**SIM808**

**Jaakko Kuivasniemi**

Projektini oli tutkia SIM808 GSM/GPRS moduulia,  
ja sain rakennettua siitä laitteen.

# Laitteen toiminta

Laite vastaa ja/tai toimii annettujen käskyjen mukaan, käskyt voi lähettää laitteelle tekstiviestillä alustuksen jälkeen. Laitteeseen on myös asennettu nappula, joka poistaa kertyneet viestit muisteista, ledi jonka voi sytyttää tekstiviestillä sekä ultraäänianturi, joka mittaa matkaa, ja ilmoittaa sen tekstiviestillä annettuun puhelinnumeroon sitä kysyttäessä ja toimii tuloksen mukaan.

**Mahdolliset tekstiviestikäskyt:**

”Korkea” Päästää korkean äänen sekunnin ajan, kirjoittaa ja lähettää annettuun numeroon viestin ”Piip!”  
  
”Matala” Päästää kaiuttimesta ensin matalan ja sitten hieman korkeamman äänen, ja lähettää viestin ”Beepboop”  
  
”Delete” Toimii kuten laitteen nappi, ja poistaa viestit muisteista  
  
”Sytyta” Sytyttää ledin ja lähettää viestin ”Led sytytetty”  
  
”Sammuta” Samuttaa ledin ja lähettää viestin ”Led sammutettu”

”Matka” Mittaa matkan ultraäänianturista, ja ilmoittaa sen viestillä. Jos matka on lyhyempi kuin 10cm, kaiuttimesta tulee Charlie Brownin teemapimputus, mutta jos matka on yli 10cm ja alle 50cm, se soittaa Metallicaa, ainakin melkein.

”Testi” Kirjoittaa sarjaporttiin ”Toimii”, Päästelee erikorkeuksia ääniä vuorotellen, kirjoittaa sarjaporttiin ”Toimii2” ja kirjoittaa sekä lähettää viestin ”Skulaa”

”Musiikki” Toistaa Metallican

”Music” Toistaa Charlie Brownin tunnarin

**Käytössä olevat AT-komennot:**

Nämä ajetaan Loopissa olevassa initModem funktiossa kerran, ja toimivat kokoajan.

AT Testaa yhteyden

AT+CMGF Valitsee missä muodossa tekstiviestit näytetään

AT+CPMS Valitsee suositun muistin viestien lukuun<mem1>, Tallennukseen<mem2> ja jos CNMItä ei ole asetettu, viestit saapuneet viestit laitetaan tänne<mem3>

AT+CMGL Listaa viestit valitusta muistista, saapuneisiin lukemattomiin, saapuneisiin luettuihin, tallennetut lähettämättömiin, tallennettuihin lähetettyihin ja kaikkiin.

AT+CMGR Itse viestin lukukomento, asetettu lukemaan saapuneet lukemattomat, ja lukemisen jälkeen muuttaamaan luetuksi

AT+CNMI Tarkistaa onko viestejä

AT+CMEE Virheen tunnistus, jos tapahtuu virhe, ilmoittaa laite sen kirjoittamalla ”+CME ERROR”sarjaporttiin ja virheen numerotunnisteen

AT+CLIP Selvittää puhelun tiedot

ATA Tämä tapahtuu vain jos laitteeseen soitetaan, sen seurauksena että jos puhelu tulee, laite kirjoittaa ”RING” sarjaporttiin, ja kun näin käy, tämä suoritetaan ja se vastaa puheluun.

# Puhelin.ino

Tässä on laitteen ’pääkoodi’ eli kerran ajettavat asetukset ja ’looppi’, kukin rivi selitettynä.

