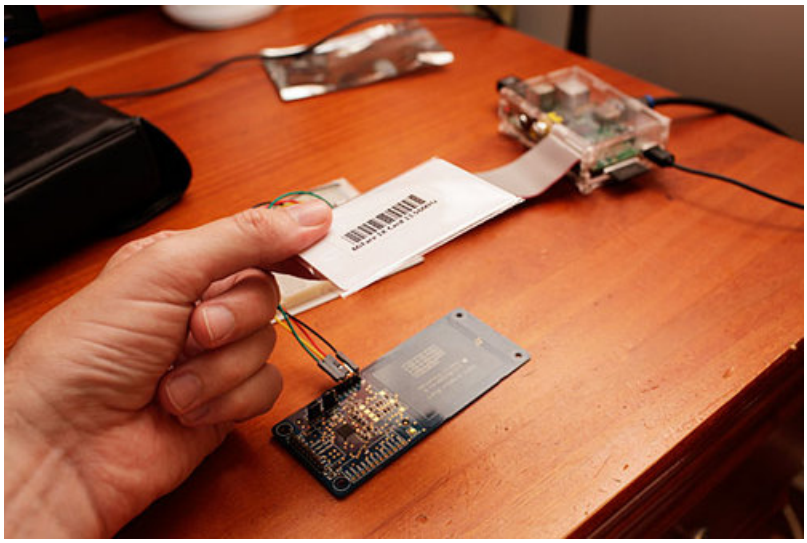
adafruit learning system

Adafruit NFC/RFID on Raspberry Pi

Created by Kevin Townsend

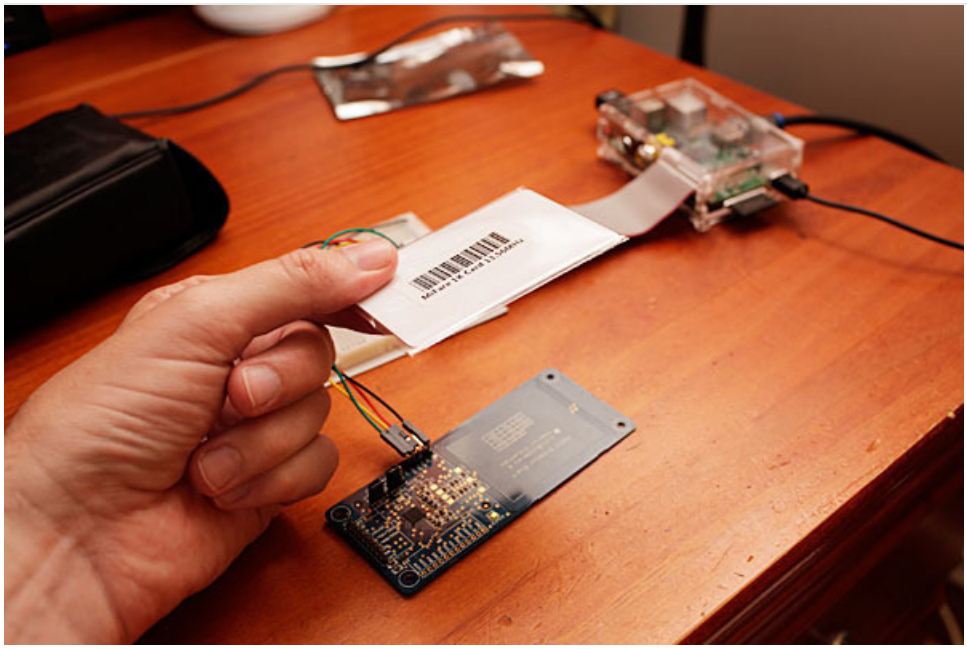


Last updated on 2018-08-22 03:31:24 PM UTC

Guide Contents

Guide Contents	2
Overview	3
Freeing UART on the Pi	4
Step One: Run raspi-conf	4
Step Two: Disable Serial Console	4
Step Three: Enable UART	4
Lastly: Reset	4
Building libnfc	5
Step One: Download libnfc	5
Step Two: Setup libnfc for the Pi	5
Step Three: Run Config	6
Step Four: Build!	7
Testing it Out	9
Hooking Everything Up	9
Read an ISO14443-A (Mifare, etc.) Card with nfc-poll	9

Overview



Interested in adding some NFC fun and excitement to your Raspberry Pi? You're in luck!

