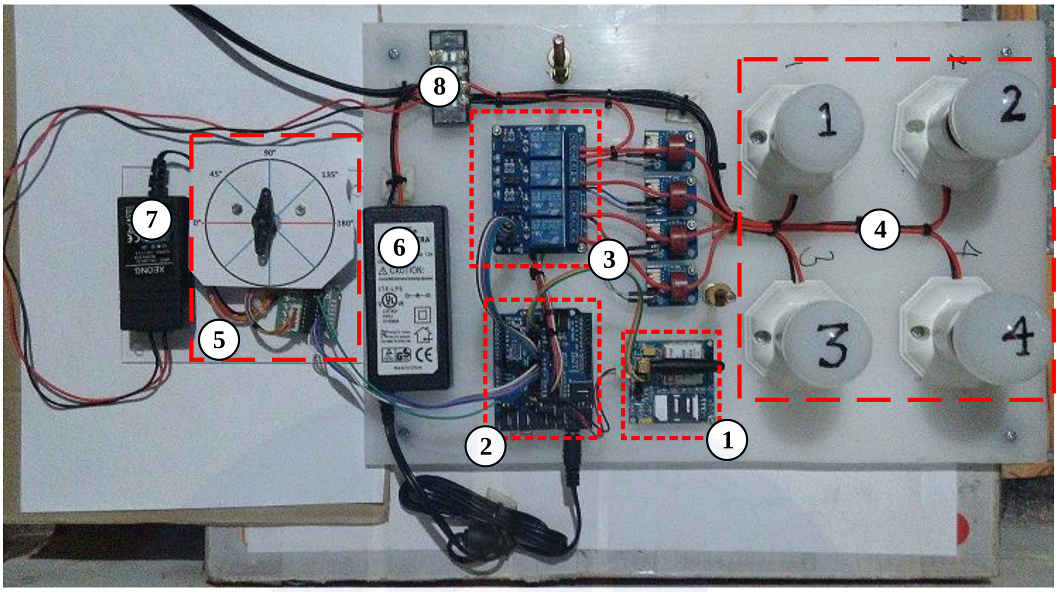
LAMPIRAN A

Layout Alat Pengendalian Lampu dan Motor Servo Keseluruhan

dan

Source Code Arduino Alat Pengendalian Lampu dan Motor Servo

Layout Alat Pengendalian Lampu dan Motor Servo Keseluruhan



Keterangan :

1. Modul SIM900A Mini
2. Board Arduino Uno R3 + board sensor shield
3. Modul relay 4 channel
4. 4 buah lampu + fitting lampu
5. Driver motor + motor servo
6. Adaptor 12Vdc 2A
7. Adaptor 12Vdc 1A
8. Sumber Tegangan AC

Source Code Arduino Alat Pengendalian Lampu dan Motor Servo

```
#include <Stepper.h>
const int stepsPerRevolution = 200;
#include "SIM900.h"
#include "sms.h"
#include "SoftwareSerial.h"
#include "sms.h"

SMSGSM sms;
Stepper myStepper(stepsPerRevolution, 8,9,10,11);

int sensorTA1 = A5;
int sensorTA2 = A4;
int sensorTA3 = A2;
int sensorTA4 = A0;

float nVPP1;
float nCurrThruResistorPP1;
float nCurrThruResistorRMS1;
float nCurrentThruWire1;

float nVPP2;
float nCurrThruResistorPP2;
float nCurrThruResistorRMS2;
float nCurrentThruWire2;

float nVPP3;
float nCurrThruResistorPP3;
float nCurrThruResistorRMS3;
float nCurrentThruWire3;

float nVPP4;
float nCurrThruResistorPP4;
float nCurrThruResistorRMS4;
float nCurrentThruWire4;

int lamp1 = 4;
int lamp2 = 5;
int lamp3 = 6;
int lamp4 = 7;

boolean started=false;
char smsbuffer[160];
char n[20];
int inc=10;

char a_str[5];
char b_str[5];
char c_str[5];
char d_str[5];
char message[180];

void setup()
{

  Serial.begin(9600);
  Serial.println("GSM Shield testing.");
   myStepper.setSpeed(50);

  pinMode(lamp1, OUTPUT);
  pinMode(lamp2, OUTPUT);
  pinMode(lamp3, OUTPUT);
  pinMode(lamp4, OUTPUT);

  pinMode(sensorTA1, INPUT);
  pinMode(sensorTA2, INPUT);
  pinMode(sensorTA3, INPUT);
  pinMode(sensorTA4, INPUT);

  digitalWrite(lamp1,HIGH);
  digitalWrite(lamp2,HIGH);
  digitalWrite(lamp3,HIGH);
  digitalWrite(lamp4,HIGH);
  if (gsm.begin(2400)){
  Serial.println("\nstatus=READY");
  started=true;
  }

  else Serial.println("\nstatus=IDLE");
  if(started){
    delsms();
  }

};

void loop()
{
  int pos=0;

  if(started){
    pos=sms.IsSMSPresent(SMS_ALL);
    if(pos){
      Serial.println("IsSMSPresent at pos ");
      Serial.println(pos);
      sms.GetSMS(pos,n,smsbuffer,100);
        Serial.println(n);
        Serial.println(smsbuffer);

        if(!strcmp(smsbuffer,"l" "a" "m" "p" "u" "1" "o" "n")){
        digitalWrite(lamp1,LOW);
        delay(2000);
        kondisi();
        }
        if(!strcmp(smsbuffer,"l" "a" "m" "p" "u" "1" "o" "f"
"f")){
        digitalWrite(lamp1,HIGH);
        delay(2000);
        kondisi();
        }
        if(!strcmp(smsbuffer,"l" "a" "m" "p" "u" "2" "o" "n")){
        digitalWrite(lamp2,LOW);
        delay(2000);
        kondisi();
        }
        if(!strcmp(smsbuffer,"l" "a" "m" "p" "u" "2" "o" "f"
"f")){
        digitalWrite(lamp2,HIGH);
        delay(2000);
        kondisi();
        }
        if(!strcmp(smsbuffer,"l" "a" "m" "p" "u" "3" "o" "n")){
        digitalWrite(lamp3,LOW);
        delay(2000);
        kondisi();
        }
        if(!strcmp(smsbuffer,"l" "a" "m" "p" "u" "3" "o" "f"
"f")){
        digitalWrite(lamp3,HIGH);
        delay(2000);
        kondisi();
        }
        if(!strcmp(smsbuffer,"l" "a" "m" "p" "u" "4" "o" "n")){
        digitalWrite(lamp4,LOW);
        delay(2000);
        kondisi();
        }
        if(!strcmp(smsbuffer,"l" "a" "m" "p" "u" "4" "o" "f"
"f")){
        digitalWrite(lamp4,HIGH);
        delay(2000);
        kondisi();
        }
        if(!strcmp(smsbuffer,"h" "i" "d" "u" "p" "k" "a" "n")){
        digitalWrite(lamp1,LOW);
        digitalWrite(lamp2,LOW);
        digitalWrite(lamp3,LOW);
        digitalWrite(lamp4,LOW);
        delay(2000);
        kondisi();
        }
        if(!strcmp(smsbuffer,"m" "a" "t" "i" "k" "a" "n")){
        digitalWrite(lamp1,HIGH);
        digitalWrite(lamp2,HIGH);
        digitalWrite(lamp3,HIGH);
        digitalWrite(lamp4,HIGH);
        delay(2000);
        kondisi();
        }
        if(!strcmp(smsbuffer,"k" "o" "n" "d" "i" "s" "i")){
        kondisi();
        }
        //Program tambahan stepper
        if(!strcmp(smsbuffer,"45")){
         myStepper.step(250);
        if(started){
        if (sms.SendSMS(n, "Motor berhasil bergerak ke 45
derajat"))
        Serial.println("\nSMS sent OK");
        }
        }
         //Program tambahan stepper
        if(!strcmp(smsbuffer,"90")){
         myStepper.step(500);
        if(started){
        if (sms.SendSMS(n, "Motor berhasil bergerak ke 90
derajat"))
        Serial.println("\nSMS sent OK");
        }
        }

         //Program tambahan stepper
        if(!strcmp(smsbuffer,"135")){
         myStepper.step(750);
        if(started){
```

```
        if (sms.SendSMS(n, "Motor berhasil bergerak ke 135
derajat"))
        Serial.println("\nSMS sent OK");
        }
        }

         //Program tambahan stepper
        if(!strcmp(smsbuffer,"180")){
         myStepper.step(1000);
        if(started){
        if (sms.SendSMS(n, "Motor berhasil bergerak ke 180
derajat"))
        Serial.println("\nSMS sent OK");
        }
        }
        delsms();
    }
  }
};

void kondisi(){
        nVPP1 = getVPP1();
        nCurrThruResistorPP1 = (nVPP1/200.0) * 1000.0;
        nCurrThruResistorRMS1 = nCurrThruResistorPP1 * 0.707;
        nCurrentThruWire1 = nCurrThruResistorRMS1 * 1000;

        nVPP2 = getVPP2();
        nCurrThruResistorPP2 = (nVPP2/200.0) * 1000.0;
        nCurrThruResistorRMS2 = nCurrThruResistorPP2 * 0.707;
        nCurrentThruWire2 = nCurrThruResistorRMS2 * 1000;

        nVPP3 = getVPP3();
        nCurrThruResistorPP3 = (nVPP3/200.0) * 1000.0;
        nCurrThruResistorRMS3 = nCurrThruResistorPP3 * 0.707;
        nCurrentThruWire3 = nCurrThruResistorRMS3 * 1000;

        nVPP4 = getVPP4();
        nCurrThruResistorPP4 = (nVPP4/200.0) * 1000.0;
        nCurrThruResistorRMS4 = nCurrThruResistorPP4 * 0.707;
        nCurrentThruWire4 = nCurrThruResistorRMS4 * 1000;

         float a = nCurrentThruWire1;
         float b = nCurrentThruWire2;
         float c = nCurrentThruWire3;
         float d = nCurrentThruWire4;

         message[0]='\0';
         strcat(message,"Lampu 1 : ");
         if (nCurrentThruWire1 == 0)
         {
         strcat(message, "OFF");
         }
         else
         {
         strcat(message, "ON");
         }
```

```
 strcat(message,"\n");
 strcat(message,"Arus 1 : ");
 itoa(a,a_str,10);
 strcat(message,a_str);
 strcat(message," mA");
 strcat(message,"\n");
 strcat(message,"Lampu 2 : ");
 if (nCurrentThruWire2 == 0)
 {
 strcat(message, "OFF");
 }
 else
 {
 strcat(message, "ON");
 }
 strcat(message,"\n");
 strcat(message,"Arus 2 : ");
 itoa(b,b_str,10);
 strcat(message,b_str);
 strcat(message," mA");
 strcat(message,"\n");
  strcat(message,"Lampu 3 : ");
 if (nCurrentThruWire3 == 0)
 {
 strcat(message, "OFF");
 }
 else
 {
 strcat(message, "ON");
 }
 strcat(message,"\n");
 strcat(message,"Arus 3 : ");
 itoa(c,c_str,10);
 strcat(message,c_str);
 strcat(message," mA");
 strcat(message,"\n");
 strcat(message,"Lampu 4 : ");
 if (nCurrentThruWire4 == 0)
 {
 strcat(message, "OFF");
 }
 else
 {
 strcat(message, "ON");
 }
 strcat(message,"\n");
 strcat(message,"Arus 4 : ");
 itoa(d,d_str,10);
 strcat(message,d_str);
 strcat(message," mA");
 strcat(message,"\n");
if(started){
if (sms.SendSMS(n, message))
Serial.println("\nSMS sent OK");
}
else
```

