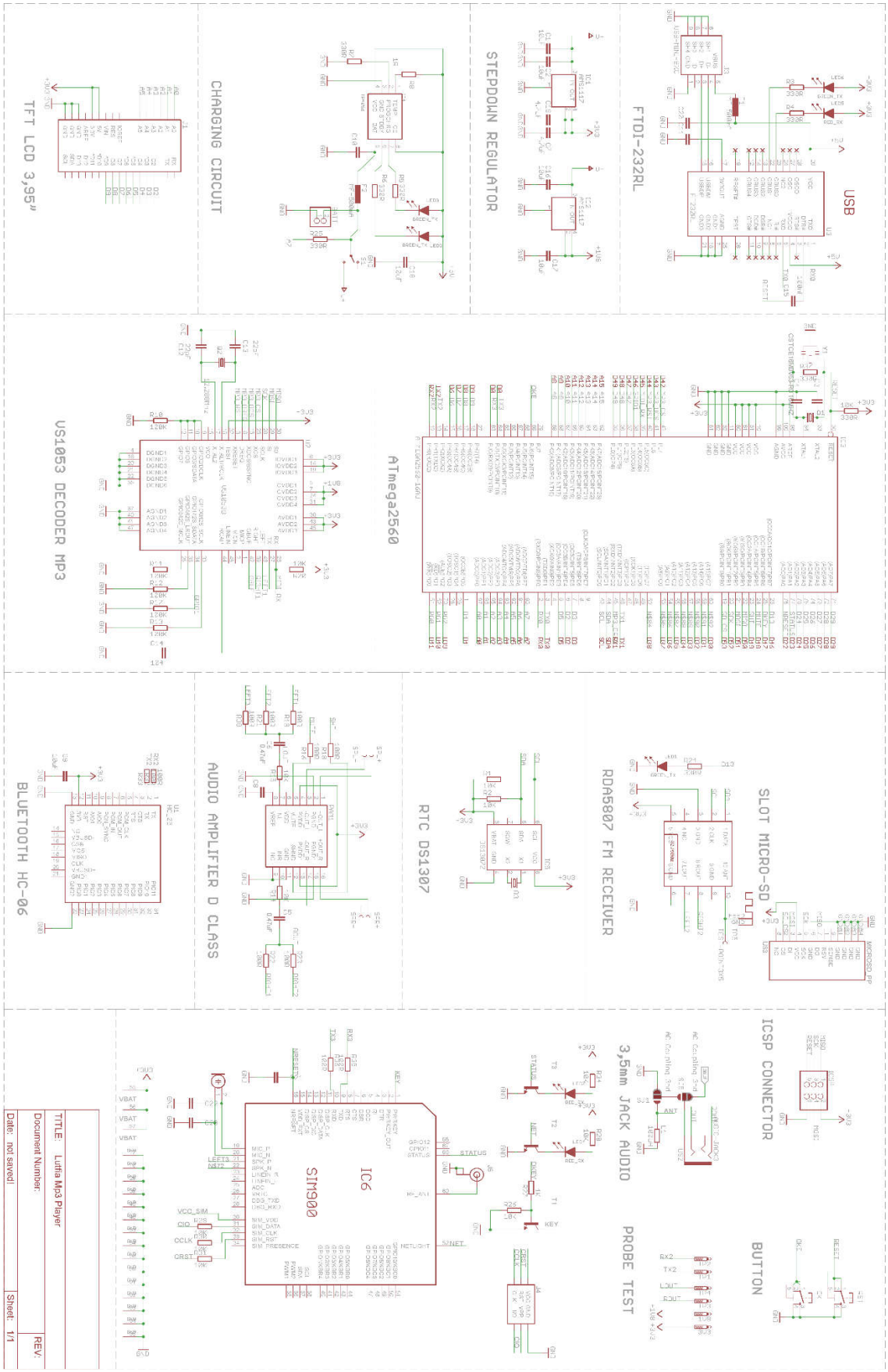
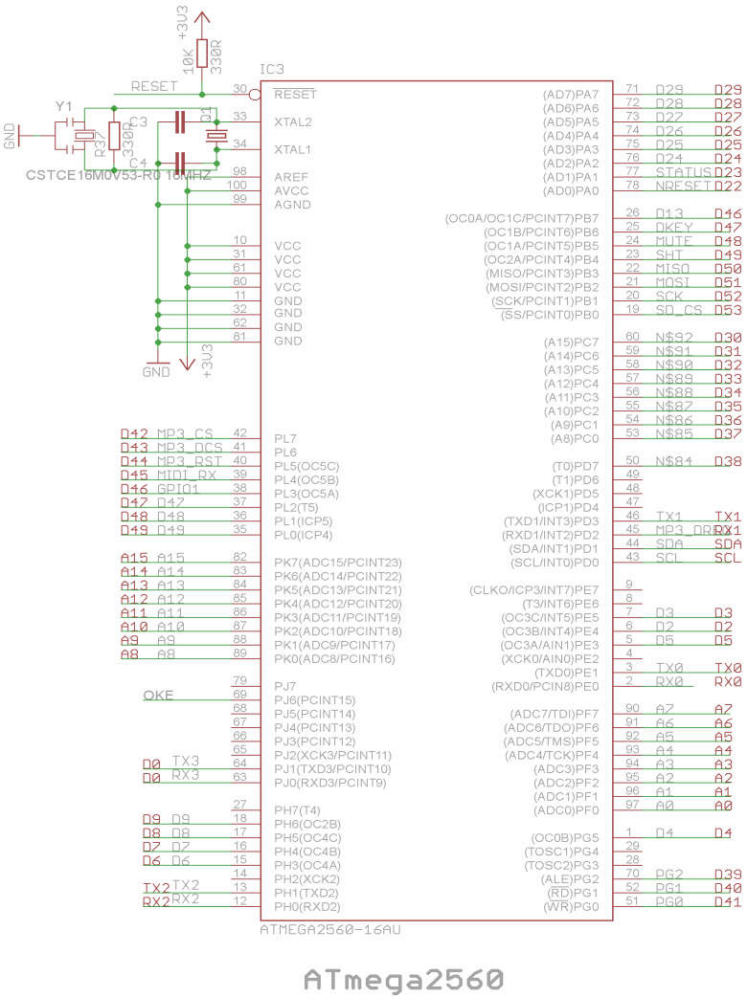
LAMPIRAN-LAMPIRAN