#include "Puhelin.h" **//Ottaa käyttöön kirjaston**

void setup() { **//Laitteen asetukset**

pinmode(); **//Suorittaa Pinmode funktion**

Serial.begin(9600); **//Alustaa sarjaportin, 9600 on nopeus (Baud)**

delay(500); **//Annetaan sille aikaa herätä**

clearBuffer();

}

void loop(){ **//Itse toiminto, joka toistuu**

initModem(); **//Modeemin alustusfunktio**

Serial.println("Init modem OK"); **//Tulostaa sarjaporttiin ”-” kun valmis**

while(1){

int x = digitalRead(Button); **//x = painikkeen luku**

if (x == LOW) { **//Jos painike on painettu toistuu seuraava**

Serial.println("Painettu"); **//Kirjoittaa sarjaporttiin ”Painettu”**

String message = "Moi"; **//Kirjoittaa viestin ”moi”**

sendSMS(message); **//Suorittaa sendSMS funktion joka lähettää viestin**

delay(1500); **//Viive että ei toistu montaa kertaa putkeen**

}

checkMessages(); **//Suorittaa checkMessages funktiota**

}

}

# Puhelin.cpp

Tästä löytyy käytettävät funktiot, joita kutsutaan Puhelin.ino tiedostossa

#include <Arduino.h>

#include "Puhelin.h"

SoftwareSerial modem(PIN\_RX,PIN\_TX); //Määritetään pinnit laitteiden väliseen keskusteluun

void clearBuffer(){ //Puskurin tyhjennys funktio

char c;

while(c = modem.read() != -1){ //Kun modeemin vastaus on muuta kun -1

(modem.read() == 0);

Serial.println("joo");

Serial.println(c); //Tämän pitäisi tulostaa mitä modeemista löytyy

}

Serial.println("Puskuri tyhjä"); //Tulostaa sarjaporttiin ”-//-”

}

bool cmdToModem(String str, String ans) { //

Serial.print("Kom: "); //Kirjoittaa sarjaporttiin "Kom:"

Serial.println(str); //Kirjoittaa sarjaporttiin komennon nimen

modem.print(str); //Kirjoittaa moodemille komennon

while(modem.available()<1){delay(2);}//Odotetaan vastausta

String msg;

msg=modem.readString();

Serial.println(msg);

if(msg.indexOf(ans)>=0){

return false;//Odotettu vastaus

}

return true;

}

void initModem(){

modem.begin(9600); //Alustetaan modeemi

if(cmdToModem("AT\r\n", "OK")){ //Testataan AT komennolla toimiiko yhteys

Serial.println("ERROR: AT"); //Tulostaa sarjaporttiin ”-//-”

while(1);

}

if(cmdToModem("AT+CMGF=1\r\n", "OK")){

Serial.println("ERROR: CMGF"); //Tulostaa sarjaporttiin ”-//-”

while(1);

}

if(cmdToModem("AT+CPMS=\"MT\",\"MT\",\"MT\"\r\n", "OK")){

Serial.println("ERROR: CPMS"); //Tulostaa sarjaporttiin ”-//-”

while(1);

}

if(cmdToModem("AT+CMGL=\"REC READ\",0\r\n", "OK")){

Serial.println("ERROR: CMGL"); //Tulostaa sarjaporttiin ”-//-”

while(1);

}

if(cmdToModem("AT+CMGR=3,0\r\n", "OK")){

Serial.println("ERROR: CMGR"); //Tulostaa sarjaporttiin ”-//-”

while(1);

}

if(cmdToModem("AT+CNMI=2,2,0,0,0\r\n", "OK")){

Serial.println("ERROR: CNMI"); //Tulostaa sarjaporttiin ”-//-”

while(1);

}

if(cmdToModem("AT+CMEE=1\r\n", "OK")){

Serial.println("ERROR: CMEE"); //Tulostaa sarjaporttiin ”-//-”

while(1);

}

if(cmdToModem("AT+CLIP=1\r\n", "OK")){

Serial.println("ERROR: CLIP"); //Tulostaa sarjaporttiin ”-//-”

while(1);

}

}

void sendSMS(String message) { //Viestin lähetys funktio

modem.println("AT+CMGS=\"+358451839030\""); //Lähettää viestin numeroon"-"

delay(100);

// Send the SMS

modem.print(message);

delay(100);

Serial.println(message);

// End AT command with a ^Z, ASCII code 26

modem.println((char)26);

delay(50);

modem.println();

delay(5000); // Annetaan modeemille aikaa lähettää viesti

}

void pinmode() {

pinMode (kaiutin, OUTPUT);

pinMode (Button, INPUT\_PULLUP);

