GSM-Phone

Ari Rankinen G8115

Juhana Suhonen G8010

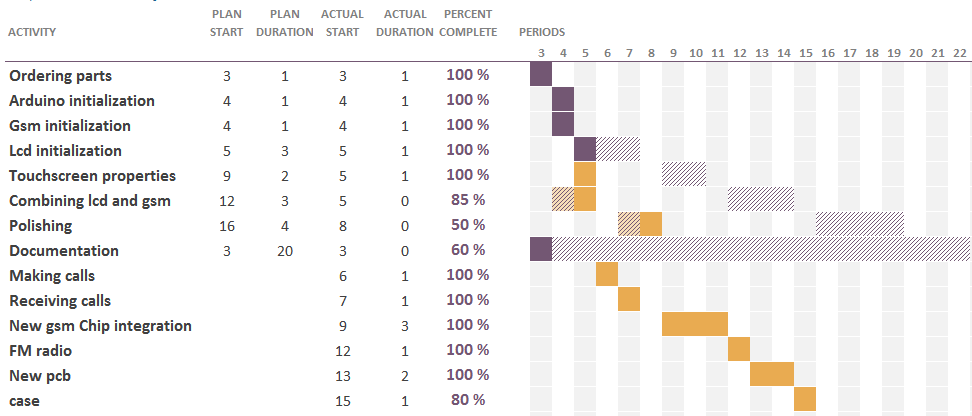
22.4.2015

## Project overview

The main target was to create a prototype of a mobile phone that uses a touchscreen interface. The main platform was chosen to be arduino uno. The project also required modules that communicated with arduino. Such as GSM module and touchscreen module. Both of which were chosen from a company that develops 3rd party arduino modules and libraries.

|  |  |
| --- | --- |
| Platform | Arduino uno (atmel 328p 8bit) |
| Gsm module | Adafruit fona uFl |
| Touchscreen module | Adafruit 2.8” tft |
| Battery | 2000mah 1s lipoly |
| speaker | 8ohm |
| microphone |  |
| Old gsm module | Arduino gsm shield |

## Scheduling



The original product plan did not anticipate how fast it was to implement the planned features and that is why half of the scheduling does not have planned start dates or durations.

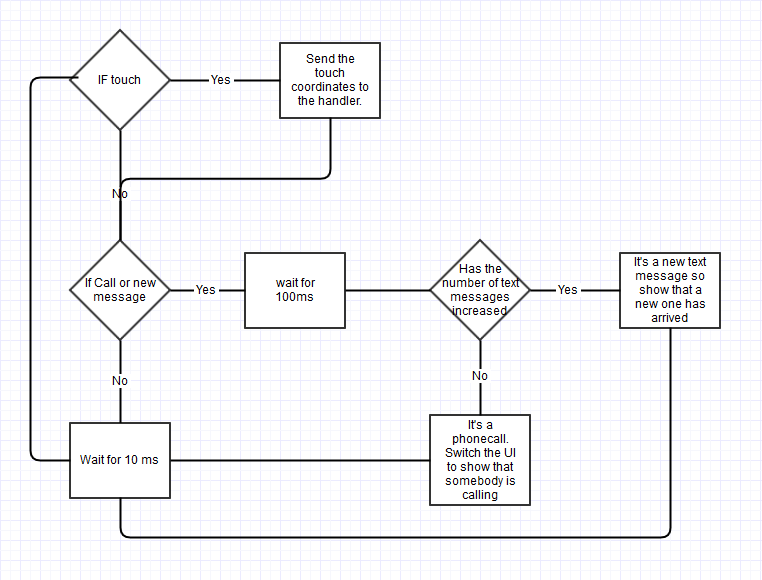
## Software overview

Arduino uses its own compiler and custom c code. The c arduino uses is quite close to ANSI c but it has some slight modifications to it. For example in arduino the main loop is simply called “loop”. In this project we used several libraries from adafruit. One library for the gsm module and a couple libraries to use the tft touchscreen.

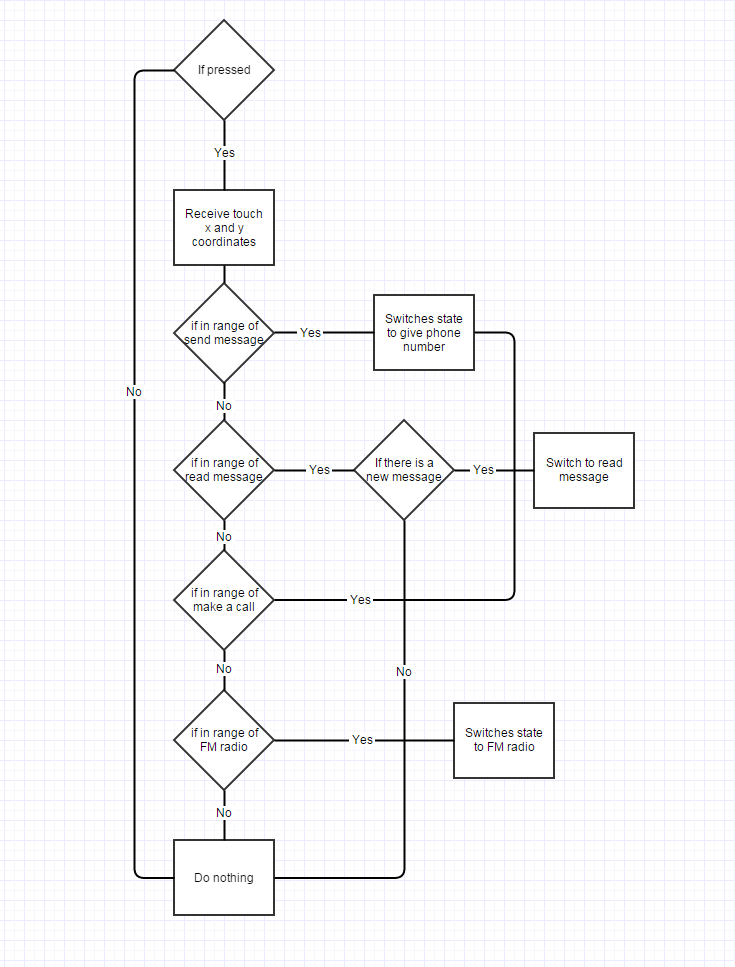
|  |  |
| --- | --- |
| Adafruit\_GFX.h | draw calls |
| Adafruit\_ILI9341.h | display driver |
| TouchScreen.h | touch control |
| SoftwareSerial.h | serial communication |
| dafruit\_FONA.h | Gsm module |