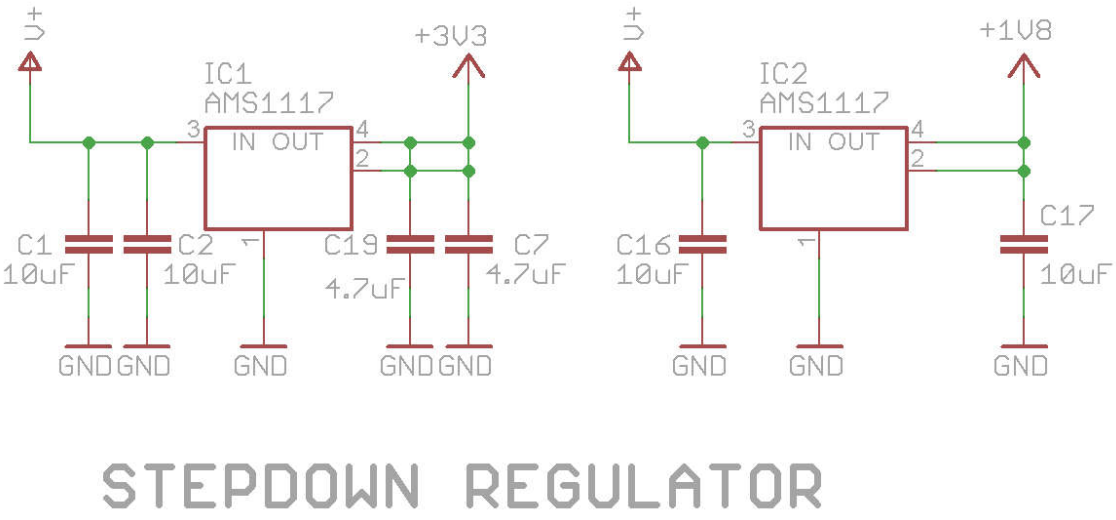
1. Skematik Keseluruhan



2. ATMega2560 Circuit

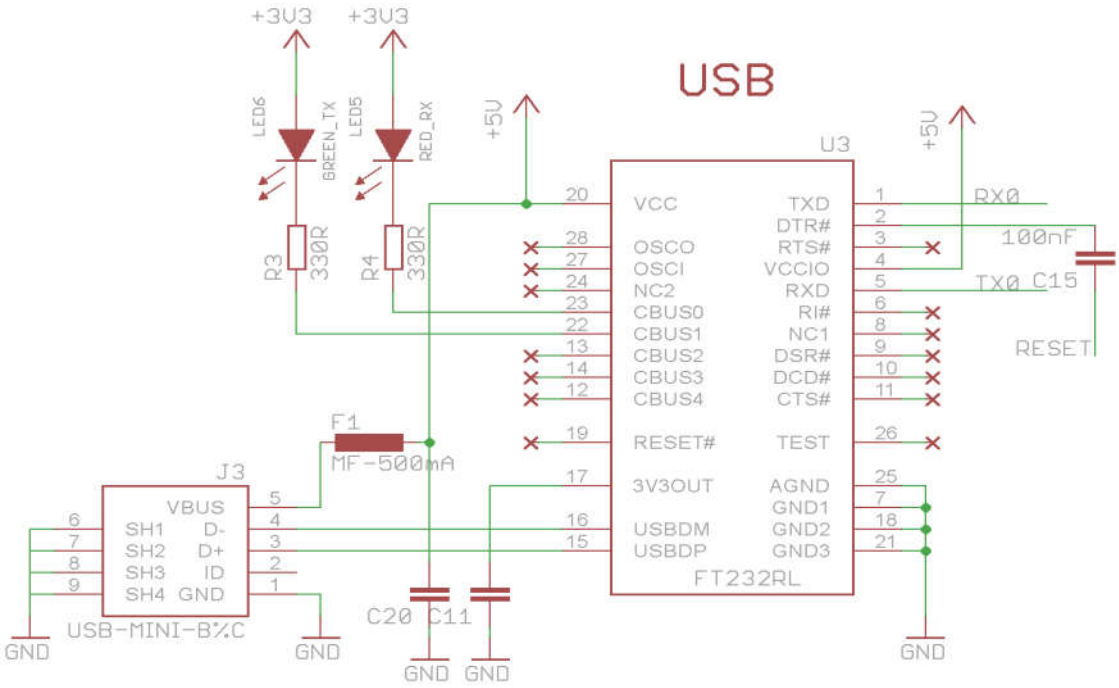


3. Catu Daya 3,3V dan 1,8V DC



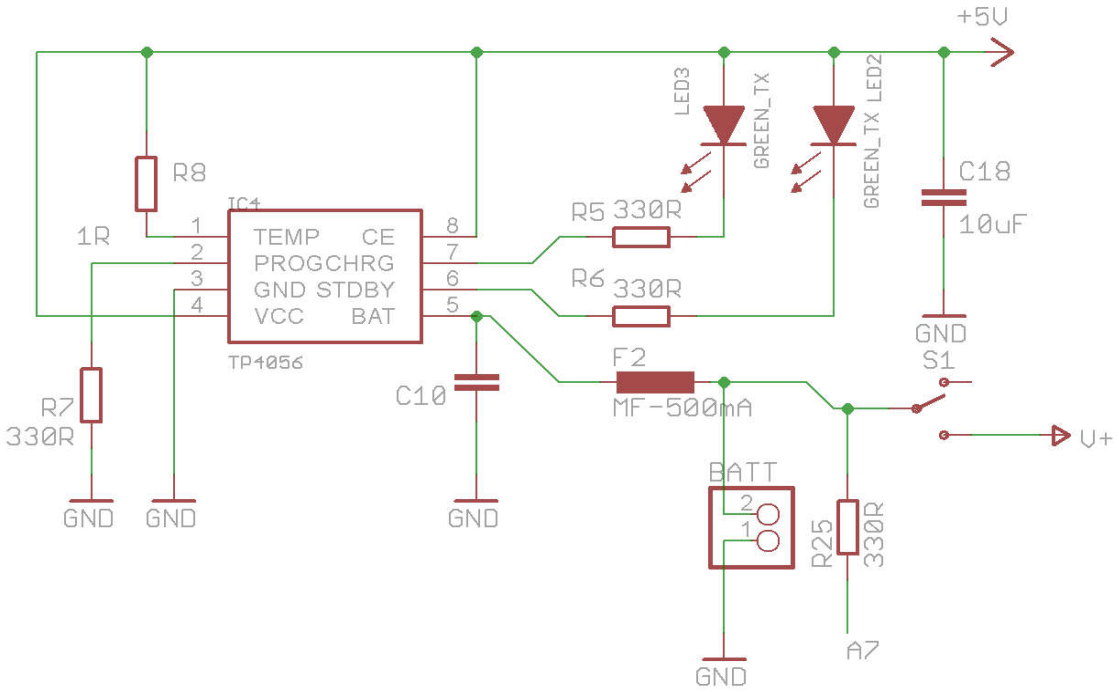
STEPDOWN REGULATOR

4. FTDI232 USB to TTL



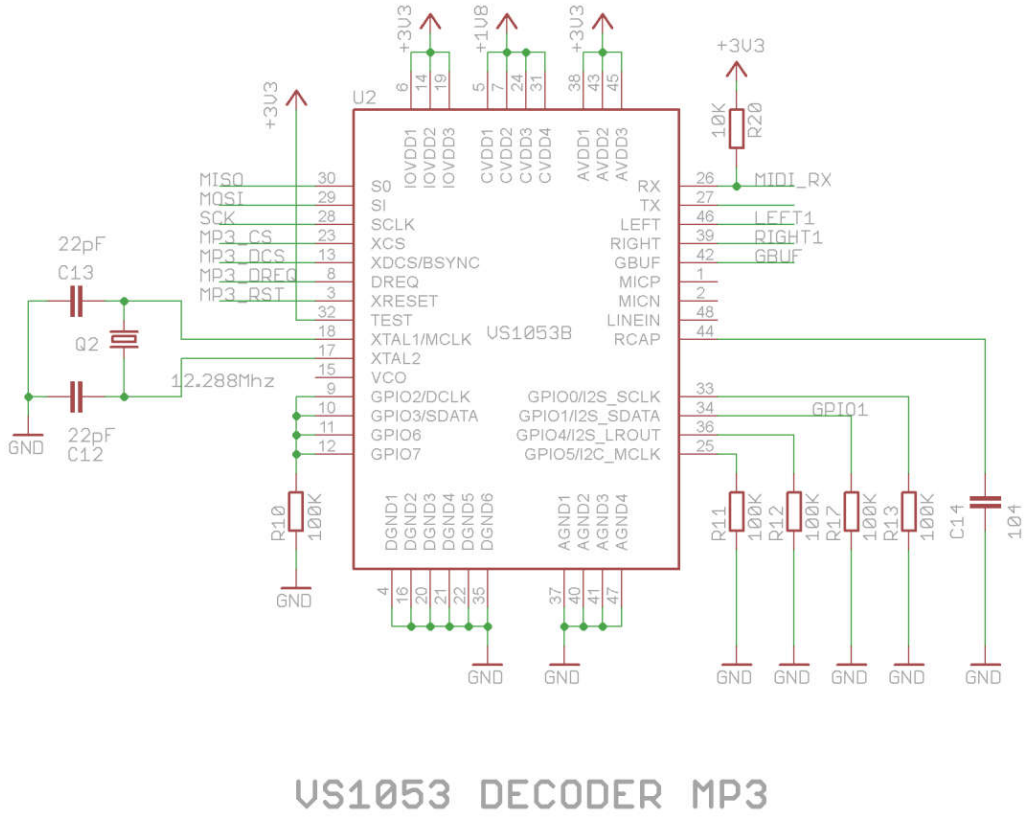
FTDI-232RL

5. TP4056 *Charger Lithium Battery*



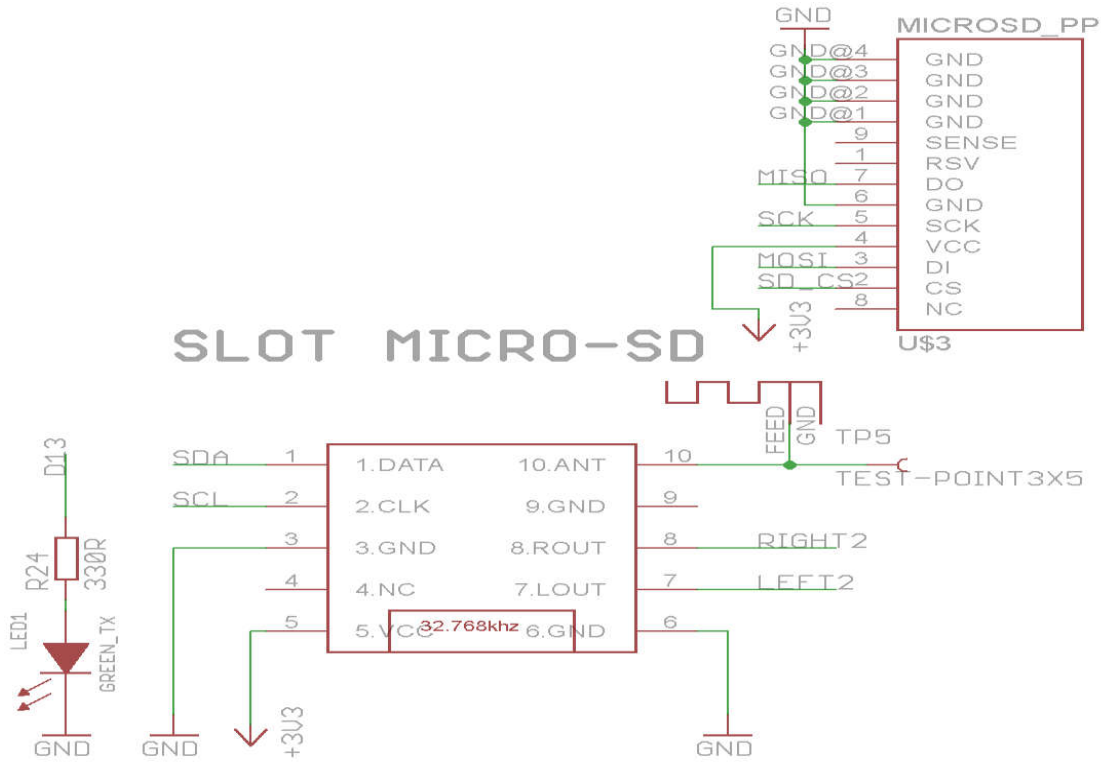
CHARGING CIRCUIT

6. **VS053** ***Mp3 Player Circuit***



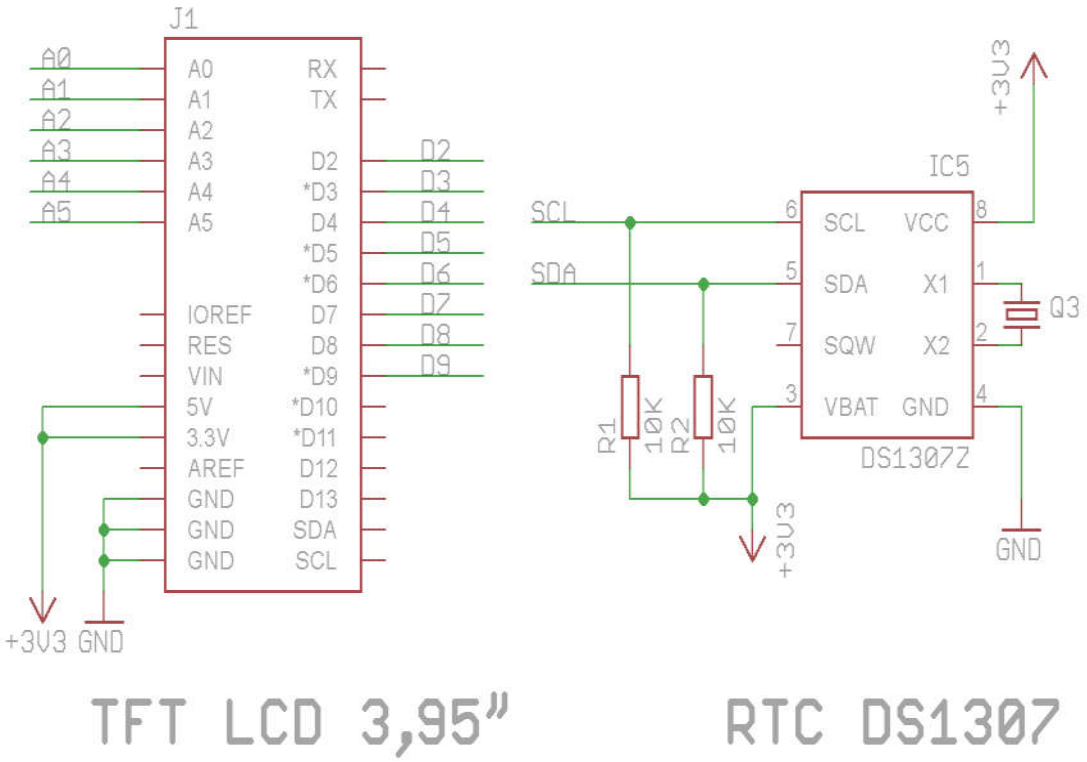
VS1053 DECODER MP3

7. **RDA5807** ***FM Radio Player Circuit*** **dan** ***MicroSD***

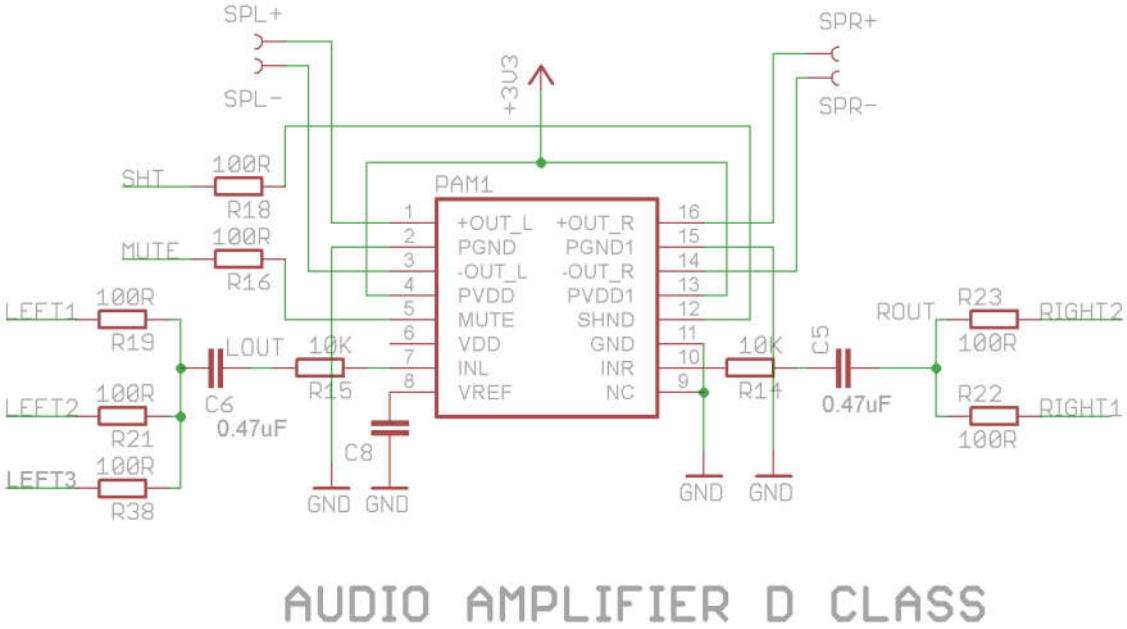


RDA5807 FM RECEIVER

8. LCD TFT 3,95 *Inch* dan RTC DS1307



9. PAM8403 *Audio Amplifier*



LAMPIRAN PROGRAM

```
/*====================== Header/Library==============================*/
#include <Adafruit_GFX.h>
#include <gfxfont.h>

#include <radio.h>
#include <RDA5807M.h>

#include <UTFTGLUE.h>
#include <TouchScreen.h>
#include <avr/pgmspace.h>
#include <SD.h>
#include <SPI.h>
#include <Wire.h>
#include <DS1307.h>
#include <EEPROM.h>

/*====================================================================*/

/*====================== DEFINISI-DEFINISI==============================*/
#define YP A1
#define YM 7
#define XM A2
#define XP 6
#define SD_CS 53
#define BUFFPIXEL 20
#define FIX_BAND     RADIO_BAND_FM
#define MINPRESSURE 3
#define MAXPRESSURE 1000
#define TOUCH_ORIENTATION  PORTRAIT
#define BUFFPIXEL 20

#define shutDwn   10
#define ampliMute 11
#define gsmPower  12
/*====================================================================*/
RDA5807M fm;    // Create an instance of Class for RDA5807M Chip

UTFTGLUE lutfia(0x9488, A2, A1, A3, A4, A0);
//UTFT lutfia(0x9488, A2, A1, A3, A4, A0);
//TouchScreen myTouch(6, A1, A2, 7, 300);
TouchScreen myTouch(XP, YP, XM, YM, 300);
TSPoint tp;

uint16_t TS_LEFT = 920;
uint16_t TS_RT  = 150;
uint16_t TS_TOP = 940;
uint16_t TS_BOT = 120;

int dispx, dispy, text_y_center, swapxy;
char buf[13];
int rtc[7];
char hour[2], minute[2];
```

```
extern const uint8_t
radio[],blue[],music[],picture[],timer[],setting[],calendar[],notes[],calcula
tor[],signal[],micro[],lcd[],logo[];
extern const uint8_t previous[],next[], power[],nextp[],
previousp[],playp[],pause[],speaker[],bass[],mono[],stereo[],favorite[],shuff
le[],repeat[],exitt[],bird01[];
const uint8_t *item_menu[12] = { radio, blue, music, picture, timer, setting,
calendar, notes, calculator, signal, micro, lcd};
char *itemImages[10]=
{"images1.bmp","images2.bmp","images3.bmp","images4.bmp","images5.bmp","image
s6.bmp","images7.bmp","images8.bmp","images9.bmp","images10.bmp"};
char *itemsDay[8]={"      ","Senin ","Selasa","Rabu  ","Kamis
","Jum'at","Sabtu ","Minggu"};

extern uint8_t SmallFont[];
extern uint8_t BigFont[];
extern uint8_t SevenSegNumFont[];


int xG=70;
int xB=70;

int xP = 319;
int yP = 100;
int yB = 30;

/*======================== FUNGSI INISIALISASI=========================*/
void setup() {
  Serial.begin(9600);
  pinMode(A0, OUTPUT);
  pinMode(shutDwn, OUTPUT);
  pinMode(ampliMute, OUTPUT);
  pinMode(gsmPower,OUTPUT);
  digitalWrite(gsmPower,HIGH);
  //setRTC(12,56,4,16,9,16);

  lutfia.InitLCD(TOUCH_ORIENTATION);
  lutfia.clrScr();
  lutfia.setFont(SmallFont);
  #if (USE_UTOUCH)
    myTouch.InitTouch(TOUCH_ORIENTATION);
  #endif

  lutfia.drawBitmap(60,200,logo,200,80, lutfia.color565(255,255,255));
  delay(2000);
  progmemPrint(PSTR("Initializing SD card..."));
  if (!SD.begin(SD_CS)) {
    progmemPrintln(PSTR("failed!"));
    return;
  }
  progmemPrintln(PSTR("OK!"));
  lutfia.clrScr();

}

/*=============================================================================*/
```

```
/*============================ FUNGSI SET RTC============================*/
void setRTC(int jam,int mnt,int day,int tgl,int bln,int thn){
  RTC.stop();
  RTC.set(DS1307_SEC,1);
  RTC.set(DS1307_MIN,mnt);
  RTC.set(DS1307_HR,jam);
  RTC.set(DS1307_DOW,day);
  RTC.set(DS1307_DATE,tgl);
  RTC.set(DS1307_MTH,bln);
  RTC.set(DS1307_YR,thn);
  RTC.start();
}
/*============================================================================*/

void rtc_set_time(int jam, int menit, int hari, int tanggal, int bulan, int
tahun){
  RTC.stop();
  RTC.set(DS1307_SEC,1);
  RTC.set(DS1307_HR, jam);
  RTC.set(DS1307_MIN, menit);
  RTC.set(DS1307_MIN, hari);
  RTC.set(DS1307_MIN, tanggal);
  RTC.set(DS1307_MIN, bulan);
  RTC.set(DS1307_MIN, tahun);
  RTC.start();
}

void setTime(){
  uint16_t xpos=0,ypos=0,dday,statDay;
  char jam=0,menit=0;
  int tanggal=0,bulan=0,tahun=0;
  char Strjam[2],Strmenit[2];
  lutfia.clrScr();
  showRTC();
  menit=rtc[1];
  jam=rtc[2];
  dday=rtc[3];
  tanggal=rtc[4];
  bulan=rtc[5];
  tahun=rtc[6];

  sprintf(Strjam,"%02d",jam);
  sprintf(Strmenit,"%02d",menit);

  //lutfia.fillTriangle(140, 60, 165, 15, 190, 60, lutfia.color565(0,255,0));
  lutfia.setTextColor(lutfia.color565(247,99,14),lutfia.color565(0,0,0));
  lutfia.setTextSize(2);
  lutfia.print("Set Time & Date",10,70,1);

  lutfia.setColor(247,99,14);
  lutfia.drawRoundRect(10,90,(lutfia.getDisplayXSize()-10),270);
                    //
  //============================== Tombol Atas=========================
  lutfia.drawTriangle(80, 130, 100, 105, 120, 130,
lutfia.color565(0,204,106));
  lutfia.drawTriangle(210, 130, 230, 105, 250, 130,
lutfia.color565(0,204,106));
```

```
  //lutfia.drawTriangle(50,100,0,0,0,0,0);//(int16_t x0, int16_t y0,int16_t
x1, int16_t y1, int16_t x2, int16_t y2, uint16_t color)
  lutfia.setTextColor(lutfia.color565(0,0,250),lutfia.color565(0,0,0));
  lutfia.setTextSize(8);
  lutfia.printNumI(jam,50,150,2);
  lutfia.print(":",140,150,2);
  lutfia.printNumI(menit,185,150,2);

  /*================ TOMBOL BAWAH JAM====================================*/
  lutfia.drawTriangle(80, 230, 100, 255, 120, 230,
lutfia.color565(0,204,106));
  lutfia.drawTriangle(210, 230, 230, 255, 250, 230,
lutfia.color565(0,204,106));

  /*================ TOMBOL BAWAH MENIT==================================*/
