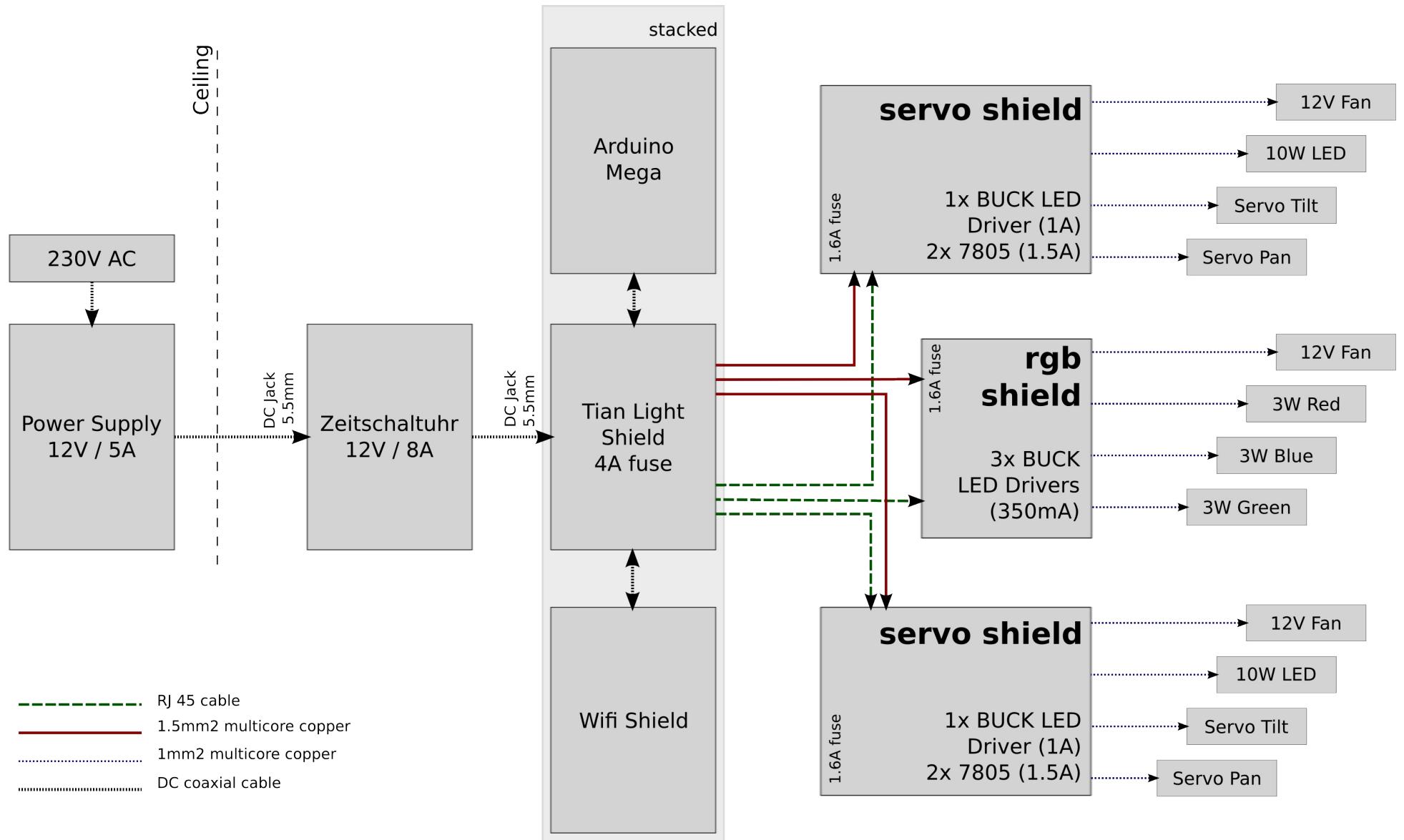
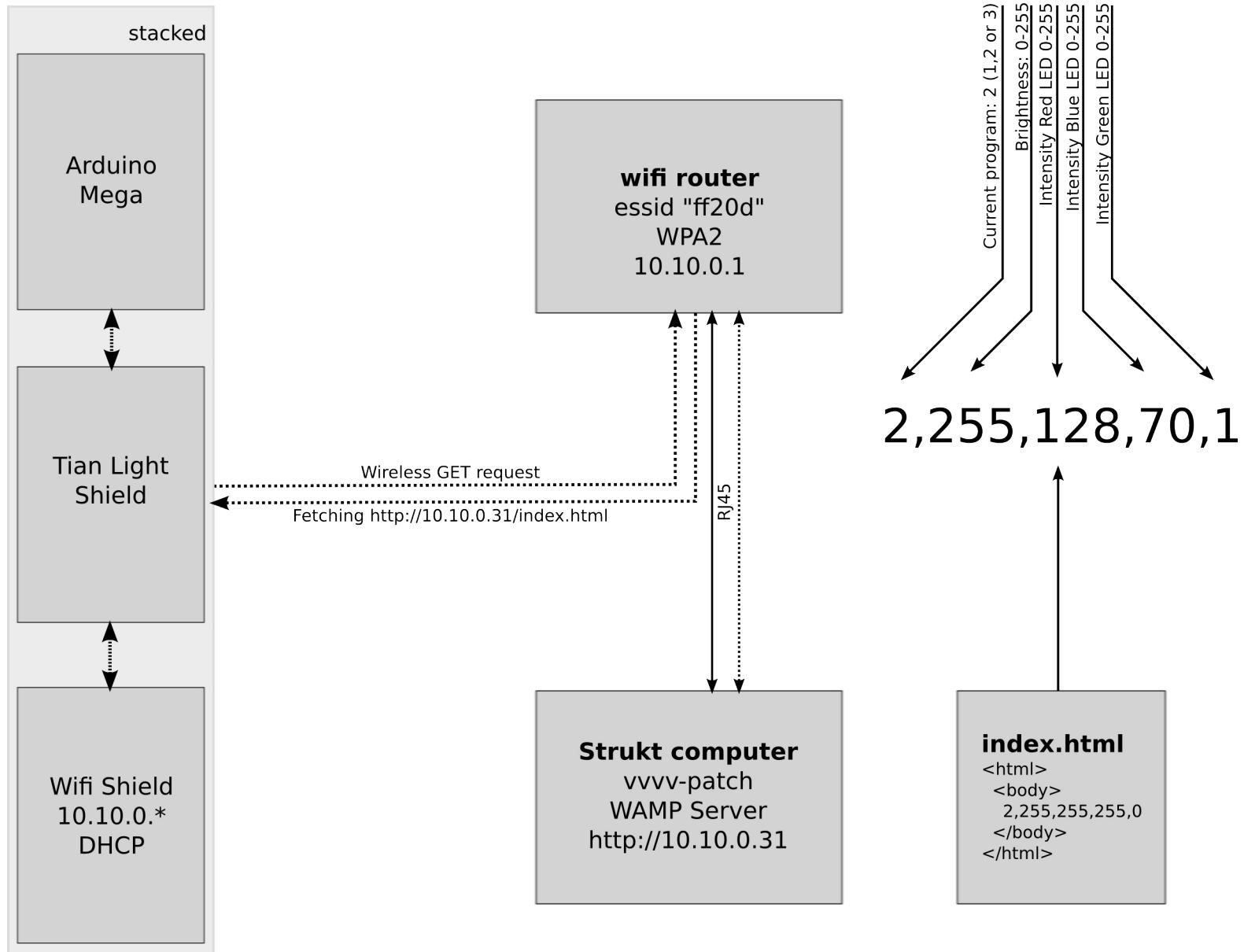