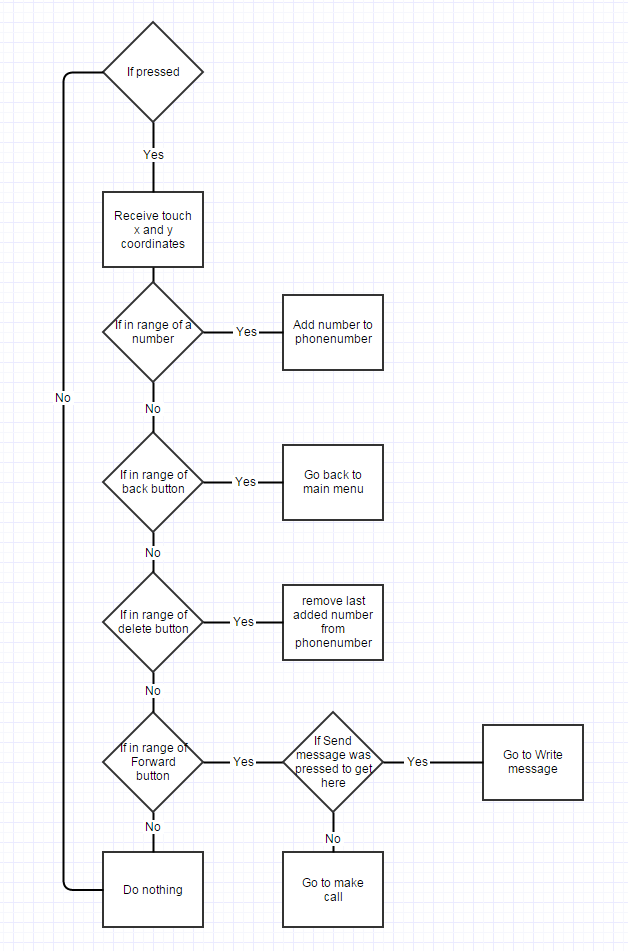
## Main functionality in software

Main loop functionality

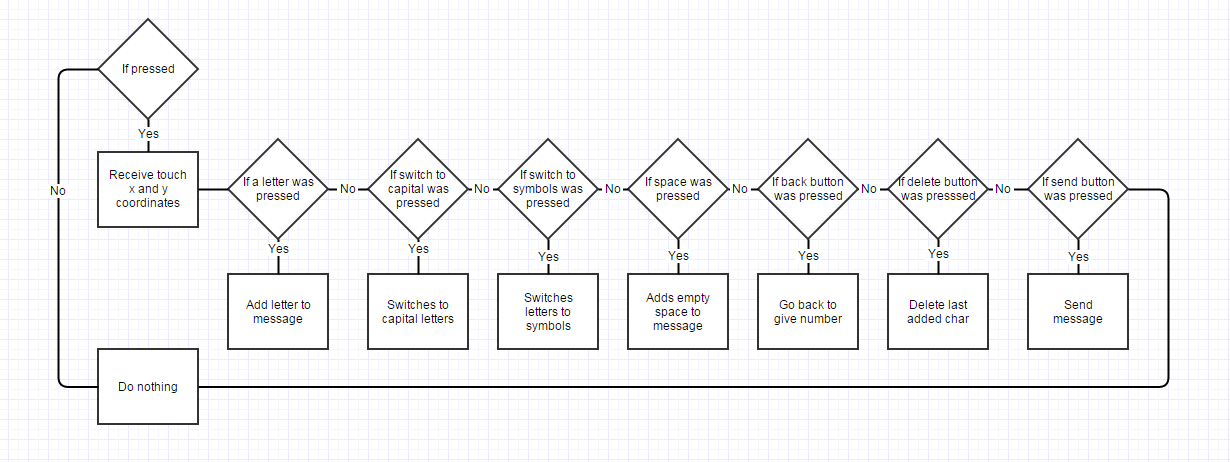


Main menu functionality

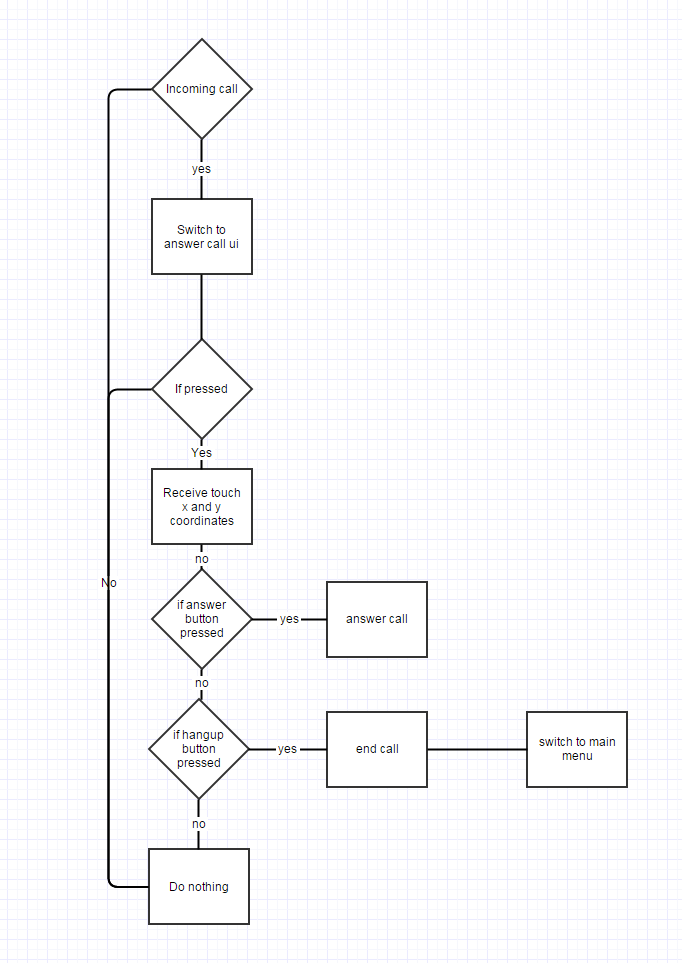


Give number functionality

Write message functionality



