/\*

\* Author: JP Meijers

\* Date: 2016-02-07

\* Previous filename: TTN-Mapper-TTNEnschede-V1

\*

\* This program is meant to be used with an Arduino UNO or NANO, conencted to an RNxx3 radio module.

\* It will most likely also work on other compatible Arduino or Arduino compatible boards, like The Things Uno, but might need some slight modifications.

\*

\* Transmit a one byte packet via TTN. This happens as fast as possible, while still keeping to

\* the 1% duty cycle rules enforced by the RN2483's built in LoRaWAN stack. Even though this is

\* allowed by the radio regulations of the 868MHz band, the fair use policy of TTN may prohibit this.

\*

\* CHECK THE RULES BEFORE USING THIS PROGRAM!

\*

\* CHANGE ADDRESS!

\* Change the device address, network (session) key, and app (session) key to the values

\* that are registered via the TTN dashboard.

\* The appropriate line is "myLora.initABP(XXX);" or "myLora.initOTAA(XXX);"

\* When using ABP, it is advised to enable "relax frame count".

\*

\* Connect the RN2xx3 as follows:

\* RN2xx3 -- Arduino

\* Uart TX -- 10

\* Uart RX -- 11

\* Reset -- 12

\* Vcc -- 3.3V

\* Gnd -- Gnd

\*

\* If you use an Arduino with a free hardware serial port, you can replace

\* the line "rn2xx3 myLora(mySerial);"

\* with "rn2xx3 myLora(SerialX);"

\* where the parameter is the serial port the RN2xx3 is connected to.

\* Remember that the serial port should be initialised before calling initTTN().

\* For best performance the serial port should be set to 57600 baud, which is impossible with a software serial port.

\* If you use 57600 baud, you can remove the line "myLora.autobaud();".

\*

\*/

#include <rn2xx3.h>

#include <SoftwareSerial.h>

SoftwareSerial mySerial(10, 11); // RX, TX

//create an instance of the rn2xx3 library,

//giving the software serial as port to use

rn2xx3 myLora(mySerial);

// the setup routine runs once when you press reset:

void setup()

{

//output LED pin

pinMode(13, OUTPUT);

led\_on();

// Open serial communications and wait for port to open:

Serial.begin(57600); //serial port to computer

mySerial.begin(9600); //serial port to radio

Serial.println("Startup");

initialize\_radio();

//transmit a startup message

myLora.tx("TTN Mapper on TTN Enschede node");

led\_off();

delay(2000);

}

void initialize\_radio()

{

//reset rn2483

pinMode(12, OUTPUT);

digitalWrite(12, LOW);

delay(500);

digitalWrite(12, HIGH);

delay(100); //wait for the RN2xx3's startup message

mySerial.flush();

//Autobaud the rn2483 module to 9600. The default would otherwise be 57600.

myLora.autobaud();

//check communication with radio

String hweui = myLora.hweui();

while(hweui.length() != 16)

{

Serial.println("Communication with RN2xx3 unsuccessful. Power cycle the board.");

Serial.println(hweui);

delay(10000);

hweui = myLora.hweui();

}

//print out the HWEUI so that we can register it via ttnctl

Serial.println("When using OTAA, register this DevEUI: ");

Serial.println(myLora.hweui());

Serial.println("RN2xx3 firmware version:");

Serial.println(myLora.sysver());

//configure your keys and join the network

Serial.println("Trying to join TTN");

bool join\_result = false;

/\*

\* ABP: initABP(String addr, String AppSKey, String NwkSKey);

\* Paste the example code from the TTN console here:

\*/

const char \*devAddr = "26011FE4";

const char \*nwkSKey = "1691F02C77FC5926C3012B6E93615349";

const char \*appSKey = "F25A5F68BF92B3E53E455C18A8864F7E";

join\_result = myLora.initABP(devAddr, appSKey, nwkSKey);

/\*

\* OTAA: initOTAA(String AppEUI, String AppKey);

\* If you are using OTAA, paste the example code from the TTN console here:

\*/

/\*const char \*appEui = "70B3D57ED0025CA9";

const char \*appKey = "CF1976248EAF647D4B59AD42D81E0BE7";

join\_result = myLora.initOTAA(appEui, appKey);\*/

while(!join\_result)

{

Serial.println("Unable to join. Are your keys correct, and do you have TTN coverage?");

delay(60000); //delay a minute before retry

join\_result = myLora.init();

}

Serial.println("Successfully joined TTN");

}

// the loop routine runs over and over again forever:

void loop()

{

led\_on();

int a = random(100);

Serial.print("TXing ");

Serial.println(a);

myLora.tx(String(a)); //one byte, blocking function

//Serial.print("SNR : ");

//Serial.println(myLora.getSNR());

//Serial.print("Message: ");

//String str = myLora.getRx();

//String decodedStr = myLora.base16decode(str);

//Serial.println(decodedStr);

led\_off();

delay(200);

}

void led\_on()

{

digitalWrite(13, 1);

}

void led\_off()

{

digitalWrite(13, 0);

}