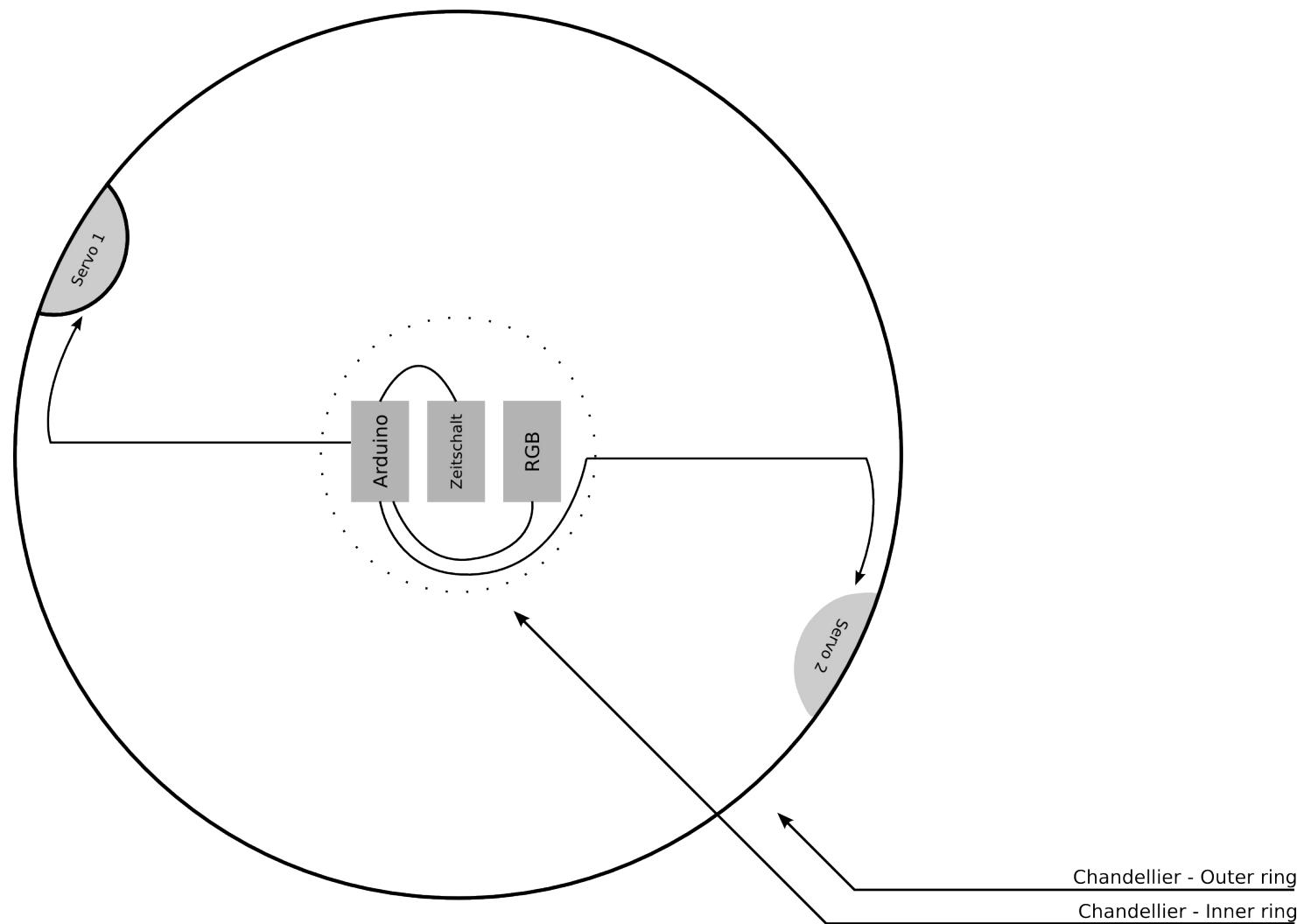
# Block Diagramm