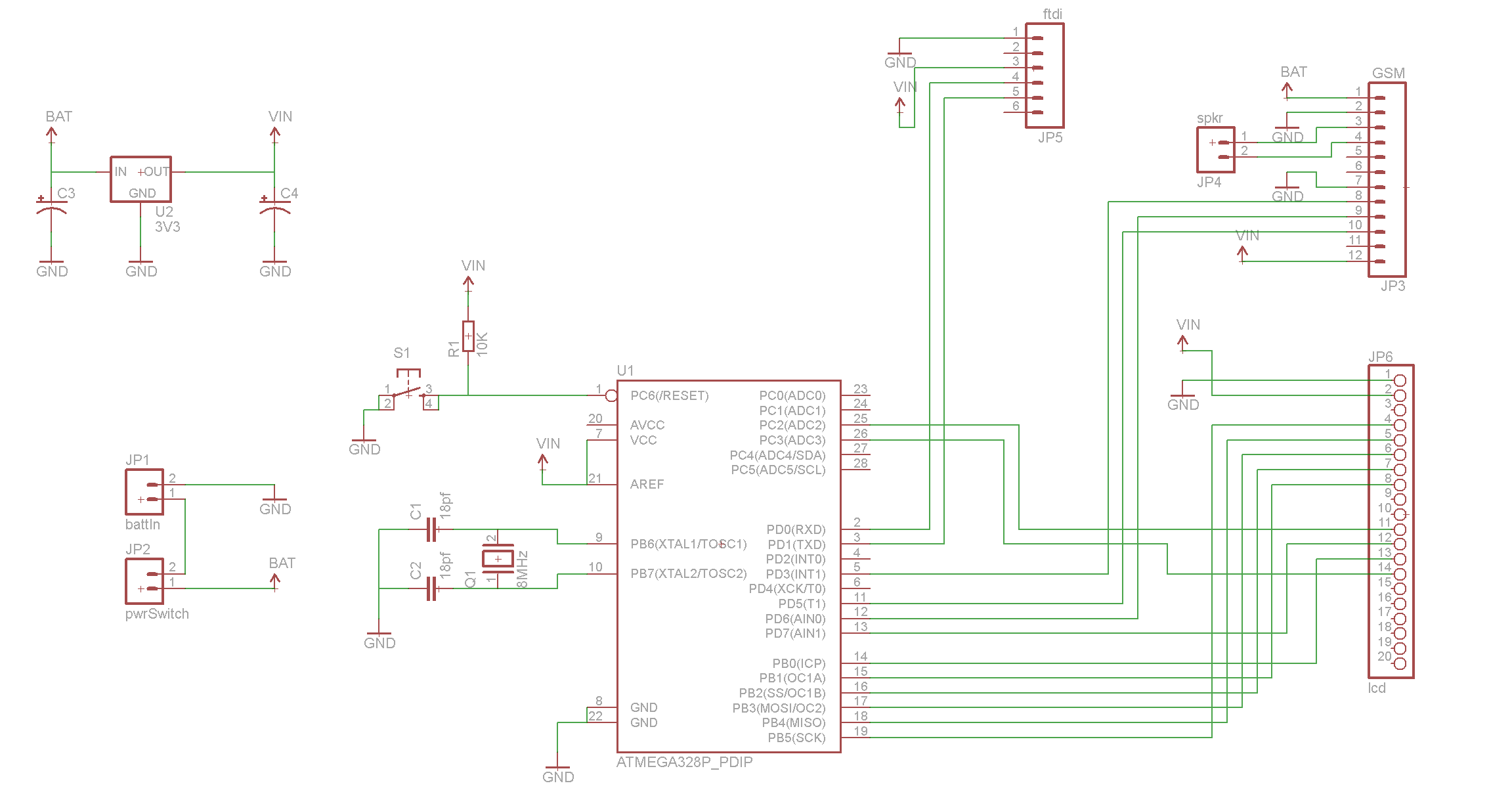
Make call functionality

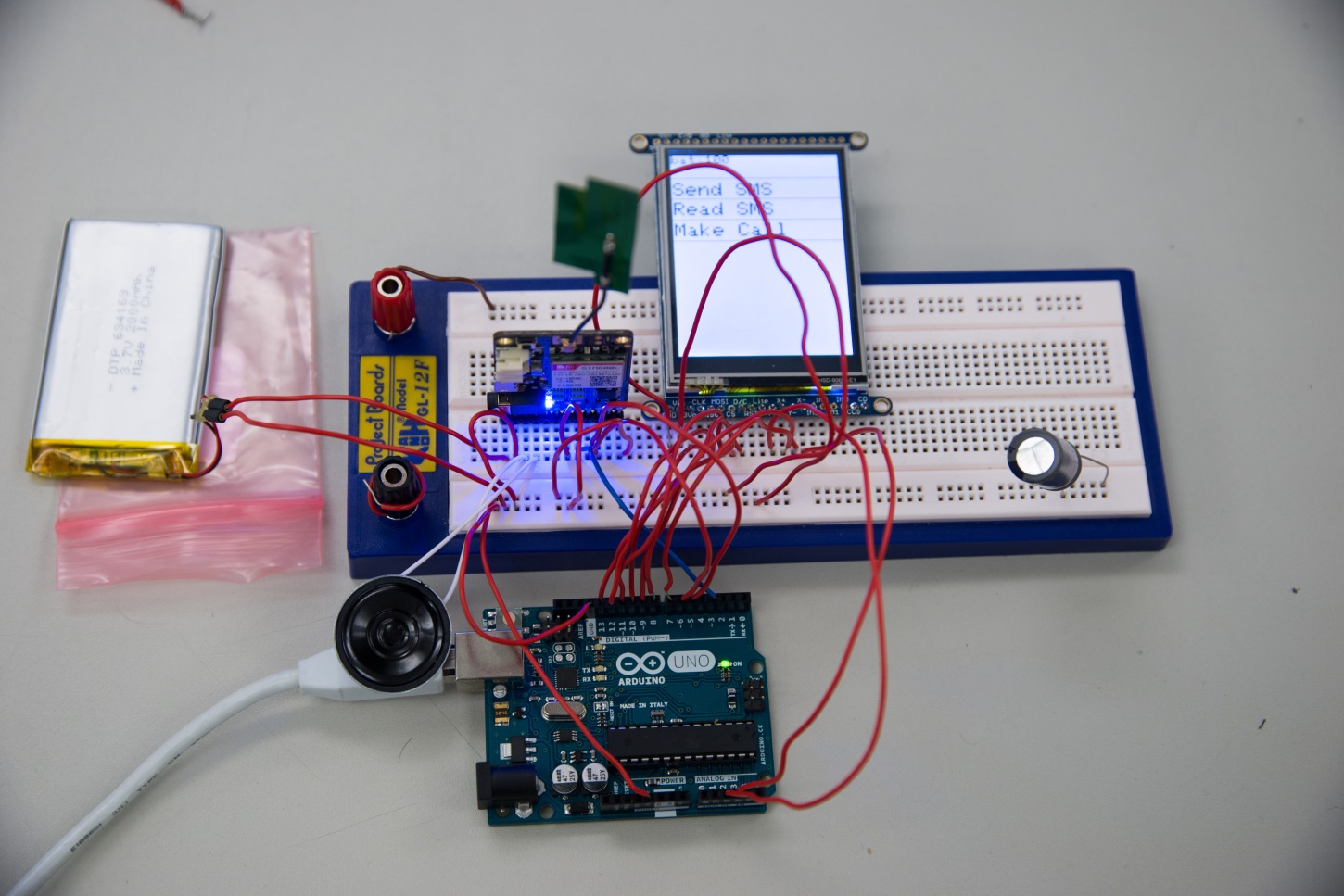


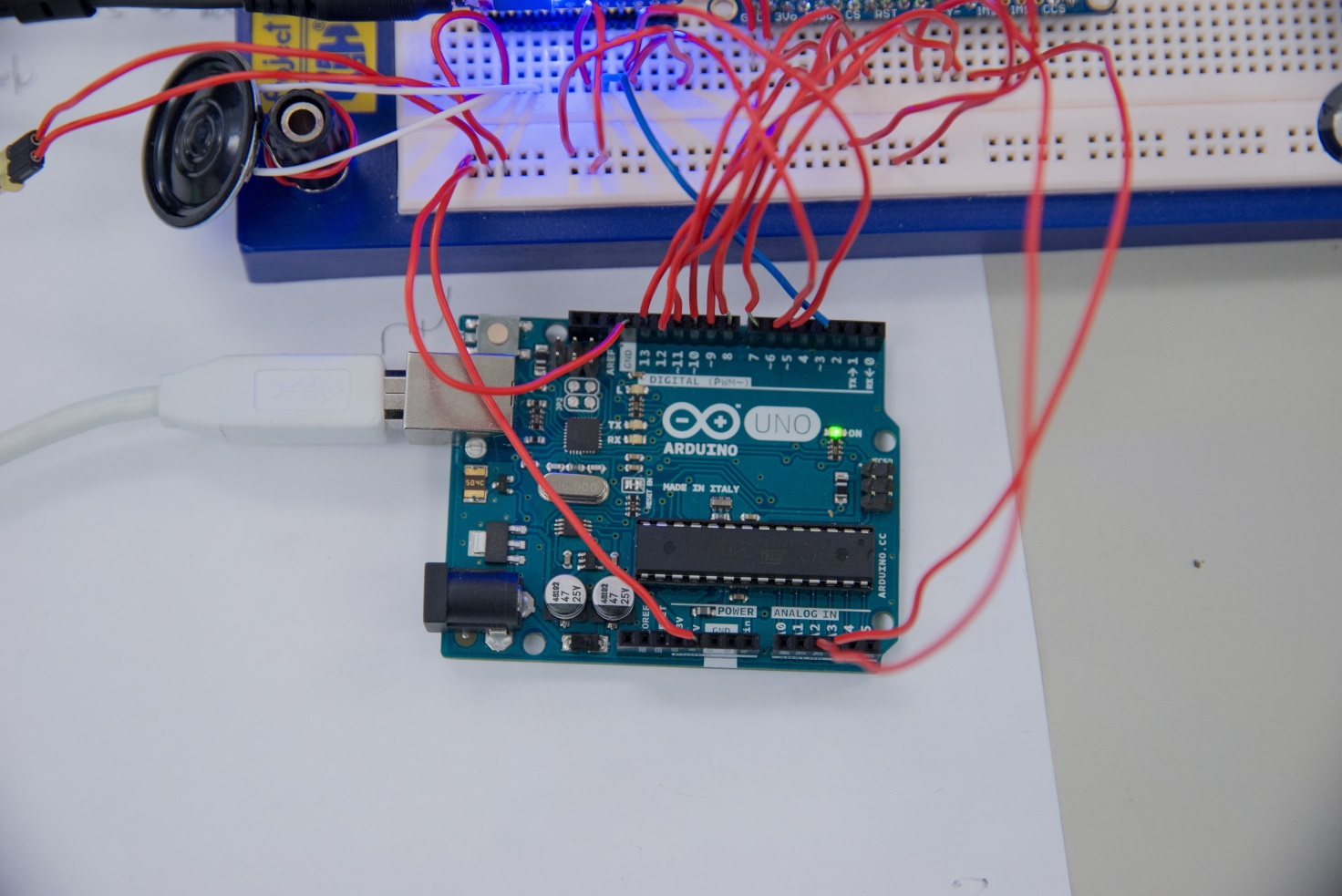
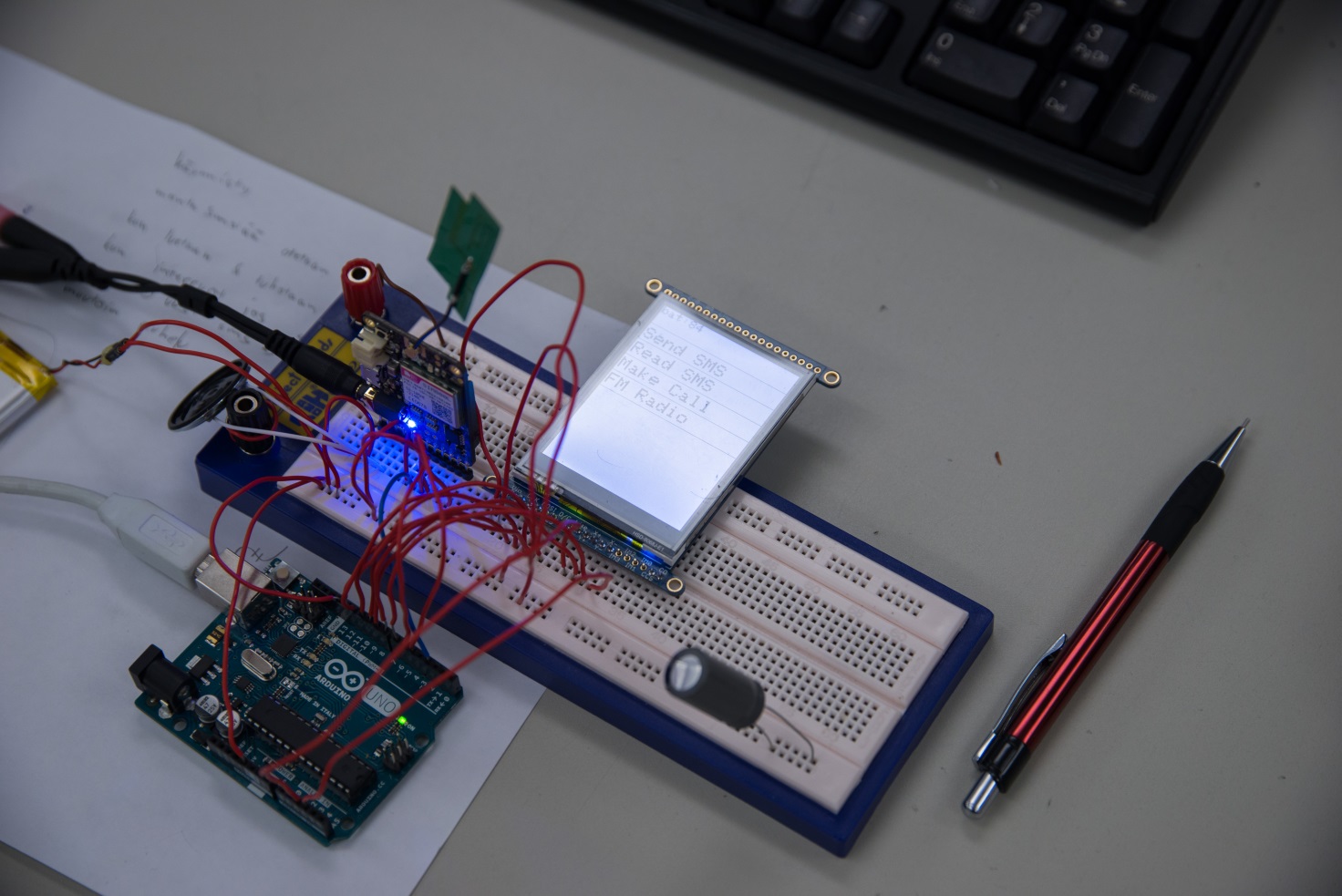
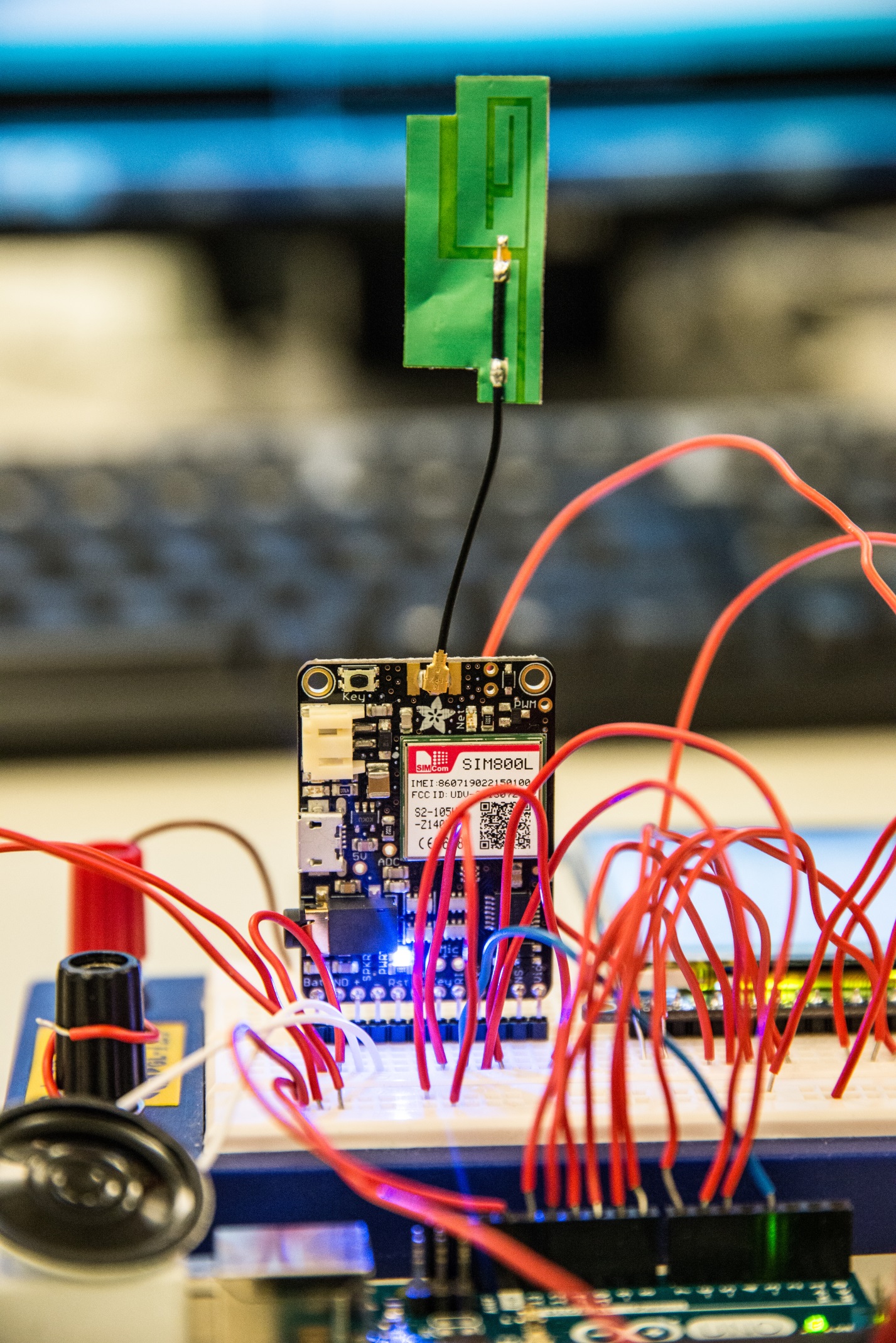
## Hardware assembly