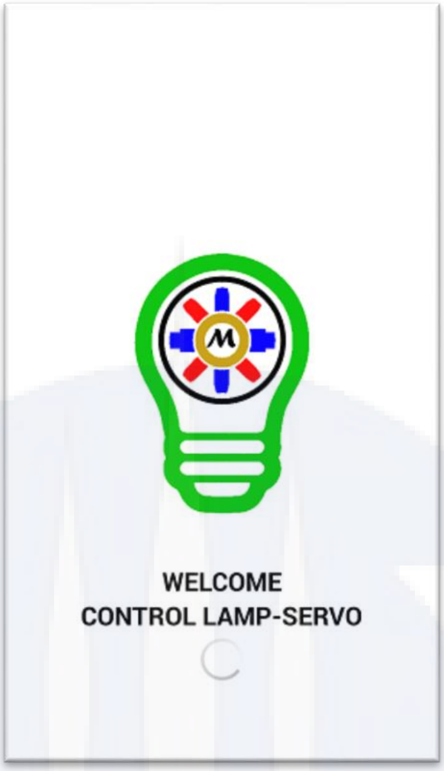
```
        {
        }
}
void delsms(){
  Serial.println("delsms");
  for (int i=0; i<10; i++){  //do it max 10 times
      int pos=sms.IsSMSPresent(SMS_ALL);
      if (pos!=0){
        Serial.print("\nFind SMS at the pos ");
        Serial.println(pos);
        if (sms.DeleteSMS(pos)==1){
          Serial.print("\nDeleted SMS at the pos ");
          Serial.println(pos);
        }
        else
        {
          Serial.print("\nCant del SMS at the pos ");
          Serial.println(pos);
        }
      }
    }
}
float getVPP1()
{
  float result1;
  int readValue1;
  int maxValue1 = 0;
   uint32_t start_time = millis();
   while((millis()-start_time) < 1000)
   {
       readValue1 = analogRead(sensorTA1);

       if (readValue1 > maxValue1)
       {
           maxValue1 = readValue1;
       }
   }
   result1 = (maxValue1 * 5.0)/1024.0;

   return result1;
 }
 float getVPP2()
{
  float result2;
  int readValue2;
  int maxValue2 = 0;
   uint32_t start_time = millis();
   while((millis()-start_time) < 1000)
   {
       readValue2 = analogRead(sensorTA2);

       if (readValue2 > maxValue2)
       {
           maxValue2 = readValue2;
       }
   }
   result2 = (maxValue2 * 5.0)/1024.0;
   return result2;
 }
 float getVPP3()
{
  float result3;
  int readValue3;
  int maxValue3 = 0;
   uint32_t start_time = millis();
   while((millis()-start_time) < 1000)
   {
       readValue3 = analogRead(sensorTA3);

       if (readValue3 > maxValue3)
       {
           maxValue3 = readValue3;
       }
   }
   result3 = (maxValue3 * 5.0)/1024.0;
   return result3;
 }
 float getVPP4()
{
  float result4;
  int readValue4;
  int maxValue4 = 0;
   uint32_t start_time = millis();
   while((millis()-start_time) < 1000)
   {
       readValue4 = analogRead(sensorTA4);

       if (readValue4 > maxValue4)
       {
           maxValue4 = readValue4;
       }
   }
   result4 = (maxValue4 * 5.0)/1024.0;

   return result4;
 }
```

LAMPIRAN B

Tampilan dan *Source Code*

Pembuatan Tampilan Antamuka Pengendali Aplikasi Android

Tampilan *Splash Screen* Aplikasi Android

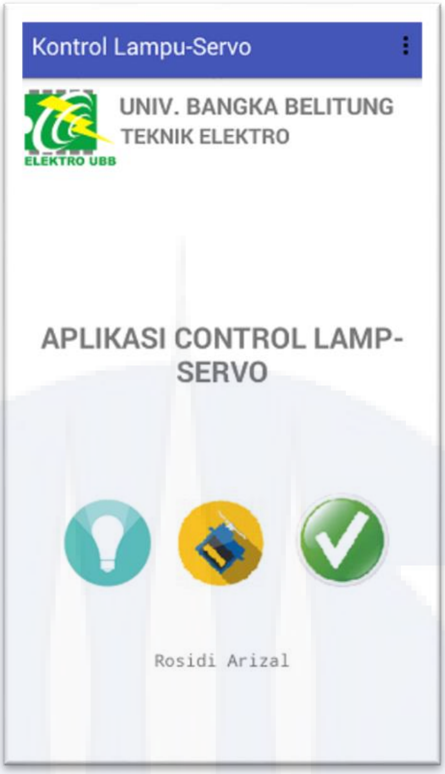


WELCOME
CONTROL LAMP-SERVO

Source Code Java Pembuatan Tampilan *Splash Screen* Aplikasi Android

```
package control.aplikasi.com.m_control;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.view.Window;
import android.view.WindowManager;

public class splashscreen extends Activity {

    //Set waktu lama splashscreen
    private static int splashInterval = 2000;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        requestWindowFeature(Window.FEATURE_NO_TITLE);

getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
                WindowManager.LayoutParams.FLAG_FULLSCREEN);

        setContentView(R.layout.splashscreen);
        new Handler().postDelayed(new Runnable() {

            @Override
            public void run() {
                // TODO Auto-generated method stub
                Intent i = new Intent(splashscreen.this,
MainActivity.class);
                startActivity(i);


                //jeda selesai Splashscreen
                this.finish();
            }
            private void finish() {
                // TODO Auto-generated method stub
            }
        }, splashInterval);
    };
}
```