One of the big advantages of Linux is that it includes a large number of stacks that have been 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/aN2) having been around for a quite some time -- in fact, it's the original reason the NFC Breakout was developed!

To get libnfc playing well with your Pi and your Adafruit NFC breakout you'll need to make some minor modification to your vanilla Wheezy distribution, and one small change to the latest NFC code (1.7.0-rc7 as of this writing), 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!

Freeing UART on the Pi

The easiest way to use libnfc with the Adafruit NFC Breakout is via UART, since it's well-supported by libnfc out of the box. Unfortunately the 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 Raspbian install via noobs) should free UART up for us:

Step One: Run raspi-conf

From the command prompt enter the following command:

```
$ sudo raspi-config
```

Step Two: Disable Serial Console

From the main menu, select **option 7 (Serial)**, then **select 'No'** to disable shell and kernel messages via UART.

Step Three: Enable UART

On the latest Jessie Raspbian, you may also need to enable the UART for user usage! Edit config.txt with `sudo nano /boot/config.txt`

and remove the line

`enable_uart=0`

if it exists. And add at the end of the file

`enable_uart=1`

Save the file

Lastly: Reset

From the command prompt enter the following commands to reboot your Raspberry Pi board, freeing UART up in the process:

```
$ sudo reboot
```

Building libnfc

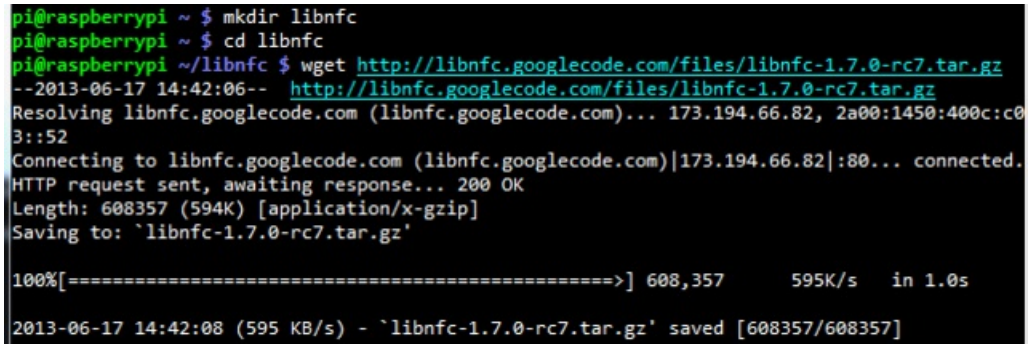
Step One: Download libnfc

Before you can do anything, you will need to get a copy of libnfc. Make sure you have an Ethernet cable connected to your Pi, and run the following commands to get libnfc 1.7.0

> UPDATE 22 Jan 2018: LIBNFC has moved to Github, and 1.7.0 can be downloaded directly from this URL: https://github.com/nfc-tools/libnfc/releases/tag/libnfc-1.7.0

```
$ cd /home/pi
$ mkdir libnfc
$ cd libnfc
$ wget https://github.com/nfc-tools/libnfc/releases/download/libnfc-1.7.0/libnfc-1.7.0.tar.bz2
$ tar -xvjf libnfc-1.7.0.tar.bz2
```

You should see something similar to the following:



Step Two: Setup 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 1.7.0 and later use a new config file, which needs to be placed at a specific location. Thankfully, libnfc 1.7.0 includes a config file for the Raspberry Pi, which you can copy to the right destination with the following commands:

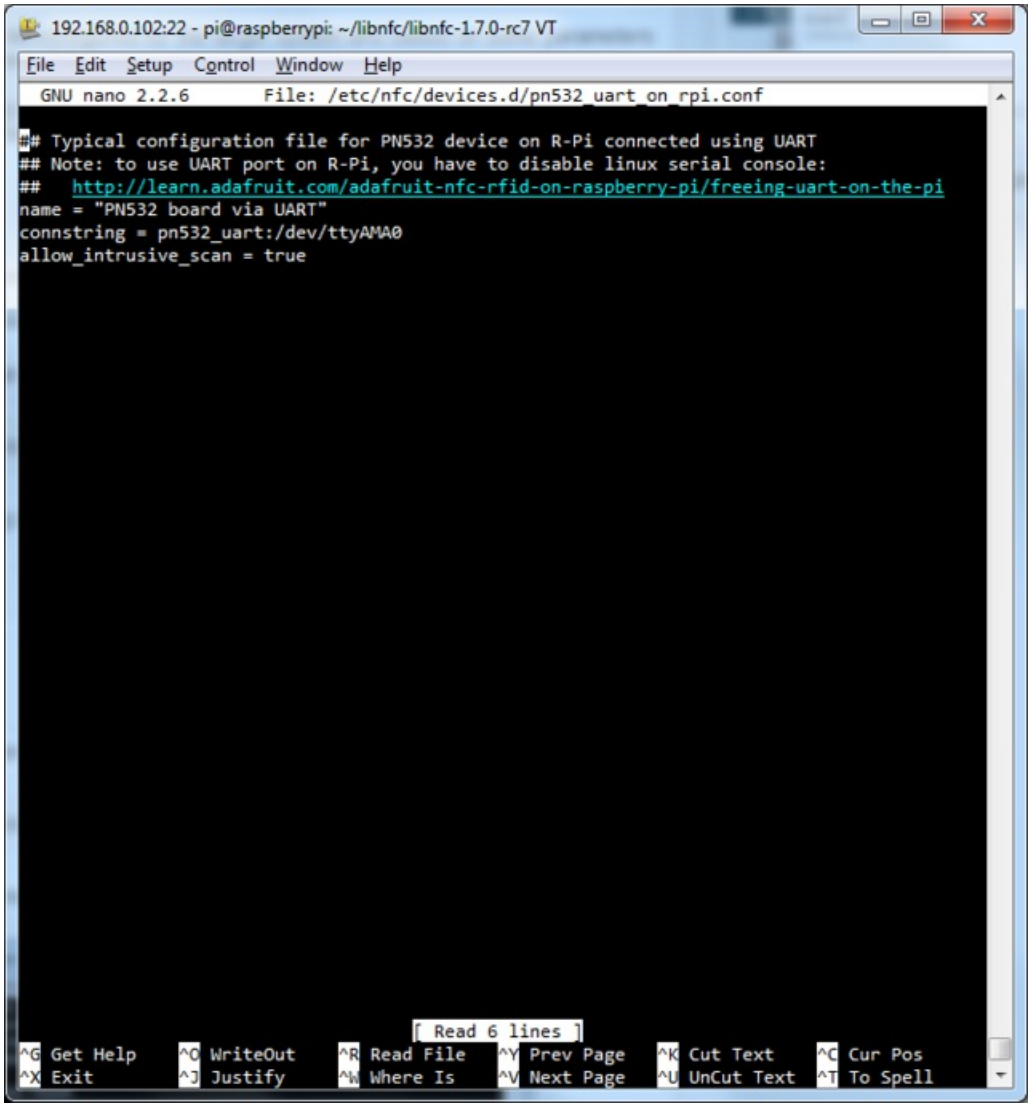
```
$ cd libnfc-1.7.0
$ sudo mkdir /etc/nfc
$ sudo mkdir /etc/nfc/devices.d
$ sudo cp contrib/libnfc/pn532_uart_on_rpi.conf.sample /etc/nfc/devices.d/pn532_uart_on_rpi.conf
```

Next, we need to make a small change to this file, which we can do by entering:

```
sudo nano /etc/nfc/devices.d/pn532_uart_on_rpi.conf
```