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

}

int distanceCm;

void mittaus(){

unsigned long d=0;

for(int i=0;i<5;i++){

digitalWrite(trigPin, HIGH);

delayMicroseconds(8);

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

d += pulseIn(echoPin, HIGH);

}

distanceCm = (d/5)\*0.034/2;

Serial.println(distanceCm); //Tulostaa sarjaporttiin ”-//-”

delay(1000);

}

void checkMessages(){

Serial.read();

if (modem.available() > 0) {

String textMessage = modem.readString();

Serial.println("Incoming:"); //Tulostaa sarjaporttiin ”-//-”

Serial.print(textMessage); //Tulostaa sarjaporttiin saapuneen viestin

delay(10);

if (textMessage.indexOf("Korkea") >= 0) { //Jos viestissä on ”-//-” tapahtuu:

tone(kaiutin, 2000, 1000);

String message = "Arsyttaako?";

sendSMS(message);

}

if (textMessage.indexOf("Delete") >= 0) { //Jos viestissä on ”-//-” tapahtuu:

cmdToModem("AT+CMGD=2,4\r\n", "OK");

textMessage = "";

String message = "Tyhjennetty";

sendSMS(message);

}

if (textMessage.indexOf("Matala") >= 0) {//Jos viestissä on ”-//-” tapahtuu:

tone(kaiutin, 100, 1000);

delay(1000);

tone(kaiutin, 300, 1000);

delay (3000);

String message = "Arsyttaako?";

sendSMS(message);

}

if (textMessage.indexOf(".") >= 0) {//Jos viestissä on ”-//-” tapahtuu:

String message = "Hävisit pelin";

sendSMS(message);

}

if (textMessage.indexOf("RING") >= 0) {//Jos viestissä on ”-//-” tapahtuu:

if (cmdToModem("ATA\r\n", "OK")) {

Serial.println ("Timeout");

}

}

if (textMessage.indexOf("NO CARRIER") >= 0)){//Jos viestissä on ”-//-” tapahtuu

String message = "Puhelu katkaistu";

sendSMS(message);

}

if (textMessage.indexOf("Matka") >=0) {){ //Jos viestissä on ”-//-” tapahtuu:

String message;

char a[4];

Serial.println("Matka: Mitataan"); //Tulostaa sarjaporttiin ”-//-”

mittaus(); //Suorittaa mittaus funktion

Serial.println("Matka: Mitattu"); //Tulostaa sarjaporttiin ”-//-”

itoa(distanceCm, a, 10);

message.concat(a);

message.concat("\r\n");

sendSMS(message);

if (distanceCm < 10){ //Jos mitattu matka on lyhyempi kuin 10cm

CharlieBrown();

}

else if (distanceCm <50){ //Jos mitattu matka on lyhyempi kuin 50cm

Metallica();

}

}

if (textMessage.indexOf("Mix") >= 0) {){ //Jos viestissä on ”-//-” tapahtuu:

Serial.println ("Toimii"); //Tulostaa sarjaporttiin ”-//-”

tone(kaiutin, 1600, 1500);

delay(1500);

tone(kaiutin, 1900, 1500);

delay(1500);

tone(kaiutin, 2100, 1500);

delay(1500);

Serial.println ("Toimii2"); //Tulostaa sarjaporttiin ”-//-”

textMessage = "";

String message = "Arsyttaako?";

sendSMS(message);

}

if (textMessage.indexOf("Musiikki") >= 0){//Jos viestissä on ”-//-” tapahtuu:

Metallica(); //Metallica funktio, Kts. Musiikki2

}

if (textMessage.indexOf("Music") >= 0){//Jos viestissä on ”-//-” tapahtuu:

CharlieBrown(); //CharlieBrown funktio, Kts. Musiikki1

}

}

}

## Musiikki 1

void CharlieBrown(){

tone(kaiutin, 196,200);

delay(200);

tone(kaiutin, 293,200);

delay(200);

tone(kaiutin, 392,300);

delay(300);

tone(kaiutin, 196,200);

delay(200);

tone(kaiutin, 293,200);

delay(200);

tone(kaiutin, 392,600);

delay(600);

tone(kaiutin, 100,200);

delay(200);

tone(kaiutin, 196,200);

delay(200);

tone(kaiutin, 293,200);

delay(200);

tone(kaiutin, 392,500);

delay(500);

tone(kaiutin, 100,100);

delay(100);

tone(kaiutin, 196,200);

delay(200);

tone(kaiutin, 293,200);

delay(200);

tone(kaiutin, 329,500);

delay(500);

tone(kaiutin, 196,200);

delay(200);

tone(kaiutin, 293,200);

delay(200);

tone(kaiutin, 392,300);

delay(300);

tone(kaiutin, 196,200);

delay(200);

tone(kaiutin, 293,200);

delay(200);

tone(kaiutin, 392,600);

delay(600);

tone(kaiutin, 100,200);

delay(200);

tone(kaiutin, 196,200);

delay(200);

tone(kaiutin, 293,200);

delay(200);

tone(kaiutin, 329,300);

delay(300);

tone(kaiutin, 100,100);

delay(100);

tone(kaiutin, 196,200);

delay(200);

tone(kaiutin, 293,300);

delay(300);

tone(kaiutin, 329,800);

delay(1300);

tone(kaiutin, 196,200);

delay(200);

tone(kaiutin, 293,200);

delay(200);

tone(kaiutin, 392,300);

delay(300);

tone(kaiutin, 196,200);

delay(200);

tone(kaiutin, 293,200);

delay(200);

tone(kaiutin, 392,600);

delay(600);

tone(kaiutin, 100,200);

delay(200);

tone(kaiutin, 196,200);

delay(200);

tone(kaiutin, 293,200);

delay(200);

tone(kaiutin, 329,300);

delay(300);

tone(kaiutin, 100,100);

delay(100);

tone(kaiutin, 196,200);

delay(200);

tone(kaiutin, 293,300);

delay(300);

tone(kaiutin, 329,800);

delay(1300);

tone(kaiutin, 196,200);

delay(200);

tone(kaiutin, 293,200);

delay(200);

tone(kaiutin, 392,300);

delay(300);

tone(kaiutin, 196,200);

delay(200);

tone(kaiutin, 293,200);

delay(200);

tone(kaiutin, 392,600);

delay(600);

tone(kaiutin, 100,200);

delay(200);

tone(kaiutin, 196,200);

delay(200);

tone(kaiutin, 293,200);

delay(200);

tone(kaiutin, 329,300);

delay(300);

tone(kaiutin, 100,100);

delay(100);

tone(kaiutin, 196,200);

delay(200);

tone(kaiutin, 293,300);

delay(300);

tone(kaiutin, 329,800);

delay(800);

tone(kaiutin, 50,30);

delay(30);

tone(kaiutin, 196,400);

delay(400);

tone(kaiutin, 220,200);

delay(200);

tone(kaiutin, 246,400);

delay(400);

tone(kaiutin, 50,30);

delay(30);

tone(kaiutin, 246,300);

delay(300);

tone(kaiutin, 50,30);

delay(30);

tone(kaiutin, 220,200);

delay(200);

tone(kaiutin, 196,200);

delay(200);

tone(kaiutin, 220,700);

delay(700);

tone(kaiutin, 196,800);

delay(800);

tone(kaiutin, 50,30);

delay(30);

tone(kaiutin, 196,200);

delay(200);

tone(kaiutin, 220,400);

delay(400);

tone(kaiutin, 246,400);

delay(400);

tone(kaiutin, 50,30);

delay(30);

tone(kaiutin, 246,1000);

delay(1500);

tone(kaiutin, 50,30);

delay(30);

tone(kaiutin, 196,400);

delay(400);

tone(kaiutin, 220,200);

delay(200);

tone(kaiutin, 246,400);

delay(400);

tone(kaiutin, 50,30);

delay(30);

tone(kaiutin, 246,300);

delay(300);

tone(kaiutin, 50,30);

delay(30);

tone(kaiutin, 220,200);

delay(200);

tone(kaiutin, 196,200);

delay(200);

tone(kaiutin, 220,700);

delay(700);

tone(kaiutin, 196,800);

delay(800);

tone(kaiutin, 50,30);

delay(30);

tone(kaiutin, 196,200);

delay(700);

tone(kaiutin, 50,30);