Tampilan Menu Utama Aplikasi Android


Kontrol Lampu-Servo

UNIV. BANGKA BELITUNG
TEKNIK ELEKTRO

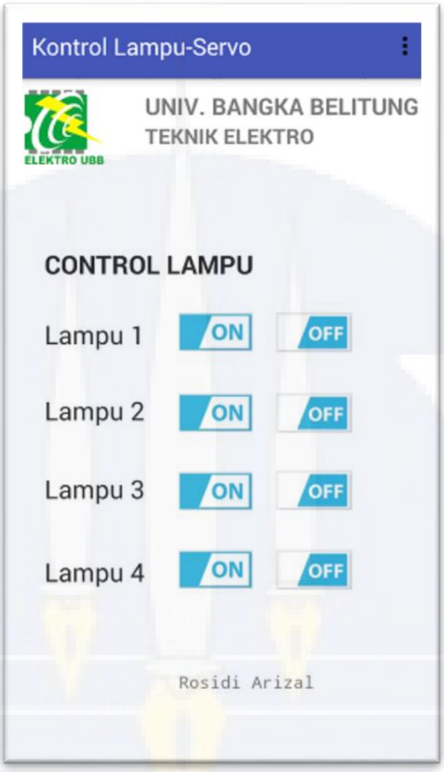
APLIKASI CONTROL LAMP-SERVO

Rosidi Arizal

Source Code Java Pembuatan Tampilan Menu Utama Aplikasi Android

```
package control.aplikasi.com.m_control;
import android.content.Intent;
import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
import android.support.design.widget.Snackbar;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.view.View;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.ImageButton;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        ImageButton BtnLampu = (ImageButton)
findViewById(R.id.btnLampu);
        BtnLampu.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View arg0) {
                Intent intent = new Intent(arg0.getContext(),
lampu.class);

                Bundle data = new Bundle();
                data.putString("", "");
                intent.putExtras(data);
                startActivityForResult(intent, 1);
            }
        });
        ImageButton BtnServo = (ImageButton)
findViewById(R.id.btnServo);
        BtnServo.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick (View arg0) {
                Intent intent = new Intent(arg0.getContext(),
servo.class);

                Bundle data = new Bundle();
                data.putString("", "");
                intent.putExtras(data);
                startActivityForResult(intent, 1);
            }
        });
        ImageButton BtnCek = (ImageButton)
findViewById(R.id.btnCek);
        BtnCek.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick (View arg0) {
                Intent intent = new Intent(arg0.getContext(),
cekkondisi.class);
                Bundle data = new Bundle();
                data.putString("", "");
                intent.putExtras(data);
                startActivityForResult(intent, 1);
            }
        });
    }
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if
it is present.
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    }
    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so
long
        // as you specify a parent activity in
AndroidManifest.xml.
        int id = item.getItemId();
        //noinspection SimplifiableIfStatement
        if (id == R.id.action_settings) {
            return true;
        }
        return super.onOptionsItemSelected(item);
    }
}
```

