Abstract

This document will demonstrate how to use nRF24L01+ RF module with Arduino™.

Introduction

Nowadays, wireless is widely adopted in different areas or fields such as mobile communications (e.g. WIFI, Mobile Phone), control system, remote car and helicopter. This document is going to demonstrate how a MCU controls a RF module for wireless communication.

In this library, the nRF24L01+ is able to work in Shock Burst Mode with max. 32 Bytes Payload, (Payload means Data), or Dynamic Payload Length Mode. There three well commented sample codes are integrated with Library. Please study the sample codes and refer to the datasheet if need.

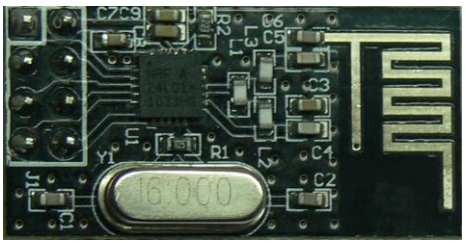
This document is only showed how to use this library as an interface to control nRF24L01+ RF Module. The operations in details will be discussed here. If you concern or are interested in the operations of nRF24L01+, please refer to the datasheet.

Specifications

VCC/VDD[1]	1.9V — 3.6V
RF	2.4GHz
Channel	125
Shock Burst Mode	Yes
Dynamic Payload Length Mode	Yes
Receiver Data Pipe	6
Built-in Auto Acknowledgement	Yes
Communication Interface	SPI

1. If input Signal > 3.6V, the VDD must be from 2.7V to 3.3V.

Pin Outs

Pin outs: 1 GND 2 VDD 3 CE 4 CSN 5 SCK 6 MOSI 7 MISO 8 IRQ	 Left Upper Corner is 1.

Install nRF24L01+ Library

Install the nRF24L01+ library

- copy nRF24L01P folder to <Arduino Folder>\libraries
  Note: If the Arduino software is running when copying library. The Arduino is needed to be restarted after copying the files.

After successful installation of nRF24L01+ library, there are some examples can be found from the menu File -> Examples -> nRF24L01P, as shown in Fig. 1.

