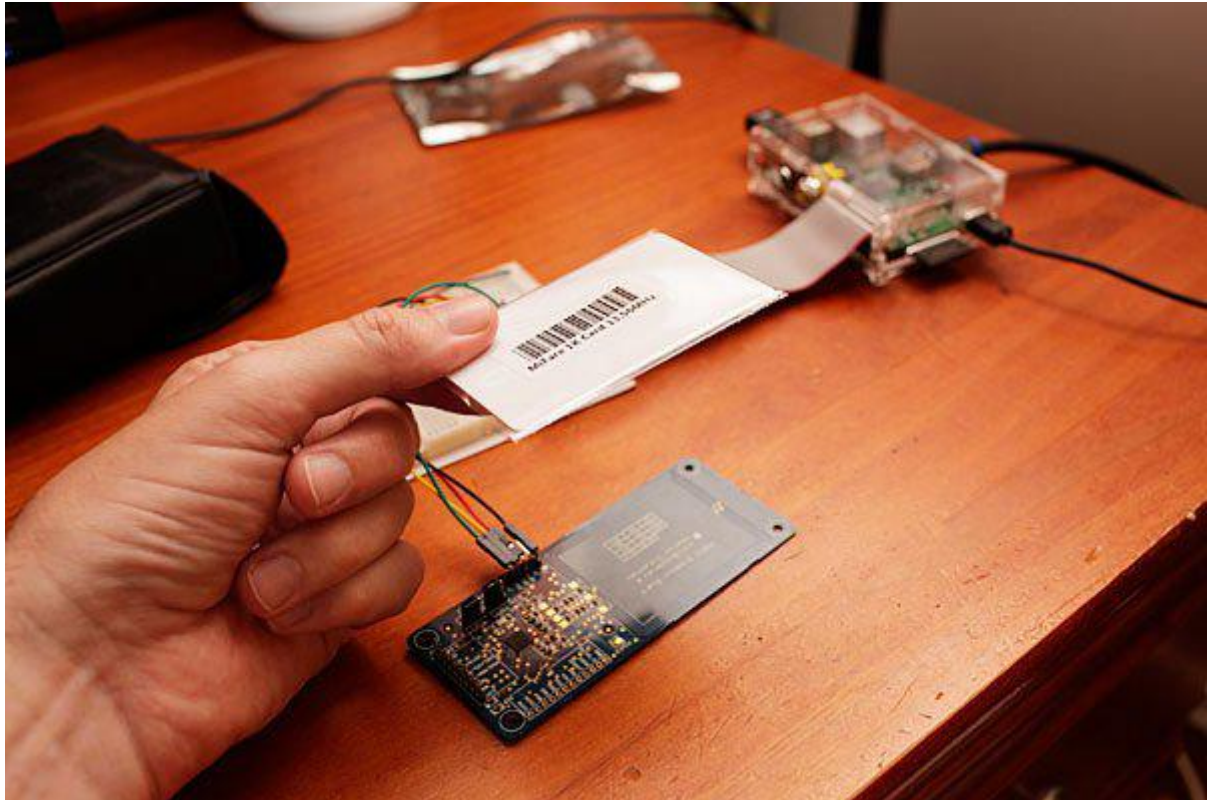




Adafruit NFC/RFID on Raspberry Pi

Created by Abigail Torres



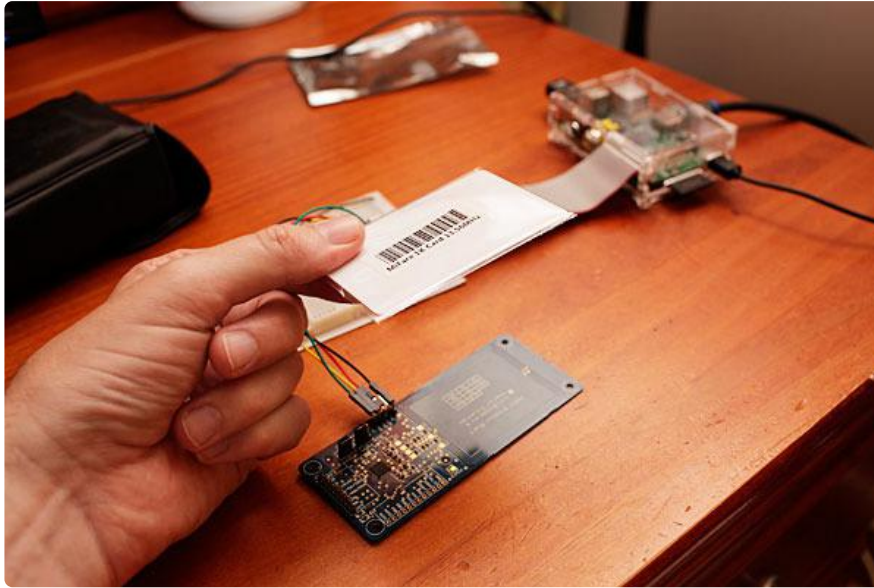
<https://learn.adafruit.com/adafruit-nfc-rfid-on-raspberry-pi>

Last updated on 2022-02-09 04:59:22 PM EST

Table of Contents

Overview	3
Pi Serial Port	4
Building libnfc	6
• Step One: Prepare for and Download libnfc	6
• Step Two: Set Up libnfc For the Pi	6
• For Recent Raspberry Pi Models	6
• For Earlier Pi Models	7
• Step Three: Configure the Library	7
• Step Four: Build and Install!	8
Testing it Out	8
• Hooking Everything Up	8
• Read an ISO14443-A (Mifare, etc.) Card with nfc-poll	10

Overview



Interested in adding some NFC (near-field communication) fun and excitement to your Raspberry Pi? You're in luck!

A big advantage of Linux is that it includes a large number of software “stacks” developed by the open source community, and you get to take advantage of all that hard work simply by using or installing the right library.

