

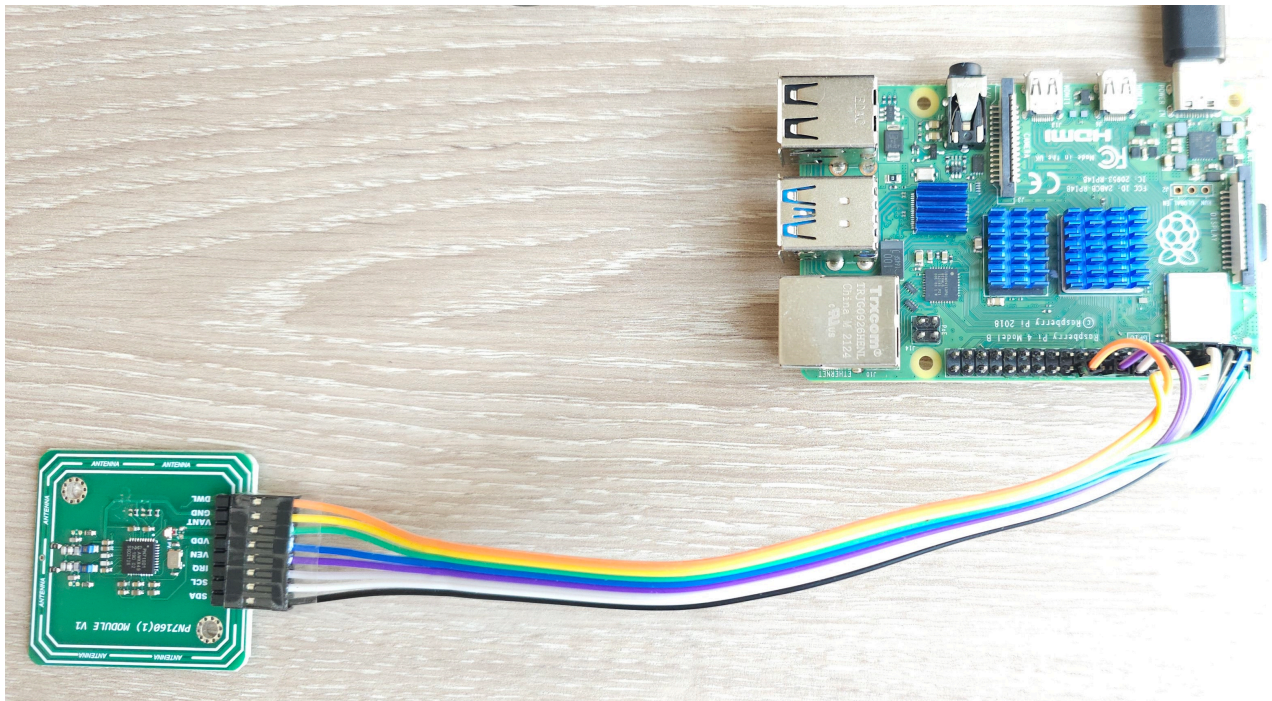
# TESTING GUIDE

## ELECHOUSE PN7160 I2C board quick start guide

This guide is based on NXP AN12991

For the following products:

1. ELECHOUSE PN7160 I2C MODULE
2. ELECHOUSE PN7161 I2C MODULE

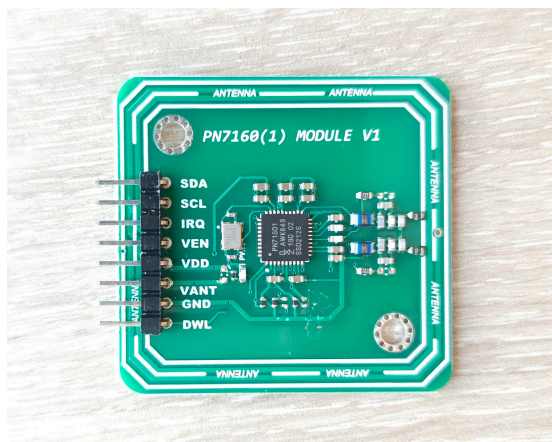


# Quick Startup with Raspberry Pi interface board

## 3.1 Required items

- Raspberry Pi [\[1\]](#) running raspbian distribution.

## 3.2 Hardware setup



		Physical Pins			
Function	BCM	pin#	pin#	BCM	Function
3.3 Volts		VDD 1	2	VANT	5 Volts
GPIO/SDA1 (I2C)	2	SDA 3	4		5 Volts
GPIO/SCL1 (I2C)	3	SCL 5	6	GND	GND
GPIO/GCLK	4	7	8	14	TX UART/GPIO
GND		9	10	15	RX UART/GPIO
GPIO	17	11	12	18	GPIO
GPIO	27	13	14		GND
GPIO	22	15	16	IRQ	GPIO
3.3 Volts		17	18	VEN	GPIO
MOSI (SPI)	10	19	20		GND
MISO (SPI)	9	21	22	DWL	GPIO
SCLK (SPI)	11	23	24		CEO_N (SPI)
GND		25	26	7	CE1_N (SPI)
RESERVED		27	28		RESERVED
GPIO	5	29	30		GND
GPIO	6	31	32	12	GPIO
GPIO	13	33	34		GND
GPIO	19	35	36	16	GPIO
GPIO	26	37	38	20	GPIO
GND		39	40	21	GPIO

### Connection

PN7160	-----Raspberry PI 4
SDA	#3 SDA
SCL	#5 SCL
IRQ	#16 GPIO23
VEN	#18 GPIO24
VDD	#1 3.3V PWR
VANT	#2 OR #4 5V PWR
GND	#6 GND
DWL	#22

## 3.3 Software setup

Use Raspbian (<https://www.raspberrypi.org/software/operating-systems/>). Guidelines to set up Linux environment on raspberry pi can be found here: <https://www.raspberrypi.org/documentation/installation/installing-images/>).

