

## 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, bluetooth must be disabled, see below

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

Teensy, with simultaneous LiquidCrystal, 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 used 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 used 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.

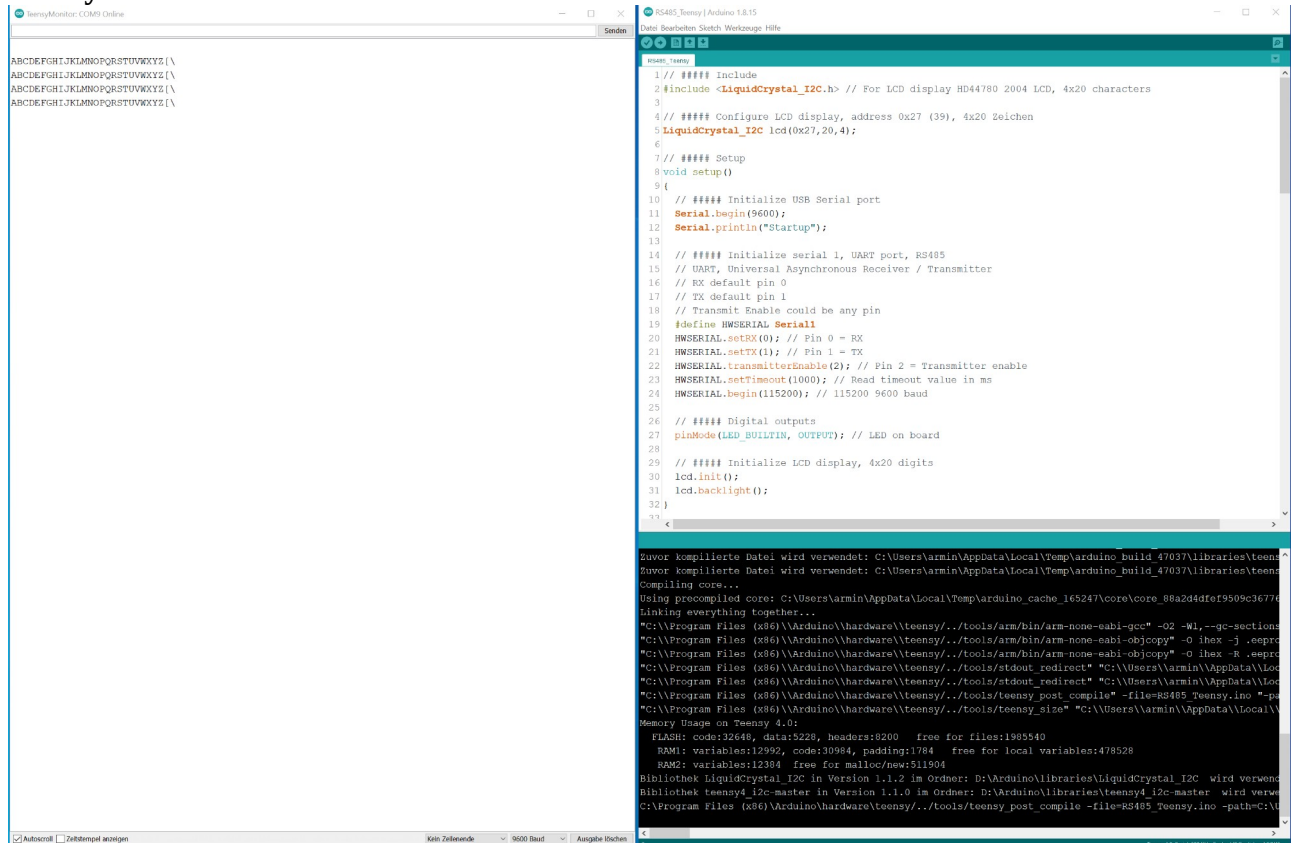
Last two Bytes are used for CRC16 checksum.