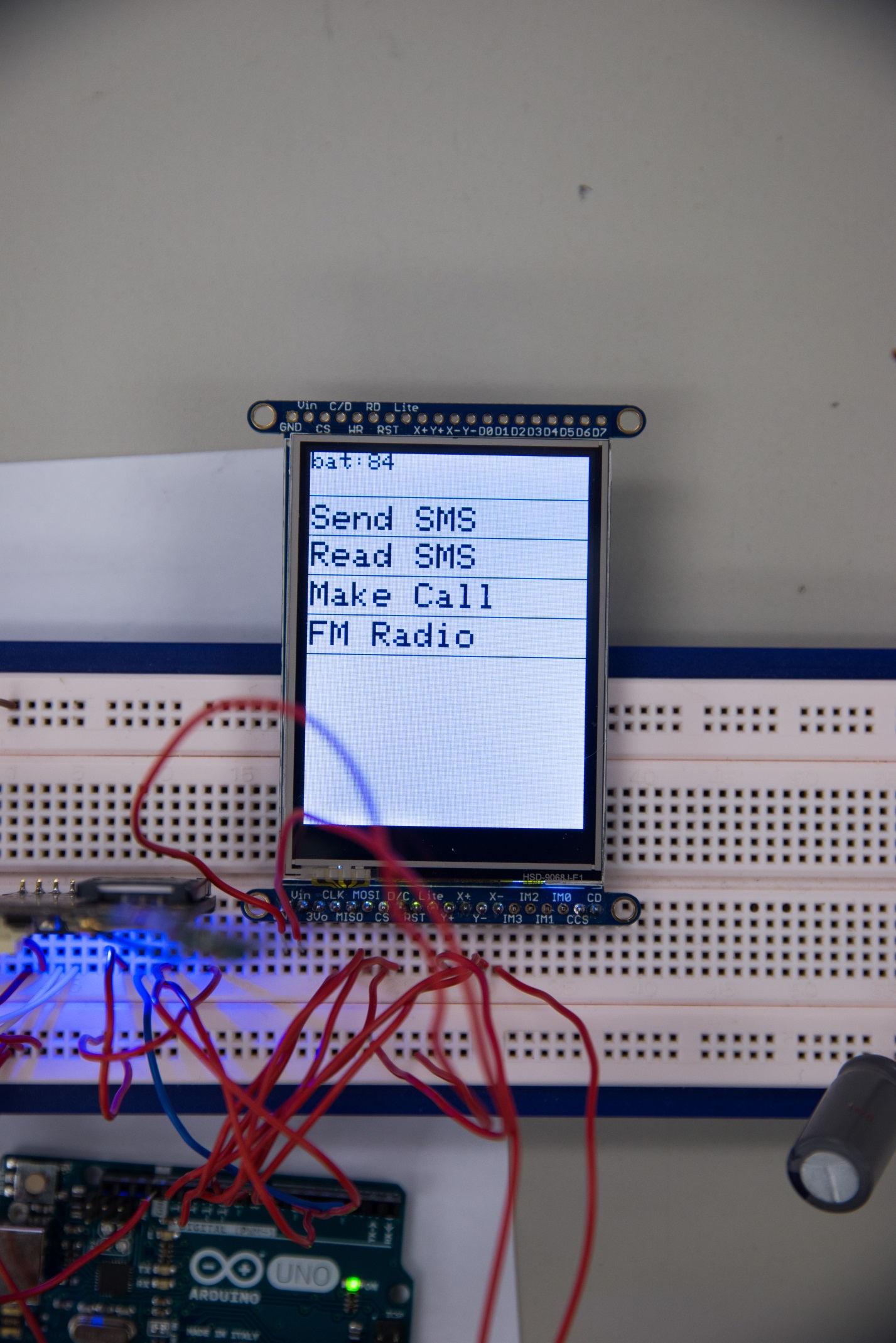
The circuit was designed by following and combining adafruit and arduino tutorials. Overall prototype circuit was created on a breadboard with headers and wires.

Final schematic



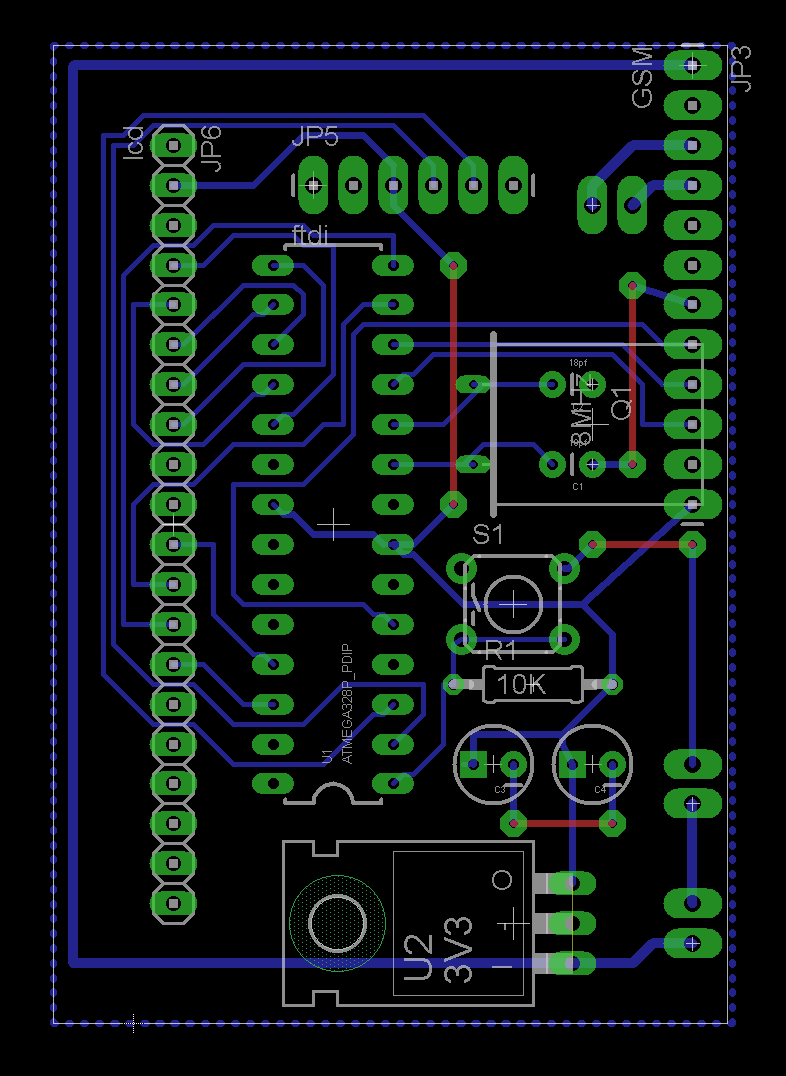
Pictures of prototype phases.