Update the file to include the following line at the bottom:

```
allow_intrusive_scan = true
```



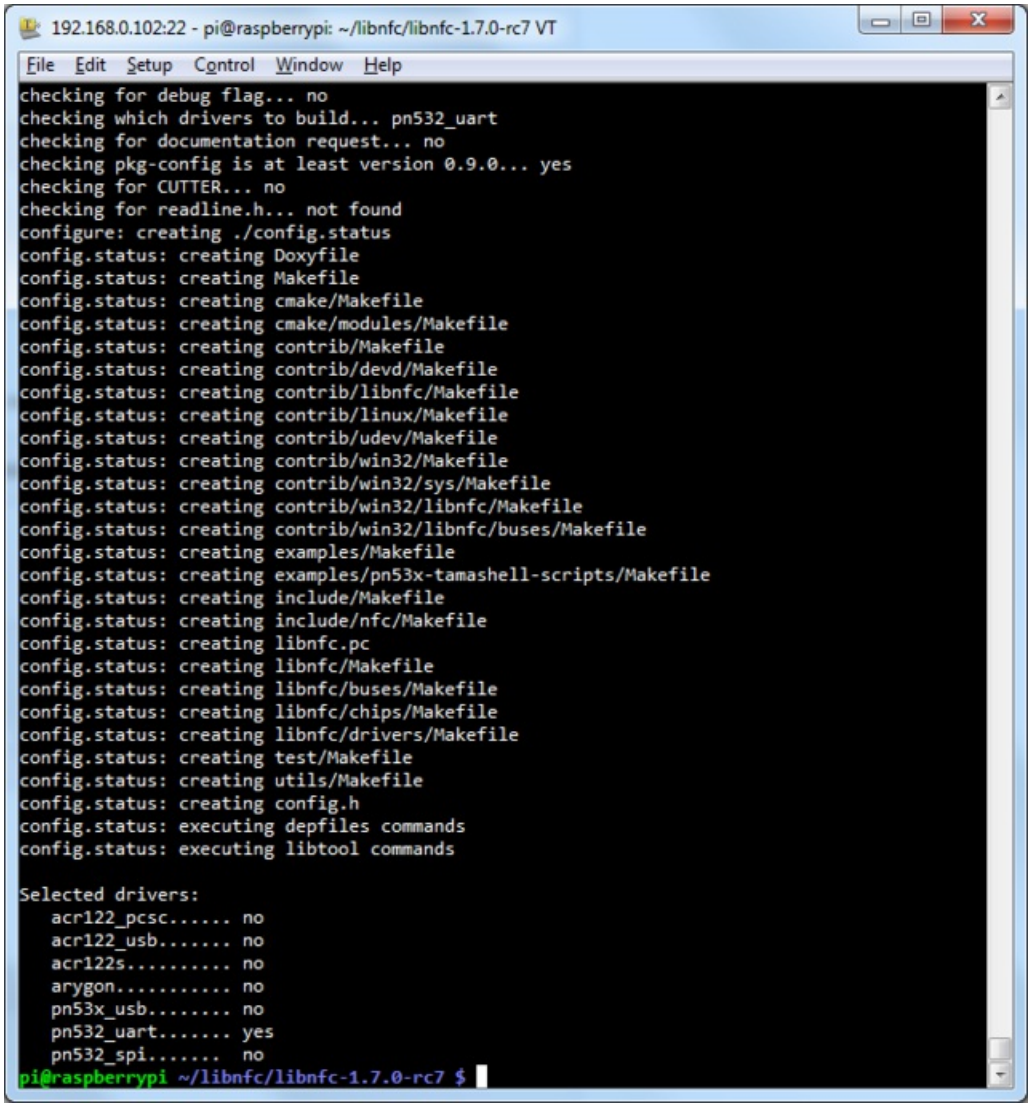
Press CTRL+O then enter to save the file, and then CTRL+X to quite nano.