NFC is no exception here, with [libnfc](https://adafru.it/YkE) (<https://adafru.it/YkE>) having been around for a quite some time—in fact, it's the original reason the NFC Breakout was developed! libnfc is a library for C programmers. For Python and CircuitPython, there's [an equivalent module](https://adafru.it/YmC) (<https://adafru.it/YmC>).

To get libnfc playing well with your Pi and your Adafruit NFC breakout you'll need to make some minor configuration changes to the system and install some code, but it's pretty painless, and this tutorial will show you everything you need to do to start writing your own NFC-enabled apps on the Pi!

This guide assumes you have some version of Raspberry Pi OS already running, that the system is network-connected and so forth. We have [a series of tutorials for first-time users](https://adafru.it/jd1) (<https://adafru.it/jd1>) if you need some help with those steps.

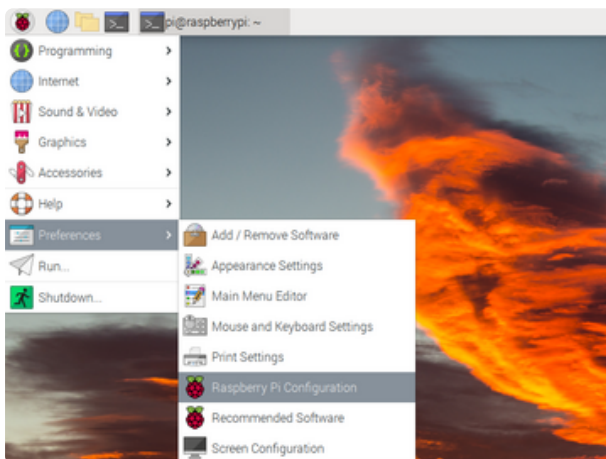
Pi Serial Port

The easiest way to use libnfc with the Adafruit NFC Breakout is via serial UART, since it's well-supported by libnfc out of the box. Unfortunately the serial UART port on the Pi is already dedicated to other purposes, and needs to be freed up for libnfc...