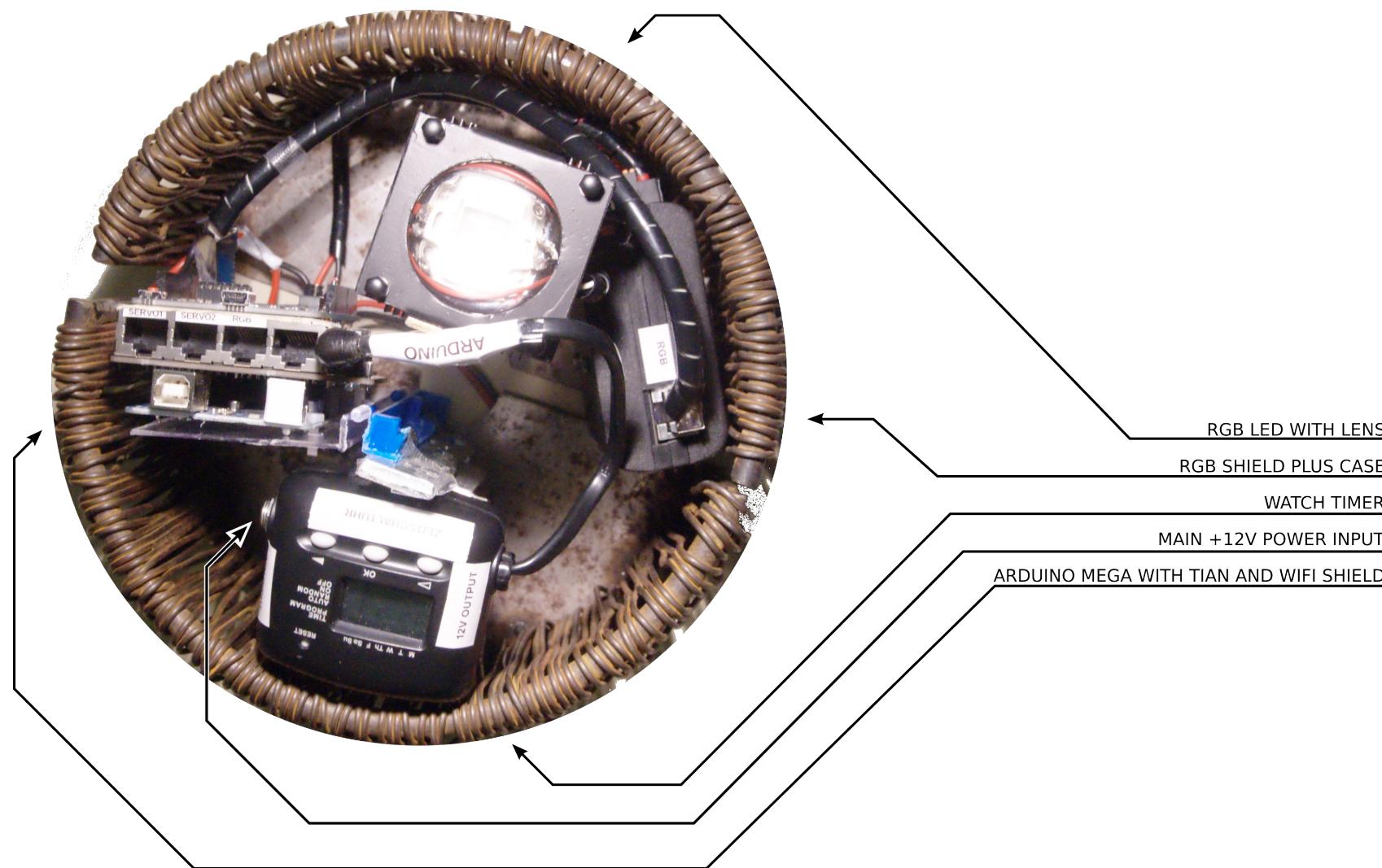
# Wifi Installation Overview



