

Raspberry Pi

RASPBERRY Pi NFC

Model:PN532

User Manual



Overview:

TEAD PN532 NFC module, as its name implies, is based on PN532 chip and used for 13.56MHz near field communication. The module is equipped with onboard antenna, thus no external antenna coil is needed. It is compatible with SPI, IIC and UART interface for communication. With NFC library support for Arduino and Raspberry Pi offered by us, it is quite convenient for development of products with NFC functions.

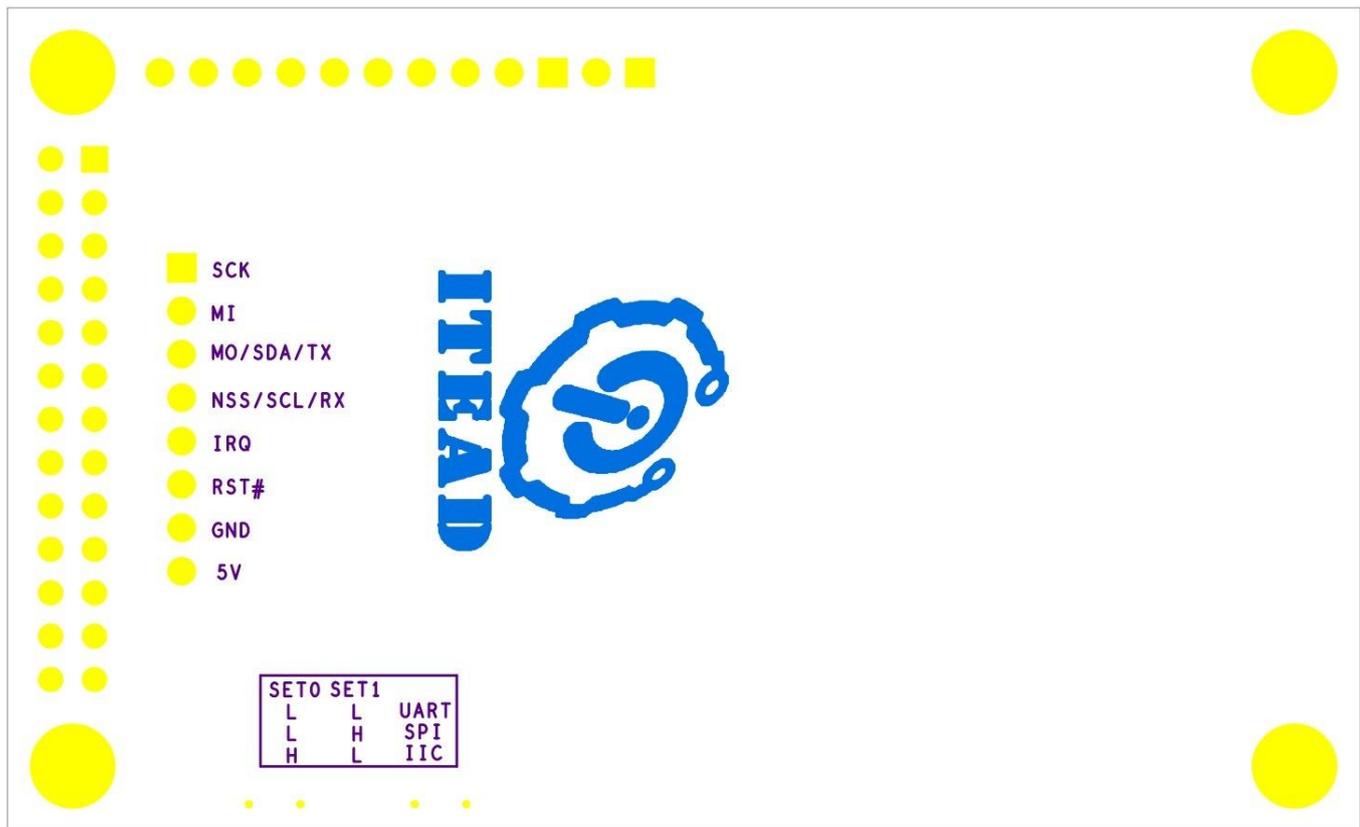
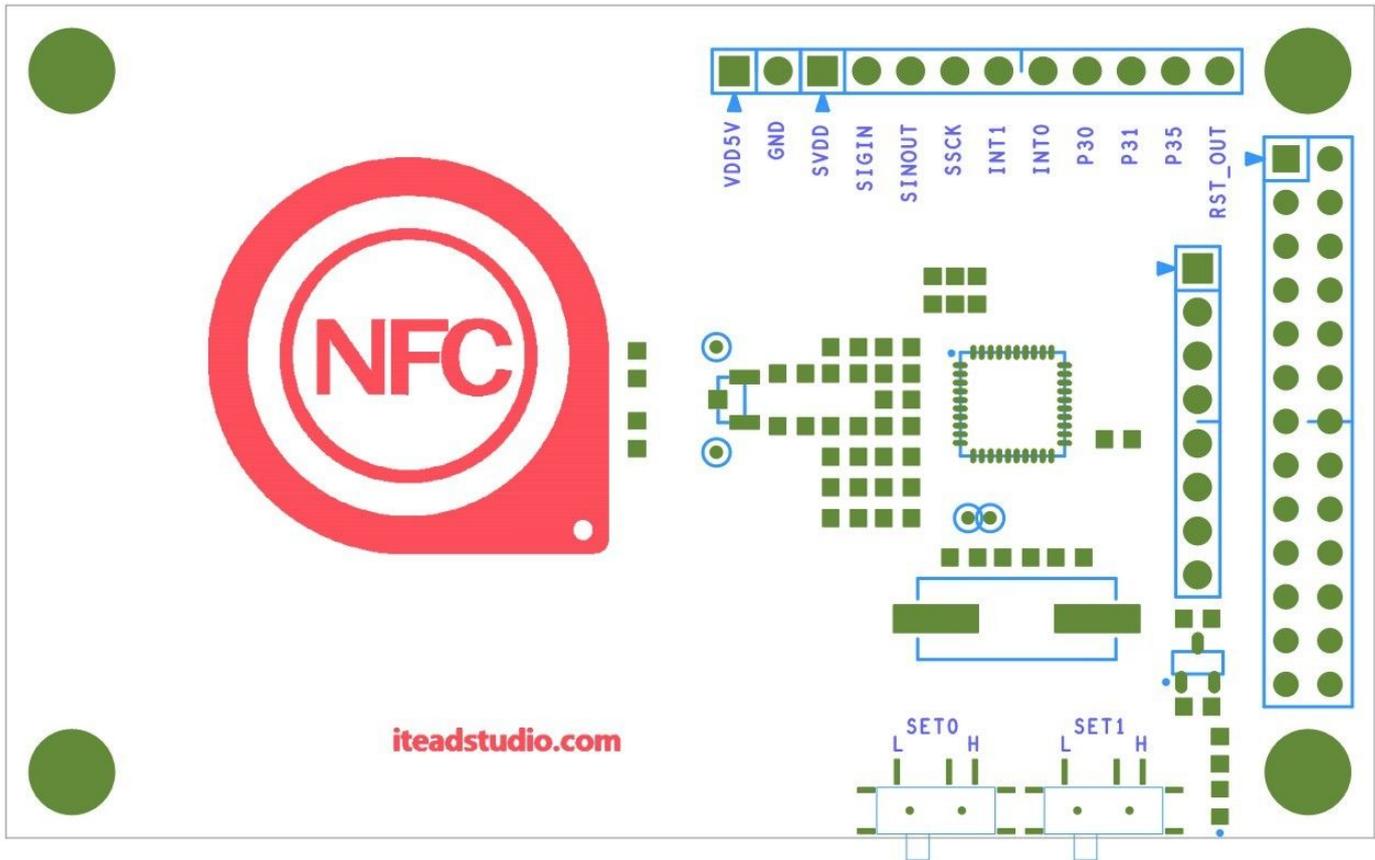
Features:

- Longest effective communication distance of 3 cm
- Supports switching of SPI, IIC and UART interface.
- Can be used for 13.56M non-contact communication
- Compatible with ISO14443 Type A and Type B standards

Specifications:

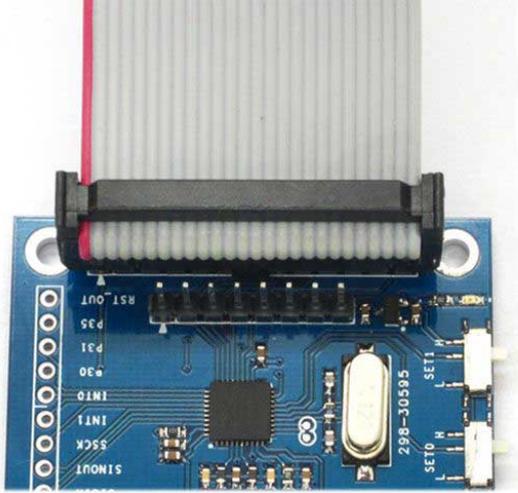
IC	NXP PN532
Operating Voltage	3.3V
Power Supply Voltage	3.3~5.5V
Max Supply Current	150mA
Working Current(Standby Mode)	100mA
Working Current(Write Mode)	120mA
Working Current(Read Mode)	120mA
Indicator	PWR
Interface	SPI Interface, Std Raspberry Pi 20pins Interface
Board Size	79 x 48.3 x 11.3 mm

Hardware:



There are two slide switches on the board for selection of interface mode:

	SET0	SET1
UART	L	L
SPI	L	H
IIC	H	L



Wiring:

NFC connection:

NFC_5V-> UNO_5V

NFC_GND-> UNO_GND

NFC_NSS-> UNO_D10

NFC_MO-> UNO_D11

NFC_MI-> UNO_D12

NFC_SCK-> UNO_D13

SD Card Connection:

SD_VCC-> UNO_5V

SD_GND-> UNO_GND

SD_CS-> UNO_D4

SD_DI-> UNO_D11

SD_DO-> UNO_D12

SD_SCK-> UNO_D13

The SD card into the SD card module.

DS1307 module wiring:

DS1307_VCC-> UNO_5V

DS1307_GND-> UNO_GND

DS1307_SDA-> UNO_A4

DS1307_SCL-> UNO_SDA

The coin cell battery holder connected to the DS1307.

Buzzer Wiring:

BUZ_S-> UNO_D5

BUZ_V-> UNO_5V

BUZ_G-> UNO_GND

[Raspberry Pi Drives ITEAD PN532 NFC Module with libnfc](#)

ITEAD PN532 NFC module is equipped with double-row pins, which can be connected directly to the Raspberry Pi via connection cable and then drive the module for non-contact near field communication operations, such as reading and writing 13.56M IC card.

libnfc is the first libre low level NFC SDK and Programmers API released under the GNU Lesser General Public License. It provides complete transparency and royalty-free use for everyone. The library currently supports modulations for ISO/IEC 14443 A and B, FeliCa, Jewel/Topaz tags and Data Exchange Protocol (P2P) as target and as initiator.

The following tutorial demonstrates how to use libnfc on Raspberry Pi to drive ITEAD PN532 module with the SPI bus.

1- Hardware connection

As Itead PN532 Module is customized for Raspberry Pi, it can be connected directly to Raspberry Pi via the adapter cable as shown in picture below

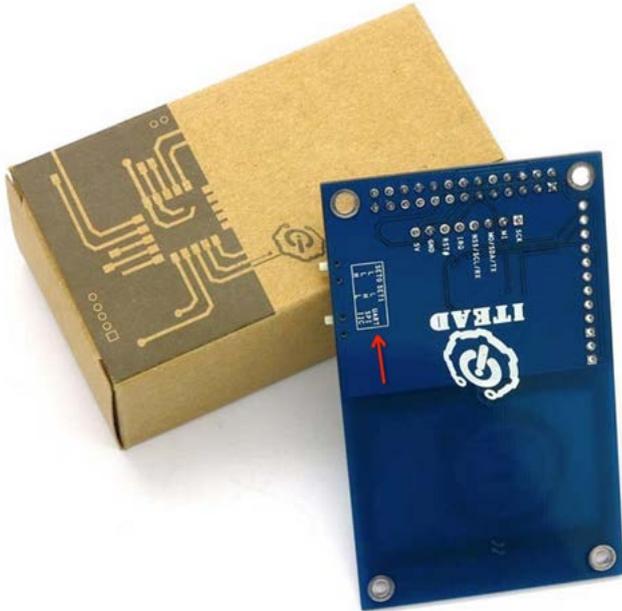


According to the connection in the picture above, Itead PN532 Module is connected via the SPI bus with raspberry pi, therefore, operating mode of PN532 Module should be set to SPI mode, as shown below:

```
SET0-->L  
SET1-->H
```