Tampilan Kendali Lampu Aplikasi Android



Kontrol Lampu-Servo

UNIV. BANGKA BELITUNG
TEKNIK ELEKTRO

CONTROL LAMPU

Lampu 1

Lampu 2

Lampu 3

Lampu 4

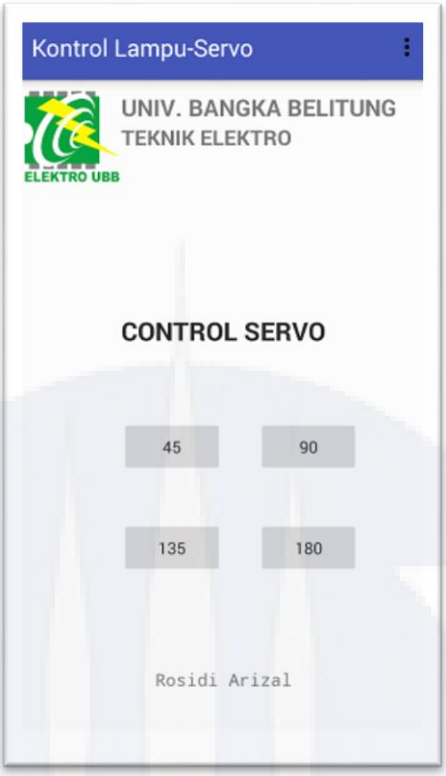
Rosidi Arizal

Source Code Java Pembuatan Tampilan Kendali Lampu Aplikasi Android

```
package control.aplikasi.com.m_control;

import android.app.Activity;
import android.app.PendingIntent;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.telephony.SmsManager;
import android.view.Menu;
import android.view.View;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.TextView;
import android.widget.Toast;
```

```
public class lampu extends AppCompatActivity {

    final String SMS_TEKIRIM = "SMS_SENT";
    final String SMS_SAMPAI = "SMS_DELIVERED";
    TextView
textViewInfo1,textViewInfo2,textViewInfo3,textViewInfo4 ;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.lampu);

        textViewInfo1 = (TextView) findViewById(R.id.info1);
        textViewInfo1.setText("");

        ImageButton tombolOn1 = (ImageButton)
findViewById(R.id.btn1on);
        tombolOn1.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View arg0) {
                PendingIntent PITerkirim =
PendingIntent.getBroadcast(
                        getApplication(), 0, new
Intent(SMS_TEKIRIM), 0);

                PendingIntent PITersampaikan =
                PendingIntent.getBroadcast(
                        getApplication(), 0, new
Intent(SMS_SAMPAI), 0);

                registerReceiver(new BroadcastReceiver() {
                    @Override
                    public void onReceive(Context context, Intent
intent)
{
                        String pesan;
                        switch (getResultCode()) {
                            case Activity.RESULT_OK:
                                pesan = "";
                                break;
                            default:
                                pesan = "" +
                                        getResultCode();
                        }

                        String isiSemula =
textViewInfo1.getText().toString();
                        textViewInfo1.setText(isiSemula +
                                "\n" + pesan);
                    }
                }, new IntentFilter(SMS_TEKIRIM));

                registerReceiver(new BroadcastReceiver() {
                    @Override
                    public void onReceive(Context context, Intent 
intent) {
                        String pesan;
                        switch (getResultCode()) {
                            case Activity.RESULT_OK:
                                pesan = "";
                                break;
                            default:
                                pesan = "" +
                                        getResultCode();
                        }
                        String isiSemula = 
textViewInfo1.getText().toString();
                        textViewInfo1.setText(isiSemula + "\n" + 
pesan);
                    }
                }, new IntentFilter(SMS_SAMPAI));

                String nomor = "085367224735";
                String pesan = "lampu1on";

                SmsManager sms = SmsManager.getDefault();
                sms.sendTextMessage(nomor, null, pesan, 
PITerkirim, PITersampaikan);
            }
        });

        ImageButton tombolOff1 = (ImageButton) 
findViewById(R.id.btn1off);
        tombolOff1.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View arg0) {
                PendingIntent PITerkirim = 
PendingIntent.getBroadcast(
                        getApplication(), 0, new 
Intent(SMS_TEKIRIM), 0);

                PendingIntent PITersampaikan = 
PendingIntent.getBroadcast(
                        getApplication(), 0, new 
Intent(SMS_SAMPAI), 0);

                registerReceiver(new BroadcastReceiver() {
                    @Override
                    public void onReceive(Context context, Intent 
intent) {
                        String pesan;
                        switch (getResultCode()) {
                            case Activity.RESULT_OK:
                                pesan = "";
                                break;
                            default:
                                pesan = "" +
                                        getResultCode();
                        }
                        String isiSemula =
textViewInfo1.getText().toString();
                        textViewInfo1.setText(isiSemula +
                                "\n" + pesan);
                    }
                }, new IntentFilter(SMS_TEKIRIM));

                registerReceiver(new BroadcastReceiver() {
                    @Override
                    public void onReceive(Context context, Intent
intent) {
                        String pesan;
                        switch (getResultCode()) {
                            case Activity.RESULT_OK:
                                pesan = "";
                                break;
                            default:
                                pesan = "" +
                                        getResultCode();
                        }
                        String isiSemula =
textViewInfo1.getText().toString();
                        textViewInfo1.setText(isiSemula + "\n" +
pesan);
                    }
                }, new IntentFilter(SMS_SAMPAI));

                String nomor = "085367224735";
                String pesan = "lampu1off";

                SmsManager sms = SmsManager.getDefault();
                sms.sendTextMessage(nomor, null, pesan,
PITerkirim, PITersampaikan);

            }
        });


        textViewInfo2 = (TextView) findViewById(R.id.info2);
        textViewInfo2.setText("");

        ImageButton tombolOn2 = (ImageButton)
findViewById(R.id.btn2on);
        tombolOn2.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View arg0) {
                PendingIntent PITerkirim =
PendingIntent.getBroadcast(
                        getApplication(), 0, new
Intent(SMS_TEKIRIM), 0);

                PendingIntent PITersampaikan =
PendingIntent.getBroadcast(
                        getApplication(), 0, new
Intent(SMS_SAMPAI), 0);

                registerReceiver(new BroadcastReceiver() {
                    @Override
                    public void onReceive(Context context, Intent
intent) {
                        String pesan;
                        switch (getResultCode()) {
                            case Activity.RESULT_OK:
                                pesan = "";
                                break;
                            default:
                                pesan = "" +
                                        getResultCode();
                        }

                        String isiSemula =
textViewInfo2.getText().toString();
                        textViewInfo2.setText(isiSemula +
                                "\n" + pesan);
                    }
                }, new IntentFilter(SMS_TEKIRIM));

                registerReceiver(new BroadcastReceiver() {
                    @Override
                    public void onReceive(Context context, Intent
intent) {
                        String pesan;
                        switch (getResultCode()) {
                            case Activity.RESULT_OK:
                                pesan = "";
                                break;
                            default:
                                pesan = "" +
                                        getResultCode();
                        }
                        String isiSemula =
textViewInfo2.getText().toString();
                        textViewInfo2.setText(isiSemula + "\n" +
pesan);
                    }
                }, new IntentFilter(SMS_SAMPAI));

                String nomor = "085367224735";
                String pesan = "lampu2on";

                SmsManager sms = SmsManager.getDefault();
                sms.sendTextMessage(nomor, null, pesan,
PITerkirim, PITersampaikan);
            }
        });

        ImageButton tombolOff2 = (ImageButton)
findViewById(R.id.btn2off);
        tombolOff2.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View arg0) {
                PendingIntent PITerkirim =
PendingIntent.getBroadcast(
                        getApplication(), 0, new
Intent(SMS_TEKIRIM), 0);

                PendingIntent PITersampaikan =
PendingIntent.getBroadcast(
                        getApplication(), 0, new
Intent(SMS_SAMPAI), 0);

                registerReceiver(new BroadcastReceiver() {
                    @Override
                    public void onReceive(Context context, Intent
intent) {
                        String pesan;
                        switch (getResultCode()) {
                            case Activity.RESULT_OK:
                                pesan = "";
                                break;
                            default:
                                pesan = "" +
                                        getResultCode();
                        }

                        String isiSemula =
textViewInfo2.getText().toString();
                        textViewInfo2.setText(isiSemula +
                                "\n" + pesan);
                    }
                }, new IntentFilter(SMS_TEKIRIM));

                registerReceiver(new BroadcastReceiver() {
                    @Override
                    public void onReceive(Context context, Intent
intent) {
                        String pesan;
                        switch (getResultCode()) {
                            case Activity.RESULT_OK:
                                pesan = "";
                                break;
                            default:
                                pesan = "" +
                                        getResultCode();
                        }
                        String isiSemula =
textViewInfo2.getText().toString();
                        textViewInfo2.setText(isiSemula + "\n" +
pesan);
                    }
                }, new IntentFilter(SMS_SAMPAI));

                String nomor = "085367224735";
                String pesan = "lampu2off";
                SmsManager sms = SmsManager.getDefault();
                sms.sendTextMessage(nomor, null, pesan,
PITerkirim, PITersampaikan);

            }
        });

        textViewInfo3 = (TextView) findViewById(R.id.info3);
        textViewInfo3.setText("");

        ImageButton tombolOn3 = (ImageButton)
findViewById(R.id.btn3on);
        tombolOn3.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View arg0) {
                PendingIntent PITerkirim =
PendingIntent.getBroadcast(
                        getApplication(), 0, new
Intent(SMS_TEKIRIM), 0);

                PendingIntent PITersampaikan =
PendingIntent.getBroadcast(
                        getApplication(), 0, new
Intent(SMS_SAMPAI), 0);

                registerReceiver(new BroadcastReceiver() {
                    @Override
                    public void onReceive(Context context, Intent
intent) {
                        String pesan;
                        switch (getResultCode()) {
                            case Activity.RESULT_OK:
                                pesan = "";
                                break;
                            default:
                                pesan = "" +
                                        getResultCode();
                        }

                        String isiSemula =
textViewInfo3.getText().toString();
                        textViewInfo3.setText(isiSemula +
                                "\n" + pesan);
                    }
                }, new IntentFilter(SMS_TEKIRIM));

                registerReceiver(new BroadcastReceiver() {
                    @Override
                    public void onReceive(Context context, Intent
intent) {
                        String pesan;
                        switch (getResultCode()) {
                            case Activity.RESULT_OK:
                                pesan = "";
                                break;
                            default:
                                pesan = "" +
                                        getResultCode();
                        }
                        String isiSemula =
textViewInfo3.getText().toString();
                        textViewInfo3.setText(isiSemula + "\n" +
pesan);
                    }
                }, new IntentFilter(SMS_SAMPAI));

                String nomor = "085367224735";
                String pesan = "lampu3on";

                SmsManager sms = SmsManager.getDefault();
                sms.sendTextMessage(nomor, null, pesan,
PITerkirim, PITersampaikan);
            }
        });

        ImageButton tombolOff3 = (ImageButton)
findViewById(R.id.btn3off);
        tombolOff3.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View arg0) {
                PendingIntent PITerkirim =
PendingIntent.getBroadcast(
                        getApplication(), 0, new
Intent(SMS_TEKIRIM), 0);

                PendingIntent PITersampaikan =
PendingIntent.getBroadcast(
                        getApplication(), 0, new
Intent(SMS_SAMPAI), 0);

                registerReceiver(new BroadcastReceiver() {
                    @Override
                    public void onReceive(Context context, Intent
intent) {
                        String pesan;
                        switch (getResultCode()) {
                            case Activity.RESULT_OK:
                                pesan = "";
                                break;
                            default:
                                pesan = "" +
                                        getResultCode();
                        }

                        String isiSemula =
textViewInfo3.getText().toString();
                        textViewInfo3.setText(isiSemula +
                                "\n" + pesan);
                    }
                }, new IntentFilter(SMS_TEKIRIM));
                registerReceiver(new BroadcastReceiver() {
                    @Override
                    public void onReceive(Context context, Intent
intent) {
                        String pesan;
                        switch (getResultCode()) {
                            case Activity.RESULT_OK:
                                pesan = "";
                                break;
                            default:
                                pesan = "" +
                                        getResultCode();
                        }
                        String isiSemula =
textViewInfo3.getText().toString();
                        textViewInfo3.setText(isiSemula + "\n" +
pesan);
                    }
                }, new IntentFilter(SMS_SAMPAI));

                String nomor = "085367224735";
                String pesan = "lampu3off";

                SmsManager sms = SmsManager.getDefault();
                sms.sendTextMessage(nomor, null, pesan,
PITerkirim, PITersampaikan);

            }
        });

        textViewInfo4 = (TextView) findViewById(R.id.info4);
        textViewInfo4.setText("");

        ImageButton tombolOn4 = (ImageButton)
findViewById(R.id.btn4on);
        tombolOn4.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View arg0) {
                PendingIntent PITerkirim =
PendingIntent.getBroadcast(
                        getApplication(), 0, new
Intent(SMS_TEKIRIM), 0);

                PendingIntent PITersampaikan =
PendingIntent.getBroadcast(
                        getApplication(), 0, new
Intent(SMS_SAMPAI), 0);

                registerReceiver(new BroadcastReceiver() {
                    @Override
           public void onReceive(Context context, Intent intent) {
                        String pesan;
                        switch (getResultCode()) {
                            case Activity.RESULT_OK:
                                pesan = "";
                                break;
                            default:
                                pesan = "" +
                                        getResultCode();
                        }

                        String isiSemula =
textViewInfo4.getText().toString();
                        textViewInfo4.setText(isiSemula +
                                "\n" + pesan);
                    }
                }, new IntentFilter(SMS_TEKIRIM));

                registerReceiver(new BroadcastReceiver() {
                    @Override
                    public void onReceive(Context context, Intent
intent) {
                        String pesan;
                        switch (getResultCode()) {
                            case Activity.RESULT_OK:
                                pesan = "";
                                break;
                            default:
                                pesan = "" +
                                        getResultCode();
                        }
                        String isiSemula =
textViewInfo4.getText().toString();
                        textViewInfo4.setText(isiSemula + "\n" +
pesan);
                    }
                }, new IntentFilter(SMS_SAMPAI));

                String nomor = "085367224735";
                String pesan = "lampu4on";

                SmsManager sms = SmsManager.getDefault();
                sms.sendTextMessage(nomor, null, pesan,
PITerkirim, PITersampaikan);
            }
        });

        ImageButton tombolOff4 = (ImageButton)
findViewById(R.id.btn4off);
        tombolOff4.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View arg0) {
                PendingIntent PITerkirim =
PendingIntent.getBroadcast(
                        getApplication(), 0, new
Intent(SMS_TEKIRIM), 0);

                PendingIntent PITersampaikan =
PendingIntent.getBroadcast(
                        getApplication(), 0, new
Intent(SMS_SAMPAI), 0);

                registerReceiver(new BroadcastReceiver() {
                    @Override
                    public void onReceive(Context context, Intent
intent) {
                        String pesan;
                        switch (getResultCode()) {
                            case Activity.RESULT_OK:
                                pesan = "";
                                break;
                            default:
                                pesan = "" +
                                        getResultCode();
                        }
                        String isiSemula =
textViewInfo4.getText().toString();
                        textViewInfo4.setText(isiSemula +
                                "\n" + pesan);
                    }
                }, new IntentFilter(SMS_TEKIRIM));

                registerReceiver(new BroadcastReceiver() {
                    @Override
           public void onReceive(Context context, Intent intent) {
                        String pesan;
                        switch (getResultCode()) {
                            case Activity.RESULT_OK:
                                pesan = "";
                                break;
                            default:
                                pesan = "" +
                                        getResultCode();
                        }
                        String isiSemula =
textViewInfo4.getText().toString();
                textViewInfo4.setText(isiSemula + "\n" + pesan);
                    }
                }, new IntentFilter(SMS_SAMPAI));

                String nomor = "085367224735";
                String pesan = "lampu4off";

                SmsManager sms = SmsManager.getDefault();
                sms.sendTextMessage(nomor, null, pesan,
PITerkirim, PITersampaikan);
            }
        });
    }
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    }
}
```