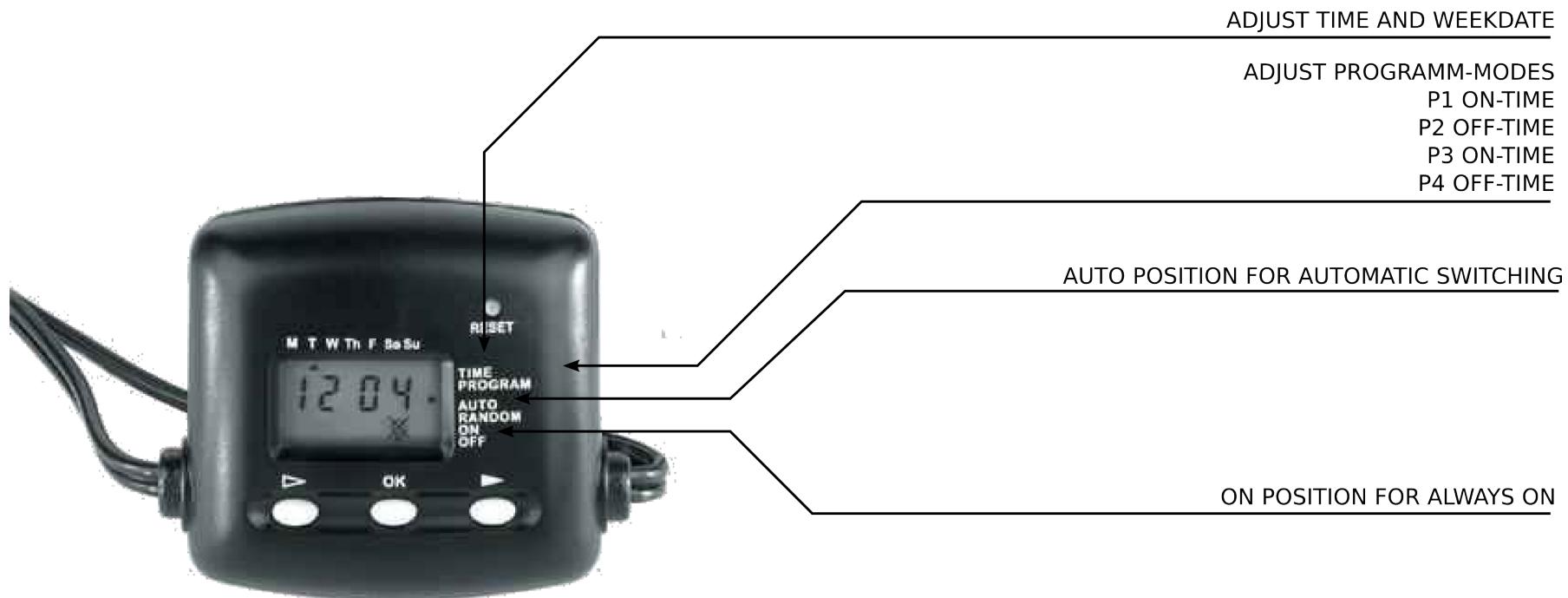
## Installation View