The following steps (based on a clean Raspberry Pi OS installation) should make the serial UART available to us:

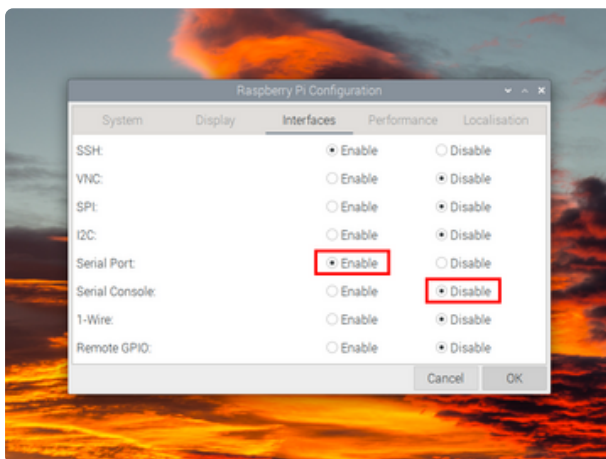
Using the Desktop/GUI “Full” OS

You'll find these settings in the Raspberry Pi Configuration tool. From the Raspberry menu at the top-left...



Go to the “Interfaces” tab and you’ll see two options “Serial Port” and “Serial Console.” Toggle both of these away from their default states. Serial Port should be enabled, Serial Console disabled. Then click “OK.” Reboot when prompted.

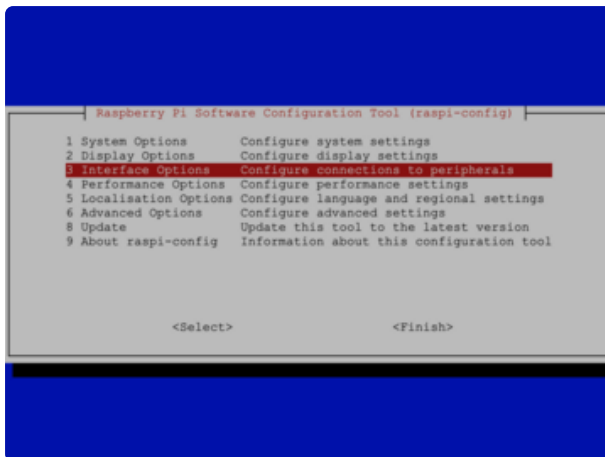
With each new OS release, it’s normal that some configuration options may move to different menus or positions. If you can’t find it where shown, check under the other top-level menu options... even if moved, the name will likely remain similar.



Using the “Lite” Command-Line OS

These options can be found in the `raspi-config` tool, which must be run as root:

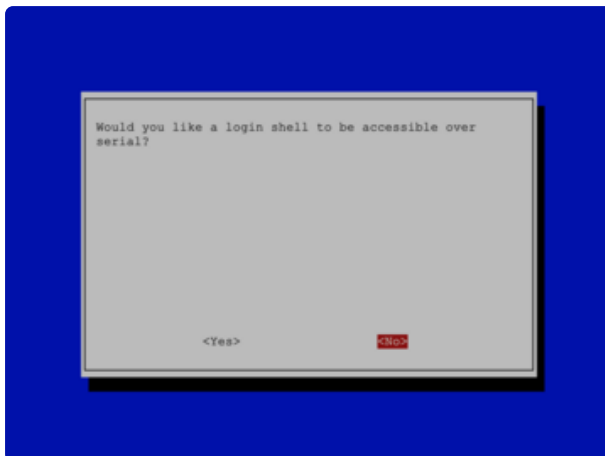
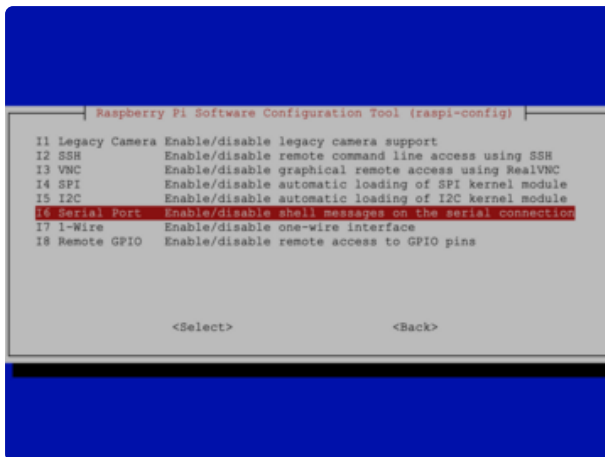
```
sudo raspi-config
```



Navigate down to “Interface Options” and then “Serial Port.” Answer “No” to the login shell question, and “Yes” to the serial port hardware.