Tampilan Kendali Motor Servo Aplikasi Android



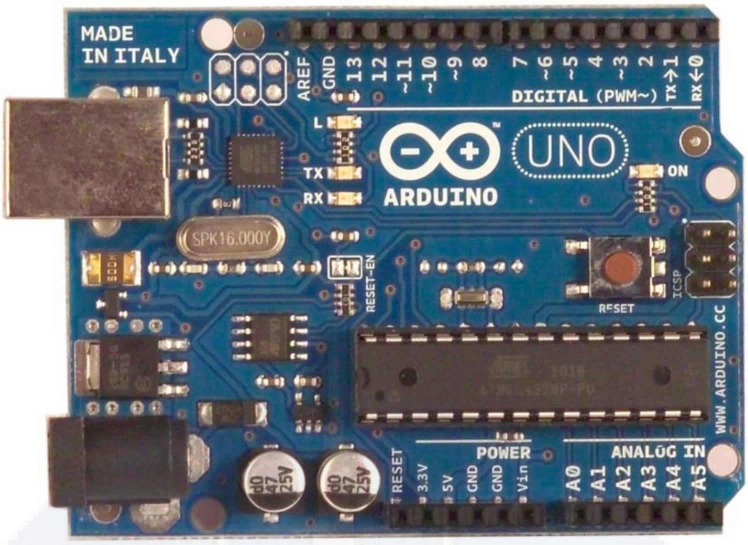
Source Code Java Pembuatan Tampilan Kendali Motor Servo Aplikasi Android

```
package control.aplikasi.com.m_control;

import android.app.Activity;
import android.app.PendingIntent;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.telephony.SmsManager;
import android.view.Menu;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.TextView;
import android.widget.Toast;

public class servo extends AppCompatActivity {
    final String SMS_TEKIRIM = "SMS_SENT";
    final String SMS_SAMPAI = "SMS_DELIVERED";
    TextView textViewInfo ;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.servo);

        textViewInfo = (TextView) findViewById(R.id.info);
        textViewInfo.setText("");

        Button tombol45 = (Button) findViewById(R.id.btn45);
        tombol45.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View arg0) {
                PendingIntent PITerkirim =
PendingIntent.getBroadcast(
                        getApplication(), 0, new
Intent(SMS_TEKIRIM), 0);

                PendingIntent PITersampaikan =
PendingIntent.getBroadcast(
                        getApplication(), 0, new
Intent(SMS_SAMPAI), 0);

                registerReceiver(new BroadcastReceiver() {
                    @Override
                    public void onReceive(Context context, Intent
intent) {
                        String pesan;
                        switch (getResultCode()) {
                            case Activity.RESULT_OK:
                                pesan = "";
                                break;
                            default:
                                pesan = "" +
                                        getResultCode();
                        }

                        String isiSemula =
textViewInfo.getText().toString();
                        textViewInfo.setText(isiSemula +
                                "\n" + pesan);
                    }
                }, new IntentFilter(SMS_TEKIRIM));

                registerReceiver(new BroadcastReceiver() {
                    @Override
                    public void onReceive(Context context, Intent
intent) {
                        String pesan;
                        switch (getResultCode()) {
                            case Activity.RESULT_OK:
                                pesan = "";
                                break;
                            default:
                                pesan = "" +
                                        getResultCode();
                        }
                        String isiSemula =
textViewInfo.getText().toString();
                        textViewInfo.setText(isiSemula + "\n" +
pesan);
                    }
                }, new IntentFilter(SMS_SAMPAI));

                String nomor = "085367224735";
                String pesan = "45";

                SmsManager sms = SmsManager.getDefault();
                sms.sendTextMessage(nomor, null, pesan,
PITerkirim, PITersampaikan);
            }
        });

        Button tombol90 = (Button) findViewById(R.id.btn90);
        tombol90.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View arg0) {
                PendingIntent PITerkirim =
PendingIntent.getBroadcast(
                        getApplication(), 0, new
Intent(SMS_TEKIRIM), 0);

                PendingIntent PITersampaikan =
PendingIntent.getBroadcast(
                        getApplication(), 0, new
Intent(SMS_SAMPAI), 0);

                registerReceiver(new BroadcastReceiver() {
                    @Override
                    public void onReceive(Context context, Intent
intent) {
                        String pesan;
                        switch (getResultCode()) {
                            case Activity.RESULT_OK:
                                pesan = "";
                                break;
                            default:
                                pesan = "" +
                                        getResultCode();
                        }

                        String isiSemula =
textViewInfo.getText().toString();
                        textViewInfo.setText(isiSemula +
                                "\n" + pesan);
                    }
                }, new IntentFilter(SMS_TEKIRIM));
                registerReceiver(new BroadcastReceiver() {
                    @Override
                    public void onReceive(Context context, Intent
intent) {
                        String pesan;
                        switch (getResultCode()) {
                            case Activity.RESULT_OK:
                                pesan = "";
                                break;
                            default:
                                pesan = "" +
                                        getResultCode();
                        }
                        String isiSemula =
textViewInfo.getText().toString();
                        textViewInfo.setText(isiSemula + "\n" +
pesan);
                    }
                }, new IntentFilter(SMS_SAMPAI));

                String nomor = "085367224735";
                String pesan = "90";

                SmsManager sms = SmsManager.getDefault();
                sms.sendTextMessage(nomor, null, pesan,
PITerkirim, PITersampaikan);

            }
        });

        Button tombol135 = (Button) findViewById(R.id.btn135);
        tombol135.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View arg0) {
                PendingIntent PITerkirim =
PendingIntent.getBroadcast(
                        getApplication(), 0, new
Intent(SMS_TEKIRIM), 0);

                PendingIntent PITersampaikan =
PendingIntent.getBroadcast(
                        getApplication(), 0, new
Intent(SMS_SAMPAI), 0);

                registerReceiver(new BroadcastReceiver() {
                    @Override
                    public void onReceive(Context context, Intent
intent) {
                        String pesan;
                        switch (getResultCode()) {
                            case Activity.RESULT_OK:
                                pesan = "";
                                break;
                            default:
                                pesan = "" +
                                        getResultCode();
                        }

                        String isiSemula =
textViewInfo.getText().toString();
                        textViewInfo.setText(isiSemula +
                                "\n" + pesan);
                    }
                }, new IntentFilter(SMS_TEKIRIM));

                registerReceiver(new BroadcastReceiver() {
                    @Override
                    public void onReceive(Context context, Intent
intent) {
                        String pesan;
                        switch (getResultCode()) {
                            case Activity.RESULT_OK:
                                pesan = "";
                                break;
                            default:
                                pesan = "" +
                                        getResultCode();
                        }
                        String isiSemula =
textViewInfo.getText().toString();
                        textViewInfo.setText(isiSemula + "\n" +

pesan);
                    }
                }, new IntentFilter(SMS_SAMPAI));

                String nomor = "085367224735";
                String pesan = "135";

                SmsManager sms = SmsManager.getDefault();
                sms.sendTextMessage(nomor, null, pesan,
PITerkirim, PITersampaikan);
            }
        });

        Button tombol180 = (Button) findViewById(R.id.btn180);
        tombol180.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View arg0) {
                PendingIntent PITerkirim =
PendingIntent.getBroadcast(
                        getApplication(), 0, new
Intent(SMS_TEKIRIM), 0);

                PendingIntent PITersampaikan =
PendingIntent.getBroadcast(
                        getApplication(), 0, new
Intent(SMS_SAMPAI), 0);

                registerReceiver(new BroadcastReceiver() {
                    @Override
                    public void onReceive(Context context, Intent
intent) {
                        String pesan;
                        switch (getResultCode()) {
                            case Activity.RESULT_OK:
                                pesan = "";
                                break;
                            default:
                                pesan = "" +
                                        getResultCode();
                        }

                        String isiSemula =
textViewInfo.getText().toString();
                        textViewInfo.setText(isiSemula +
                                "\n" + pesan);
                    }
                }, new IntentFilter(SMS_TEKIRIM));

                registerReceiver(new BroadcastReceiver() {
                    @Override
                    public void onReceive(Context context, Intent
intent) {
                        String pesan;
                        switch (getResultCode()) {
                            case Activity.RESULT_OK:
                                pesan = "";
                                break;
                            default:
                                pesan = "" +
                                        getResultCode();
                        }
                        String isiSemula =
textViewInfo.getText().toString();
                        textViewInfo.setText(isiSemula + "\n" +

pesan);
                    }
                }, new IntentFilter(SMS_SAMPAI));

                String nomor = "085367224735";
                String pesan = "180";

                SmsManager sms = SmsManager.getDefault();
                sms.sendTextMessage(nomor, null, pesan,
PITerkirim, PITersampaikan);
            }
        });
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    }
}
```