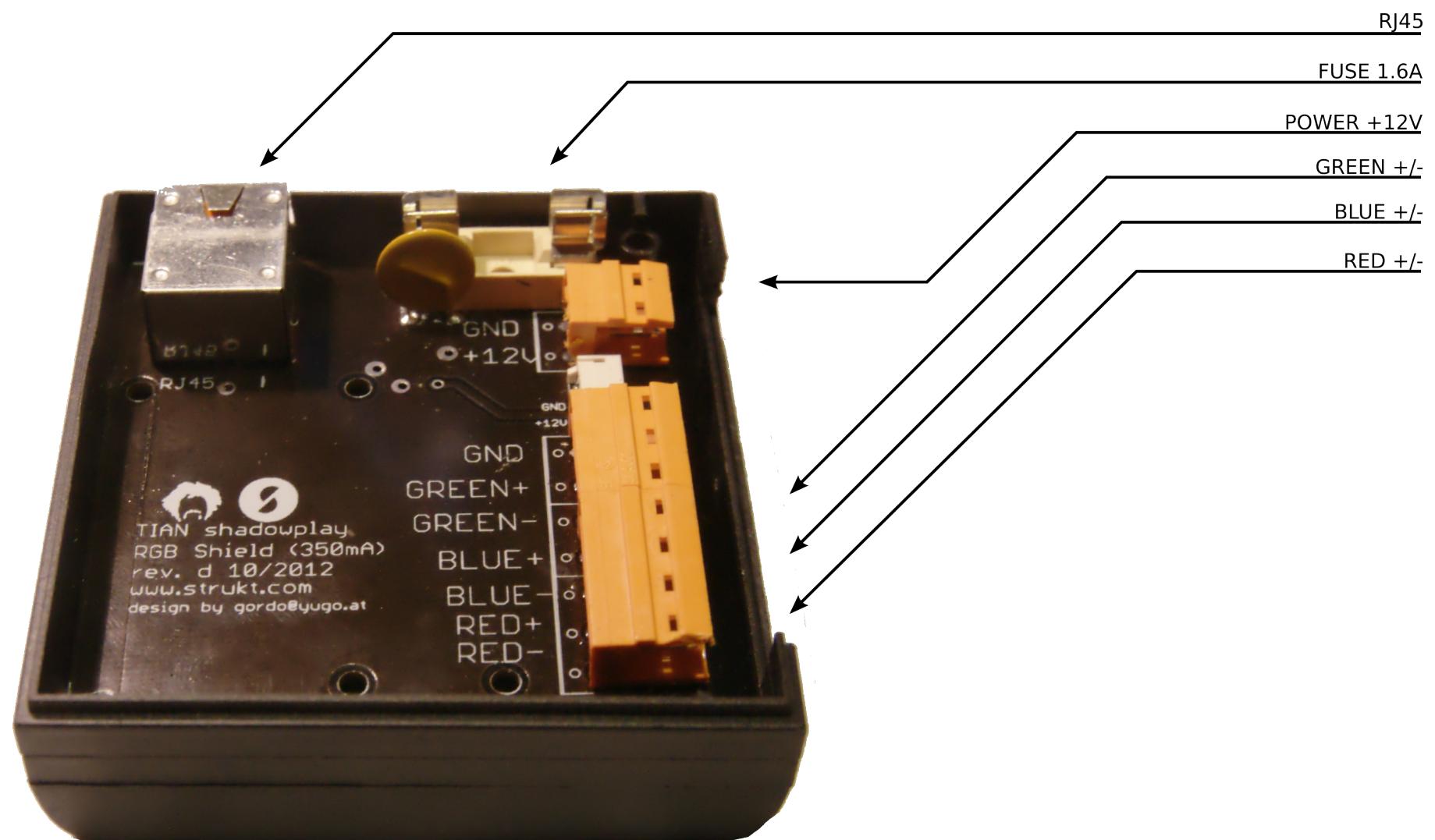
## Installation View - Photo



