**Sender(Verici) Görevindeki Cihaz:**

#include <SPI.h>

#include <LoRa.h>

#include "SSD1306.h"

#include<Arduino.h>

#include <TinyGPS++.h>

#include <HardwareSerial.h>

SSD1306 display(0x3c, 4, 15);

#define SS 18

#define RST 14

#define DI0 26

#define BAND 433E6 //915E6

#define gpsTx 16

#define gpsRx 17

#define extInt 22

int counter = 0;

const int sleepTimes = 52;

static const int RXPin = 4, TXPin = 3;

static const uint32\_t GPSBaud = 9600;

String imei\_id=" XXXXXXXXXXXXXXXXX ";

String baslangic={"@L0XX"};

String bitis={"X"};

int data\_durumu;

String status1={" XXXXXX XX"};

String distance={" XXXXXX "};

String alarm={" XXX"};

TinyGPSPlus gps;

HardwareSerial gpsSerial(1);

int ignitionState = HIGH;

void setup()

{

pinMode(extInt, INPUT);

Serial.begin(115200);

while (!Serial);

gpsSerial.begin(9600, SERIAL\_8N1, gpsTx, gpsRx);

attachInterrupt(digitalPinToInterrupt(extInt), ignitionCallback, CHANGE);

ignitionState = digitalRead(extInt);

SPI.begin(5,19,27,18);

LoRa.setPins(SS,RST,DI0);

Serial.println("LoRa Sender");

if (!LoRa.begin(BAND))

{

Serial.println("Starting LoRa failed!");

while (1);

}

LoRa.setSpreadingFactor(12);

LoRa.setSignalBandwidth(62.5E3);

LoRa.crc();

Serial.println("LoRa Initial OK!");

Serial.print("Sending packet: ");

Serial.println(counter);

Serial.println ("@C;XXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX");

LoRa.beginPacket();

LoRa.println ("@C;XXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX");

LoRa.endPacket();

delay(2000);

}

void loop()

{

int deg;

int deg1;

float arcMinutes;

float arcMinutes1;

int arcSeconds;

int arcSeconds1;

float minutesRemainder;

float minutesRemainder1;

int hiz;

int course1;

char data[512]={deg,arcMinutes,arcSeconds,minutesRemainder,deg1,arcMinutes1,arcSeconds1

,minutesRemainder1,hiz,course1};

while(true)

{

Serial.print("Sending packet: ");

Serial.println(counter);

Serial.print("ignitionState: ");

Serial.println(ignitionState);

if(gps.location.lat()==0.0 && gps.location.lng()==0.0)

{

data\_durumu=8;

}

else

{

data\_durumu=0;

}

deg=gps.location.lat();

minutesRemainder = abs(gps.location.lat() - deg) \* 60;

arcMinutes = minutesRemainder;

arcSeconds = (minutesRemainder - arcMinutes) \* 60;

deg1=gps.location.lng();

minutesRemainder1 = abs(gps.location.lng() - deg1) \* 60;

arcMinutes1 = minutesRemainder1;

arcSeconds1 = (minutesRemainder1 - arcMinutes1) \* 60;

hiz=((gps.speed.kmph()/1.852)\*100);

course1=gps.course.deg();

if(ignitionState==LOW)

{

status1=" XXXXXXXX ";

}

else

{

status1=" XXXXXXXX ";

}

sprintf(data,"%s%s%06d%02d%02d%02d%d%d%2.0f%03d%2.0f%s%04X%s%03X%s%s",baslangic.c\_str(),

imei\_id.c\_str(),gps.date.value(),gps.time.hour(),gps.time.minute(),gps.time.second(),data\_durumu,

deg,arcMinutes\*10000,deg1,arcMinutes1\*10000,status1.c\_str(),

hiz,distance.c\_str(),course1,alarm.c\_str(),bitis.c\_str());

Serial.println(data);

LoRa.beginPacket();

LoRa.print(data);

LoRa.endPacket();

smartDelay(1000);

if (millis() > 5000 && gps.charsProcessed() < 10)

Serial.println(F("No GPS data received: check wiring"));

}

}

void ignitionCallback() {

ignitionState = digitalRead(extInt);

Serial.print("ignitionState: ");

Serial.println(ignitionState);

}

static void smartDelay(unsigned long ms)

{

unsigned long start = millis();

do

{

while (gpsSerial.available())

gps.encode(gpsSerial.read());

} while (millis() - start < ms);

}

static void printFloat(float val, bool valid, int len, int prec)

{

if (!valid)

{

while (len-- > 1)

Serial.print('\*');

Serial.print(' ');

}

else

{

Serial.print(val, prec);

int vi = abs((int)val);

int flen = prec + (val < 0.0 ? 2 : 1); // . and -

flen += vi >= 1000 ? 4 : vi >= 100 ? 3 : vi >= 10 ? 2 : 1;

for (int i=flen; i<len; ++i)

Serial.print(' ');

}

smartDelay(0);

}

static void printInt(unsigned long val, bool valid, int len)

{

char sz[32] = "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

if (valid)

sprintf(sz, "%ld", val);

sz[len] = 0;

for (int i=strlen(sz); i<len; ++i)

sz[i] = ' ';

if (len > 0)

sz[len-1] = ' ';

Serial.print(sz);

smartDelay(0);

}

static void printDateTime(TinyGPSDate &d, TinyGPSTime &t)

{

if (!d.isValid())

{

Serial.print(F("\*\*\*\*\*\*\*\*\*\* "));

}

else

{

char sz[32];

sprintf(sz, "%02d/%02d/%02d ", d.month(), d.day(), d.year());

Serial.print(sz);

}

if (!t.isValid())

{

Serial.print(F("\*\*\*\*\*\*\*\* "));

}

else

{

char sz[32];

sprintf(sz, "%02d:%02d:%02d ", t.hour(), t.minute(), t.second());

Serial.print(sz);

}

printInt(d.age(), d.isValid(), 5);

smartDelay(0);

}

static void printStr(const char \*str, int len)

{

int slen = strlen(str);

for (int i=0; i<len; ++i)

Serial.print(i<slen ? str[i] : ' ');

smartDelay(0);

}