**Code for lpc2148 receiver part: -**

#include "nrf24l01.h"

#include "systemInit.h"

#include "SPI.h"

#include "uart.h"

#include "delays.h"

#include "gpio.h"

#define \_CH 1 // Channel 0..125

#define \_Address\_Width 5 // 3..5

#define \_Buffer\_Size 5 // 1..32

void print\_AllRegisters();

void NRF24L01\_Receive(char Buf[\_Buffer\_Size]) {

NRF24L01\_CE\_HIGH;

DelayUS(130);

while ((NRF24L01\_Get\_Status() & \_RX\_DR) != \_RX\_DR);

NRF24L01\_CE\_LOW;

NRF24L01\_Read\_RX\_Buf(Buf, \_Buffer\_Size);

NRF24L01\_Clear\_Interrupts();

}

void NRF24L01\_Send(char Buf[\_Buffer\_Size]) {

NRF24L01\_Write\_TX\_Buf(Buf, \_Buffer\_Size);

NRF24L01\_RF\_TX();

while ((NRF24L01\_Get\_Status() & \_TX\_DS) != \_TX\_DS);

NRF24L01\_Clear\_Interrupts();

}

void main(void)

{

char Buf[\_Buffer\_Size]; //use this for receiving

char Address[\_Address\_Width] = { 0x11, 0x22, 0x33, 0x44, 0x55 };

SystemInit();// initialize the system with 60Mhz clock

UART0\_Init\_115200();

SPI1\_Init();

NRF24L01\_CE\_OUT;

NRF24L01\_CSN\_OUT;

NRF24L01\_CE\_LOW;

NRF24L01\_CSN\_HIGH;

NRF24L01\_Init(\_RX\_MODE, \_CH, \_1Mbps, Address, \_Address\_Width, \_Buffer\_Size);

print\_AllRegisters();

GPIO\_PinDirection(LED\_PIN, OUTPUT);

GPIO\_PinWrite(LED\_PIN,HIGH); // LED OFF INITIALLY HIGH = LED OFF, In the Board Schematic it is an ACTIVE LOW Design ckt

DelayMS(1000);

while (1) {

/\*

UART\_TxString(0,"\nsending started...");

NRF24L01\_Send(Buf);

UART\_TxString(0,"\nsent...");

DelayMS(500);

\*/

//UART\_TxString(0,"\nWaiting for data...");

NRF24L01\_Receive(Buf);

//UART\_TxString(0,"\nData Received = ");

//UART\_TxString(0,Buf);

if(Buf[0] == 'R')

{

GPIO\_PinWrite(LED\_PIN,LOW);//LED ON

}

else

{

GPIO\_PinWrite(LED\_PIN,HIGH); //LED OFF

}

Buf[0] = ' ';//Clear buffer

}

}

void print\_AllRegisters()

{

unsigned char data1;

unsigned char data2[5];

UART\_TxString(0,"\n========================== Configurations are ======================== ");

NRF24L01\_ReadRegBuf(CONFIG, &data1, 1);

UART\_TxString(0,"\nCONFIG REG = ");

UART\_Printf(0,"%x", data1);

NRF24L01\_ReadRegBuf(EN\_AA, &data1, 1);

UART\_TxString(0,"\nEN\_AA REG = ");

UART\_Printf(0,"%x", data1);

NRF24L01\_ReadRegBuf(EN\_RXADDR, &data1, 1);

UART\_TxString(0,"\nEN\_RXADDR REG = ");

UART\_Printf(0,"%x", data1);

NRF24L01\_ReadRegBuf(SETUP\_AW, &data1, 1);

UART\_TxString(0,"\nSETUP\_AW REG = ");

UART\_Printf(0,"%x", data1);

NRF24L01\_ReadRegBuf(SETUP\_RETR, &data1, 1);

UART\_TxString(0,"\nnrf24l01\_SETUP\_RETR REG = ");

UART\_Printf(0,"%x", data1);

NRF24L01\_ReadRegBuf(RF\_CH, &data1, 1);

UART\_TxString(0,"\nRF\_CH REG = ");

UART\_Printf(0,"%x", data1);

NRF24L01\_ReadRegBuf(RF\_SETUP, &data1, 1);

UART\_TxString(0,"\nRF\_SETUP REG = ");

UART\_Printf(0,"%x", data1);

NRF24L01\_ReadRegBuf(STATUS, &data1, 1);

UART\_TxString(0,"\nSTATUS REG = ");

UART\_Printf(0,"%x", data1);

NRF24L01\_ReadRegBuf(OBSERVE\_TX, &data1, 1);

UART\_TxString(0,"\nOBSERVE\_TX REG = ");

UART\_Printf(0,"%x", data1);

NRF24L01\_ReadRegBuf(RX\_ADDR\_P0, data2, 5);

UART\_TxString(0,"\nRX P0 Address reg = ");

UART\_Printf(0,"%x", data2[0]);

UART\_Printf(0,"%x", data2[1]);

UART\_Printf(0,"%x", data2[2]);

UART\_Printf(0,"%x", data2[3]);

UART\_Printf(0,"%x", data2[4]);

NRF24L01\_ReadRegBuf(RX\_ADDR\_P1, data2, 5);

UART\_TxString(0,"\nRX P1 Address reg = ");

UART\_Printf(0,"%x", data2[0]);

UART\_Printf(0,"%x", data2[1]);

UART\_Printf(0,"%x", data2[2]);

UART\_Printf(0,"%x", data2[3]);

UART\_Printf(0,"%x", data2[4]);

NRF24L01\_ReadRegBuf(RX\_ADDR\_P2, data2, 5);

UART\_TxString(0,"\nRX P2 Address reg = ");

UART\_Printf(0,"%x", data2[0]);

UART\_Printf(0,"%x", data2[1]);

UART\_Printf(0,"%x", data2[2]);

UART\_Printf(0,"%x", data2[3]);

UART\_Printf(0,"%x", data2[4]);

NRF24L01\_ReadRegBuf(RX\_ADDR\_P3, data2, 5);

UART\_TxString(0,"\nRX P3 Address reg = ");

UART\_Printf(0,"%x", data2[0]);

UART\_Printf(0,"%x", data2[1]);

UART\_Printf(0,"%x", data2[2]);

UART\_Printf(0,"%x", data2[3]);

UART\_Printf(0,"%x", data2[4]);

NRF24L01\_ReadRegBuf(RX\_ADDR\_P4, data2, 5);

UART\_TxString(0,"\nRX P4 Address reg = ");

UART\_Printf(0,"%x", data2[0]);

UART\_Printf(0,"%x", data2[1]);

UART\_Printf(0,"%x", data2[2]);

UART\_Printf(0,"%x", data2[3]);

UART\_Printf(0,"%x", data2[4]);

NRF24L01\_ReadRegBuf(RX\_ADDR\_P5, data2, 5);

UART\_TxString(0,"\nRX P5 Address reg = ");

UART\_Printf(0,"%x", data2[0]);

UART\_Printf(0,"%x", data2[1]);

UART\_Printf(0,"%x", data2[2]);

UART\_Printf(0,"%x", data2[3]);

UART\_Printf(0,"%x", data2[4]);

NRF24L01\_ReadRegBuf(TX\_ADDR, data2, 5);

UART\_TxString(0,"\nTX Address reg = ");

UART\_Printf(0,"%x", data2[0]);

UART\_Printf(0,"%x", data2[1]);

UART\_Printf(0,"%x", data2[2]);

UART\_Printf(0,"%x", data2[3]);

UART\_Printf(0,"%x", data2[4]);

}