If you want to use I2C mode of PN532 module, you should set the code as below:

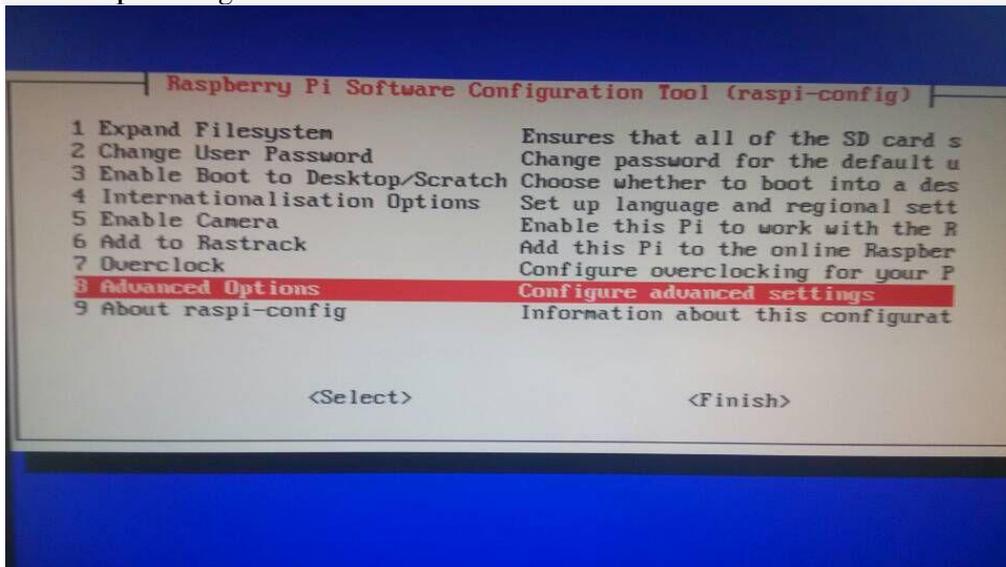
```
SET0-->H  
SET1-->L
```



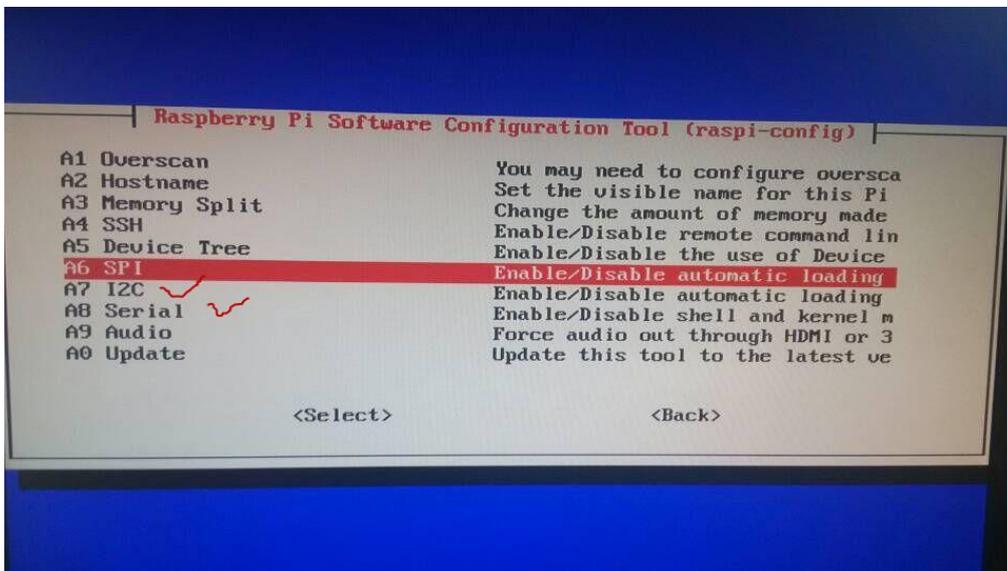
Before install the needed software, please do the configuration first.

Raspberry Pi Software Configuration

```
sudo raspi-config
```



Chosse to configure SPI, I2C, and Serial.