Step Three: Run Config

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

```
$ sudo apt-get install autoconf
$ sudo apt-get install libtool
$ sudo apt-get install libpcsclite-dev libusb-dev
$ autoreconf -vis
$ ./configure --with-drivers=pn532_uart --sysconfdir=/etc --prefix=/usr
```

This should give you the following screen once the configuration process is complete (it takes a few minute though):

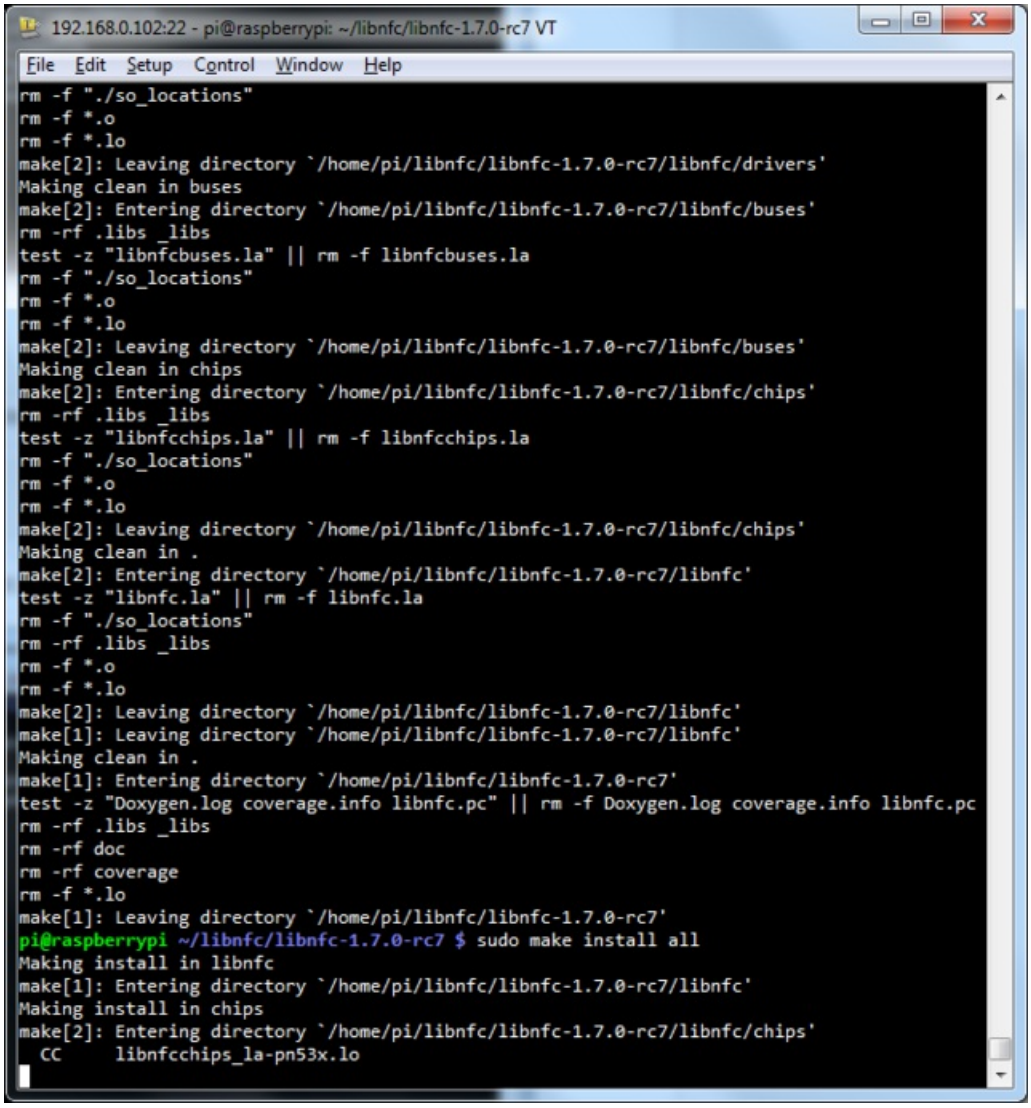


Step Four: Build!