Navigate back to the main menu, tab to the “Finish” button and reboot when prompted.

With each new OS release, it’s normal that some configuration options may move to different menus or positions. If you can’t find it where shown, check under the other top-level menu options... even if moved, the name will likely remain similar.



Building libnfc

Step One: Prepare for and Download libnfc

Before you can do anything, you will need to get the libnfc library. Make sure you have internet access on your Pi, through Ethernet or WiFi.

If using the full/GUI Raspberry Pi OS, open a terminal window for typing commands.

```
cd
sudo apt-get install git autoconf libtool libusb-dev
git clone https://github.com/nfc-tools/libnfc
```

This will create a new directory “libnfc” in your home directory.

Step Two: Set Up libnfc For the Pi

Before libnfc can be built, it needs to be configured for the target system and based on some parameters specific the NFC device you have connected.

libnfc requires a configuration file in a specific location. We'll start by creating the required folder, then (unrelated) change to the libnfc directory for subsequent steps.

```
sudo mkdir -p /etc/nfc/devices.d
cd libnfc
```

Next step varies by target system. This refers to the machine where the NFC reader will be connected...occasionally one might set up the OS on one machine, then move the card to a different Pi for use. Follow one or other, not both! If you upgrade or downgrade the Raspberry Pi to a different model later, you might need to recompile everything, depending on which Pi models.

For Recent Raspberry Pi Models

This step works for Raspberry Pi 400, Pi 4, and Pi 3 models B and B+. It almost certainly works on Compute Module 4 and Pi Zero 2W, but did not have hardware on-hand to confirm (worst case, if later tests fail, you can “make clean” and re-try the alternate instructions below).

Enter this command to install the libnfc serial configuration file for these Pi models:

```
sudo cp contrib/libnfc/pn532_uart_on_rpi_3.conf.sample /etc/nfc/devices.d/  
pn532_uart_on_rpi_3.conf
```

This should be entered as one continuous long line...it might appear wrapped to two lines in your browser, but enter it as one with a space character in-between. Or use the “Copy Code” button.

For Earlier Pi Models

This step works for Raspberry Pi 2, Pi 1, and initial Pi Zero models (not Pi Zero 2W). This may work on the wireless Pi Zero W models, but did not have hardware on-hand to confirm (worst case, if later tests fail, you can “make clean” and try the alternate instructions above).

Enter this command to install the libnfc serial configuration file for these systems:

```
sudo cp contrib/libnfc/pn532_uart_on_rpi.conf.sample /etc/nfc/devices.d/  
pn532_uart_on_rpi.conf
```

This should be entered as one continuous long line...it might appear wrapped to two lines in your browser, but enter it as one with a space character in-between. Or use the “Copy Code” button.

Step Three: Configure the Library

The next step is to configure the project itself using the 'configure' tool, as follows:

```
autoreconf -vis  
./configure --with-drivers=pn532_uart --sysconfdir=/etc --prefix=/usr
```

This may take a minute or two to complete, that's normal. The output will resemble the following:

```
config.status: creating contrib/win32/Makefile
config.status: creating contrib/win32/sys/Makefile
config.status: creating contrib/win32/libnfc/Makefile
config.status: creating contrib/win32/libnfc/buses/Makefile
config.status: creating examples/Makefile
config.status: creating examples/pn53x-tamashell-scripts/Makefile
config.status: creating include/Makefile
config.status: creating include/nfc/Makefile
config.status: creating libnfc.pc
config.status: creating libnfc/Makefile
config.status: creating libnfc/buses/Makefile
config.status: creating libnfc/chips/Makefile
config.status: creating libnfc/drivers/Makefile
config.status: creating test/Makefile
config.status: creating utils/Makefile
config.status: creating config.h
config.status: config.h is unchanged
config.status: executing depfiles commands
config.status: executing libtool commands

Selected drivers:
pcsc..... no
acr122_pcsc..... no
acr122_usb..... no
acr122s..... no
arygon..... no
pn53x_usb..... no
pn532_uart..... yes
pn532_spi..... no
pn532_i2c..... no
pn71xx..... no
pi@raspberrypi:~/libnfc $
```

Step Four: Build and Install!