  lutfia.setColor(247,99,14);
  lutfia.drawRoundRect(10,280,(lutfia.getDisplayXSize()-10),400);

  /*================== MENAMPILKAN KALENDER=============================*/
  lutfia.drawTriangle(145, 315, 175, 315, 160, 295,
lutfia.color565(0,204,106));

  lutfia.setTextColor(lutfia.color565(254,0,86),lutfia.color565(0,0,0));
  lutfia.setTextSize(3);
  lutfia.print(itemsDay[rtc[3]],20,330,2);
  lutfia.printNumI(tanggal,130,330,2);
  lutfia.print("/",165,330,2);
  lutfia.printNumI(bulan,185,330,2);
  lutfia.print("/",220,330,2);
  lutfia.printNumI(tahun,240,330,2);
  lutfia.drawTriangle(145, 370, 175, 370, 160, 390,
lutfia.color565(0,204,106));

  lutfia.setColor(247,99,14);
  lutfia.drawRoundRect(100,410,(lutfia.getDisplayXSize()-100),450);
  lutfia.setTextColor(lutfia.color565(247,99,14),lutfia.color565(0,0,0));
  lutfia.print("SAVE",125,420,2);
  while(1){
    showRTC();
    readResistiveTouch();

    lutfia.setTextColor(lutfia.color565(0,0,250),lutfia.color565(0,0,0));
    lutfia.setTextSize(8);
    lutfia.printNumI(jam,50,150,2);
    if(jam<10){
      lutfia.print("0",50,150,2);
      lutfia.printNumI(jam,95,150,2);
    }
    lutfia.print(":",140,150,2);
    lutfia.printNumI(menit,185,150,2);
    if(menit<10){
      lutfia.print("0",183,150,2);
      lutfia.printNumI(menit,230,150,2);
    }

    //lutfia.setTextColor(lutfia.color565(254,0,86),lutfia.color565(0,0,0));
```

```
    if((xpos>=35 && xpos<=120) && (ypos>=355 && ypos<=380)){

lutfia.setTextColor(lutfia.color565(255,255,255),lutfia.color565(0,0,0));
      lutfia.setTextSize(3);
      lutfia.print(itemsDay[dday],20,330,2);
      delay(200);
      statDay=1;
    }

    if((xpos>=135 && xpos<=160) && (ypos>=355 && ypos<=380)){

lutfia.setTextColor(lutfia.color565(255,255,255),lutfia.color565(0,0,0));
      lutfia.setTextSize(3);
      lutfia.printNumI(tanggal,130,330,2);
      delay(200);
      statDay=2;
    }

    if((xpos>=200 && xpos<=220) && (ypos>=355 && ypos<=380)){

lutfia.setTextColor(lutfia.color565(255,255,255),lutfia.color565(0,0,0));
      lutfia.setTextSize(3);
      lutfia.printNumI(bulan,185,330,2);
      delay(200);
      statDay=3;
    }


    if((xpos>=245 && xpos<=290) && (ypos>=355 && ypos<=380)){

lutfia.setTextColor(lutfia.color565(255,255,255),lutfia.color565(0,0,0));
      lutfia.setTextSize(3);
      lutfia.printNumI(tahun,240,330,2);
      delay(200);
      statDay=4;
    }


    lutfia.setTextColor(lutfia.color565(254,0,86),lutfia.color565(0,0,0));
    lutfia.setTextSize(3);
    lutfia.print(itemsDay[dday],20,330,2);
    lutfia.printNumI(tanggal,130,330,2);
    lutfia.printNumI(bulan,185,330,2);
    lutfia.printNumI(tahun,240,330,2);

    xpos = map(tp.x, TS_LEFT, TS_RT, 0, lutfia.getDisplayXSize());
    ypos = map(tp.y, TS_TOP, TS_BOT, 0, lutfia.getDisplayYSize());

    lutfia.setTextColor(lutfia.color565(255,255,255),lutfia.color565(0,0,0));
    lutfia.setTextSize(1.5);
    lutfia.printNumI(xpos, 135,10,10);
    lutfia.printNumI(ypos, 180,10,10);

    /*==================== SETTING ATAS JAM
================================*/
    if((xpos>=55 && xpos <=175) && (ypos>=145 && ypos<=175) &&
tp.z>MINPRESSURE){
```

```
      jam++;
      lutfia.setColor(247,99,14);
      lutfia.drawRoundRect(75,100,125,135);
      delay(100);
      lutfia.setColor(0,0,0);
      lutfia.drawRoundRect(75,100,125,135);

      if(jam>23)jam=0;

    }

/*=============================================================================*/
    /*================== SETTING ATAS MENIT============================*/
    if((xpos>=235 && xpos<=275) && (ypos>=145 && ypos<=175)){
      menit++;
      lutfia.setColor(247,99,14);
      lutfia.drawRoundRect(205,100,255,135);
      delay(100);
      lutfia.setColor(0,0,0);
      lutfia.drawRoundRect(205,100,255,135);

      if(menit>59)menit=0;
    }

/*=============================================================================*/
    /*======================== SETTING BAWAH JAM ========================*/
    if((xpos>=55 && xpos <=175) && (ypos>=250 && ypos<=290)){
      jam--;
      lutfia.setColor(247,99,14);
      lutfia.drawRoundRect(75,225,125,260);
      delay(100);
      lutfia.setColor(0,0,0);
      lutfia.drawRoundRect(75,225,125,260);

      if(jam<0)jam=23;

    }

/*=============================================================================*/

    /*======================== SETTING BAWAH MENIT ======================*/
    if((xpos>=235 && xpos<=275) && (ypos>=250 && ypos<=290)){
      menit--;
      lutfia.setColor(247,99,14);
      lutfia.drawRoundRect(205,225,255,260);
      delay(100);
      lutfia.setColor(0,0,0);
      lutfia.drawRoundRect(205,225,255,260);
      if(menit<0)menit=59;
    }

    /*========================= SETTING HARI ============================*/
    if((xpos>=160 && xpos<=190) && (ypos>=320 && ypos<=345) && statDay==1){
          lutfia.fillTriangle(145, 315, 175, 315, 160, 295,
lutfia.color565(0,204,106));
          dday++;
          delay(100);
```

```
          lutfia.setColor(0,0,0);
          lutfia.fillRect(140, 290, 180, 320);
          lutfia.drawTriangle(145, 315, 175, 315, 160, 295,
lutfia.color565(0,204,106));
          if(dday>7){
            dday=1;
          }
     }

     if((xpos>=160 && xpos<=190) && (ypos>=390 && ypos<=410) && statDay==1){
          lutfia.fillTriangle(145, 370, 175, 370, 160, 390,
lutfia.color565(0,204,106));
          dday--;
          delay(100);
          lutfia.setColor(0,0,0);
          lutfia.fillRect(140, 365, 180, 395);
          lutfia.drawTriangle(145, 370, 175, 370, 160, 390,
lutfia.color565(0,204,106));
          if(dday<1){
            dday=7;
          }
     }

      /*========================= SETTING TANGGAL   ====================*/
      if((xpos>=160 && xpos<=190) && (ypos>=320 && ypos<=345) && statDay==2){
          lutfia.fillTriangle(145, 315, 175, 315, 160, 295,
lutfia.color565(0,204,106));
          tanggal++;
          delay(500);
          lutfia.setColor(0,0,0);
          lutfia.fillRect(140, 290, 180, 320);