2- Install the software pack needed

```
sudo apt-get update
sudo apt-get install libusb-dev libpcsclite-dev
```

3, Download and compress source code pack of libnfc

```
cd ~
wget http://dl.bintray.com/nfc-tools/sources/libnfc-1.7.1.tar.bz2
tar -xf libnfc-1.7.1.tar.bz2
```

4, Compile and install

```
cd libnfc-1.7.1
./configure --prefix=/usr --sysconfdir=/etc
make
sudo make install
```

5, Modify configuration file

```
cd /etc
sudo mkdir nfc
sudo nano /etc/nfc/libnfc.conf
```

Copy and paste the following contents to file /etc/nfc/libnfc.conf:

```
# Allow device auto-detection (default: true)
# Note: if this auto-detection is disabled, user has to set manually a device
# configuration using file or environment variable
allow_autoscan = true

# Allow intrusive auto-detection (default: false)
# Warning: intrusive auto-detection can seriously disturb other devices
# This option is not recommended, user should prefer to add manually his device.
allow_intrusive_scan = false

# Set log level (default: error)
# Valid log levels are (in order of verbosity): 0 (none), 1 (error), 2 (info), 3 (debug)
# Note: if you compiled with --enable-debug option, the default log level is "debug"
log_level = 1

# Manually set default device (no default)
```

```
# To set a default device, you must set both name and connstring for your device
# Note: if autoscan is enabled, default device will be the first device available in device list.
device.name = "Itead_PN532_SPI"
device.connstring = "pn532_spi:/dev/spidev0.0:500000"
```

6, As Raspberry Pi defaults to disable the driver for SPI module, we need to enable it

```
sudo nano /etc/modprobe.d/raspi-blacklist.conf
```

Add # in front of blacklist spi-bcm2708 and it will become #blacklist spi-bcm2708
As shown in the picture below



```
Jerry — ssh — Solarized Dark ansi — 80x24
GNU nano 2.2.6 File: /etc/modprobe.d/raspi-blacklist.conf
#
blacklist spi and i2c by default (many users don't need them)
#blacklist spi-bcm2708
#blacklist i2c-bcm2708

[ Read 4 lines ]
^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text    ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page  ^U UnCut Text ^I To Spell
```

You can see two spi devices under /dev after rebooting.



```
Jerry — ssh — Solarized Dark ansi — 80x24
pi@raspberrypi ~ $ ls /dev/spi*
/dev/spidev0.0 /dev/spidev0.1
pi@raspberrypi ~ $
```

7, Up to now, preparations are done. And then we can use command nfc-list to check if it is successfully installed.

```
Jerry — ssh — Solarized Dark ansi — 80x24
pi@raspberrypi ~ $ ls /dev/spi*
/dev/spidev0.0 /dev/spidev0.1
pi@raspberrypi ~ $ nfc-list
nfc-list uses libnfc 1.7.1
NFC device: pn532_spi:/dev/spidev0.0 opened
pi@raspberrypi ~ $
```

Let's try swiping the card.

```
Jerry — ssh — Solarized Dark ansi — 80x24
pi@raspberrypi ~ $ ls /dev/spi*
/dev/spidev0.0 /dev/spidev0.1
pi@raspberrypi ~ $ nfc-list
nfc-list uses libnfc 1.7.1
NFC device: pn532_spi:/dev/spidev0.0 opened
pi@raspberrypi ~ $ nfc-poll
nfc-poll uses libnfc 1.7.1
NFC reader: pn532_spi:/dev/spidev0.0 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): 8d 90 e9 00
  SAK (SEL_RES): 08

```

8, Configurations for connection via I2C bus

8.1 Software configuration

Enable I2C bus on Raspberry Pi, and change blacklist `i2c-bcm2708` in `/etc/modprobe.d/raspi-blacklist.conf` to `#blacklist i2c-bcm2708`