To build libnfc, you simply need to enter the following commands:

```
$ sudo make clean
$ sudo make install all
```

Which should start the (slowish!) build process as follows:



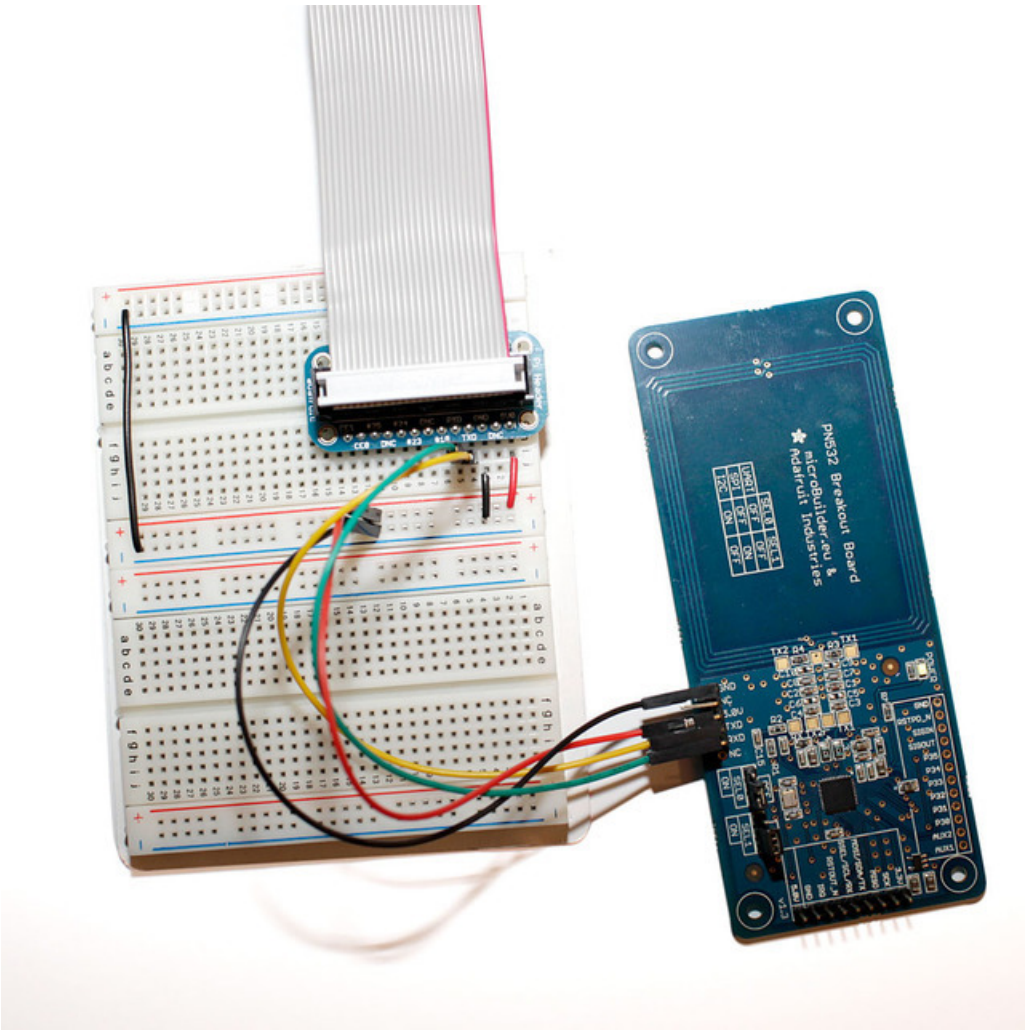
Once the build process is complete, you're ready to go on to testing the library on actual HW ...

Testing it Out

Hooking Everything Up

The Adafruit NFC Breakout (http://adafru.it/364) board is much more appropriate with the Pi than the NFC Shield (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), 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.