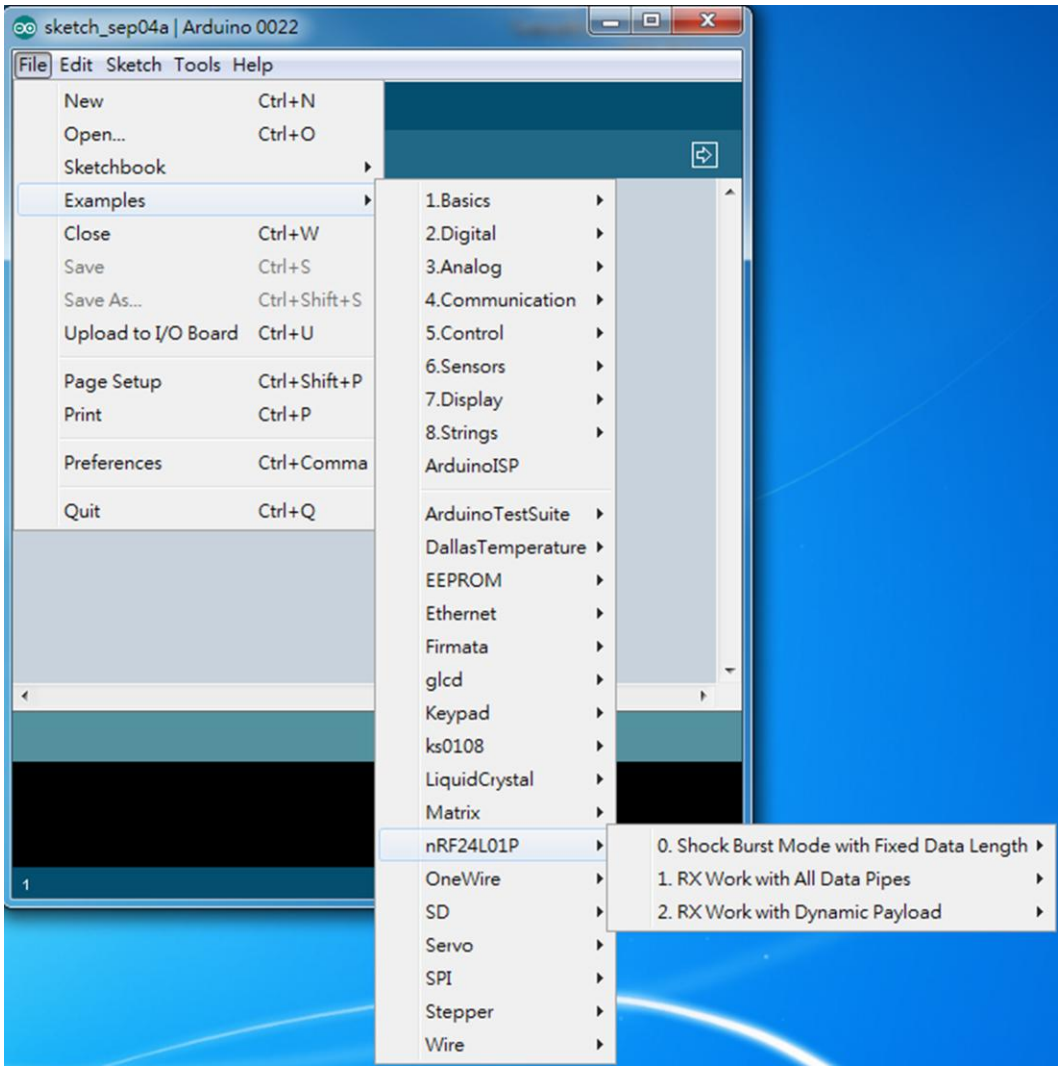


Fig. 1 — Examples from nRF24L01+ Library

Set nRF24L01+ as a Transmitter

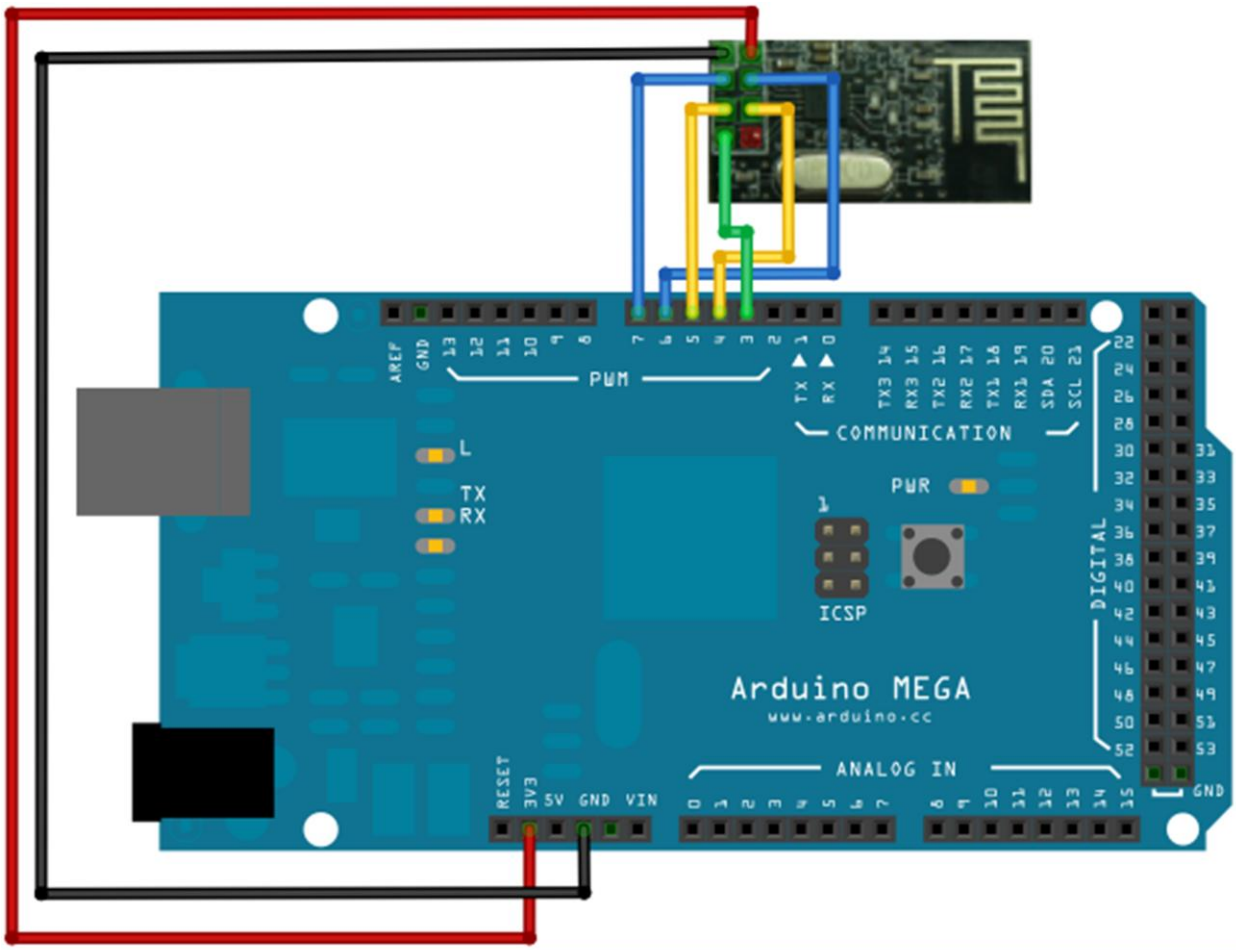


Fig. 2 — Connection of nRF24L01+ RF Module with Arduino

1. Connect the nRF24L01+ RF Module with Arduino as shown in Fig. 2.
2. Upload TX (Transmitter) Example Codes to Arduino, menu File -> Examples -> nRF24L01P -> 0. Shock Burst Mode with Fixed Data Length -> TX

Set nRF24L01+ as a Receiver

1. Connect the nRF24L01+ RF Module with another Arduino, shown in Fig. 2.