          lutfia.drawTriangle(145, 315, 175, 315, 160, 295,
lutfia.color565(0,204,106));
          if(tanggal>31){
            tanggal=0;
          }
     }

     if((xpos>=160 && xpos<=190) && (ypos>=390 && ypos<=410) && statDay==2){
          lutfia.fillTriangle(145, 370, 175, 370, 160, 390,
lutfia.color565(0,204,106));
          tanggal--;
          delay(100);
          lutfia.setColor(0,0,0);
          lutfia.fillRect(140, 365, 180, 395);
          lutfia.drawTriangle(145, 370, 175, 370, 160, 390,
lutfia.color565(0,204,106));
          if(tanggal<0){
            tanggal=31;
          }

     }

     if((xpos>=160 && xpos<=190) && (ypos>=320 && ypos<=345) && statDay==3){
          lutfia.fillTriangle(145, 315, 175, 315, 160, 295,
lutfia.color565(0,204,106));
          bulan++;
```

```
          delay(100);
          lutfia.setColor(0,0,0);
          lutfia.fillRect(140, 290, 180, 320);
          lutfia.drawTriangle(145, 315, 175, 315, 160, 295,
lutfia.color565(0,204,106));
          if(bulan>12){
            bulan=0;
          }
     }

     if((xpos>=160 && xpos<=190) && (ypos>=390 && ypos<=410) && statDay==3){
          lutfia.fillTriangle(145, 370, 175, 370, 160, 390,
lutfia.color565(0,204,106));
          bulan--;
          delay(100);
          lutfia.setColor(0,0,0);
          lutfia.fillRect(140, 365, 180, 395);
          lutfia.drawTriangle(145, 370, 175, 370, 160, 390,
lutfia.color565(0,204,106));
          if(bulan<0){
            bulan=12;
          }

     }

     if((xpos>=160 && xpos<=190) && (ypos>=320 && ypos<=345) && statDay==4){
          lutfia.fillTriangle(145, 315, 175, 315, 160, 295,
lutfia.color565(0,204,106));
          tahun++;
          delay(100);
          lutfia.setColor(0,0,0);
          lutfia.fillRect(140, 290, 180, 320);
          lutfia.drawTriangle(145, 315, 175, 315, 160, 295,
lutfia.color565(0,204,106));
          if(tahun>100){
            tahun=0;
          }
     }

     if((xpos>=160 && xpos<=190) && (ypos>=390 && ypos<=410) && statDay==4){
          lutfia.fillTriangle(145, 370, 175, 370, 160, 390,
lutfia.color565(0,204,106));
          tahun--;
          delay(100);
          lutfia.setColor(0,0,0);
          lutfia.fillRect(140, 365, 180, 395);
          lutfia.drawTriangle(145, 370, 175, 370, 160, 390,
lutfia.color565(0,204,106));
          if(tahun<0){
            tahun=100;
          }

     }


    /*======================= EXIT SETTING================================*/
    if((xpos>=100 && xpos<=230) && (ypos>=425 && ypos<=465)){
```

```
      lutfia.setColor(247,99,14);
      lutfia.fillRoundRect(100,410,(lutfia.getDisplayXSize()-100),450);
      lutfia.setTextSize(3);
      lutfia.setTextColor(lutfia.color565(0,0,0),lutfia.color565(247,99,14));
      lutfia.print("SAVE",125,420,2);
      rtc_set_time(jam,menit,dday, tanggal, bulan, tahun);
      menu();
    }
  }

}

void showRTC(){
  int adcV=0;
  RTC.get(rtc,true);
  adcV= analogRead(A7);
  adcV=adcV/1023*100;
  lutfia.setColor(255,255,255);
  lutfia.drawLine(0,32,319,32);

  lutfia.setTextColor(lutfia.color565(255,255,255),lutfia.color565(0,0,0));
  lutfia.setTextSize(2);

  lutfia.printNumI(rtc[2], 235,10,10);
  lutfia.print(":",260,10,100);
  lutfia.printNumI(rtc[1], 275,10,10);

  lutfia.fillRect(22, 10, 23, 15);

  lutfia.drawRect(24, 4, 52, 22);
  lutfia.setColor(0,255,0);
  lutfia.fillRect(26, 6, 50, 20);
  lutfia.setTextColor(lutfia.color565(255,255,255), lutfia.color565(0,0,0));
  lutfia.setTextSize(2);
  lutfia.printNumI(adcV,58,7);
  lutfia.print("%",98,7);

  lutfia.setColor(255,255,255);
  lutfia.setBackColor(255, 255, 255);
}

void songTimer(int x,int y){
  lutfia.setTextColor(lutfia.color565(255,255,255),lutfia.color565(0, 0, 0));
  lutfia.setTextSize(2);
  lutfia.printNumI(rtc[2], x,y,10);
  lutfia.print(":",x+20,y,1);
  lutfia.printNumI(rtc[1], x+40,y,10);
}

void readResistiveTouch(void){
  tp = myTouch.getPoint();
  pinMode(YP, OUTPUT);
  pinMode(XM, OUTPUT);
  pinMode(XP, OUTPUT);
  pinMode(YM, OUTPUT);
  //digitalWrite(YP, HIGH);
  //digitalWrite(XM, HIGH);
```

```
}

void playFm(){
  uint16_t xpos,ypos,xR=70;
  unsigned int dataFreq=8800,signalFreq;
  uint8_t statSpeaker,statBass, statPower;
  uint8_t currentState1=LOW,currentState2=LOW,currentState3=LOW;
  uint8_t lastState1=LOW,lastState2=LOW,lastState3=LOW;
  uint8_t ledState1=LOW,ledState2=LOW,ledState3=LOW;
  lutfia.clrScr();
  lutfia.setColor(0,79,238);
  lutfia.fillRect(0,33,320,70);
  lutfia.setTextColor(lutfia.color565(255, 255,
255),lutfia.color565(0,79,238));
  lutfia.setTextSize(3);
  lutfia.print("FM RADIO",95,38);
  lutfia.setColor(0,79,238);
  lutfia.fillRoundRect(50, 190, 270, 190);
  lutfia.fillRoundRect(50, 300, 270, 300);
  lutfia.setColor(255,255,255);
  lutfia.fillRect(70, 340, 250, 342);
  lutfia.drawBitmap(35, 80, favorite, 30,30, lutfia.color565(0,79,238));
  lutfia.setColor(0,79,238);
  lutfia.drawRect(150,80,190,110);
  lutfia.setTextColor(lutfia.color565(0,79,238),lutfia.color565(0,0,0));
  lutfia.setTextSize(3);
  lutfia.print("X",163,85);
  lutfia.drawBitmap(265,80, stereo, 30,30, lutfia.color565(0,79,238));
  lutfia.drawBitmap(35, 330, speaker, 30,30, lutfia.color565(255, 255, 255));
  lutfia.drawBitmap(260, 330, bass, 30,30, lutfia.color565(255, 255, 255));
  lutfia.drawBitmap(35, 380, previous, 50,50, lutfia.color565(0,79,238));
  lutfia.drawBitmap(135, 380, power, 70,70, lutfia.color565(255, 255, 255));
  lutfia.drawBitmap(235, 380, next, 50,50, lutfia.color565(0,79,238));
  fm.setBandFrequency(FIX_BAND, dataFreq);
  fm.setVolume(0);
  fm.setMono(false);
  signalFreq=8800;
  while (1) {
    showRTC();
    readResistiveTouch();

    xpos = map(tp.x, TS_LEFT, TS_RT, 0, lutfia.getDisplayXSize());
    ypos = map(tp.y, TS_TOP, TS_BOT, 0, lutfia.getDisplayYSize());
    //if (tp.z < MINPRESSURE || tp.z > MAXPRESSURE) continue;

    lutfia.setTextColor(lutfia.color565(255,255,255),lutfia.color565(0,0,0));
    lutfia.setTextSize(1.5);
    lutfia.printNumI(xpos, 135,10,10);