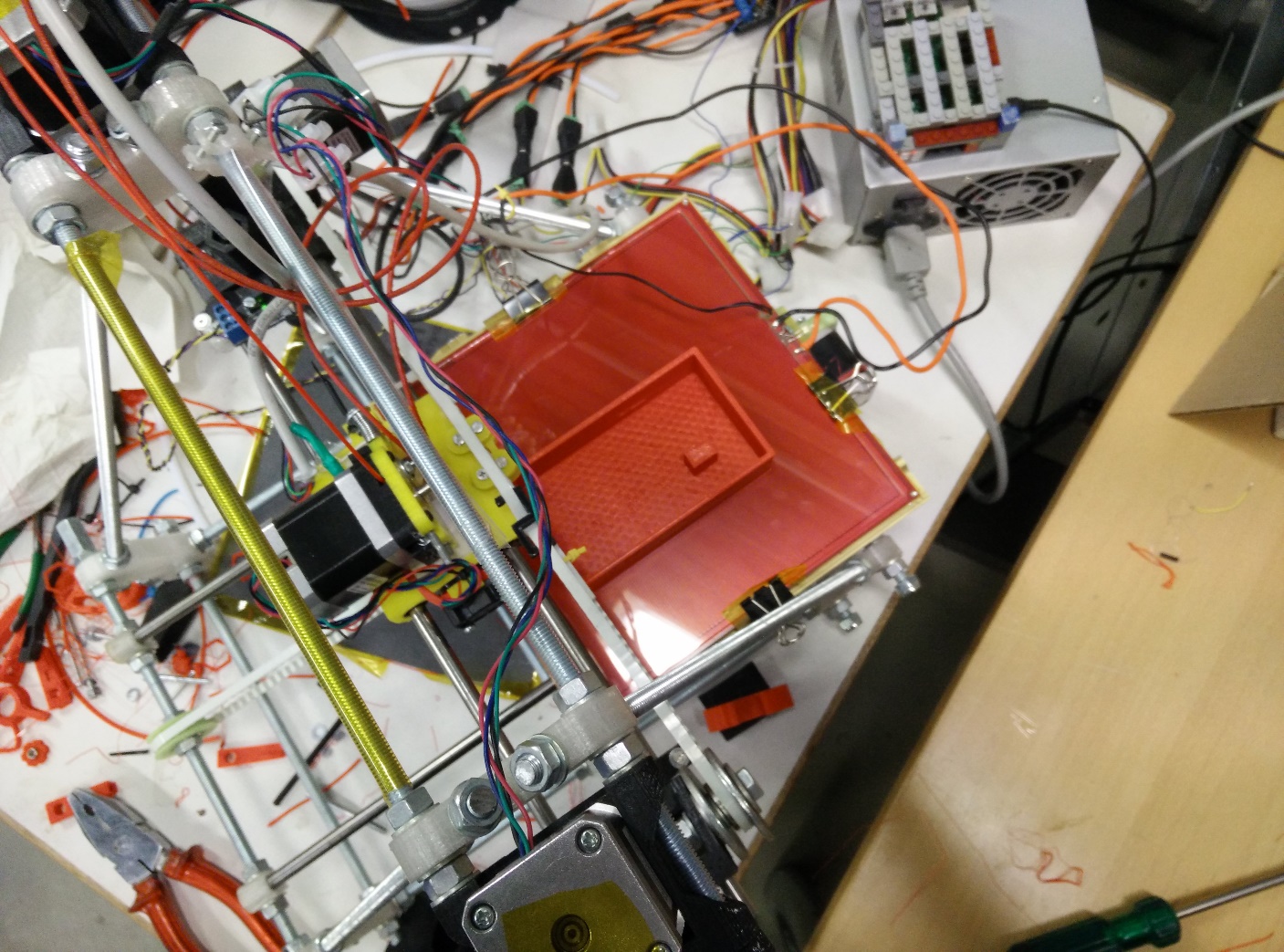
## Making it mobile

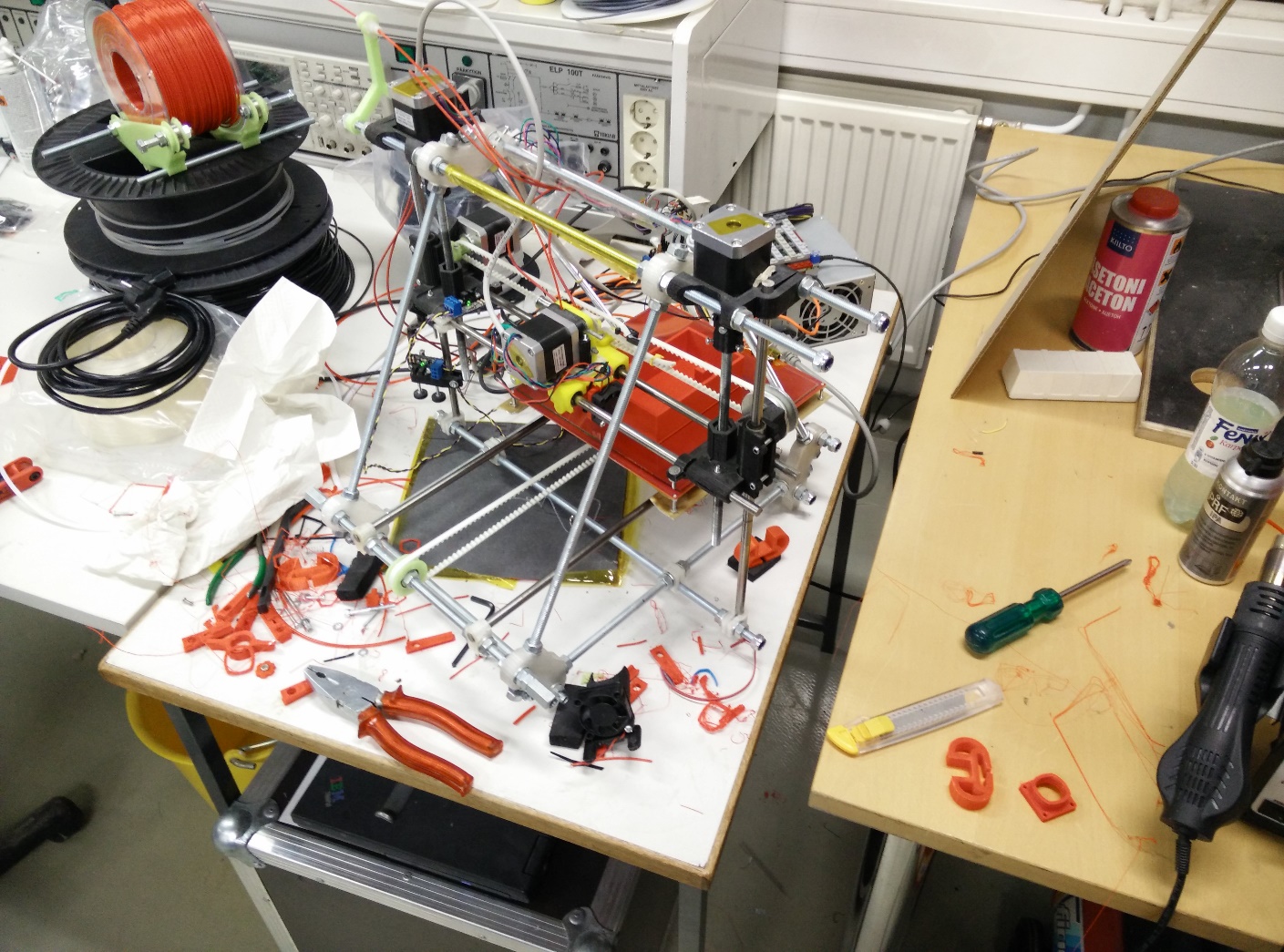
The project progressed so rapidly that it was decided to create our own pcb so that the phone would be completely mobile.