delay(30);

tone(kaiutin, 196,500);

delay(500);

tone(kaiutin, 220,400);

delay(400);

tone(kaiutin, 50,30);

delay(30);

tone(kaiutin, 220,1000);

delay(1000);

tone(kaiutin, 50,30);

delay(530);

tone(kaiutin, 50,30);

delay(30);

tone(kaiutin, 196,400);

delay(400);

tone(kaiutin, 220,200);

delay(200);

tone(kaiutin, 246,400);

delay(400);

tone(kaiutin, 50,30);

delay(30);

tone(kaiutin, 246,300);

delay(300);

tone(kaiutin, 50,30);

delay(30);

tone(kaiutin, 220,200);

delay(200);

tone(kaiutin, 196,200);

delay(200);

tone(kaiutin, 220,700);

delay(700);

tone(kaiutin, 196,800);

delay(800);

tone(kaiutin, 50,30);

delay(30);

tone(kaiutin, 196,200);

delay(200);

tone(kaiutin, 220,400);

delay(400);

tone(kaiutin, 246,400);

delay(400);

tone(kaiutin, 50,30);

delay(30);

tone(kaiutin, 246,1000);

delay(1500);

tone(kaiutin, 50,30);

delay(30);

tone(kaiutin, 196,400);

delay(400);

tone(kaiutin, 220,200);

delay(200);

tone(kaiutin, 246,400);

delay(400);

tone(kaiutin, 50,30);

delay(30);

tone(kaiutin, 246,300);

delay(300);

tone(kaiutin, 50,30);

delay(30);

tone(kaiutin, 220,200);

delay(200);

tone(kaiutin, 196,200);

delay(200);

tone(kaiutin, 220,700);

delay(700);

tone(kaiutin, 196,800);

delay(800);

tone(kaiutin, 50,30);

delay(30);

tone(kaiutin, 196,200);

delay(700);

tone(kaiutin, 50,30);

delay(30);

tone(kaiutin, 196,500);

delay(250);

tone(kaiutin, 220,400);

delay(400);

tone(kaiutin, 50,30);

delay(30);

tone(kaiutin, 220,1000);

delay(1000);

tone(kaiutin, 50,30);

delay(530);

tone(kaiutin, 50,30);

delay(30);

tone(kaiutin, 196,600);

delay(600);

tone(kaiutin, 220,300);

delay(300);

tone(kaiutin, 50,30);

delay(30);

tone(kaiutin, 220,300);

delay(300);

tone(kaiutin, 220,800);

delay(1300);

tone(kaiutin, 50,30);

delay(30);

tone(kaiutin, 196,2000);

delay(2500);

tone(kaiutin, 50,30);

delay(30);

tone(kaiutin, 190,10);

delay(10);

tone(kaiutin, 196,1500);

delay(2000);

tone(kaiutin, 258,200);

delay(200);

tone(kaiutin, 50,30);

delay(30);

tone(kaiutin, 261,400);

delay(400);

tone(kaiutin, 50,30);

delay(30);

tone(kaiutin, 261,400);

delay(400);

tone(kaiutin, 50,30);

delay(30);

tone(kaiutin, 261,400);

delay(900);

tone(kaiutin, 50,30);

delay(30);

tone(kaiutin, 293,700);

delay(1200);

tone(kaiutin, 329,500);

delay(500);

tone(kaiutin, 293,200);

delay(200);

tone(kaiutin, 329,500);

delay(1000);

tone(kaiutin, 258,200);

delay(200);

tone(kaiutin, 50,30);

delay(30);

tone(kaiutin, 261,400);

delay(400);

tone(kaiutin, 50,30);

delay(30);

tone(kaiutin, 261,400);

delay(400);

tone(kaiutin, 50,30);

delay(30);

tone(kaiutin, 261,400);

delay(900);

tone(kaiutin, 50,30);

delay(30);

tone(kaiutin, 293,700);

delay(1200);

tone(kaiutin, 329,500);

delay(500);

tone(kaiutin, 293,200);

delay(200);

tone(kaiutin, 329,500);

delay(1000);

tone(kaiutin, 50,30);

delay(30);

tone(kaiutin, 196,600);

delay(600);

tone(kaiutin, 220,300);

delay(300);

tone(kaiutin, 50,30);

delay(30);

tone(kaiutin, 220,300);

delay(300);

tone(kaiutin, 220,800);

delay(1300);

tone(kaiutin, 50,30);

delay(30);

tone(kaiutin, 196,2000);

delay(2500);

tone(kaiutin, 50,30);

delay(30);

tone(kaiutin, 190,10);

delay(10);

tone(kaiutin, 196,1500);

delay(1500);

return 0;