Once configured, the following commands then build and install the library:

```
make
sudo make install all
```

This may take a few minutes to complete. Some compiler warning messages might be generated along the way...these can be safely ignored.

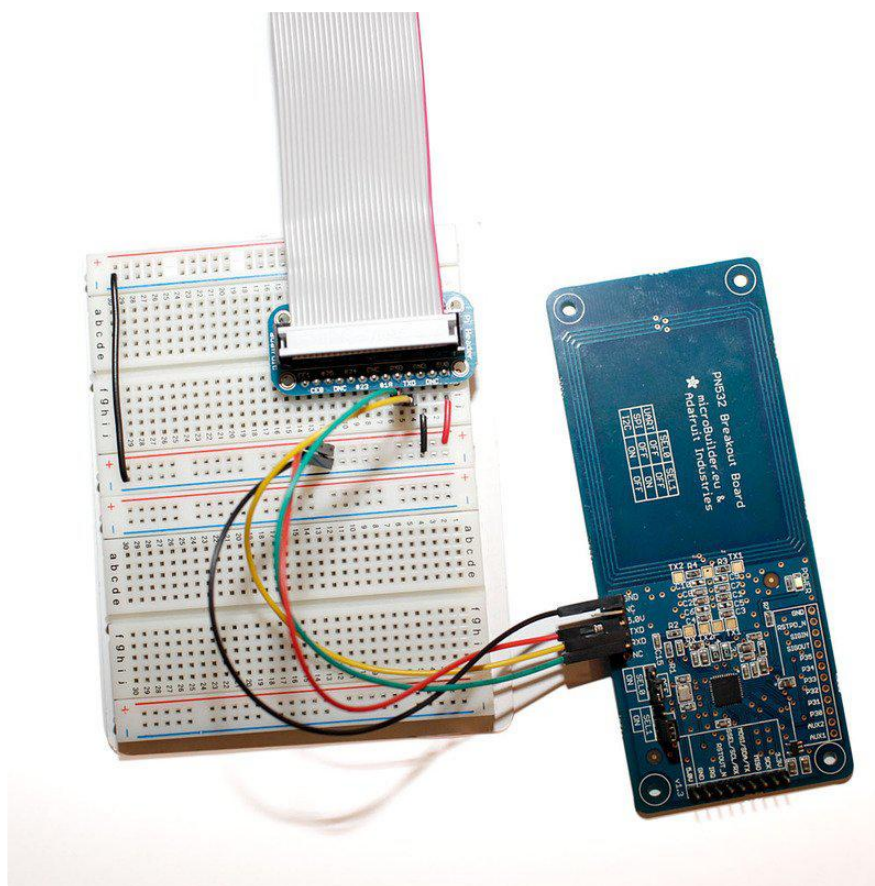
Once the process is complete, you're ready to test on actual hardware...

Testing it Out

Hooking Everything Up

The Adafruit [NFC Breakout](http://adafru.it/364) (<http://adafru.it/364>) board is much more appropriate with the Pi than the [NFC Shield](http://adafru.it/789) (<http://adafru.it/789>), since the breakout doesn't have 5V level shifting (which means you won't accidentally damage your Pi!), and you have easier access to the bus select pins, etc.

If it isn't already hooked up, you can connect your breakout now using a convenient [Pi Cobbler](http://adafru.it/914) (<http://adafru.it/914>), following the image below:



Note: Make sure that the SEL0 and SEL1 jumpers on the NFC breakout are set to OFF, which will cause the PN532 to boot into UART mode (rather than SPI and I2C, which aren't currently supported by libnfc). You will need to reset the breakout after changing these pins, which you can do by cycling the power pin.

Use a 5V supply pin from the Pi, and the 5V input on either the FTDI header or board end header, rather than the Pi's 3.3V supply, since the 3.3V supply is used by the core on the Raspberry Pi and you don't want to pull sharp, heavy loads from it, like when you first enable and charge the near field.