    lutfia.printNumI(ypos, 180,10,10);

    lutfia.setTextColor(lutfia.color565(255,255,255),lutfia.color565(0,0,0));
    lutfia.setTextSize(5);
    lutfia.printNumI(signalFreq, 90,230,1);

    lutfia.setTextSize(3);
    lutfia.setTextColor(lutfia.color565(255,255,255),lutfia.color565(0,0,0));
```

```
    lutfia.print("MHz", 210,275,1);

    /*============================== SETTING VOLUME ======================*/
    if((ypos>=360) && (ypos<=375) && (xpos>=70) && (xpos<=250) ) {
      xR=xpos; // Stores the X value where the screen has been pressed in to
variable xR
      if (xR<=70) { // Confines the area of the slider to be above 38 pixels
        xR=70;
        fm.setVolume(0);
      }
      if (xR>=250){ /// Confines the area of the slider to be under 310
pixels
        xR=250;
      }
      fm.setVolume(xR/100);
    }

    /*============================== DRAW POSTIONER ======================*/
    int xRC = map(xR,70,310,0,255);
    // Draws the positioners
//    lutfia.setColor(255, 255, 255);
//    lutfia.fillRect(xR,340,(xR+1),345); // Positioner
//    lutfia.setColor(xRC, 0, 0);
//    lutfia.fillRect(71, 341, (xR-1), 344);
//    lutfia.setColor(0, 0, 0);
//    lutfia.fillRect((xR+5), 341, 249, 344);

    lutfia.setColor(0,79,238);
    lutfia.fillRect(xR,336,xR+5,346);

    lutfia.setColor(0,0,0);
    lutfia.fillRect(70, 336, (xR-1), 339);
    lutfia.setColor(0,0,0);
    lutfia.fillRect(70, 343, (xR-1), 350);

    lutfia.setColor(0,79,238);
    lutfia.fillRoundRect(70, 340, (xR-1), 342);

    lutfia.setColor(255, 255, 255);
    lutfia.fillRoundRect((xR+6), 340, 250, 342);

    lutfia.setColor(0,0,0);
    lutfia.fillRect(xR+5, 336, 250, 339);

    lutfia.setColor(0,0,0);
    lutfia.fillRect(xR+5, 343, 250, 350);

    /*======== Button Speaker untuk ON/OFF Power Ampli===*/
    if((xpos>=30 && xpos<=60) && (ypos>=350 && ypos<=380)){
      statSpeaker=HIGH;
    }

    else{
      statSpeaker=LOW;
    }
    currentState1=statSpeaker;
    if(currentState1== HIGH && lastState1==LOW){
```

```
      delay(1);
      if (ledState1 == HIGH){
        lutfia.drawBitmap(35, 330, speaker, 30,30,
lutfia.color565(0,79,238));
        digitalWrite(shutDwn,HIGH);
        ledState1 = LOW;
      }
      else {
        lutfia.drawBitmap(35, 330, speaker, 30,30,
lutfia.color565(255,255,255));
        digitalWrite(shutDwn,LOW);
        ledState1= HIGH;
      }
    }

    /*======== Button Headset untuk ON/OFF Bass Booster===*/
    if((xpos>=260 && xpos<=290) && (ypos>=350 && ypos<=380)){
      statBass=HIGH;
    }

    else{
      statBass=LOW;
    }
    currentState2=statBass;
    if(currentState2== HIGH && lastState2==LOW){
      delay(1);
      if (ledState2 == HIGH){
        lutfia.drawBitmap(260, 330, bass, 30,30, lutfia.color565(0,79,238));
        fm.setBassBoost(true);
        ledState2= LOW;
      }
      else {
        lutfia.drawBitmap(260, 330, bass, 30,30,
lutfia.color565(255,255,255));
        fm.setBassBoost(false);
        ledState2= HIGH;
      }
    }

    /*======== Button Power untuk ON/OFF RDA5807===*/
    if ((xpos>145 && xpos<190) && (ypos>400 && ypos<450)) {
      statPower=1;
    }
    else{
      statPower=0;
    }
    currentState3=statPower;

    if (currentState3 == HIGH && lastState3 == LOW){
      delay(1);//crude form of button debouncing
      if (ledState3 == HIGH){
        lutfia.drawBitmap(135, 380, power, 70,70, lutfia.color565(0,79,238));
        fm.setMute(false);
        ledState3 = LOW;
      }
      else {
```

```
        lutfia.drawBitmap(135, 380, power, 70,70,
lutfia.color565(255,255,255));
        fm.setMute(true);
        ledState3 = HIGH;
      }
    }

    if((xpos>=200 && xpos<=230) && (ypos>=130 && ypos<=160)){
      lutfia.setColor(0,79,238);
      lutfia.fillRect(150,80,190,110);
      lutfia.setTextColor(lutfia.color565(0,0,0),lutfia.color565(0,79,238));
      lutfia.setTextSize(3);
      lutfia.print("X",163,85);
      menu();
    }

    /*========================== PENGURANGAN
FREKUENSI==========================*/
    if ((xpos>50 && xpos<90) && (ypos>400 && ypos<450)) {
      fm.setBandFrequency(FIX_BAND, dataFreq);
      lutfia.drawBitmap(35, 380, previous, 50,50,
lutfia.color565(0,100,104));
      dataFreq=dataFreq-10;
      signalFreq=dataFreq;
      if(EEPROM.read(1)!=signalFreq)
        EEPROM.write(1,signalFreq);

      delay(10);
      lutfia.drawBitmap(35, 380, previous, 50,50, lutfia.color565(0,79,238));

    }

    /*========================== PENAMBAHAN
FREKUENSI==========================*/
    if ((xpos>240 && xpos<290) && (ypos>400 && ypos<450)) {
      fm.setBandFrequency(FIX_BAND, dataFreq);
      lutfia.drawBitmap(235, 380, next, 50,50, lutfia.color565(0,100,104));
      dataFreq=dataFreq+10;
      signalFreq=dataFreq;
      if(EEPROM.read(1)!=signalFreq)
        EEPROM.write(1,signalFreq);
      delay(10);
      lutfia.drawBitmap(235, 380, next, 50,50, lutfia.color565(0,79,238));
    }
  }
}

void playMp3(){
  uint16_t xpos,ypos,xR=80;
  uint8_t z;
  uint8_t statPlay;
  uint8_t currentState=LOW,lastState=LOW;
  uint8_t ledState=LOW;
  uint8_t volMp3;
  lutfia.clrScr();
  lutfia.setColor(255,119,0);
  lutfia.fillRect(0,33,320,70);
```

```
  lutfia.setTextColor(lutfia.color565(255, 255,
255),lutfia.color565(255,119,0));
  lutfia.setTextSize(3);
  lutfia.print("MP3 PLAYER",80,38);

  /*======================== DRAW ICON ATAS ============================*/
  lutfia.drawBitmap(35, 80, shuffle, 30,30, lutfia.color565(255,255,255));
  lutfia.drawBitmap(115, 80, repeat, 30,30, lutfia.color565(255,255,255));
  lutfia.drawBitmap(175, 80, favorite, 30,30, lutfia.color565(255,255,255));
  lutfia.drawBitmap(235, 80, exitt, 30,30, lutfia.color565(255,255,255));


  lutfia.setColor(255, 255, 255);
  lutfia.fillRoundRect(270, 320, 50, 190);
  lutfia.drawBitmap(40,370, speaker, 30,30, lutfia.color565(255,255,255));
  lutfia.setColor(255,255,255);

  lutfia.fillRect(80, 382, 230, 384); // R - Slider


  lutfia.drawLine(20,360,300,360);