Below is the step-by-step procedure run from the Raspberry Pi to add software support for PN7160:

### 3.3.1 Enable I<sup>2</sup>C interface

1. Run command:

```
sudo raspi-config
```

2. Use the down arrow to select "Interface Options"
3. Arrow down to "P5 I<sup>2</sup>C"
4. Select "yes" when it asks you to enable I<sup>2</sup>C
5. Use the right arrow to select the <Finish> button

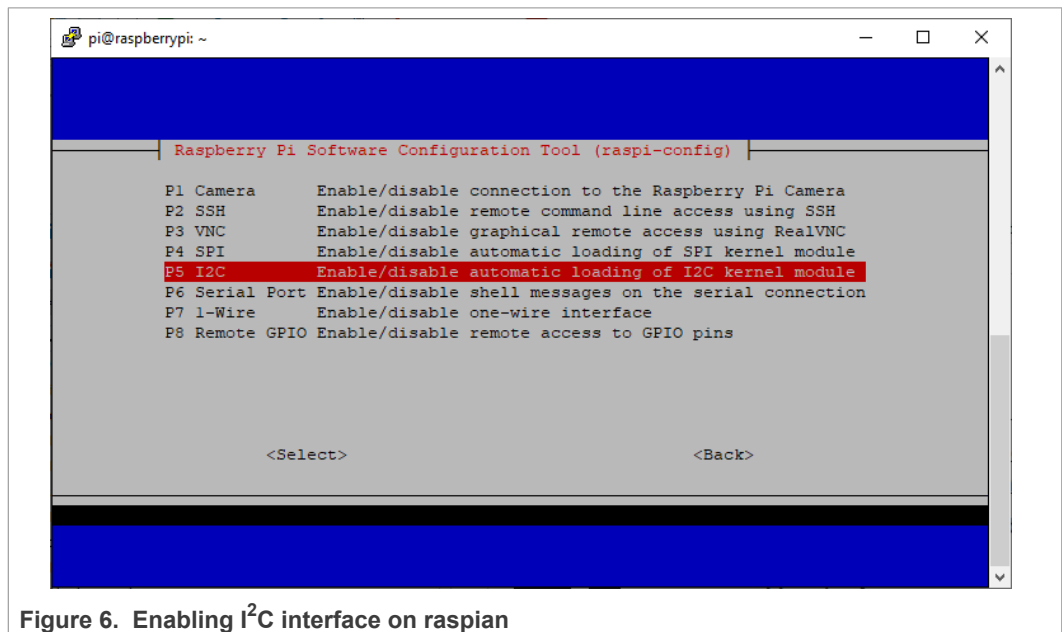


Figure 6. Enabling I<sup>2</sup>C interface on raspian

To verify the I<sup>2</sup>C interface is enabled, enter the following command `ls /dev/i2c*`.

The Pi should respond with `/dev/i2c-1` which represents the user-mode I<sup>2</sup>C interface to which is connected the PN7160.

### 3.3.3 Install necessary tools

Execute the command:

```
sudo apt-get install autoconf automake libtool git
```

### 3.3.4 Clone Linux libnfc-nci library repository

Execute the command:

```
git clone https://github.com/NXPnfcLinux/linux_libnfc-nci.git -b NCI2.0_PN7160
```

### 3.3.5 Configure the library

Execute the commands:

```
cd linux_libnfc-nci
./bootstrap
./configure
```

### 3.3.6 Set the library to map I<sup>2</sup>C interface (only for OM27160A1EVK)

Edit `linux_libnfc-nci/conf/libnfc-nxp.conf` file to update `NXP_TRANSPORT` and `NXP_NFC_DEV_NODE` settings as shown below:

```
#####
# TRANSPORT Type
# 0x00 - I2C /SPI for noraml nxpnfc driver
# 0x01 - Not Used, kept to align with Android code
# 0x02 - ALT_I2C
# 0x03 - ALT_SPI
```

```
NXP_TRANSPORT=0x02
```

```
#####
# NXP HW Device Node information
NXP_NFC_DEV_NODE="/dev/i2c-1"
```

### 3.3.8 Build and install the library

Execute the commands:

```
make
sudo make install
export LD_LIBRARY_PATH=/usr/local/lib
```

To make this last setting permanent, run the following command:

```
echo "export LD_LIBRARY_PATH=/usr/local/lib" >> .bashrc
```

### 3.3.9 Run the demo application (built and installed together with the library during previous step)

To simply display all data collected from remote NFC device (Peer, reader/writer or card), run the demo application in "poll mode" executing the command:

```
nfcDemoApp poll
```

For more details about the demo application modes execute command:

```
nfcDemoApp --help
```

For more detailed information about the demo application, but also for additional example applications, please refer to [\[2\]](#).

```
pi@raspberrypi: ~/linux_libnfc X + v
pi@raspberrypi:~/linux_libnfc-nci $ ./nfcDemoApp poll
#####
##                                NFC demo                                ##
#####
##                                Poll mode activated                        ##
#####
                                ... press enter to quit ...

Waiting for a Tag/Device...

NFC Tag Found

Type :          'Type A - Mifare Classic'
NFCID1 :        '35 43 9F 93 '
NDEF Content : NO, mode=1, tech=8

Mifare Authenticate command sent
Response :
00

Mifare Read command sent
Response :
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Mifare Write command sent
Response :
00 0A 14
```