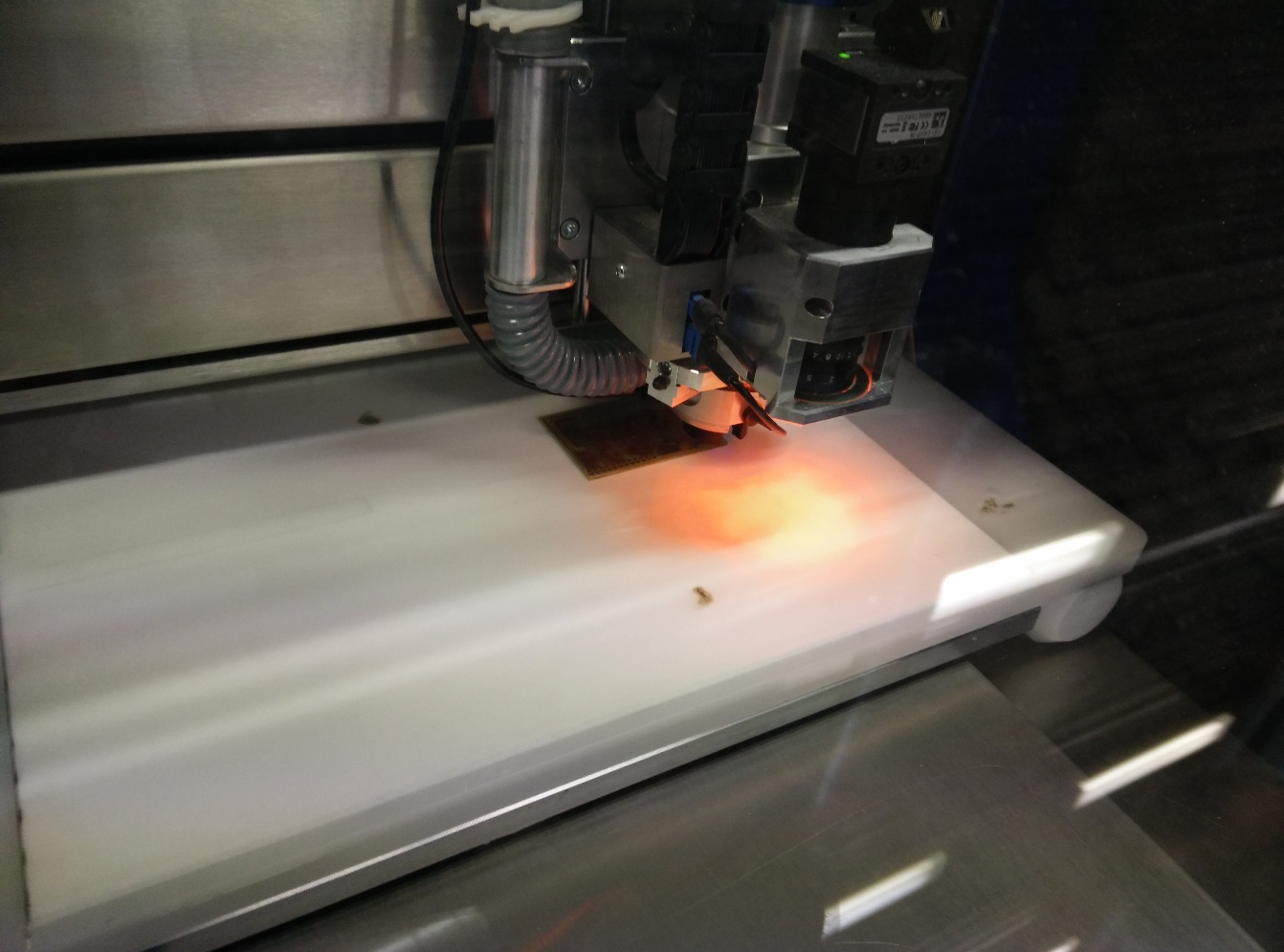
A casing was also designed and printed using Jussi Lemmetty’s self-made 3D printer.

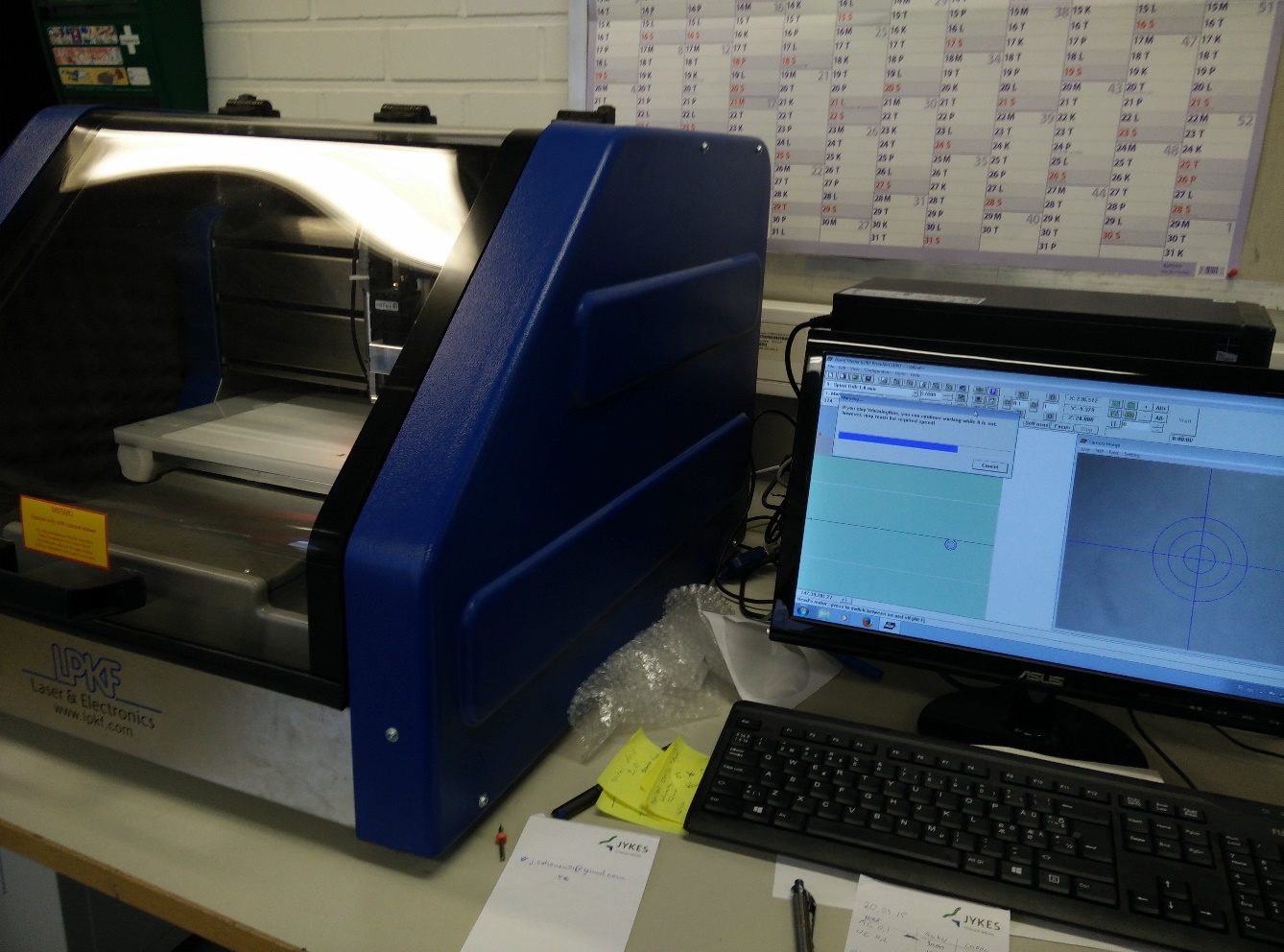






The pcb was etched at school and the holes were drilled in HackLab JKL using LPKF pcb cnc drill.





What possibly went wrong

When all components were assembled the circuit did not boot correctly.

Because the new phone with it’s own pcb had to use a one cell lipo battery regulated to 3.3V. Because of this the microcontroller’s bootloader had to be changed to use 8Mhz clock instead of 16Mhz on Arduino uno. There were problems uploading the software to the Arduino after the bootloader was changed. There also might be a problem with the power supply to deliver enough power to the circuit.

## Final word

We think that this was a great project. And we would recommend it to others as well. The beginning was super smooth and the only problems appeared at the end when we tried to switch to our own pcb.

## Source code for the project

#include "SPI.h"

#include "Adafruit\_GFX.h"

#include "Adafruit\_ILI9341.h"

#include "TouchScreen.h"

#include <SoftwareSerial.h>

#include "Adafruit\_FONA.h"

// For the Adafruit shield, these are the default.

#define TFT\_DC 9

#define TFT\_CS 10

#define PINNUMBER "1620"

// These are the four touchscreen analog pins

#define YP A2 // must be an analog pin, use "An" notation!

#define XM A3 // must be an analog pin, use "An" notation!