2. Upload TX (Transmitter) Example Codes to Arduino, menu File -> Examples -> nRF24L01P -> 0. Shock Burst Mode with Fixed Data Length -> RX

3. Click menu Tools -> Serial Monitor and Set the baud rate to be 115200

If all settings are correct, the Serial Monitor of Arduino Board with a nRF24L01+ RF Module working as a Receiver will display some data, as shown in Fig. 3.

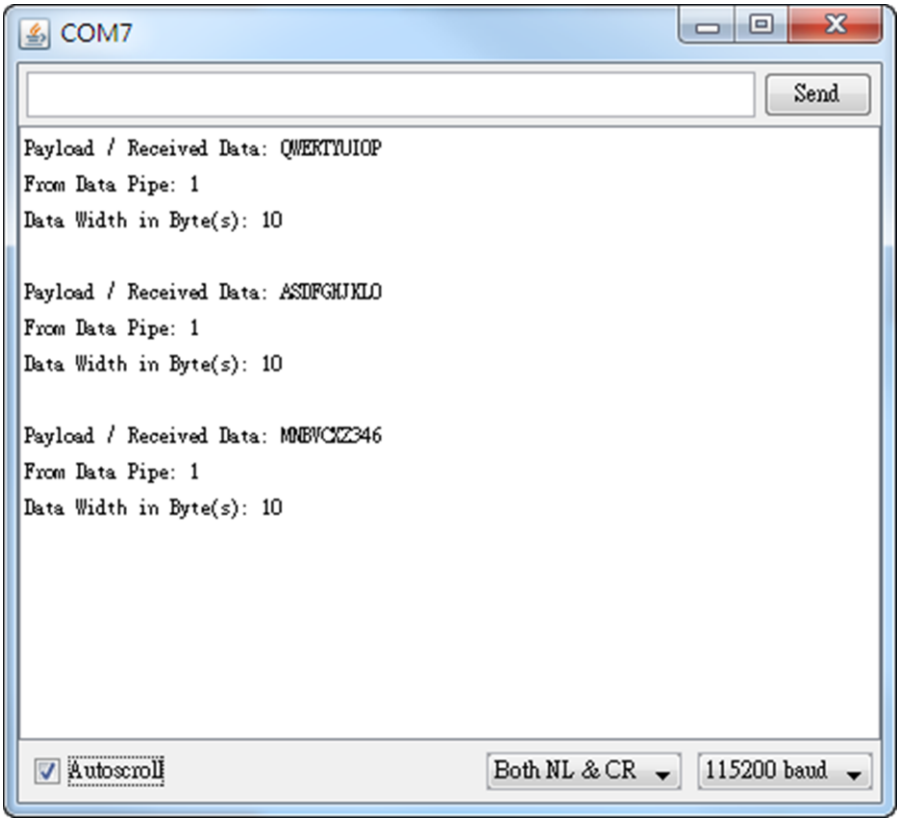


Fig. 3 — Data Received and Display with Serial Monitor

Setting of nRF24L01+

Some parameters or setting should be set before any wireless communication operations.