Add `i2c-dev` at the end of `/etc/modules`

Modify last line of `/etc/nfc/libnfc.conf` to `pn532_i2c:/dev/i2c-1` as shown in the picture below

```
Jerry — ssh — Solarized Dark ansi — 80x24

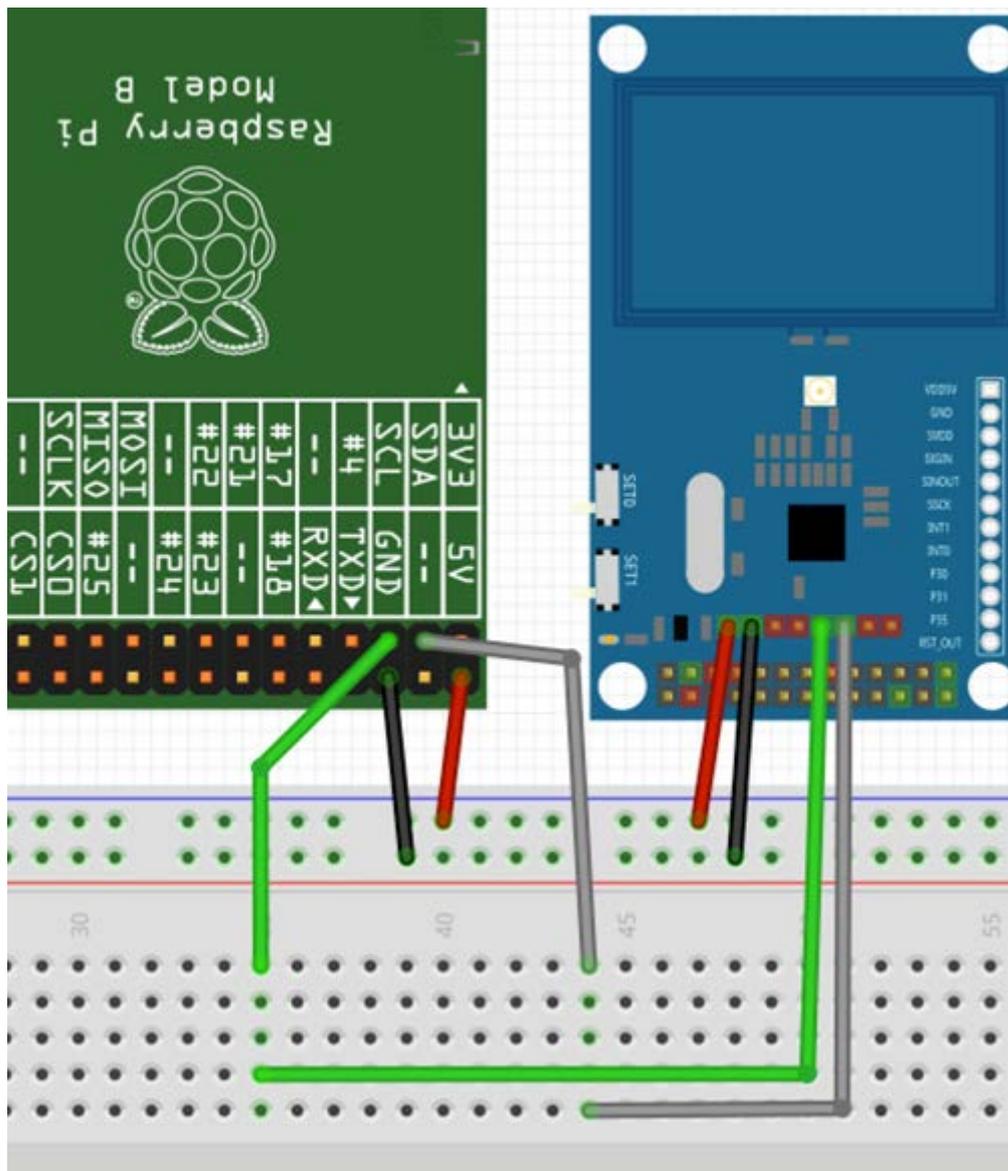
# Set log level (default: error)
# Valid log levels are (in order of verbosity): 0 (none), 1 (error), 2 (info), 3
(debug)
# Note: if you compiled with --enable-debug option, the default log level is "de
bug"
log_level = 1

# Manually set default device (no default)
# To set a default device, you must set both name and connstring for your device
# Note: if autoscan is enabled, default device will be the first device availabl
e in device list.
device.name = "Itead_PN532"
device.connstring = "pn532_i2c:/dev/i2c-1"
pi@raspberrypi ~ $ less /etc/modules
# /etc/modules: kernel modules to load at boot time.
#
# This file contains the names of kernel modules that should be loaded
# at boot time, one per line. Lines beginning with "#" are ignored.
# Parameters can be specified after the module name.

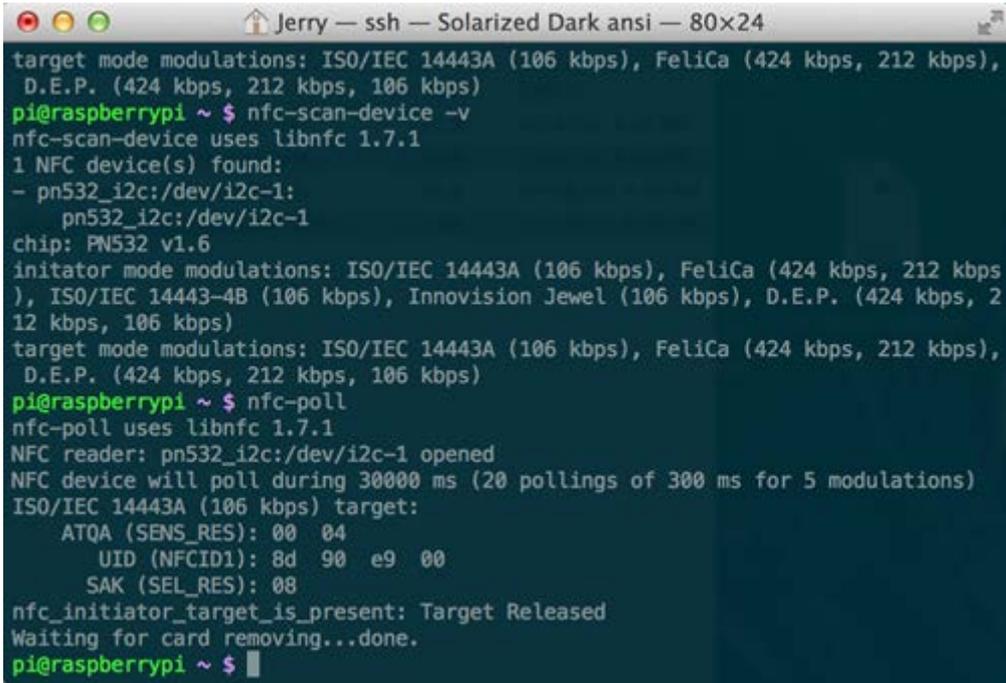
snd-bcm2835
i2c-dev
pi@raspberrypi ~ $
```