LAMPIRAN C

Datasheet Arduino Uno

Arduino UNO



Product Overview

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduno, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the index of Arduino boards.

Index

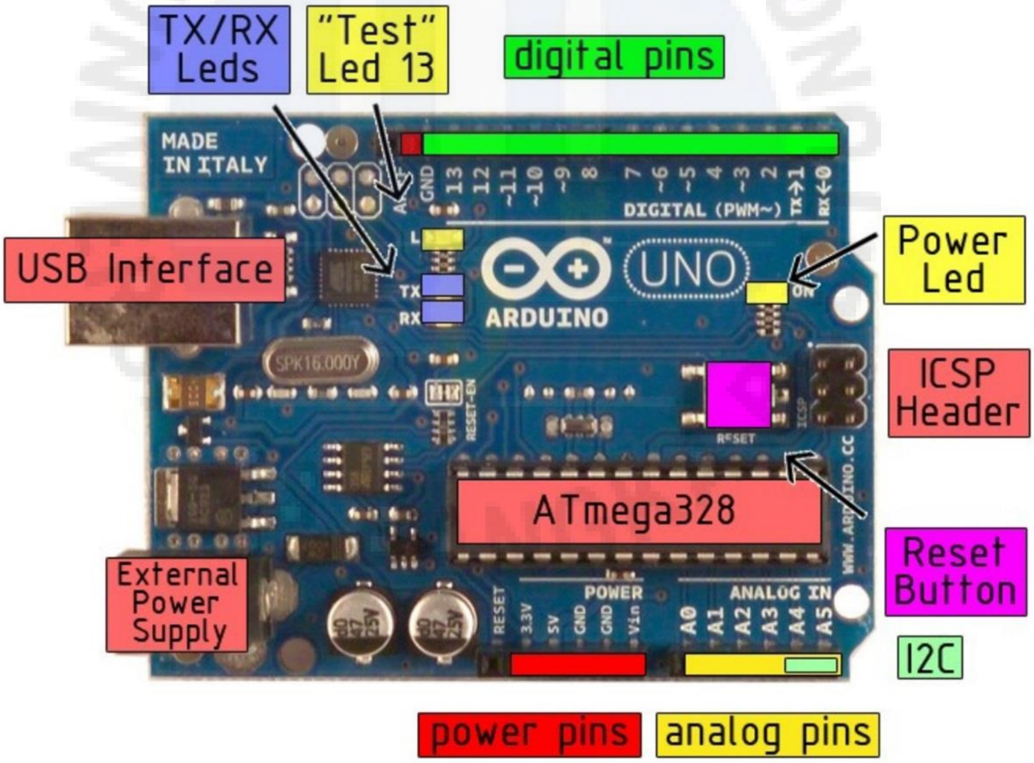
Technical Specification

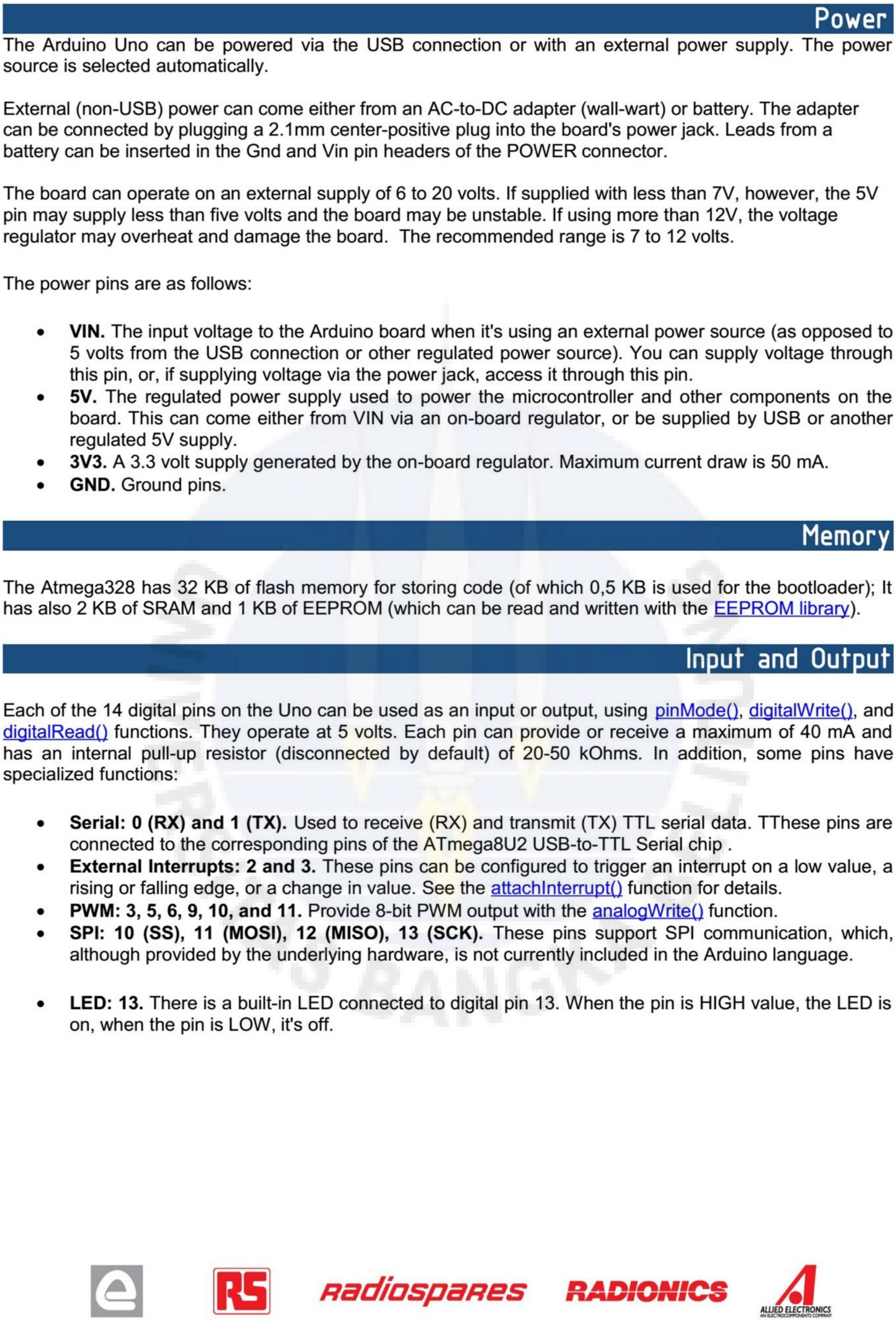
EAGLE files: arduino-duemilanove-uno-design.zip Schematic: arduino-uno-schematic.pdf

Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB of which 0.5 KB used by bootloader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

the board





Power

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The Atmega328 has 32 KB of flash memory for storing code (of which 0,5 KB is used for the bootloader); It has also 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. TThese pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip .
- **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.
- **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the analogWrite() function.
- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.

- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference() function. Additionally, some pins have specialized functionality:

- **I²C: 4 (SDA) and 5 (SCL).** Support I²C (TWI) communication using the Wire library.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with analogReference().
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

See also the mapping between Arduino pins and Atmega328 ports.

Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega8U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '8U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, an *.inf file is required..

The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A SoftwareSerial library allows for serial communication on any of the Uno's digital pins.

The ATmega328 also support I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation for details. To use the SPI communication, please see the ATmega328 datasheet.

Programming

The Arduino Uno can be programmed with the Arduino software (download). Select "Arduino Uno w/ ATmega328" from the **Tools > Board** menu (according to the microcontroller on your board). For details, see the reference and tutorials.

The ATmega328 on the Arduino Uno comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details.

The ATmega8U2 firmware source code is available . The ATmega8U2 is loaded with a DFU bootloader, which can be activated by connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2. You can then use Atmel's FLIP software (Windows) or the DFU programmer (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader).

Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

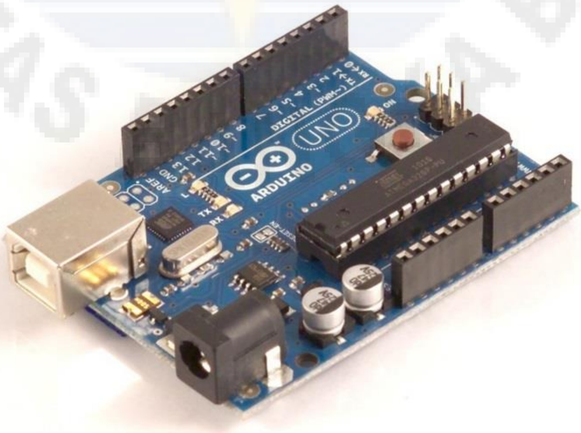
The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see this forum thread for details.

USB Overcurrent Protection

The Arduino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.



How to use Arduino

Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the Arduino programming language (based on Wiring) and the Arduino development environment (based on Processing). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP).

Arduino is a cross-platoform program. You'll have to follow different instructions for your personal OS. Check on the Arduino site for the latest instructions. *http://arduino.cc/en/Guide/HomePage*

Linux Install

Windows Install

Mac Install

Once you have downloaded/unzipped the arduino IDE, you can Plug the Arduino to your PC via USB cable.

Blink led

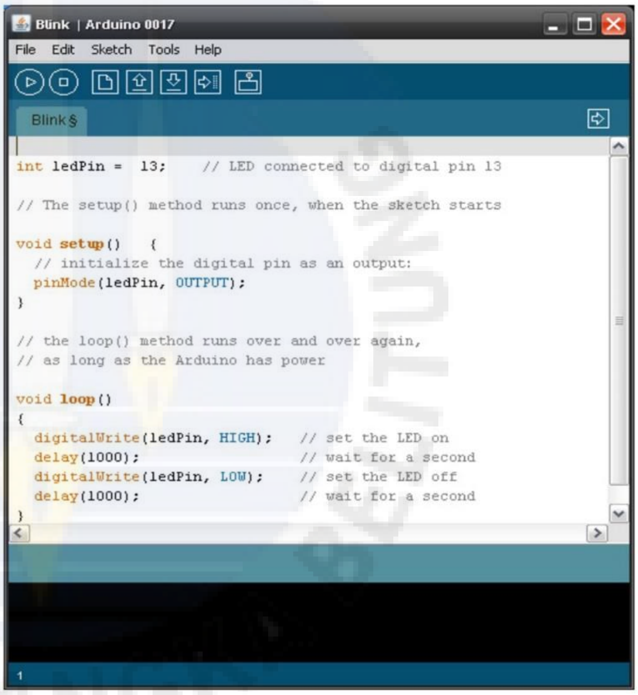
Now you're actually ready to "burn" your first program on the arduino board. To select "blink led", the physical translation of the well known programming "hello world", select

File>Sketchbook>
Arduino-0017>Examples>
Digital>Blink

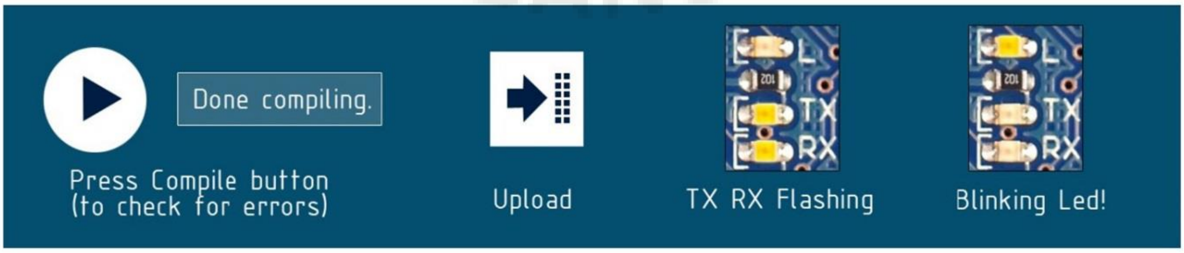
Once you have your skecth you'll see something very close to the screenshot on the right.

In **Tools>Board** select

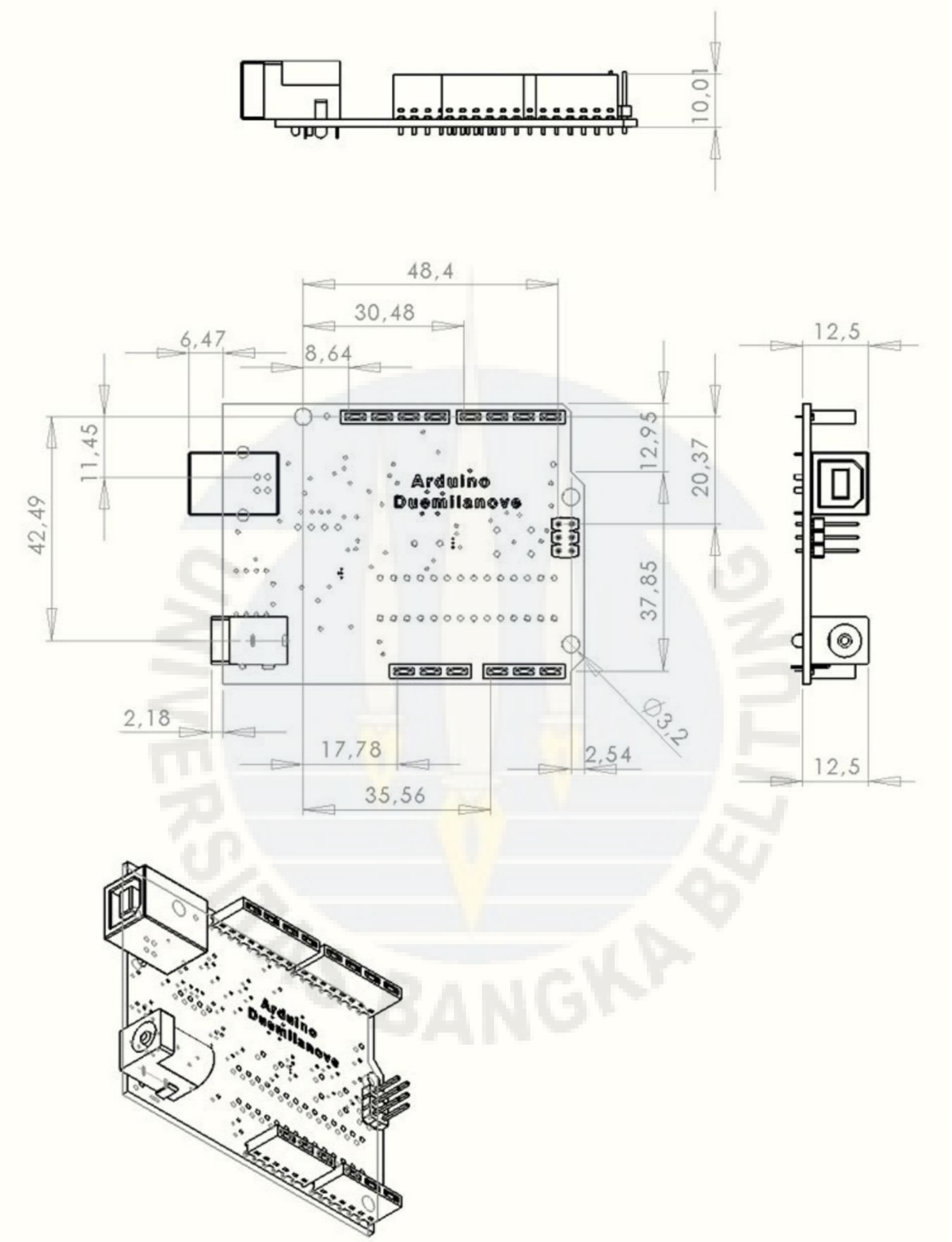
Now you have to go to
Tools>SerialPort
and select the right serial port, the one arduino is attached to.