- Set nRF24L01+ working mode, Shock Burst Mode or Dynamic Payload Mode
- Set the width of address, 3 Bytes to 5 Bytes
- Set the address for Transmitter and Receiver

  Note: Receiver Address of Data Pipe 0 must be same as Transmitter's Address for activating Auto Acknowledgment Function.
- Set operating channel which is from 0 to 125
- Set the Payload width. The payload width of Transmitter and Receiver must be same, otherwise there is malfunction.

Library of nRF24L01+

Function : nRF24L01P(int cePin, int csnPin, int clkPin, int mosiPin, int misoPin)
Parameters : cePin — CE Pin of nRF24L01+ is connected to Arduino Board.
csnPin — CSN Pin of nRF24L01+ is connected to Arduino Board.
clkPin — CLK Pin of nRF24L01+ is connected to Arduino Board.
mosiPin — MOSI Pin of nRF24L01+ is connected to Arduino Board.
misoPin — MISO Pin of nRF24L01+ is connected to Arduino Board.
Description : Constructor of nRF24L01P Class
Set up the connection pins of nRF24L01+ RF module
Return : none
Notes : none

Function : void DefaultShockBurstSetting(void)
Parameters : none
Description : Set nRF24L01+ RF Module works in
Shock Burst Mode,
Switch to Receiver Mode,
Enable Receiver Address Data Pipe 0 and 1
Set RX and TX Address Width 5 Bytes
Set Retransmission Delay 4 ms
Set Retransmission Count 15
Set Operating Channel 23
Set Speed Air Rate 2Mbps
Set RF Power 0dBm
Flush TX and RX Buffer
Clear Status Register
Return : none
Notes : none

Function : void DefaultDynamicPayloadSetting(void)
Parameters : none
Description : Initial Setting as same as Default Shock Burst Setting and Enable Data Pipe 0 and 1 operates in Dynamic Payload Mode.
Return : none
Notes : none

Function : void PowerUP(void)
Parameters : none
Description : Power Up nRF24L01+ Device
Return : none
Notes : none

Function : void PowerDown (void)
Parameters : none
Description : Power Down nRF24L01+ Device
Return : none
Notes : none

Function : void DisableRxDrInterrupt (void)
Parameters : none
Description : Disable nRF24L01+ Interrupt when data are received.
Return : none
Notes : none

Function : void EnableRxDrInterrupt (void)
Parameters : none
Description : Enable nRF24L01+ Interrupt when data are received.
Return : none
Notes : none

Function : void DisableTxDsInterrupt (void)
Parameters : none
Description : Disable nRF24L01+ Interrupt when data are transmitted successfully.
Return : none
Notes : none

Function : void EnableTxDsInterrupt (void)
Parameters : none
Description : Enable nRF24L01+ Interrupt when data are transmitted successfully.
Return : none
Parameters : none
Note : none

Function : void DisableMaxRtInterrupt(void
Description : Disable nRF24L01+ Interrupt when counts of data retransmitted are reaching to maximum counts.
Return : none
Notes : none

Function : void EnableMaxRtInterrupt (void)
Parameters : none
Description : Enable nRF24L01+ Interrupt when counts of data retransmitted are reaching to maximum counts.
Return : none
Notes : none

Function : void DisableCRC (void)
Parameters : none
Description : Disable nRF24L01+'s internal CRC function
Return : none
Notes : none

Function : void EnableCRC(unsigned char scheme)
Parameters : scheme — 0 for 1 Byte CRC
1 for 2 Bytes CRC
Description : Enable nRF24L01+'s internal CRC function
Return : none
Notes : none

Function : void EnableAutoAck(unsigned char pipe)
Parameters : pipe — Auto Acknowledgement of Data Pipe to be enable
Which is from 0 to 5
Description : Enable Data Pipe 0 to 5 Auto Acknowledgement Function
Return : none
Notes : none

