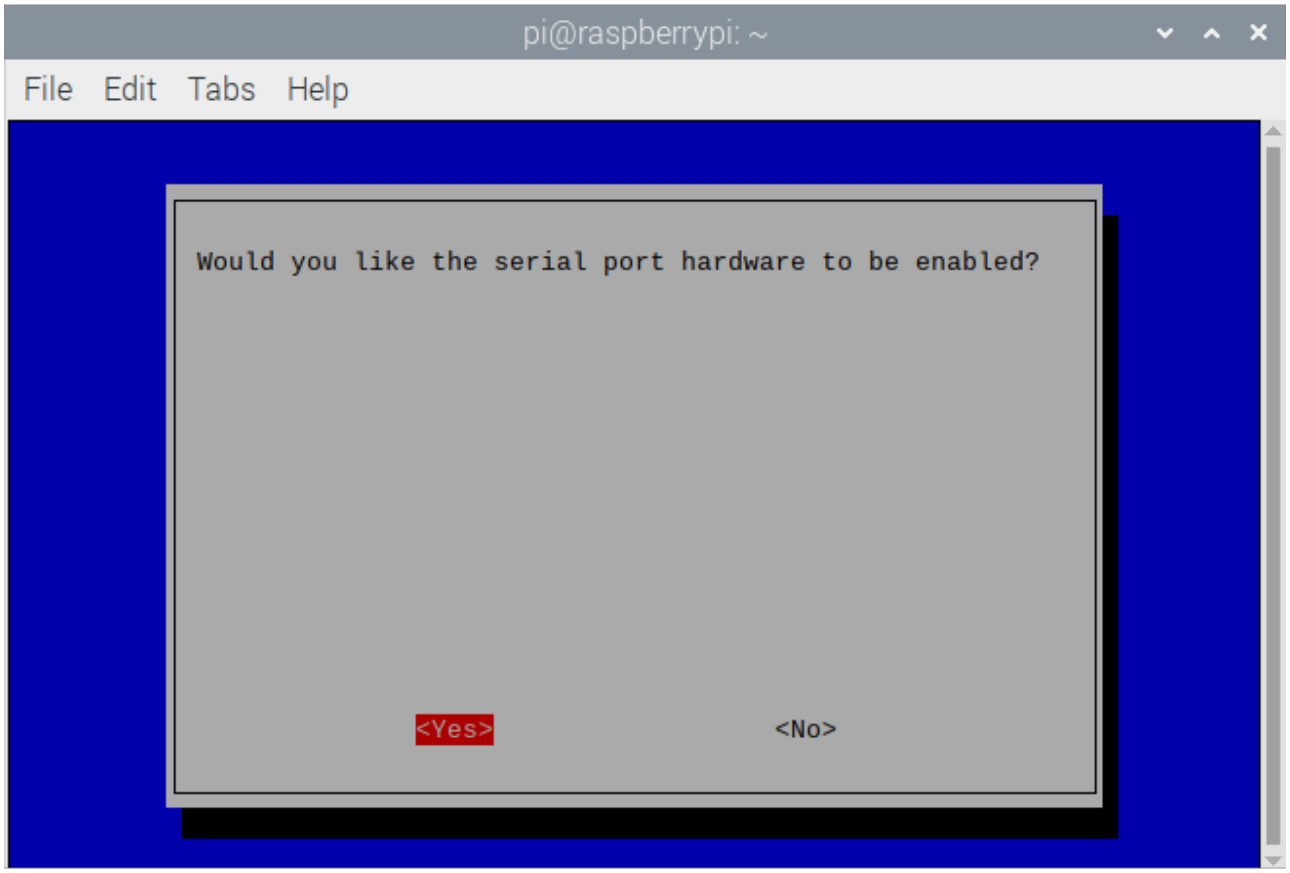
Open a terminal and type sudo raspi-config