}

## Musiikki 2

void Metallica(){

tone(kaiutin, 329, 300);//E

delay(300);

tone(kaiutin, 493, 300); //B

delay(300);

tone(kaiutin, 698, 300); //F^

delay(300);

tone(kaiutin, 659, 300); //E^

delay(400);

tone(kaiutin, 329, 150); //G^

delay(150);

tone(kaiutin, 493, 150); //F^

delay(300);

tone(kaiutin, 783, 300); //E^

delay(300);

tone(kaiutin, 698, 300);

delay(400);

tone(kaiutin, 329, 300);

delay(300);

tone(kaiutin, 493, 300);

delay(300);

tone(kaiutin, 698, 300);

delay(300);

tone(kaiutin, 590, 300);

delay(400);

tone(kaiutin, 392, 150);

delay(150);

tone(kaiutin, 440, 150);

delay(150);

tone(kaiutin, 587, 300);

delay(400);

tone(kaiutin, 349, 300);

delay(300);

tone(kaiutin, 587, 300);

delay(600);

tone(kaiutin, 329, 300);

delay(300);

tone(kaiutin, 493, 300);

delay(300);

tone(kaiutin, 698, 300);

delay(300);

tone(kaiutin, 659, 300);

delay(400);

tone(kaiutin, 329, 150);

delay(150);

tone(kaiutin, 493, 150);

delay(300);

tone(kaiutin, 783, 300);

delay(300);

tone(kaiutin, 698, 300);

delay(400);

tone(kaiutin, 329, 300);

delay(300);

tone(kaiutin, 493, 300);

delay(300);

tone(kaiutin, 698, 300);

delay(300);

tone(kaiutin, 590, 300);

delay(400);

tone(kaiutin, 392, 150);

delay(150);

tone(kaiutin, 440, 150);

delay(150);

tone(kaiutin, 587, 300);

delay(400);

tone(kaiutin, 349, 300);

delay(300);

tone(kaiutin, 587, 300);

delay(600);

}

# Puhelin.h

#ifndef Puhelin\_h

#define Puhelin\_h

#include <Arduino.h> //Arduinon oma kirjasto pitää ottaa erikseen käyttöön

#include <SoftwareSerial.h> //Laitteiden välisen keskustelun kirjasto

#define PIN\_RX 7 //Määritetään pinnit modeemin ja arduinon keskusteluun

#define PIN\_TX 8

#define kaiutin 11 //Määritetään pinni kaiuttimelle

#define Button 2 //Määritetään pinni nappulalle

#define LED 3

const int trigPin = 5; //Määritetään sisään ja ulostulopinnit ’silmille’

const int echoPin = 4;

Käytettävät funktiot:

void pinmode();

void clearBuffer();

void initModem();

void sendSMS(String);

void mittaus();

void checkMessages();

bool cmdToModem(String str, String ans);

#endif

# Omat tavoitteet

Ennen kurssia en ymmärtänyt koodausta ollenkaan, enkä ollut perehtynyt kurssin sisältöön ennen sen alkua, joten tavoitteita kurssin suhteen ei oikeastaan ollut. Mutta lopputulos on positiivisesti yllättävä. Monesti olen miettinyt että miten asiat toimii, ja nyt on ainakin jonkinlainen idea koodista, elektroniikasta ja joidenkin ohjelmointikielien rakenteesta.

En saanut dokumentointia ihan valmiiksi, ja koodissa ja dokumentissa on pieniä eroja.

# Osaluettelo

* Nappula
* Kaiutin
* HC-SR04 mittari
* SIM808 Moduuli
* SIM kortti
* Arduino uno
* Hyppylankoja
* Ledi