Function : void DisableAutoAck (unsigned char pipe)
Parameters : pipe — Auto Acknowledgement of Data Pipe to be disable
Which is from 0 to 5
Description : Diable Data Pipe 0 to 5 Auto Acknowledgement Function
Return : none
Notes : none

Function : void DisableAutoAck (unsigned char pipe)
Parameters : pipe — Auto Acknowledgement of Data Pipe to be disable
Which is from 0 to 5
Description : Diable Data Pipe 0 to 5 Auto Acknowledgement Function
Return : none
Notes : none

Function : void DisableDataPipe (unsigned char pipe)
Parameters : pipe — Data Pipe to be disable, which is from 0 to 5
Description : Diable Data Pipe 0 to 5
Return : none
Notes : none

Function : void EnableDataPipe (unsigned char pipe)
Parameters : pipe — Data Pipe to be enable, which is from 0 to 5
Description : Enable Data Pipe 0 to 5
Return : none
Notes : none

Function : void DisableRXAddr (unsigned char pipe)
Parameters : pipe — Data Pipe to be disable, which is from 0 to 5
Description : Diable Data Pipe 0 to 5
Return : none
Notes : note

Function : void EnableRXAddr (unsigned char pipe)
Parameters : pipe — Data Pipe to be enable, which is from 0 to 5
Description : Enable Data Pipe 0 to 5
Return : none
Notes : none

Function	:	void setAddressWidth(unsigned char AddrWidth)
Parameters	:	AddrWidth — Set Address Width which is from 3 to 5 Bytes
Description	:	Set Address Width
Return	:	none
Notes	:	Default is 5 Bytes

Function	:	unsigned char getAddressWidth(void)
Parameters	:	none
Description	:	Get Address Width
Return	:	Address Width, which is from 3 to 5 Bytes
Notes	:	none

Function	:	void setAutoRD(unsigned char AutoRD)
Parameters	:	AutoRD — Auto Re-transmission Delay, which is from 0 to 15
Description	:	Set Auto Re-transmission Delay Auto Re-transmission Delay: (AutoRD + 1) * 250us
Return	:	none
Notes	:	Please refer to data sheet for the details

Function	:	void setAutoReCount(unsigned char AutoCount)
Parameters	:	AutoCount — Auto Re-transmission Count, which is from 0 to 15
Description	:	Set Auto Re-transmission Count
Return	:	none
Notes	:	Please refer to data sheet for the details If the Auto Retransmission Count is reached and the data is not transmitted out successfully, an interrupt will be occurred. This can be checked by MaxRt() function.

Function	:	void setSpeedAirRate(unsigned char Speed)
Parameters	:	Speed — 0 for 1Mbps, 1 for 2 Mbps, 2 for 250Kbps Or use SpeedAirRate_1MBPS SpeedAirRate_2MBPS SpeedAirRate_250KBPS
Description	:	Set Air Speed Data Rate
Return	:	none
Notes	:	none

Function	:	void setRFPower(unsigned char PWR)
Parameters	:	PWR — 0 for -18dBm 1 for -12dBm 2 for -6dBm 3 for 0dBm
Description	:	Set RF Power
Return	:	none
Notes	:	none

Function	:	void WriteRegByte(unsigned char reg, unsigned char value)
Parameters	:	reg — Register to be written value — Value to be written to register
Description	:	Write value to register
Return	:	none
Notes	:	Refer to Datasheet for details of registers and parameters

---

Function	:	unsigned char ReadRegByte(unsigned char reg)
Parameters	:	reg — Register to be read
Description	:	Read a byte from a register
Return	:	Value of register to be read
Notes	:	Refer to Datasheet for details of registers and parameters

---

Function	:	void WriteRegBuf(unsigned char reg, unsigned char *buf, char bytes)
Parameters	:	reg — Register to be written *buf — Char Array to be written to a register bytes — Bytes of *buf
Description	:	Write a char array to a register
Return	:	none
Notes	:	Refer to Datasheet for details of registers and parameters

---

Function	:	void ReadRegBuf(unsigned char reg, unsigned char *buf, char bytes)
Parameters	:	reg — Register to be read *buf — Char Array to be written from a register bytes — Bytes of *buf
Description	:	Read a char array from a register to *buf
Return	:	none
Notes	:	Refer to Datasheet for details of registers and parameters