  lutfia.setColor(255,119,0);
  lutfia.fillCircle(z+20,360,5);
  songTimer(240,330);

  lutfia.drawBitmap(lutfia.getDisplayXSize()-300, 400, previousp, 50,50,
lutfia.color565(255,119,0));
  lutfia.drawBitmap(130, 400, playp, 70,70, lutfia.color565(255,119,0));
  lutfia.drawBitmap(lutfia.getDisplayXSize()-70, 400, nextp, 50,50,
lutfia.color565(255,119,0));
  showRTC();

  while (1) {
    readResistiveTouch();
    showRTC();
    xpos = map(tp.x, TS_LEFT, TS_RT, 0, lutfia.getDisplayXSize());
    ypos = map(tp.y, TS_TOP, TS_BOT, 0, lutfia.getDisplayYSize());

    //if (tp.z < MINPRESSURE || tp.z > MAXPRESSURE) continue;

    lutfia.setTextColor(lutfia.color565(255,255,255),lutfia.color565(0,0,0));
    lutfia.setTextSize(1.5);
    lutfia.printNumI(xpos, 135,10,10);
    lutfia.printNumI(ypos, 180,10,10);

    lutfia.setTextColor(lutfia.color565(255,255,255),lutfia.color565(0,0,0));
    lutfia.setTextSize(2);
    lutfia.printNumI(volMp3,250,380);

    if((ypos>=400) && (ypos<=410) && (xpos>=80) && (xpos<=230) ) {
      xR=xpos; // Stores the X value where the screen has been pressed in to
variable xR
      if (xR<=60) { // Confines the area of the slider to be above 38 pixels
        xR=80;
        //fm.setVolume(0);
      }
```

```
      if (xR>=230){ /// Confines the area of the slider to be under 310
pixels
        xR=230;
      }
      volMp3=(xR/10);
    }
    int xRC = map(xR,60,230,0,255);

    lutfia.setColor(255,119,0);
    //lutfia.fillRoundRect(xR,382,(xR+1),387); // Positioner
    lutfia.fillRect(xR,378,xR+5,388);

    lutfia.setColor(0,0,0);
    lutfia.fillRect(80, 372, (xR-1), 381);

    lutfia.setColor(0,0,0);
    lutfia.fillRect(80, 385, (xR-1), 389);

    lutfia.setColor(255,119,0);
    lutfia.fillRoundRect(80, 382, (xR-1), 384);

//    lutfia.setColor(0,0,0);
//    lutfia.fillRoundRect(60, 382, (xR-1), 393);
//
    lutfia.setColor(255, 255, 255);
    lutfia.fillRoundRect((xR+6), 382, 235, 384);

    lutfia.setColor(0,0,0);
    lutfia.fillRect(xR+5, 372, 235, 381);

    lutfia.setColor(0,0,0);
    lutfia.fillRect(xR+5, 385, 235, 389);

    //======================== Tombol Previous Track=======================
    if ((xpos>50 && xpos<90) && (ypos>430 && ypos<460)) {
      lutfia.drawBitmap(lutfia.getDisplayXSize()-300, 400, previousp, 50,50,
lutfia.color565(255,255,255));
      z=z+10;
      delay(100);
      lutfia.drawBitmap(lutfia.getDisplayXSize()-300, 400, previousp, 50,50,
lutfia.color565(255,119,0));

    }

    //======================== Tombol Play & Pause=======================
    if ((xpos>140 && xpos<200) && (ypos>420 && ypos<470) && tp.z>MINPRESSURE)
{
      statPlay=1;
    }
    else{
      statPlay=0;
    }
    currentState=statPlay;

    if (currentState == HIGH && lastState == LOW){
      delay(1);//crude form of button debouncing
      if (ledState == HIGH){
```

```
        lutfia.setColor(0,0,0);
        lutfia.fillCircle(165,435,30);

        lutfia.drawBitmap(130, 400, pause, 70,70,
lutfia.color565(255,119,0));
        fm.setMute(false);
        ledState = LOW;
      }
      else {
        lutfia.setColor(0,0,0);
        lutfia.fillCircle(165,435,30);

        lutfia.drawBitmap(130, 400, playp, 70,70,
lutfia.color565(255,119,0));
        fm.setMute(true);
        ledState = HIGH;
      }
    }

    //====================== Tombol Next Track======================
    if ((xpos>240 && xpos<290) && (ypos>430 && ypos<460)) {
      lutfia.drawBitmap(lutfia.getDisplayXSize()-70, 400, nextp, 50,50,
lutfia.color565(255,255,255));
      delay(100);
      lutfia.drawBitmap(lutfia.getDisplayXSize()-70, 400, nextp, 50,50,
lutfia.color565(255,119,0));

    }

    if((xpos>=250 && xpos<=280) && (ypos>=130 && ypos<=160) &&
tp.z>MINPRESSURE){
      lutfia.drawBitmap(235, 80, exitt, 30,30, lutfia.color565(255,119,0));
      delay(500);
      menu();
    }
  }

}

void Calculator(){
  int x=0;
  // Draw the upper row of buttons
  for (x=0; x<4; x++)
  {
    lutfia.setColor(33, 37, 43);
    lutfia.fillRoundRect (10+(x*75), 5, 75+(x*75), 75);
    //lutfia.setColor(255, 255, 255);
    //lutfia.drawRoundRect (10+(x*80), 5, 80+(x*80), 80);
    lutfia.setTextSize(4);
    lutfia.setTextColor(lutfia.color565(255,255,255),lutfia.color565(33, 37,
43));
    lutfia.printNumI(x+1, 27+(x*80), 27);
  }
// Draw the center row of buttons
//   for (x=0; x<3; x++)
//   {
//     lutfia.setColor(0, 0, 255);
```

```
//     lutfia.fillRoundRect (10+(x*60), 70, 60+(x*60), 120);
//     lutfia.setColor(255, 255, 255);
//     lutfia.drawRoundRect (10+(x*60), 70, 60+(x*60), 120);
//     if (x<4)
//       lutfia.printNumI(x+6, 27+(x*60), 87);
//   }
//   lutfia.print("0", 267, 87);
// // Draw the lower row of buttons
//   lutfia.setColor(0, 0, 255);
//   lutfia.fillRoundRect (10, 130, 150, 180);
//   lutfia.setColor(255, 255, 255);
//   lutfia.drawRoundRect (10, 130, 150, 180);
//   lutfia.print("Clear", 40, 147);
//   lutfia.setColor(0, 0, 255);
//   lutfia.fillRoundRect (160, 130, 300, 180);
//   lutfia.setColor(255, 255, 255);
//   lutfia.drawRoundRect (160, 130, 300, 180);
//   lutfia.print("Enter", 190, 147);
//   lutfia.setBackColor (0, 0, 0);
}

void bmpDraw(char *filename, int x, int y) {
  File     bmpFile;
  int      bmpWidth, bmpHeight;   // W+H in pixels
  uint8_t  bmpDepth;              // Bit depth (currently must be 24)
  uint32_t bmpImageoffset;        // Start of image data in file
  uint32_t rowSize;               // Not always = bmpWidth; may have padding
  uint8_t  sdbuffer[3*BUFFPIXEL]; // pixel in buffer (R+G+B per pixel)
  uint16_t lcdbuffer[BUFFPIXEL];  // pixel out buffer (16-bit per pixel)
  uint8_t  buffidx = sizeof(sdbuffer); // Current position in sdbuffer
  boolean  goodBmp = false;       // Set to true on valid header parse
  boolean  flip    = true;        // BMP is stored bottom-to-top
  int      w, h, row, col;
  uint8_t  r, g, b;
  uint32_t pos = 0, startTime = millis();
  uint8_t  lcdidx = 0;
  boolean  first = true;

