|  |  |  |  |
| --- | --- | --- | --- |
| **10** | **Interfacing Raspberry Pi with RFID.** | | |
|  |  | | |
| C:\Users\Hiren\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\BFFDF6A9.tmp  I2C Communication Instructions for Raspberry Pi  1. Open I2C of the Raspberry Pi : | |  |
| sudo raspi-config  Select **5 Interfacing Options** -> **I2C** -> **yes**. | |
|  | 2. **Install some dependent packages** |
| sudo apt-get update  sudo apt-get install libusb-dev libpcsclite-dev i2c-tools | |
|  | 3. **Download and unzip the source code package 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 | |

T.Y.B.Sc. (I.T.) 2019

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | 4. **Compile and install** |  |
| cd libnfc-1.7.1  ./configure --prefix=/usr --sysconfdir=/etc make  sudo make install | |
|  | 5. **Write the configuration file for NFC communication** |
| cd /etc  sudo mkdir nfc  sudo nano /etc/nfc/libnfc.conf  Check the following details of the 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 = "\_PN532\_SPI"  #device.connstring = "pn532\_spi:/dev/spidev0.0:500000" | |

T.Y.B.Sc. (I.T.) 2019

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | device.name = "\_PN532\_I2c" device.connstring = "pn532\_i2c:/dev/i2c-1" | | | | | |  |
|  | 6. **Wiring** | | | | |
| Toggle the switch to the **I2C mode** | | | | | |
|  | SEL 0 | SEL 1 | |  | |
| H | L | |
| Connect the devices: | | | | | |
|  | PN532 | | Raspberry | |  |
| 5V | | 5V 4 | |
| GND | | GND 6 | |
| SDA | | SDA0 3 | |
| SCL | | SCL0 5 | |
| **7. Run *i2cdetect –yes 1* to check whether the I2C device is recognized.** | | | | | |
| If yes, it means both the module and the wiring work well. Then type in *nfc-list* to check the NFC module: | | | | | |

T.Y.B.Sc. (I.T.) 2019

|  |  |  |
| --- | --- | --- |
|  | Run *nfc-poll* to scan the RFID tag and you can read information on the card:    **SPI Communication Instructions for Raspberry Pi**  **1. Open SPI of the Raspberry Pi:** |  |
| sudo raspi-config |
| Select **9 Advanced Options** -> **SPI** -> **yes**.  **2. Install some dependent packages** |
| sudo apt-get update  sudo apt-get install libusb-dev libpcsclite-dev i2c-tools |
| **3. Download and unzip the source code package 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 |

|  |  |
| --- | --- |
|  |  |
|  |  |

T.Y.B.Sc. (I.T.) 2019

|  |  |  |
| --- | --- | --- |
|  | **5. Write the configuration file for NFC communication**  cd /etc |  |
| sudo mkdir nfc  sudo nano /etc/nfc/libnfc.conf |
| Check the following details of the 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 = "\_PN532\_SPI"  device.connstring = "pn532\_spi:/dev/spidev0.0:500000" #device.name = "\_PN532\_I2c"  #device.connstring = "pn532\_i2c:/dev/i2c-1" |
| **6. Wiring**  Toggle the switch to the **SPI mode**  SEL0 SEL1  L H  Connect the devices: |

T.Y.B.Sc. (I.T.) 2019

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | PN532 | Raspberry |  |  |
| 5V | 5V |
| GND | GND |
| SCK | SCKL |
| MISO | MISO |
| MOSI | MOSI |
| NSS | CE0 |
| **7. Run *ls /dev/spidev0.\** to check whether the SPI is opened or not.**  If yes, it means both the module and the wiring work well. Then type in *nfc-list* to check the NFC module:  */dev/spidev0.0 /dev/spidev0.1*  If two devices are detected, it means the SPI is already opened. Then type in *nfc-list* to check the NFC module:    For Raspberry Pi 3, you may be appear the following error    You should modifiy the *libnfc.conf* | | | |
| sudo nano /etc/nfc/libnfc.conf | | | |
| then modify 500000 to 50000: | | | |
| device.connstring = "pn532\_spi:/dev/spidev0.0:50000" | | | |
| Run *nfc-poll* to scan the RFID tag and you can read information on the card: | | | |

T.Y.B.Sc. (I.T.) 2019

|  |  |  |
| --- | --- | --- |
|  | **Code:**  import subprocess import time  def nfc\_raw():  lines=subprocess.check\_output("/usr/bin/nfc-poll", stderr=open('/dev/null','w'))  return lines  def read\_nfc():  lines=nfc\_raw() return lines  try:  while True:  myLines=read\_nfc() buffer=[]  for line in myLines.splitlines():  line\_content=line.split()  if(not line\_content[0] =='UID'):  pass  else:  buffer.append(line\_content)  str=buffer[0] id\_str=str[2]+str[3]+str[4]+str[5] print (id\_str)  except KeyboardInterrupt: pass |  |