Function	:	void setMode(char mode)
Parameters	:	mode — TX for transmitter RX for receiver
Description	:	Switch the nRF24L01+ to Transmitter or Receiver Mode
Return	:	none
Notes	:	none

Function	:	void Transmit(unsigned char *pBuf, unsigned char plWidth) void Transmit(char *pBuf, unsigned char plWidth)
Parameters	:	*pBuf — Bytes to be sent out plWidth — Number of bytes in pBuf will be sent
Description	:	Transmit the data
Return	:	none
Notes	:	none

Function	:	void WriteTXPayload(unsigned char *pBuf, unsigned char plWidth)
Parameters	:	*pBuf — Bytes to be written to TX Payload Register plWidth — Number of bytes in pBuf will be written
Description	:	Write Payload to the TX Payload Width Ack. Register
Return	:	none
Notes	:	none

Function	:	void setChannel(unsigned char RF_Channel)
Parameters	:	RF_Channel — Set RF Channel which is from 0 to 125
Description	:	Set the RF operating channel
Return	:	none
Notes	:	none

Function	:	unsigned char getChannel(void)
Parameters	:	none
Description	:	Get the RF operating channel
Return	:	RF operating Channel which is from 0 to 125
Notes	:	none

Function	:	unsigned char getStatus (void)
Parameters	:	none
Description	:	Get the value of Status Register
Return	:	The Value of Status Register
Notes	:	Refer to Datasheet for details

Function	:	unsigned char getRXPayloadWidth(unsigned char pipe)
Parameters	:	pipe — Data Pipe which is from 0 to 5
Description	:	Get the RX payload width of a Data Pipe
Return	:	Payload Width of Data Pipe, which is from 1 to 32
Notes	:	none

Function	:	void setRXPayloadWidth(unsigned char pipe, unsigned char plWidth)
Parameters	:	pipe — Data Pipe which is from 0 to 5
		plWidth — Set the Payload Width
Description	:	Set the RX Payload width of a Data Pipe
Return	:	none
Notes	:	none

Function	:	unsigned char getCurrentPipeNum(void)
Parameters	:	none
Description	:	Get the Current Pipe No., that Data is received
Return	:	Data Pipe received data, which is from 0 to 5
Notes	:	none

Function	:	void getRXPload(unsigned char *pBuf)
		void getData(unsigned char *pBuf)
		void getData(char *pBuf)
Parameters	:	*pBuf — Get Received Data
Description	:	Get Received Data
Return	:	none
Notes	:	none

Function	:	void getRXPload(unsigned char *pBuf, unsigned char *pipe, unsigned char *plWidth) void getData(unsigned char *pBuf, unsigned char *pipe, unsigned char *plWidth) void getData(char *pBuf, unsigned char *pipe, unsigned char *plWidth)
Parameters	:	*pBuf — Get Received Data *pipe — Get Data Pipe, the data are came from which data pipe *plWidth — Get Payload Width
Description	:	Get Received Data
Return	:	none
Notes	:	none

Function	:	unsigned char getFIFO(void)
Parameters	:	none
Description	:	Get the value of FIFO register status
Return	:	the value of FIFO register status
Notes	:	Refer to Datasheet for details

Function	:	void setTXAddress(unsigned char *pBuf, unsigned char addrWidth)
Parameters	:	*pBuf — Char Array to be written to TX Address Register addrWidth — Number of Bytes of *pBuf to be written
Description	:	Set TX Address
Return	:	none
Notes	:	none

Function	:	void setRXAddress(unsigned char *pBuf, unsigned char addrWidth, unsigned char pipe) void setRXAddress(unsigned char pBuf, unsigned char addrWidth, unsigned char pipe)
Parameters	:	*pBuf — Char Array to be written to RX Address Register addrWidth — Number of Bytes of *pBuf to be written pBuf — Char to be written RX Address Register, Data pipe 2 to 5 pipe — Data Pipe to be set
Description	:	Set RX Address
Return	:	none
Notes	:	none

Function : void getTXAddress(unsigned char *pBuf)
Parameters : *pBuf — Char Array to be written from a TX Address Register
Description : Get TX Address
Return : none
Notes : none

Function : void getRXAddress(unsigned char *pBuf, unsigned char pipe)
Parameters : *pBuf — Char Array to be written from a RX Address Register
pipe — Data Pipe to be select
Description : Get RX Address
Return : none
Notes : none