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



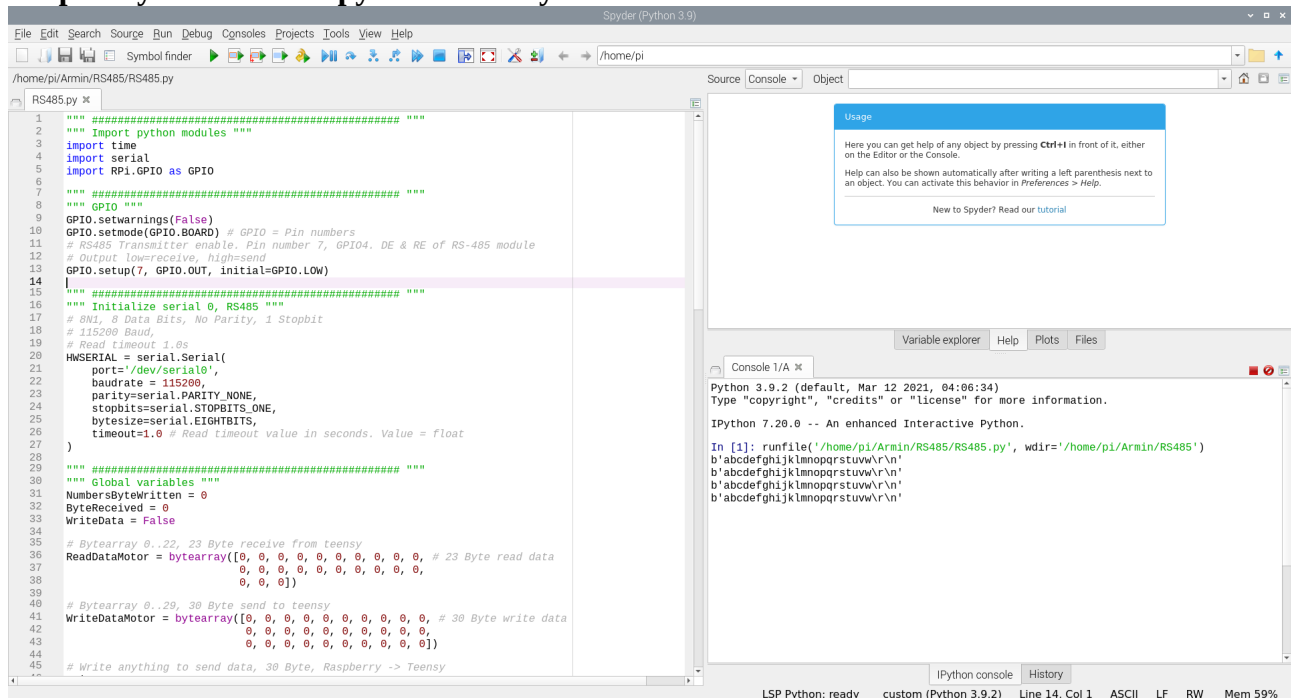
```
Arduino IDE - RS485_Teensy (Arduino 1.8.15)
Date: 2020-08-04 10:00:00
Send

ABCDEF GHIJ KLMNOPQRSTUVWXYZ\
ABCDEF GHIJ KLMNOPQRSTUVWXYZ\
ABCDEF GHIJ KLMNOPQRSTUVWXYZ\
ABCDEF GHIJ KLMNOPQRSTUVWXYZ\

1 // ##### Include
2 #include <LiquidCrystal_I2C.h> // For LCD display HD44780 2004 LCD, 4x20 characters
3
4 // ##### Configure LCD display, address 0x27 (39), 4x20 Zeichen
5 LiquidCrystal_I2C lcd(0x27,20,4);
6
7 // ##### Setup
8 void setup()
9 {
10
11 // ##### Initialize USB Serial port
12 Serial.begin(9600);
13 Serial.println("Startup");
14
15 // ##### Initialize serial 1, UART port, RS485
16 // UART, Universal Asynchronous Receiver / Transmitter
17 // RX default pin 0
18 // TX default pin 1
19 // Transmit Enable could be any pin
20 #define HWSERIAL Serial1
21 HWSERIAL.setRX(0); // Pin 0 = RX
22 HWSERIAL.setTX(1); // Pin 1 = TX
23 HWSERIAL.transmitEnable(2); // Pin 2 = Transmitter enable
24 HWSERIAL.setTimeout(1000); // Read timeout value in ms
25 HWSERIAL.begin(115200); // 115200 9600 baud
26
27 // ##### Digital outputs
28 pinMode(LED_BUILTIN, OUTPUT); // LED on board
29
30 // ##### Initialize LCD display, 4x20 digits
31 lcd.init();
32 lcd.backlight();
33 }
```

avor kompilierte Datei wird verwendet: C:\Users\armin\AppData\Local\Temp\arduino\_build\_47637\libraries\Teensy...  
zuerst kompilierte Datei wird verwendet: C:\Users\armin\AppData\Local\Temp\arduino\_build\_47637\libraries\Teensy...  
Compiling core...  
Using precompiled core: C:\Users\armin\AppData\Local\Temp\arduino\_cache\_165247\core\core\_8ba2d4dfe19509c36776...  
Linking everything together...  
C:\Program Files (x86)\Arduino\hardware\teensy\tools\arm\bin\arm-none-eabi-gcc -O2 -Wl,--gc-sections...  
FLASH: code:32648, data:5228, headers:8200 free for files:1985540  
RAM: variables:12992, code:30984, padding:1784 free for local variables:478528  
Bibliothek LiquidCrystal\_I2C in Version 1.1.2 im Ordner: D:\Arduino\libraries\LiquidCrystal\_I2C wird verwendet  
Bibliothek teensy\_i2c-master in Version 1.1.0 im Ordner: D:\Arduino\libraries\teensy\_i2c-master wird verwendet  
C:\Program Files (x86)\Arduino\hardware\teensy\tools\teensy\_post\_compile -file=RS485\_Teensy.ino -path=C:\U...