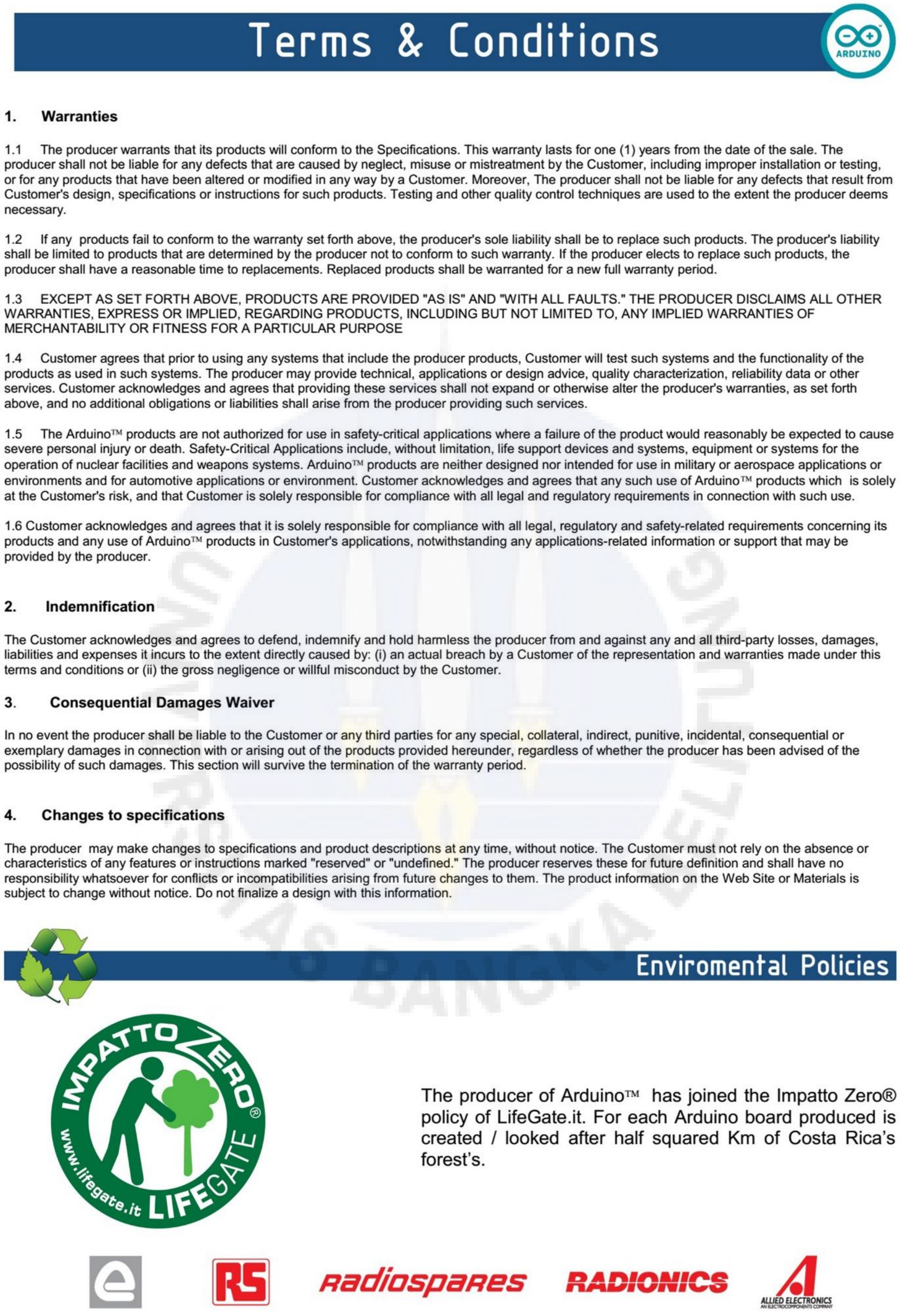


```
int ledPin =  13;    // LED connected to digital pin 13

// The setup() method runs once, when the sketch starts

void setup()   {
  // initialize the digital pin as an output:
  pinMode(ledPin, OUTPUT);
}

// the loop() method runs over and over again,
// as long as the Arduino has power

void loop()
{
  digitalWrite(ledPin, HIGH);   // set the LED on
  delay(1000);                  // wait for a second
  digitalWrite(ledPin, LOW);    // set the LED off
  delay(1000);                  // wait for a second
}
```



Dimensioned Drawing





Terms & Conditions

1. Warranties

1.1 The producer warrants that its products will conform to the Specifications. This warranty lasts for one (1) years from the date of the sale. The producer shall not be liable for any defects that are caused by neglect, misuse or mistreatment by the Customer, including improper installation or testing, or for any products that have been altered or modified in any way by a Customer. Moreover, The producer shall not be liable for any defects that result from Customer's design, specifications or instructions for such products. Testing and other quality control techniques are used to the extent the producer deems necessary.

1.2 If any products fail to conform to the warranty set forth above, the producer's sole liability shall be to replace such products. The producer's liability shall be limited to products that are determined by the producer not to conform to such warranty. If the producer elects to replace such products, the producer shall have a reasonable time to replacements. Replaced products shall be warranted for a new full warranty period.

1.3 EXCEPT AS SET FORTH ABOVE, PRODUCTS ARE PROVIDED "AS IS" AND "WITH ALL FAULTS." THE PRODUCER DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE

1.4 Customer agrees that prior to using any systems that include the producer products, Customer will test such systems and the functionality of the products as used in such systems. The producer may provide technical, applications or design advice, quality characterization, reliability data or other services. Customer acknowledges and agrees that providing these services shall not expand or otherwise alter the producer's warranties, as set forth above, and no additional obligations or liabilities shall arise from the producer providing such services.

1.5 The Arduino™ products are not authorized for use in safety-critical applications where a failure of the product would reasonably be expected to cause severe personal injury or death. Safety-Critical Applications include, without limitation, life support devices and systems, equipment or systems for the operation of nuclear facilities and weapons systems. Arduino™ products are neither designed nor intended for use in military or aerospace applications or environments and for automotive applications or environment. Customer acknowledges and agrees that any such use of Arduino™ products which is solely at the Customer's risk, and that Customer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

1.6 Customer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products and any use of Arduino™ products in Customer's applications, notwithstanding any applications-related information or support that may be provided by the producer.

2. Indemnification

The Customer acknowledges and agrees to defend, indemnify and hold harmless the producer from and against any and all third-party losses, damages, liabilities and expenses it incurs to the extent directly caused by: (i) an actual breach by a Customer of the representation and warranties made under this terms and conditions or (ii) the gross negligence or willful misconduct by the Customer.

3. Consequential Damages Waiver

In no event the producer shall be liable to the Customer or any third parties for any special, collateral, indirect, punitive, incidental, consequential or exemplary damages in connection with or arising out of the products provided hereunder, regardless of whether the producer has been advised of the possibility of such damages. This section will survive the termination of the warranty period.

4. Changes to specifications

The producer may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." The producer reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.

Enviromental Policies

The producer of Arduino™ has joined the Impatto Zero® policy of LifeGate.it. For each Arduino board produced is created / looked after half squared Km of Costa Rica's forest's.