  if((x >= lutfia.width()) || (y >= lutfia.height())) return;

  Serial.println();
  progmemPrint(PSTR("Loading image '"));
  Serial.print(filename);
  Serial.println('\'');
  // Open requested file on SD card
  if ((bmpFile = SD.open(filename)) == NULL) {
    progmemPrintln(PSTR("File not found"));
    return;
  }

  // Parse BMP header
  if(read16(bmpFile) == 0x4D42) { // BMP signature
    progmemPrint(PSTR("File size: ")); Serial.println(read32(bmpFile));
    (void)read32(bmpFile); // Read & ignore creator bytes
    bmpImageoffset = read32(bmpFile); // Start of image data
    progmemPrint(PSTR("Image Offset: ")); Serial.println(bmpImageoffset,
DEC);
```

```
// Read DIB header
progmemPrint(PSTR("Header size: ")); Serial.println(read32(bmpFile));
bmpWidth  = read32(bmpFile);
bmpHeight = read32(bmpFile);
if(read16(bmpFile) == 1) { // # planes -- must be '1'
  bmpDepth = read16(bmpFile); // bits per pixel
  progmemPrint(PSTR("Bit Depth: ")); Serial.println(bmpDepth);
  if((bmpDepth == 24) && (read32(bmpFile) == 0)) { // 0 = uncompressed

    goodBmp = true; // Supported BMP format -- proceed!
    progmemPrint(PSTR("Image size: "));
    Serial.print(bmpWidth);
    Serial.print('x');
    Serial.println(bmpHeight);

    // BMP rows are padded (if needed) to 4-byte boundary
    rowSize = (bmpWidth * 3 + 3) & ~3;

    // If bmpHeight is negative, image is in top-down order.
    // This is not canon but has been observed in the wild.
    if(bmpHeight < 0) {
      bmpHeight = -bmpHeight;
      flip      = false;
    }

    // Crop area to be loaded
    w = bmpWidth;
    h = bmpHeight;
    if((x+w-1) >= lutfia.width())  w = lutfia.width()  - x;
    if((y+h-1) >= lutfia.height()) h = lutfia.height() - y;

    // Set TFT address window to clipped image bounds
    lutfia.setAddrWindow(x, y, x+w-1, y+h-1);

    for (row=0; row<h; row++) { // For each scanline...
      // Seek to start of scan line.  It might seem labor-
      // intensive to be doing this on every line, but this
      // method covers a lot of gritty details like cropping
      // and scanline padding.  Also, the seek only takes
      // place if the file position actually needs to change
      // (avoids a lot of cluster math in SD library).
      if(flip) // Bitmap is stored bottom-to-top order (normal BMP)
        pos = bmpImageoffset + (bmpHeight - 1 - row) * rowSize;
      else     // Bitmap is stored top-to-bottom
        pos = bmpImageoffset + row * rowSize;
      if(bmpFile.position() != pos) { // Need seek?
        bmpFile.seek(pos);
        buffidx = sizeof(sdbuffer); // Force buffer reload
      }

      for (col=0; col<w; col++) { // For each column...
        // Time to read more pixel data?
        if (buffidx >= sizeof(sdbuffer)) { // Indeed
          // Push LCD buffer to the display first
          if(lcdidx > 0) {
            lutfia.pushColors(lcdbuffer, lcdidx, first);
            lcdidx = 0;
```

```
              first  = false;
            }
            bmpFile.read(sdbuffer, sizeof(sdbuffer));
            buffidx = 0; // Set index to beginning
          }

          // Convert pixel from BMP to TFT format
          b = sdbuffer[buffidx++];
          g = sdbuffer[buffidx++];
          r = sdbuffer[buffidx++];
          lcdbuffer[lcdidx++] = lutfia.color565(r,g,b);
        } // end pixel
      } // end scanline
      // Write any remaining data to LCD
      if(lcdidx > 0) {
        lutfia.pushColors(lcdbuffer, lcdidx, first);
      }
      progmemPrint(PSTR("Loaded in "));
      Serial.print(millis() - startTime);
      Serial.println(" ms");
    } // end goodBmp
  }
 }

 bmpFile.close();
 if(!goodBmp) progmemPrintln(PSTR("BMP format not recognized."));
}

uint16_t read16(File f) {
 uint16_t result;
 ((uint8_t *)&result)[0] = f.read(); // LSB
 ((uint8_t *)&result)[1] = f.read(); // MSB
 return result;
}

uint32_t read32(File f) {
 uint32_t result;
 ((uint8_t *)&result)[0] = f.read(); // LSB
 ((uint8_t *)&result)[1] = f.read();
 ((uint8_t *)&result)[2] = f.read();
 ((uint8_t *)&result)[3] = f.read(); // MSB
 return result;
}

void progmemPrint(const char *str) {
 char c;
 while(c = pgm_read_byte(str++)) Serial.print(c);
}

void progmemPrintln(const char *str) {
 progmemPrint(str);
 Serial.println();
}

void showImages(){
 uint16_t xpos,ypos;
 uint8_t status1=1,status2=1,pointer;
```

```
  int i=1;
  lutfia.clrScr();
  showRTC();

  lutfia.drawTriangle(80, 430, 100, 415, 100, 445,
lutfia.color565(0,204,106));
  lutfia.setColor(0,204,106);
  lutfia.drawRect(140,415,185,445);
  lutfia.setTextColor(lutfia.color565(0,204,106),lutfia.color565(0,0,0));
  lutfia.setTextSize(3);
  lutfia.print("X",155,420);
  lutfia.drawTriangle(240, 430, 220, 415, 220, 445,
lutfia.color565(0,204,106));

  while(1){
    showRTC();
    readResistiveTouch();
    xpos = map(tp.x, TS_LEFT, TS_RT, 0, lutfia.getDisplayXSize());
    ypos = map(tp.y, TS_TOP, TS_BOT, 0, lutfia.getDisplayYSize());
    lutfia.setTextColor(lutfia.color565(255,255,255),lutfia.color565(0,0,0));
    lutfia.setTextSize(1.5);
    lutfia.printNumI(xpos, 135,10,10);
    lutfia.printNumI(ypos, 180,10,10);

    if((xpos >=75 && xpos<=120) && (ypos>=430 && ypos<=470)){
      lutfia.setColor(23, 161, 165);
      lutfia.drawRoundRect(75,410,105,450);
      lutfia.setColor(0,0,0);
      lutfia.fillRect(40,60,280,380);
      status1=0;
      i--;
      delay(500);
      lutfia.setColor(0,0,0);
      lutfia.drawRoundRect(75,410,105,450);
      if(i<1){
        i=10;
      }
    }

    if((xpos >=140 && xpos<=195) && (ypos>=430 && ypos<=470)){
      //lutfia.setColor(23, 161, 165);
      //lutfia.drawRoundRect(135,410,190,450);

      lutfia.setColor(0,204,106);
      lutfia.fillRect(140,415,185,445);
      lutfia.setTextColor(lutfia.color565(0,0,0),lutfia.color565(0,204,106));
      lutfia.setTextSize(3);
      lutfia.print("X",155,420);
      menu();
    }

    if((xpos >=220 && xpos<=250) && (ypos>=430 && ypos<=470)){
      lutfia.setColor(23, 161, 165);
      lutfia.drawRoundRect(215,410,245,450);