## Raspberry Pi 3 B with Spyder V4.2.1 Python 3.9.2



```
File Edit Search Source Run Debug Consoles Projects Tools View Help
/home/pi/Armin/RS485/RS485.py

1 """ ##### """
2 """ Import python modules """
3 import time
4 import serial
5 import RPi.GPIO as GPIO
6
7 """ ##### """
8 GPIO = GPIO
9 GPIO.setwarnings(False)
10 GPIO.setmode(GPIO.BOARD) # GPIO = Pin numbers
11 # RS485 Transmitter enable. Pin number 7, GPIO4. DE & RE of RS-485 module
12 # Output low=receive, high=send
13 GPIO.setup(7, GPIO.OUT, initial=GPIO.LOW)
14
15 """ ##### """
16 """ Initialize serial 0, RS485 """
17 # 8N1, 8 Data Bits, No Parity, 1 Stopbit
18 # 115200 Baud,
19 # Read timeout 1.0s
20 HWSERIAL = serial.Serial(
21     port='/dev/serial0',
22     baudrate=115200,
23     parity=serial.PARITY_NONE,
24     stopbits=serial.STOPBITS_ONE,
25     bytesize=serial.EIGHTBITS,
26     timeout=1.0 # Read timeout value in seconds. Value = float
27 )
28
29 """ ##### """
30 """ Global variables """
31 NumbersByteWritten = 0
32 ByteReceived = 0
33 WriteData = False
34
35 # Bytearray 0..22, 23 Byte receive from teensy
36 ReadDataMotor = bytearray([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
37
38 # Bytearray 0..29, 30 Byte send to teensy
39 WriteDataMotor = bytearray([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
40
41 # Write anything to send data, 30 Byte, Raspberry -> Teensy
```

Usage  
Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.  
Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in **Preferences > Help**.  
New to Spyder? Read our tutorial

Variable explorer Help Plots Files

Console I/A x

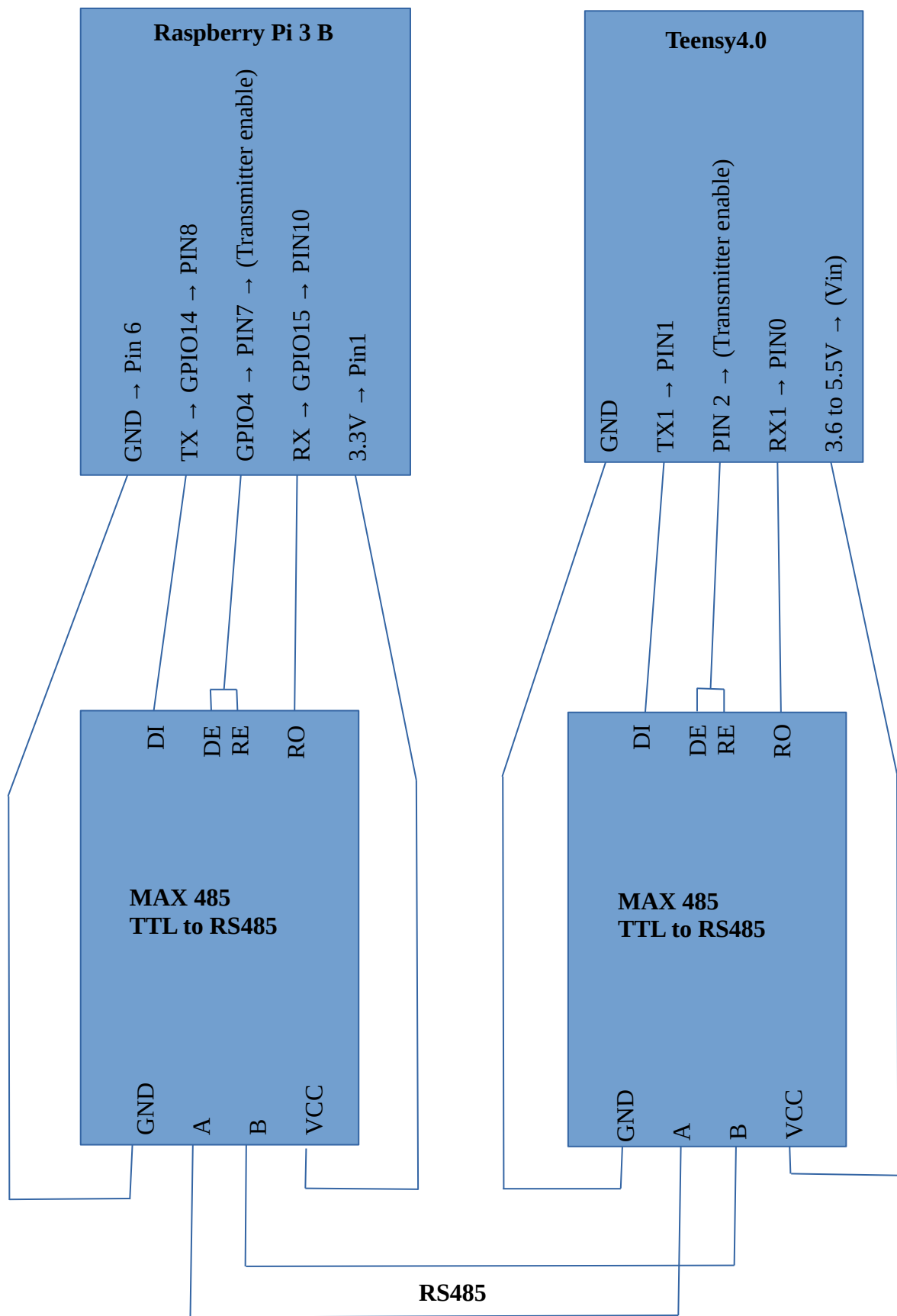
```
Python 3.9.2 (default, Mar 12 2021, 04:06:34)
Type "copyright", "credits" or "license()" for more information.

IPython 7.20.0 -- An enhanced Interactive Python.

In [1]: runfile('/home/pi/Armin/RS485/RS485.py', wdir='/home/pi/Armin/RS485')
b'abcdefghijklmnopqrstuvwxyz\r\n'
b'abcdefghijklmnopqrstuvwxyz\r\n'
b'abcdefghijklmnopqrstuvwxyz\r\n'
b'abcdefghijklmnopqrstuvwxyz\r\n'
```