#define YM 8 // can be a digital pin

#define XP 7 // can be a digital pin

// This is calibration data for the raw touch data to the screen coordinates

#define TS\_MINX 120//150

#define TS\_MINY 120//120

#define TS\_MAXX 920//920

#define TS\_MAXY 900//940

#define FONA\_RI\_INTERRUPT 1

#define MINPRESSURE 5

#define MAXPRESSURE 1000

#define FONA\_RX 5

#define FONA\_TX 6

#define FONA\_RST 4

SoftwareSerial fonaSS = SoftwareSerial(FONA\_TX, FONA\_RX);

Adafruit\_FONA fona = Adafruit\_FONA(FONA\_RST);

// Use hardware SPI (on Uno, #13, #12, #11) and the above for CS/DC

Adafruit\_ILI9341 tft = Adafruit\_ILI9341(TFT\_CS, TFT\_DC);

// If using the breakout, change pins as desired

//Adafruit\_ILI9341 tft = Adafruit\_ILI9341(TFT\_CS, TFT\_DC, TFT\_MOSI, TFT\_CLK, TFT\_RST, TFT\_MISO);

TouchScreen ts = TouchScreen(XP, YP, XM, YM, 300);

char nappaimisto;

char vaihto[2];

int SideLength=30;

int startPoint\_X=0;

int startPoint\_Y=135;

//alignaa kirjaimet gridiin

int x\_offset=8;

int y\_offset=5;

int akku=0;

int pointteri=0;

boolean vastaa = false;

int touchTila=0;

int prevTouch;

char callingNumber [20];

char PhoneNumber[20];

int NewMessage=0;

char textMessage[160];

int smsInterval=0;

boolean puhelu = false;

int numberofsms;

int prev=0;

int freqpoi=0;

int frequency;

char valifreq[5];

void setup() {

Serial.begin(115200);

fonaSS.begin(4800); // if you're using software serial

tft.begin();

tft.setTextColor(ILI9341\_BLACK);

tft.fillScreen(ILI9341\_WHITE);

tft.setTextSize(3);

tft.setCursor(80,260);

tft.print("LOADING");

attachInterrupt(1, Incoming, FALLING);

fona.begin(fonaSS);

fona.unlockSIM("1620");

//fona.callerIdNotification(true, FONA\_RI\_INTERRUPT);

numberofsms=fona.getNumSMS();

ScreenState();

Serial.println(numberofsms);

Serial.println(fona.getNumSMS());

fona.setAudio(FONA\_EXTAUDIO);

fona.FMradio(false);

}

void loop(void) {

TSPoint p = ts.getPoint();

/\*

if (p.z > MINPRESSURE && p.z < MAXPRESSURE) {

if(smsInterval-prev>19){

prev=smsInterval;

p.x = map(p.x, TS\_MINX, TS\_MAXX, 0, tft.width());

p.y = map(p.y, TS\_MINY, TS\_MAXY, 0, tft.height());

TouchControls(p.x,p.y);

}

}

if(smsInterval==1000){

smsInterval=0;

prev=0;

if( fona.getNumSMS()>0 ){

NewMessage=1;

if(touchTila==0)

ScreenState();

}

}\*/

if (p.z > MINPRESSURE && p.z < MAXPRESSURE) {

if(smsInterval-prev>15){

prev=smsInterval;

p.x = map(p.x, TS\_MINX, TS\_MAXX, 0, tft.width());

p.y = map(p.y, TS\_MINY, TS\_MAXY, 0, tft.height());

TouchControls(p.x,p.y);

}

}

if(puhelu)

{

puhelu = false;

//Serial.println(numberofsms);

//Serial.println(fona.getNumSMS());

//fona.getNumSMS();

delay(100);

if( numberofsms < fona.getNumSMS())

{

//tekstiviestihandlaus

numberofsms=fona.getNumSMS();

NewMessage=1;

if(touchTila==0)

ScreenState();

//Serial.println("message");

}

else{

//tahan puhelu handlaus

//Serial.println(touchTila);

fona.FMradio(false);

fona.setAudio(FONA\_EXTAUDIO);

prevTouch=touchTila;

touchTila=6;

//Serial.println(touchTila);

ScreenState();

//Serial.println("call");

}

}

if(smsInterval==1000){

prev=0;

smsInterval=0;

if(touchTila==0||touchTila==7){

printStatuses();

}

}

/\*

if(fona.incomingCallNumber(callingNumber)&&puhelu){

puhelu = false;

prevTouch=touchTila;

touchTila=6;

ScreenState();

}

\*/

delay(10);

smsInterval++;

}

void Incoming(){

puhelu = true;

}

void answeringCallUi()

{

tft.setTextSize(2);

tft.setCursor(0,80);

tft.print(callingNumber);

tft.setCursor(0,100);

tft.print("Calling");

tft.setCursor(0,140);

tft.setTextColor(ILI9341\_GREEN);

tft.print("Answer");

tft.setCursor(0,180);

tft.setTextColor(ILI9341\_RED);

tft.print("Hang Up");

tft.setTextColor(ILI9341\_BLACK);

}

void TouchControlAnsweringCall(int x, int y)

{

if(y > 120 && y < 160){

tft.setCursor(85,230);

tft.print("DELETE");

delay(100);

fona.pickUp();

}

if(y > 160 && y < 200)

{

fona.hangUp();

touchTila = prevTouch;

Serial.println(touchTila);

ScreenState();

}

}

void LoadMakeCall()

{

tft.setCursor(0,80);

tft.print("Fingerpori");

}

void TouchControlMakeCall(int x,int y)

{

if( x > -1 && x < 241 && y > 0 && y < 340 )

{

EmptyNumber();

fona.hangUp();

touchTila=0;

ScreenState();

}

}

void TouchControls(int x,int y)

{

switch(touchTila)

{

case 0:

//mainmenu control

TouchControlMainMenu(x,y);

break;

case 1:

//Control for setting phonenumber

TouchControlGiveNumber(x,y);

break;

case 2:

//control for writing message

TouchControlKeyboard(x,y);

break;

case 3:

//control for reading message

TouchControlReadMessage(x,y);

break;

case 4:

TouchControlGiveNumber(x,y);

break;

case 5:

TouchControlMakeCall(x,y);

break;

case 6:

//answer call

TouchControlAnsweringCall(x, y);

break;

case 7:

//radio

TouchControlFMRadio(x, y);

break;

}

}

void ScreenState()

{

tft.fillScreen(ILI9341\_WHITE);

/\*

tft.setCursor(0,0);

tft.setTextSize(2);

tft.setTextColor(ILI9341\_GREEN);

tft.println(scanner.getCurrentCarrier());

\*/

tft.setTextSize(3);

tft.setTextColor(ILI9341\_BLACK);

switch(touchTila)

{

case 0:

//Load Main menu graphics

EmptyMessage();

EmptyNumber();

LoadMainMenu();

break;

case 1:

//Load write number graphics

pointteri=3;

SetPhoneNumberMenu();

break;

case 2:

//Load write message graphics

pointteri=0;

EmptyMessage();

SetUpLargeText();

PrintNumber(0);

NappaimistoTulostus(26);

break;

case 3:

LoadMessageRead();

//Load read message graphics

break;

case 4:

//Load write number graphics

pointteri=3;

SetPhoneNumberMenu();

break;

case 5:

//make call

LoadMakeCall();

fona.callPhone(PhoneNumber);

break;

case 6:

answeringCallUi();

break;

case 7:

//radio

LoadRadioMenu();

break;

}

}

void SetPhoneNumberMenu(){

nappaimisto='0';

PhoneNumber[0]='0';

PhoneNumber[1]='5';

PhoneNumber[2]='0';

tft.setTextSize(3);

tft.setCursor(2,45);

tft.println("Give Number");

PrintNumber(75);

/\*int xxx=15;

while(xxx<240){

tft.drawLine(xxx, 0, xxx, 340, ILI9341\_BLACK);

xxx+=40;

}

\*/

for(int i=0;i<10;i++)

{

tft.setCursor( 30 + ( i - ( i / 5 ) \* 5 ) \* 40 , 130 + ( i / 5 ) \* 40 );

tft.print(char(nappaimisto+i));

}

tft.setCursor(85,230);

tft.print("DELETE");

tft.drawLine(0, 220, 240, 220, ILI9341\_BLACK);

tft.setCursor(80,260);

tft.print("PROCEED");

tft.drawLine(0, 252, 240, 252, ILI9341\_BLACK);

tft.setCursor(95,290);

tft.print("BACK");

tft.drawLine(0, 282, 240, 282, ILI9341\_BLACK);

}

void TouchControlFMRadio(int x, int y){

//numbers

if( x > 15 && x < 221 && y > 120 && y < 200 )

{

if(freqpoi<4){

valifreq[freqpoi]=nappaimisto + ( ( y - 125 ) / 40 ) \* 5 + ( x - 15 ) / 40;

freqpoi++;

tft.setCursor(0,75);

tft.print(valifreq);

}

}

//delete

else if( x > -1 && x < 240 && y > 220 && y < 252 )

{

if(freqpoi>0)

{

freqpoi--;

valifreq[freqpoi]=' ';

for(int i=(1+freqpoi)\*12;i>=pointteri\*12;i--){

tft.drawLine(i, 75, i, 89, ILI9341\_WHITE);

}

tft.setCursor(0,75);

tft.print(valifreq);

}

}

//setfreq

else if( x > -1 && x < 240 && y > 252 && y < 282 )

{

//conversio 0=48 9=57

frequency = 0;

frequency += ((int)valifreq[0]-48)\*1000;

frequency += ((int)valifreq[1]-48)\*100;

frequency += ((int)valifreq[2]-48)\*10;

frequency += ((int)valifreq[3]-48);

// tähän radioiinti

if(frequency > 875 && frequency < 1080){

fona.FMradio(true, FONA\_EXTAUDIO);

fona.tuneFMradio(frequency);

fona.setFMVolume(6);

}

else

{

//fail!