8.2 Hardware connection

Connection is as shown in the picture below:



Running result is as shown in the picture below:

A terminal window titled "Jerry — ssh — Solarized Dark ansi — 80x24" showing the execution of NFC-related commands on a Raspberry Pi. The terminal output includes the results of 'nfc-scan-device -v' and 'nfc-poll'. The scan command identifies a PN532 i2c device at /dev/i2c-1 with a PN532 v1.6 chip. The poll command shows the device is opened and will poll for 30000 ms. The poll results show a target in ISO/IEC 14443A mode with ATQA (SENS_RES) 00 04, UID (NFCID1) 8d 90 e9 00, and SAK (SEL_RES) 08. The terminal ends with the prompt 'pi@raspberrypi ~ \$' and a cursor.

```
Jerry — ssh — Solarized Dark ansi — 80x24
target mode modulations: ISO/IEC 14443A (106 kbps), FeliCa (424 kbps, 212 kbps),
D.E.P. (424 kbps, 212 kbps, 106 kbps)
pi@raspberrypi ~ $ nfc-scan-device -v
nfc-scan-device uses libnfc 1.7.1
1 NFC device(s) found:
- pn532_i2c:/dev/i2c-1:
  pn532_i2c:/dev/i2c-1
chip: PN532 v1.6
initiator mode modulations: ISO/IEC 14443A (106 kbps), FeliCa (424 kbps, 212 kbps
), ISO/IEC 14443-4B (106 kbps), Innovision Jewel (106 kbps), D.E.P. (424 kbps, 2
12 kbps, 106 kbps)
target mode modulations: ISO/IEC 14443A (106 kbps), FeliCa (424 kbps, 212 kbps),
D.E.P. (424 kbps, 212 kbps, 106 kbps)
pi@raspberrypi ~ $ nfc-poll
nfc-poll uses libnfc 1.7.1
NFC reader: pn532_i2c:/dev/i2c-1 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): 8d 90 e9 00
  SAK (SEL_RES): 08
nfc_initiator_target_is_present: Target Released
Waiting for card removing...done.
pi@raspberrypi ~ $
```