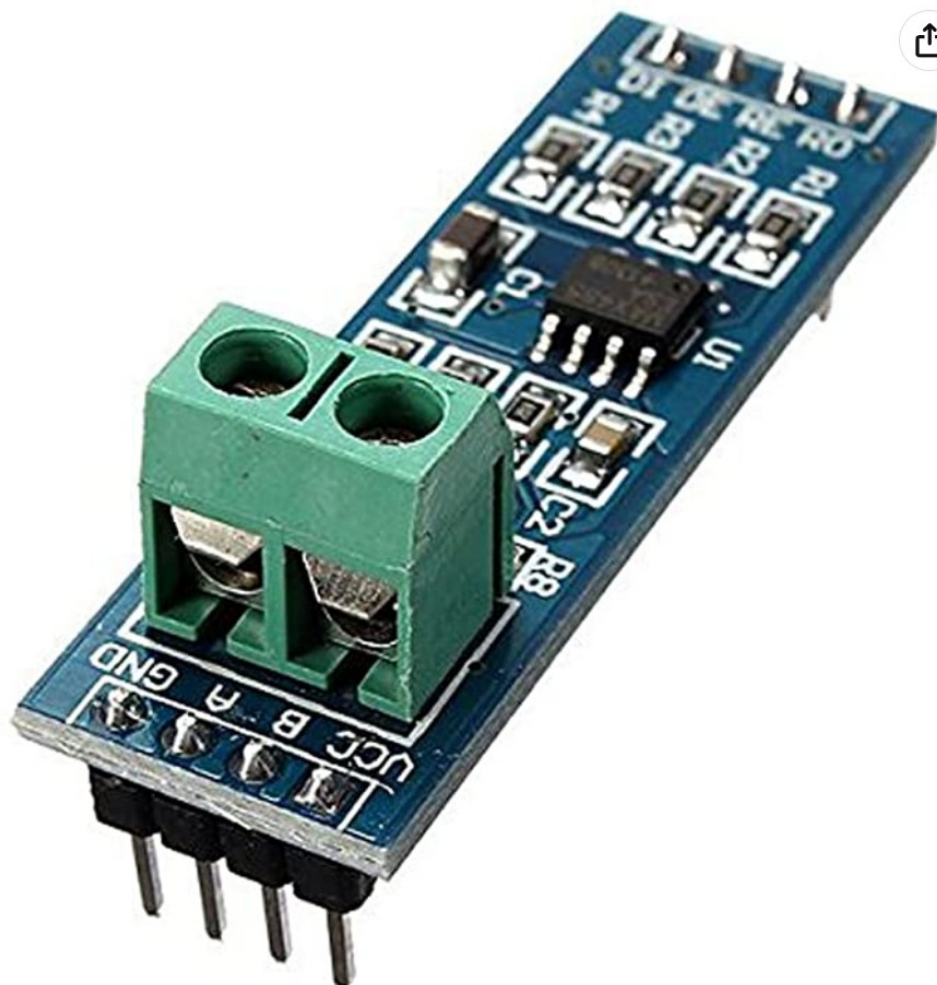
LSP Python: ready custom (Python 3.9.2) Line 14, Col 1 ASCII LF RW Mem 59%