}

}

//back

else if( x > -1 && x < 240 && y > 282 && y < 300 )

{

fona.FMradio(false);

touchTila=0;

ScreenState();

}

}

void LoadRadioMenu(){

nappaimisto='0';

tft.setTextSize(3);

tft.setCursor(2,35);

tft.println("Give Freq");

PrintNumber(75);

for(int i=0;i<10;i++)

{

tft.setCursor( 30 + ( i - ( i / 5 ) \* 5 ) \* 40 , 130 + ( i / 5 ) \* 40 );

tft.print(char(nappaimisto+i));

}

tft.setCursor(85,230);

tft.print("DELETE");

tft.drawLine(0, 220, 240, 220, ILI9341\_BLACK);

tft.setCursor(50,260);

tft.print("Tune and play");

tft.drawLine(0, 252, 240, 252, ILI9341\_BLACK);

tft.setCursor(95,290);

tft.print("BACK");

tft.drawLine(0, 282, 240, 282, ILI9341\_BLACK);

}

void TouchControlGiveNumber(int x,int y){

//numbers

if( x > 15 && x < 221 && y > 120 && y < 200 )

{

if(pointteri<12){

PhoneNumber[pointteri]=nappaimisto + ( ( y - 125 ) / 40 ) \* 5 + ( x - 15 ) / 40;

pointteri++;

PrintNumber(75);

}

}

//delete

else if( x > -1 && x < 240 && y > 220 && y < 252 )

{

if(pointteri>0)

{

pointteri--;

PhoneNumber[pointteri]=' ';

for(int i=(1+pointteri)\*12;i>=pointteri\*12;i--){

tft.drawLine(i, 75, i, 89, ILI9341\_WHITE);

}

PrintNumber(75);

}

}

//proceed

else if( x > -1 && x < 240 && y > 252 && y < 282 )

{

if(touchTila==1)

touchTila=2;

if(touchTila==4)

touchTila=5;

ScreenState();

}

//back

else if( x > -1 && x < 240 && y > 282 && y < 300 )

{

touchTila=0;

ScreenState();

}

}

void LoadMessageRead()

{

NewMessage=0;

getTxtMessage();

PrintNumber(0);

PrintMessage(14);

tft.drawLine(0, 252, 240, 252, ILI9341\_BLACK);

tft.setCursor(80,260);

tft.print("ANSWER");

tft.drawLine(0, 282, 240, 282, ILI9341\_BLACK);

tft.setCursor(95,290);

tft.print("BACK");

}

void TouchControlReadMessage(int x, int y)

{

if( x > -1 && x < 240 && y > 252 && y < 282 )

{

touchTila=2;

ScreenState();

}

else if( x > -1 && x < 240 && y > 282 && y < 300 )

{

touchTila=0;

ScreenState();

}

}

void getTxtMessage(){

boolean isMessage=true;

int i=1;

while(!fona.getSMSSender(i, PhoneNumber, 20) && isMessage)

{

if(i==100)

{

isMessage=false;

}

i++;

}

if(isMessage)

{

uint16\_t smslen;

fona.readSMS(i, textMessage, 250, &smslen);

fona.deleteSMS(i);

numberofsms--;

}

}

void PrintNumber(int height){

tft.setTextSize(2);

tft.setCursor(0,height);

tft.print(PhoneNumber);

}

void printStatuses(){

tft.fillRect(50, 0, 36, 15, ILI9341\_WHITE);

uint16\_t vbat;

if (! fona.getBattPercent(&vbat)) {

akku=0;

}

else {

//akku=100;

akku=vbat;

}

tft.setTextSize(2);

tft.setCursor(1,0);

tft.print("bat:");

tft.setCursor(50,0);

tft.print(akku);

}

void LoadMainMenu(){

printStatuses();

tft.setTextSize(3);

tft.setCursor(2,45);

tft.print("Send SMS");

tft.setCursor(2,77);

if(NewMessage)

tft.print("Read SMS NEW!");

else

tft.print("Read SMS");

tft.setCursor(2,111);

tft.print("Make Call");

tft.setCursor(2,145);

tft.print("FM Radio");

tft.drawLine(0, 37, 240, 37, ILI9341\_BLACK);

tft.drawLine(0, 71, 240, 71, ILI9341\_BLACK);

tft.drawLine(0, 105, 240, 105, ILI9341\_BLACK);

tft.drawLine(0, 139, 240, 139, ILI9341\_BLACK);

tft.drawLine(0, 173, 240, 173, ILI9341\_BLACK);

}

void TouchControlMainMenu(int x, int y){

if( x > -1 && x < 241 && y > 37 && y < 71 )

{

//todo

touchTila = 1;

ScreenState();

}

else if( x > -1 && x < 241 && y > 71 && y < 105 )

{

// this should contain the logic for reading the message & stuff

if(fona.getNumSMS()>0){

touchTila = 3;

ScreenState();

}

}

else if( x > -1 && x < 241 && y > 105 && y < 140 )

{

// make a call

touchTila = 4;

ScreenState();

}

else if( x > -1 && x < 241 && y > 140 && y <173 )

{

// radio

touchTila = 7;

ScreenState();

}

}

void TouchControlKeyboard(int x,int y){

if( x > -1 && x < 241 && y > startPoint\_Y && y < 90 + startPoint\_Y )

{

textMessage[pointteri]=nappaimisto + ( ( y - startPoint\_Y ) / 30 ) \* 8 + x / 30;

pointteri++;

PrintMessage(14);

}

else if( x > -1 && x < 61 && y > 90 + startPoint\_Y && y < 120 + startPoint\_Y )

{

textMessage[pointteri]=nappaimisto + 24 + x / 30;

pointteri++;

PrintMessage(14);

}

else if( x > 75 && x < 145 && y > 100 + startPoint\_Y && y < 125 + startPoint\_Y )

{

if(vaihto[0]=='a')

{

SetUpSmallText();

NappaimistoTulostus(26);

}

else

{

SetUpLargeText();

NappaimistoTulostus(26);

}

}

else if( x > 164 && x < 226 && y > 100 + startPoint\_Y && y < 125 + startPoint\_Y )

{

if(vaihto[1]=='1')

{

SetUpNumbers();

NappaimistoTulostus(26);

}

else

{

SetUpLargeText();

NappaimistoTulostus(26);

}

}

//spacebar ja delete toiminta

else if( x > -1 && x < 150 && y > 120 + startPoint\_Y && y < 150 + startPoint\_Y )

{

textMessage[pointteri]=' ';

pointteri++;

}

else if( x > 160 && x < 230 && y > 120 + startPoint\_Y && y < 150 + startPoint\_Y )

{

if(pointteri!=0)

{

pointteri--;

textMessage[pointteri]=' ';

for(int i=(1+pointteri)\*12;i>=pointteri\*12;i--){

tft.drawLine( i - ( ( i / 240 ) \* 240 ), 14 + ( pointteri / 20 )\*16 , i - (( i / 240 ) \* 240) , 28 + ( pointteri / 20 )\*16 , ILI9341\_WHITE);

}

PrintMessage(14);

}

}

//pitää tarkistaaa koordinaatit

else if( x > -1 && x < 100 && y > 150 + startPoint\_Y && y < 180 + startPoint\_Y )

{

touchTila = 1;

ScreenState();

}

else if( x > 101 && x < 200 && y > 150 + startPoint\_Y && y < 180 + startPoint\_Y )

{

//This should contain the logic for sending the message!

SendTextMessage();

touchTila = 0;

ScreenState();

}

}

void EmptyMessage(){

for(int i=0;i<160;i++)

textMessage[i]=0;

}

void EmptyNumber(){

for(int i=0;i<20;i++)

PhoneNumber[i]=0;

}

void SendTextMessage(){

tft.fillScreen(ILI9341\_WHITE);

tft.setTextSize(3);

tft.setCursor(80,260);

tft.print("Sending");

fona.sendSMS(PhoneNumber, textMessage);

}

void PrintMessage(int height){

tft.setTextSize(2);

tft.setCursor(0,height);

tft.print(textMessage);

}

void NappaimistoTulostus(int amount) {

//tft.fillScreen(ILI9341\_WHITE);

tft.fillRect(0, startPoint\_Y-5, 240, startPoint\_Y+150, ILI9341\_WHITE);

tft.setTextSize(3);

int linty=startPoint\_Y;

int i=0;

for(;i<amount;i++)

{

//saattaa joutua laskemaan osissa.

tft.setCursor( ( i - ( ( i / 8 ) \* 8 ) ) \* SideLength + x\_offset , SideLength \* ( i / 8 ) + linty + y\_offset);

tft.print(char(nappaimisto+i));

}

linty += SideLength \* ( i / 8 );

// voi sisältää virheen jos toisenkin

for(int index=0;index<3;index++)

{

tft.setCursor( ( 4 + index ) \* ( SideLength - 10 ) + x\_offset , linty + y\_offset );

tft.print(char(vaihto[0]+index));

}

for(int index=0;index<3;index++)

{

tft.setCursor( ( 8 + index ) \* ( SideLength - 10 ) + x\_offset , linty + y\_offset );

tft.print(char(vaihto[1]+index));

}

linty+=SideLength;

tft.setCursor(x\_offset,linty+y\_offset);

tft.print("SPACEBAR DEL");

linty+=SideLength;

tft.setCursor(x\_offset,linty+y\_offset);

tft.print("BACK SEND");

}

void SetUpLargeText(){

nappaimisto='A';

vaihto[0]='a';

vaihto[1]='1';

}

void SetUpSmallText(){

nappaimisto='a';

vaihto[0]='A';

vaihto[1]='1';

}

void SetUpNumbers(){

nappaimisto='!';

vaihto[0]='a';

vaihto[1]='A';

}