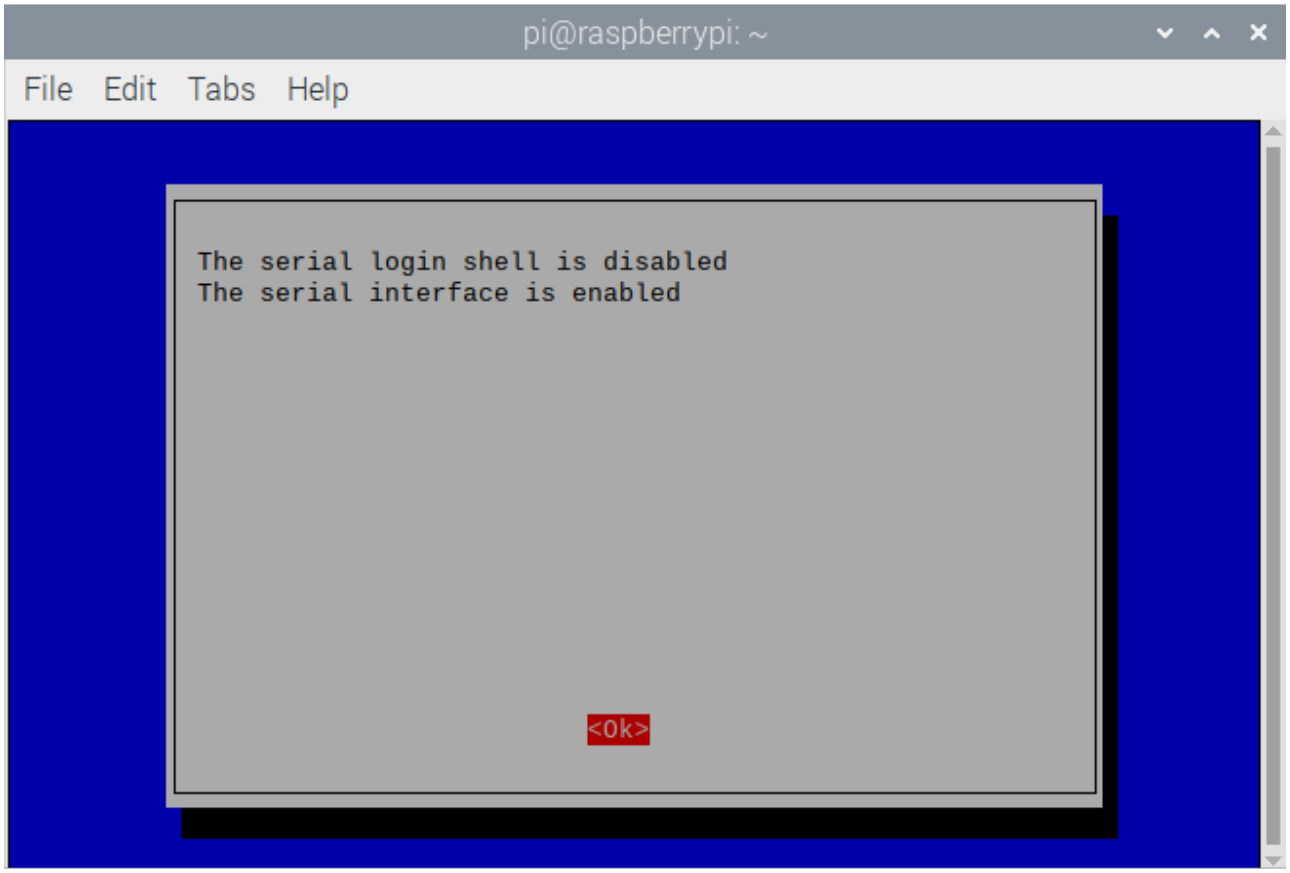





pi@raspberrypi: ~
File Edit Tabs Help
Raspberry Pi Software Configuration Tool (raspi-config)
I1 Legacy Camera Enable/disable legacy camera support
I2 SSH Enable/disable remote command line access using SSH
I3 VNC Enable/disable graphical remote access using RealVNC
I4 SPI Enable/disable automatic loading of SPI kernel module
I5 I2C Enable/disable automatic loading of I2C kernel module
I6 Serial Port Enable/disable shell messages on the serial connection
I7 1-Wire Enable/disable one-wire interface
I8 Remote GPIO Enable/disable remote access to GPIO pins
<Select>
<Back>


pi@raspberrypi: ~
File Edit Tabs Help
Would you like a login shell to be accessible over serial?
<Yes>
<No>


pi@raspberrypi: ~
File Edit Tabs Help
Would you like the serial port hardware to be enabled?
<Yes>
<No>


pi@raspberrypi: ~
File Edit Tabs Help
The serial login shell is disabled
The serial interface is enabled
<Ok>