      lutfia.setColor(0,0,0);
      lutfia.fillRect(40,60,280,380);
      status1=0;
```

```
      i++;
      delay(500);
      lutfia.setColor(0,0,0);
      lutfia.drawRoundRect(215,410,245,450);
      if(i>10){
        i=1;
      }
    }

    if(status1==0){
      bmpDraw(itemImages[i],40,60);
      delay(1000);
      status1=1;
    }
  }
}

void osciloscope(){
  int buf[478];
  int x, x2;
  int y, y2;
  int r;
  lutfia.InitLCD(LANDSCAPE);
  lutfia.clrScr();

  lutfia.setColor(255, 0, 0);
  lutfia.fillRect(0, 0, 479, 13);
  lutfia.setColor(64, 64, 64);
  lutfia.fillRect(0, 306, 479, 319);
  lutfia.setColor(255, 255, 255);
  lutfia.setBackColor(255, 0, 0);
  lutfia.print("* Universal Color TFT Display Library *", CENTER, 1);
  lutfia.setBackColor(64, 64, 64);
  lutfia.setColor(255,255,0);
  lutfia.print("<http://electronics.henningkarlsen.com>", CENTER, 307);

  lutfia.setColor(0, 0, 255);
  lutfia.drawRect(0, 14, 479, 305);

// Draw crosshairs
  lutfia.setColor(0, 0, 255);
  lutfia.setBackColor(0, 0, 0);
  lutfia.drawLine(239, 15, 239, 304);
  lutfia.drawLine(1, 159, 478, 159);
  for (int i=9; i<470; i+=10)
    lutfia.drawLine(i, 157, i, 161);
  for (int i=19; i<220; i+=10)
    lutfia.drawLine(237, i, 241, i);

// Draw sin-, cos- and tan-lines
  lutfia.setColor(0,255,255);
  lutfia.print("Sin", 5, 15);
  for (int i=1; i<478; i++)
  {
    lutfia.drawPixel(i,159+(sin(((i*1.13)*3.14)/180)*95));
  }
```

```
  lutfia.setColor(255,0,0);
  lutfia.print("Cos", 5, 27);
  for (int i=1; i<478; i++)
  {
    lutfia.drawPixel(i,159+(cos(((i*1.13)*3.14)/180)*95));
  }

  lutfia.setColor(255,255,0);
  lutfia.print("Tan", 5, 39);
  for (int i=1; i<478; i++)
  {
    lutfia.drawPixel(i,159+(tan(((i*1.13)*3.14)/180)));
  }

  delay(2000);

  lutfia.setColor(0,0,0);
  lutfia.fillRect(1,15,478,304);
  lutfia.setColor(0, 0, 255);
  lutfia.setBackColor(0, 0, 0);
  lutfia.drawLine(239, 15, 239, 304);
  lutfia.drawLine(1, 159, 478, 159);

// Draw a moving sinewave
  x=1;
  for (int i=1; i<(478*15); i++)
  {
    x++;
    if (x==479)
      x=1;
    if (i>479)
    {
      if ((x==239)||(buf[x-1]==159))
        lutfia.setColor(0,0,255);
      else
        lutfia.setColor(0,0,0);
      lutfia.drawPixel(x,buf[x-1]);
    }
    lutfia.setColor(0,255,255);
    y=159+(sin(((i*0.7)*3.14)/180)*(90-(i / 100)));
    lutfia.drawPixel(x,y);
    buf[x-1]=y;
  }

  delay(2000);

  lutfia.setColor(0,0,0);
  lutfia.fillRect(1,15,478,304);

}


void menu(){
  uint16_t xpos,ypos;
  lutfia.clrScr();
  lutfia.setColor(255,255,255);
  lutfia.drawLine(0,32,319,32);
```

```
  showRTC();
  for(int i = 0; i < 12; i++) {
    int x = i%3;
    int y = i/3;
    lutfia.drawBitmap((x*90)+35, (y*100)+50, item_menu[i], 70, 70,
lutfia.color565(23, 161, 165));
  }
  while(1){
    showRTC();
    readResistiveTouch();
    xpos = map(tp.x, TS_LEFT, TS_RT, 0, lutfia.getDisplayXSize());
    ypos = map(tp.y, TS_TOP, TS_BOT, 0, lutfia.getDisplayYSize());
    if (tp.z < MINPRESSURE || tp.z > MAXPRESSURE) continue;
    lutfia.setTextColor(lutfia.color565(255,255,255),lutfia.color565(0,0,0));
    lutfia.setTextSize(1.5);
    lutfia.printNumI(xpos, 135,10,10);
    lutfia.printNumI(ypos, 180,10,10);
    /*====================== Menu FM==================*/
    if((xpos>35 && xpos<100) && (ypos>50 && ypos<150)){
      lutfia.drawBitmap(35, 50, item_menu[0], 70, 70, lutfia.color565(255,
255, 255));
      playFm();
    }
    /*====================== Menu Bluetooth==================*/
    if((xpos>165 && xpos<230) && (ypos>50 && ypos<150)){
      lutfia.drawBitmap(125, 50, item_menu[1], 70, 70, lutfia.color565(255,
255, 255));
    }
    /*====================== Menu Play Mp3==================*/
    if((xpos>245 && xpos<295) && (ypos>50 && ypos<150)){
      lutfia.drawBitmap(215, 50, item_menu[2], 70, 70, lutfia.color565(255,
255, 255));
      playMp3();
    }
    /*====================== Menu View Image==================*/
    if((xpos>20 && xpos<100) && (ypos>190 && ypos<250)){
      lutfia.drawBitmap(35, 150, item_menu[3], 70, 70, lutfia.color565(255,
255, 255));
      showImages();
    }
    /*====================== Set Time & Date==================*/
    if((xpos>165 && xpos<230) && (ypos>190 && ypos<250)){
      lutfia.drawBitmap(125, 150, item_menu[4], 70, 70, lutfia.color565(255,
255, 255));
      setTime();
    }
    if((xpos>245 && xpos<295) && (ypos>190 && ypos<250)){
      lutfia.drawBitmap(215, 150, item_menu[5], 70, 70, lutfia.color565(255,
255, 255));
    }
    if((xpos>20 && xpos<100) && (ypos>280 && ypos<345)){//if((xpos>(bx*90+35)
&& xpos<(bx*90+100)) && (ypos>(by*90+50) && ypos<(by*90+150))){
      lutfia.drawBitmap(35, 250, item_menu[6], 70, 70, lutfia.color565(255,
255, 255));
    }
```

```
    if((xpos>145 && xpos<200) && (ypos>280 &&
ypos<345)){//if((xpos>(bx*90+35) && xpos<(bx*90+100)) && (ypos>(by*90+50) &&
ypos<(by*90+150))){
      lutfia.drawBitmap(125, 250, item_menu[7], 70, 70, lutfia.color565(255,
255, 255));
    }
    if((xpos>230 && xpos<280) && (ypos>280 &&
ypos<345)){//if((xpos>(bx*90+35) && xpos<(bx*90+100)) && (ypos>(by*90+50) &&
ypos<(by*90+150))){
      lutfia.drawBitmap(215, 250, item_menu[8], 70, 70, lutfia.color565(255,
255, 255));

    }
    if((xpos>20 && xpos<100) && (ypos>370 && ypos<435)){//if((xpos>(bx*90+35)
&& xpos<(bx*90+100)) && (ypos>(by*90+50) && ypos<(by*90+150))){
      lutfia.drawBitmap(35, 350, item_menu[9], 70, 70, lutfia.color565(255,
255, 255));
      osciloscope();
    }
    if((xpos>145 && xpos<200) && (ypos>370 &&
ypos<435)){//if((xpos>(bx*90+35) && xpos<(bx*90+100)) && (ypos>(by*90+50) &&
ypos<(by*90+150))){
      lutfia.drawBitmap(125, 350, item_menu[10], 70, 70, lutfia.color565(255,
255, 255));
    }
    if((xpos>230 && xpos<280) && (ypos>370 &&
ypos<435)){//if((xpos>(bx*90+35) && xpos<(bx*90+100)) && (ypos>(by*90+50) &&
ypos<(by*90+150))){
      lutfia.drawBitmap(215, 350, item_menu[11], 70, 70, lutfia.color565(255,
255, 255));
    }
  }
}

void loop(){
  //setTime();
  menu();
}
```