/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Programs for Arduino - Copyright of the author Stuart Robinson - 21/09/21

This program is supplied as is, it is up to the user of the program to decide if the program is

suitable for the intended purpose and free from errors.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Program Operation - This is a minimum setup LoRa test transmitter. A packet containing the ASCII text

"Hello World 1234567890" is sent using the frequency and LoRa settings specified in the LT.setupLoRa()

command. The pins to access the lora device need to be defined at the top of the program also.

This program does not need the DIO1 pin on the LoRa device connected.

The details of the packet sent and any errors are shown on the Arduino IDE Serial Monitor, together with

the transmit power used and the packet length. The matching receiver program, '4\_LoRa\_Receiver' can be used

to check the packets are being sent correctly, the frequency and LoRa settings (in the LT.setupLoRa()

commands) must be the same for the transmitter and receiver programs. Sample Serial Monitor output;

10dBm Packet> Hello World 1234567890\* BytesSent,23 PacketsSent,6

This is a version of example 3\_LoRa\_Transmitter.ino that does not require the use of the DIO1 pin to

check for transmit done. In addition no NRESET pin is needed either, so its a program for use with a

minimum pin count Arduino. Leave the DIO1 and NRESET pins on the LoRa device not connected.

For an example of a more detailed configuration for a transmitter, see program 103\_LoRa\_Transmitter.

Serial monitor baud rate is set at 9600

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <SPI.h> //the lora device is SPI based so load the SPI library

#include <SX128XLT.h> //include the appropriate library

SX128XLT LT; //create a library class instance called LT

#define NSS PA4 //select pin on LoRa device

#define NRESET PB0 //reset pin on LoRa device

#define RFBUSY PB1 //busy pin on LoRa device

#define LORA\_DEVICE DEVICE\_SX1280 //we need to define the device we are using

#define TXpower 0 //LoRa transmit power in dBm

uint8\_t TXPacketL;

uint32\_t TXPacketCount;

uint8\_t buff[] = "Hello World234567890Hello World234567890Hello World234567890Hello World234567890Hello World234567890Hello World234567890Hello World234567890"; //the message to send

int ledState = LOW; // ledState used to set the LED

void loop()

{

Serial.print(TXpower); //print the transmit power defined

Serial.print(F("dBm "));

Serial.print(F("Packet> "));

Serial.flush();

TXPacketL = sizeof(buff); //set TXPacketL to length of array

buff[TXPacketL - 1] = '\*'; //replace null character at buffer end so its visible on receiver

LT.printASCIIPacket(buff, TXPacketL); //print the buffer (the sent packet) as ASCII

if (LT.transmitIRQ(buff, TXPacketL, 10000, TXpower, WAIT\_TX)) //will return packet length sent if OK, otherwise 0 if transmit error

{

TXPacketCount++;

packet\_is\_OK();

}

else

{

packet\_is\_Error(); //transmit packet returned 0, there was an error

}

Serial.println();

delay(1000); //have a delay between packets

}

void packet\_is\_OK()

{

// if the LED is off turn it on and vice-versa:

if (ledState == LOW) {

ledState = HIGH;

} else {

ledState = LOW;

}

// set the LED with the ledState of the variable:

digitalWrite(PC13, ledState);

//if here packet has been sent OK

Serial.print(F(" BytesSent,"));

Serial.print(TXPacketL); //print transmitted packet length

Serial.print(F(" PacketsSent,"));

Serial.print(TXPacketCount); //print total of packets sent OK

}

void packet\_is\_Error()

{

//if here there was an error transmitting packet

uint16\_t IRQStatus;

IRQStatus = LT.readIrqStatus(); //read the the interrupt register

Serial.print(F(" SendError,"));

Serial.print(F("Length,"));

Serial.print(TXPacketL); //print transmitted packet length

Serial.print(F(",IRQreg,"));

Serial.print(IRQStatus, HEX); //print IRQ status

LT.printIrqStatus(); //prints the text of which IRQs set

}

void setup()

{

delay(5000);

pinMode(PC13, OUTPUT);

Serial.begin(9600);

Serial.println();

Serial.println(F("3\_LoRa\_TransmitterIRQ Starting"));

SPI.begin();

if (LT.begin(NSS, NRESET, RFBUSY, LORA\_DEVICE))

{

Serial.println(F("LoRa Device found"));

delay(1000);

}

else

{

Serial.println(F("No LoRa device responding"));

while (1);

}

LT.setupLoRa(2424500000, 0, LORA\_SF11, LORA\_BW\_0200, LORA\_CR\_4\_5); //configure frequency and LoRa settings

Serial.print(F("Transmitter ready"));

Serial.println();

}