Function : void FlushTX(void);
Parameters : none
Description : Flush TX FIFO Buffer
Return : none
Notes : none

Function : void FlushRX (void);
Parameters : none
Description : Flush RX FIFO Buffer
Return : none
Notes : none

Function : void clearRxDr (void);
Parameters : none
Description : Clear Data Received Interrupt Flag
Return : none
Notes : none

Function : void clearMaxRt (void);
Parameters : none
Description : Clear Max. Re-transmission Interrupt Flag
Return : none
Notes : none

Function	:	void clearStatusReg(void);
Parameters	:	none
Description	:	Clear Status Register
Return	:	none
Notes	:	none

Function	:	unsigned char RxDR(void) unsigned char Available(void)
Parameters	:	none
Description	:	Check whether Data is received
Return	:	1 for Data Received is ready. 0 for NO Data is received.
Notes	:	none

Function	:	unsigned char MaxRt (void)
Parameters	:	none
Description	:	Check whether Max. Retransmission Interrupt is occurred
Return	:	1 for Maximum number of TX retransmits interrupt. 0 for otherwise
Notes	:	If 1 is returned, it is meant that the transmission is fialed. The data cannot be sent to target device.

Function	:	unsigned char TxDs(void) unsigned char isTransmitted(void)
Parameters	:	none
Description	:	Check whether Data is transmitted successfully.
Return	:	1 for Data is transmitted successfully. 0 for otherwise
Notes	:	none

Function	:	unsigned char PacketLostCounter(void)
Parameters	:	none
Description	:	How many packets are lost during transmission.
Return	:	How many packets are lost
Notes	:	The maximum number of packet lost is 15. Overflow will be occurred when it is greater than 15

Function	:	void ClearPacketLostCounter(void)
Parameters	:	none
Description	:	Clear Packet Lost Counter Register
Return	:	none
Notes	:	none

Function	:	unsigned char RetransmittedPackets(void)
Parameters	:	none
Description	:	Get how many Packets are retransmitted.
Return	:	How Many Packets are retransmitted.
Notes	:	none

Function	:	bool isTxFIFOFull(void)
Parameters	:	none
Description	:	Check whether TX FIFO buffer is full.
Return	:	true for TX FIFO Full false for Available locations in TX FIFO
Notes	:	none

Function	:	bool isTxFIFOEmpty (void)
Parameters	:	none
Description	:	Check whether TX FIFO buffer is empty.
Return	:	true for TX FIFO is empty false for TX FIFO is not empty
Notes	:	none

Function	:	bool isRxFIFOFull (void)
Parameters	:	none
Description	:	Check whether RX FIFO buffer is full.
Return	:	true for RX FIFO Full false for Available locations in RX FIFO
Notes	:	none

Function	:	bool isRxFIFOEmpty(void)
Parameters	:	none
Description	:	Check whether RX FIFO buffer is empty.
Return	:	true for RX FIFO is empty false for RX FIFO is not empty
Notes	:	none

Function	:	void EnableDynamicPayloadLength(void)
Parameters	:	none
Description	:	Enable Dynamic Payload Length Function. Let nRF24L01+ work in Dynamic Payload Mode.
Return	:	none
Notes	:	none

Function	:	void DisableDynamicPayloadLength(void)
Parameters	:	none
Description	:	Disable Dynamic Payload Length Function.
Return	:	none
Notes	:	none

Function	:	void DataPipeEnableDynamicPldLen(unsigned char pipe)
Parameters	:	pipe — Data Pipe to be enable in Dynamic Payload Length
Description	:	Set which Data Pipe is enabled in Dynamic Payload Length Mode
Return	:	none
Notes	:	none

Function	:	void DataPipeDisableDynamicPldLen (unsigned char pipe)
Parameters	:	pipe — Data Pipe to be disable in Dynamic Payload Length
Description	:	Set which Data Pipe is disabled in Dynamic Payload Length Mode
Return	:	none
Notes	:	none

Function	:	bool isDynamicPDEnabled(void)
Parameters	:	none
Description	:	Set which Data Pipe is disabled in Dynamic Payload Length Mode
Return	:	true for Enabled Dynamic Payload Length Mode false for not
Notes	:	none