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

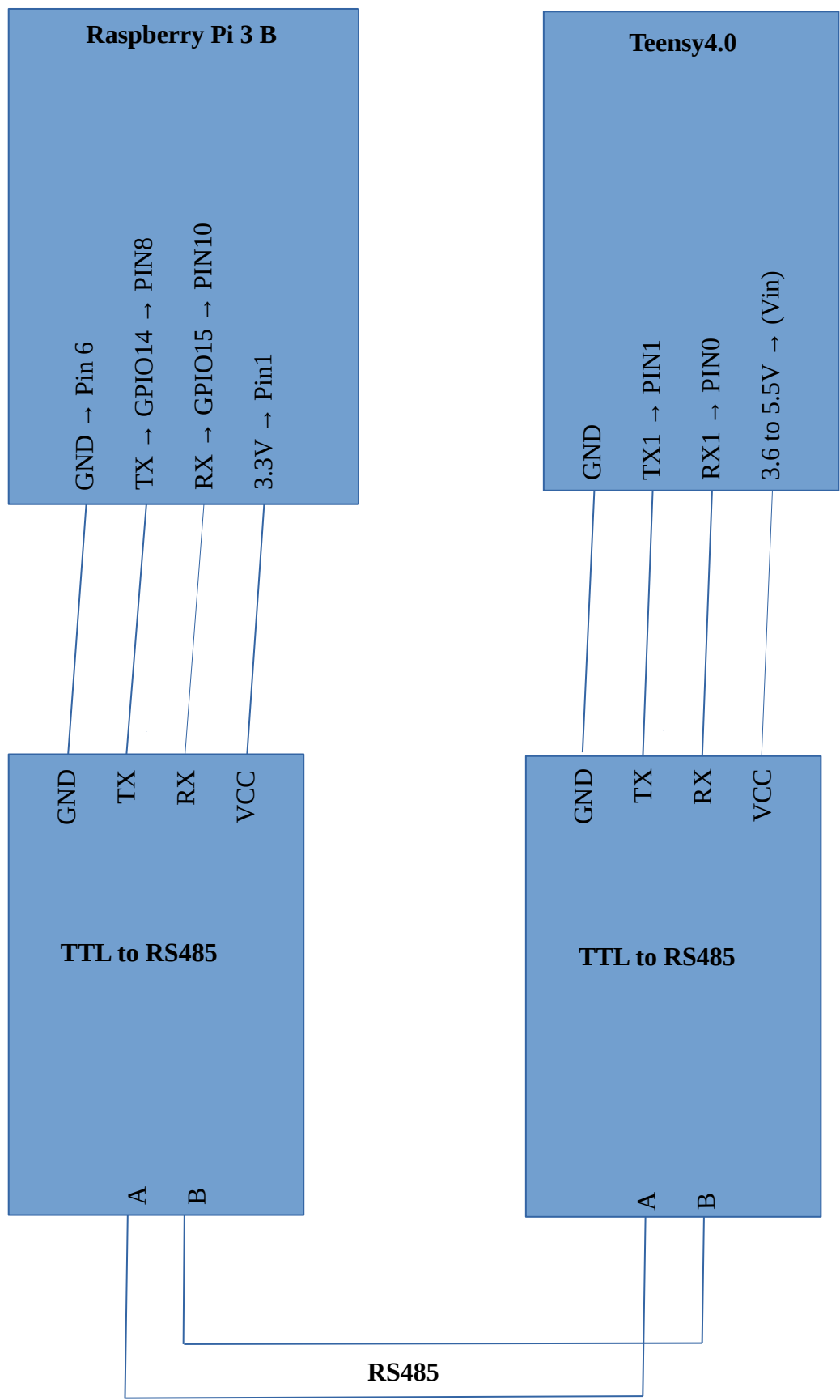


## MAX485 TTL RS485 Adapter. UART Serial 3.3V or 5.0V

With hardware Flow control

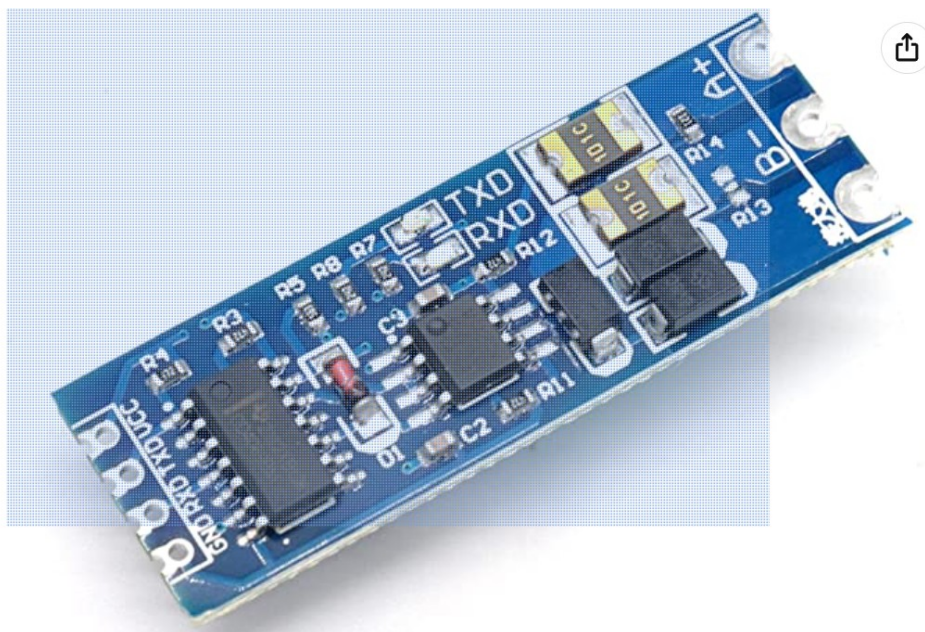


**Wiring Raspberry Pi 3 B → TTL-RS485 → TTL-RS485 → Teensy4.0**  
With automatic flow control



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

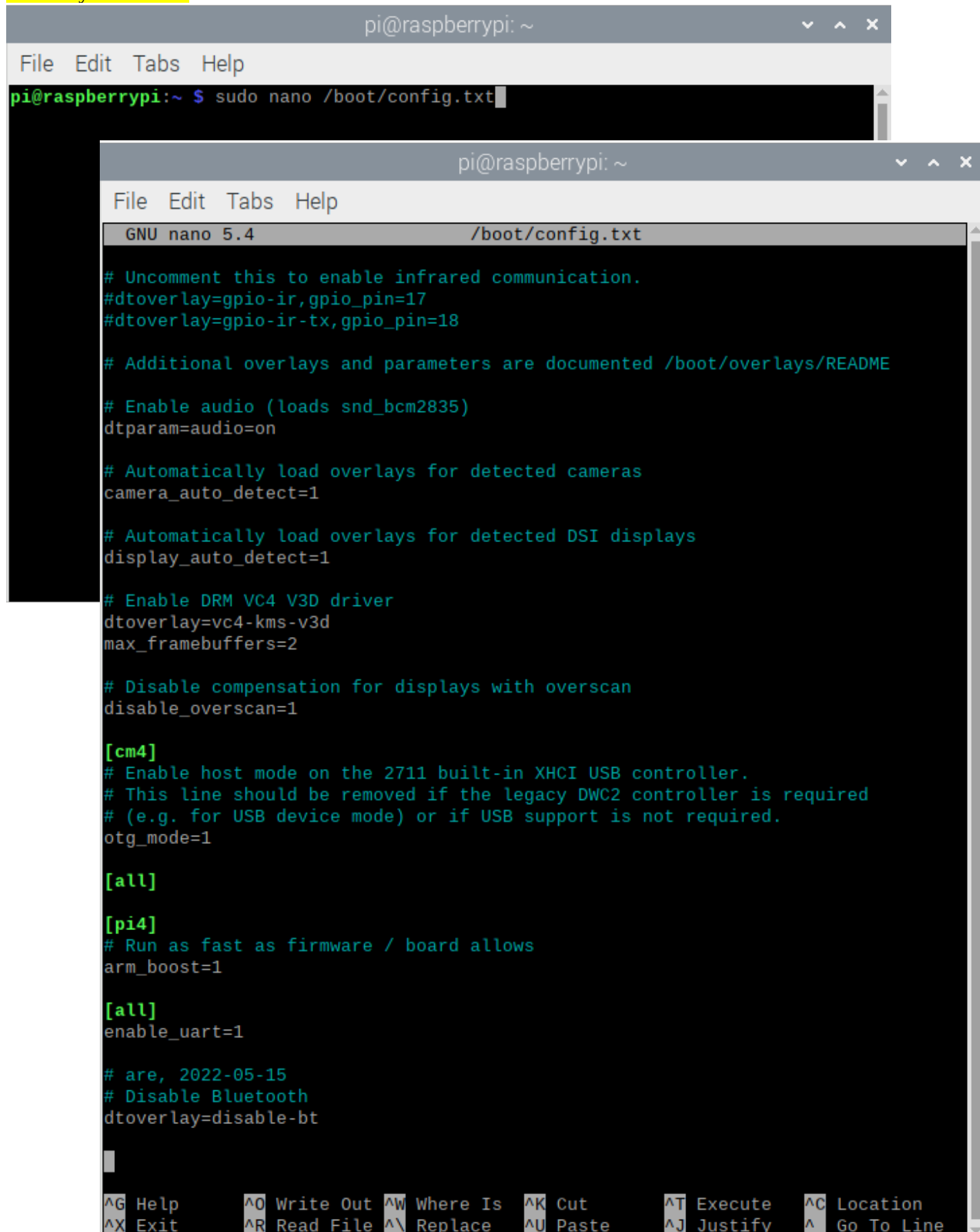
Automatic Flow control



## Raspberry Pi 3 B, config, disable bluetooth

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

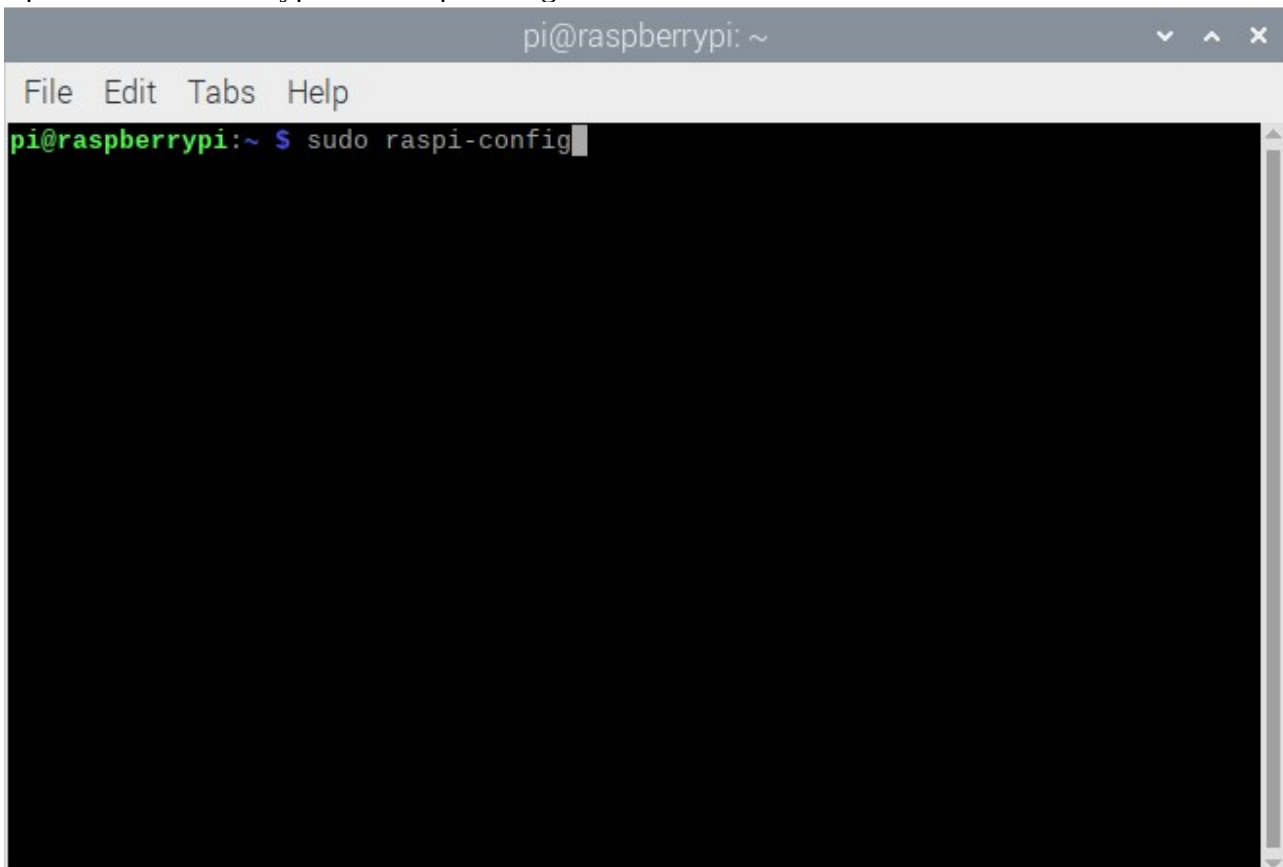
`dtoverlay=disable-bt`



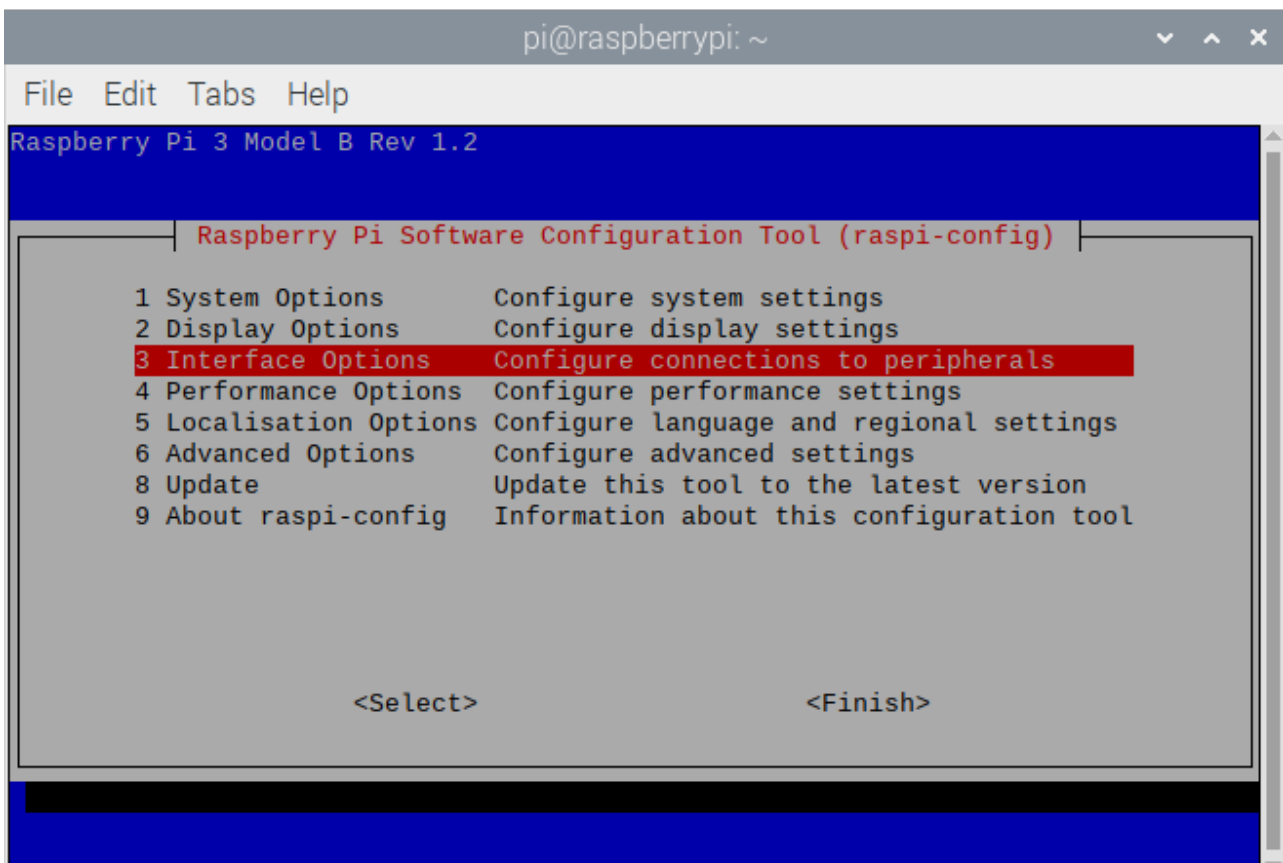
```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~ $ sudo nano /boot/config.txt  
GNU nano 5.4 /boot/config.txt  
  