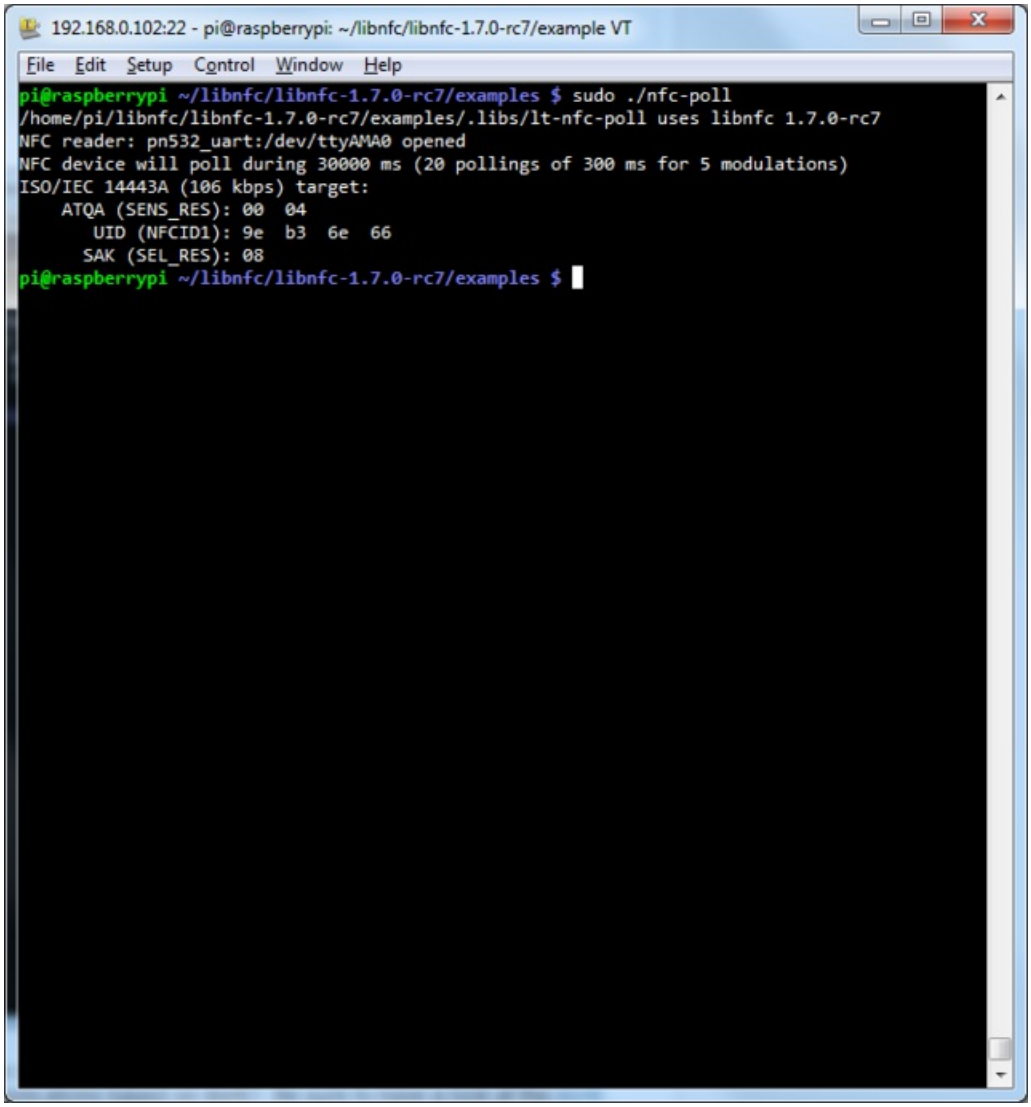
Use the 5V supply on the Pi Cobbler, and the 5V input on the FTDI header rather than the 3.3V supply, since the 3.3V supply is used by the core on the rasberry Pi and you don't want to pull sharp, heavy loads from it, like when you first enable and charge the near field.

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

With libnfc built and properly configure, 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:

```
$ cd examples
$ sudo ./nfc-poll
```

Which should results in the following:



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/aN3) which also contains a useful and active forum.