Raspberry Pi	NFC Breakout Board
5V	5.0V
GND	GND
GPIO14 (TXD0)	RX

GPIO15 (RXD0)

TX

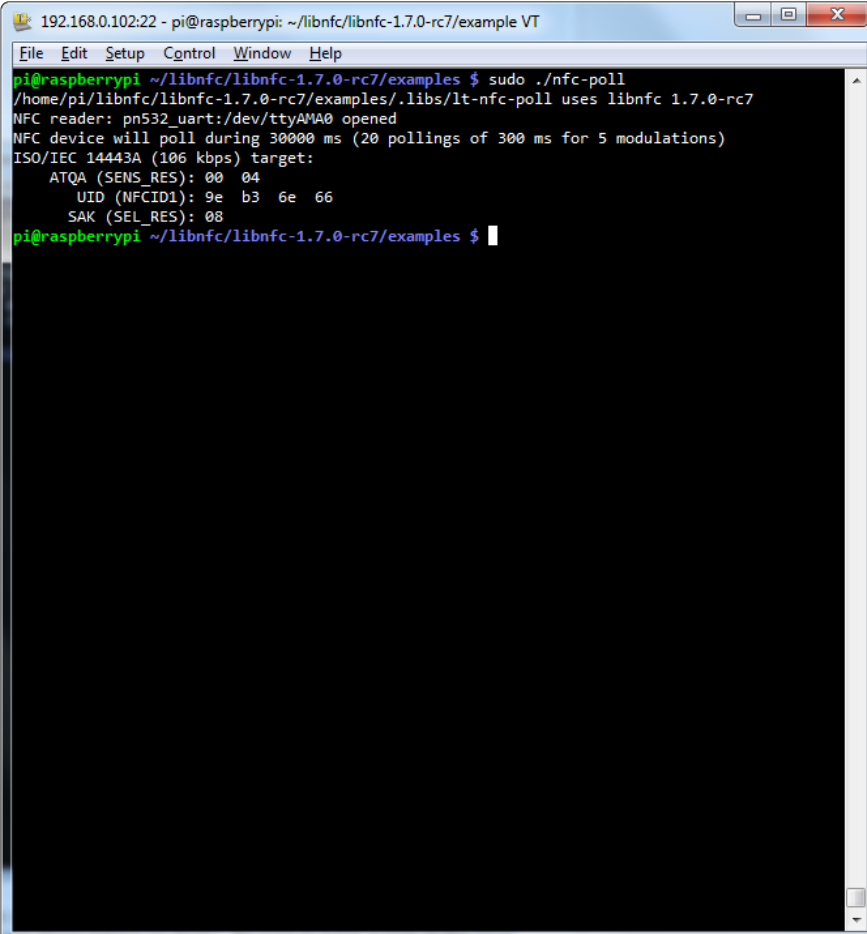
Read an ISO14443-A (Mifare, etc.) Card with nfc-poll

With libnfc configured, built and installed, you can go back to the command-line, place a card on the reader, and run the following command to get the tags unique ID:

```
nfc-poll
```

nfc-poll should be able to run this way thanks to the “make install” step on the prior page, which puts it in the /user/bin directory. If it doesn’t run, check if you did the install step, or you can find the executable program in ~/libnfc/examples

A successful run will yield the following:



```
192.168.0.102:22 - pi@raspberrypi: ~/libnfc/libnfc-1.7.0-rc7/example VT
File Edit Setup Control Window Help
pi@raspberrypi ~/libnfc/libnfc-1.7.0-rc7/examples $ sudo ./nfc-poll
/home/pi/libnfc/libnfc-1.7.0-rc7/examples/.libs/lt-nfc-poll uses libnfc 1.7.0-rc7
NFC reader: pn532_uart:/dev/ttyAMA0 opened
NFC device will poll during 30000 ms (20 pollings of 300 ms for 5 modulations)
ISO/IEC 14443A (106 kbps) target:
  ATQA (SENS_RES): 00 04
  UID (NFCID1): 9e b3 6e 66
  SAK (SEL_RES): 08
pi@raspberrypi ~/libnfc/libnfc-1.7.0-rc7/examples $
```

That's it! From here, you can explore some of the other examples in the 'examples' folder, and figure out how to get started writing your own applications based on libnfc! Be sure to have a look at the [libnfc project page](https://adafru.it/YkE) (<https://adafru.it/YkE>).