# Uncomment this to enable infrared communication.  
#dtoverlay=gpio-ir,gpio_pin=17  
#dtoverlay=gpio-ir-tx,gpio_pin=18  
  
# Additional overlays and parameters are documented /boot/overlays/README  
  
# Enable audio (loads snd_bcm2835)  
dtparam=audio=on  
  
# Automatically load overlays for detected cameras  
camera_auto_detect=1  
  
# Automatically load overlays for detected DSI displays  
display_auto_detect=1  
  
# Enable DRM VC4 V3D driver  
dtoverlay=vc4-kms-v3d  
max_framebuffers=2  
  
# Disable compensation for displays with overscan  
disable_overscan=1  
  
[cm4]  
# Enable host mode on the 2711 built-in XHCI USB controller.  
# This line should be removed if the legacy DWC2 controller is required  
# (e.g. for USB device mode) or if USB support is not required.  
otg_mode=1  
  
[all]  
  
[pi4]  
# Run as fast as firmware / board allows  
arm_boost=1  
  
[all]  
enable_uart=1  
  
# are, 2022-05-15  
# Disable Bluetooth  
dtoverlay=disable-bt  
  
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location  
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

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

Open a terminal and type `sudo raspi-config`



A terminal window titled 'pi@raspberrypi: ~' with a menu bar containing 'File', 'Edit', 'Tabs', and 'Help'. The command prompt shows 'pi@raspberrypi:~ \$ sudo raspi-config' with a cursor at the end of the command.



A terminal window titled 'pi@raspberrypi: ~' with a menu bar containing 'File', 'Edit', 'Tabs', and 'Help'. The window displays the 'Raspberry Pi Software Configuration Tool (raspi-config)' menu. The first line of the menu is 'Raspberry Pi 3 Model B Rev 1.2'. The menu options are listed in a table-like format, with the third option, '3 Interface Options', highlighted in red. At the bottom of the menu, there are two options: '<Select>' and '<Finish>'. The background of the menu is blue.

Raspberry Pi Software Configuration Tool (raspi-config)	
1 System Options	Configure system settings
2 Display Options	Configure display settings
3 Interface Options	Configure connections to peripherals
4 Performance Options	Configure performance settings
5 Localisation Options	Configure language and regional settings
6 Advanced Options	Configure advanced settings
8 Update	Update this tool to the latest version
9 About raspi-config	Information about this configuration tool

<Select> <Finish>