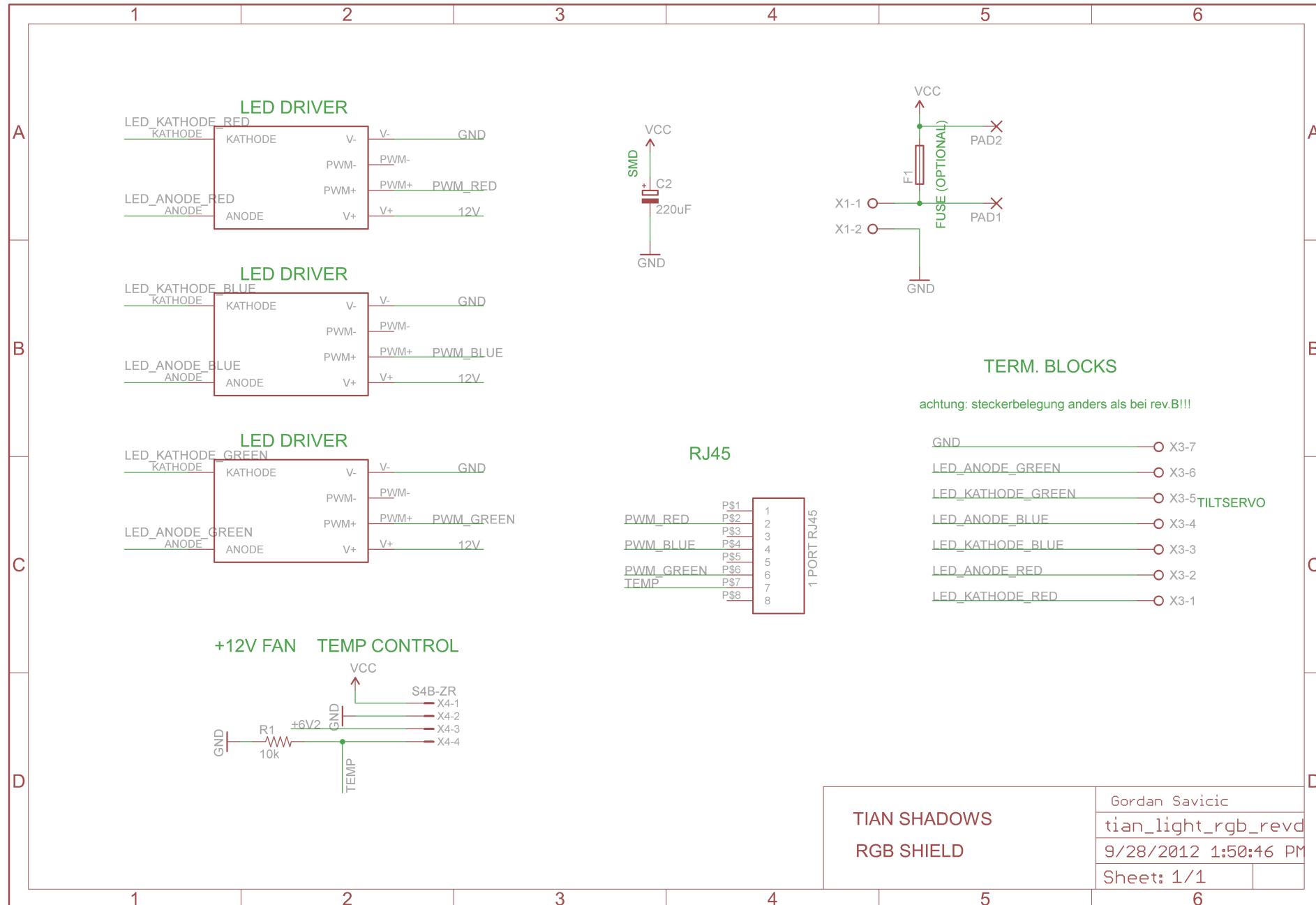
# Timer Watch aka Zeitschaltuhr



# RGB Shield

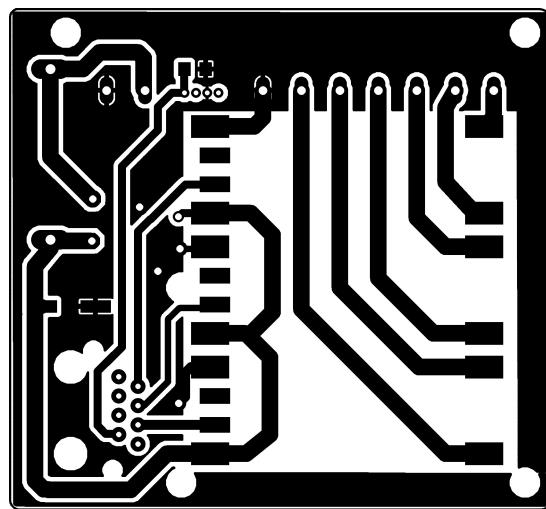


# RGB Shield Schematics

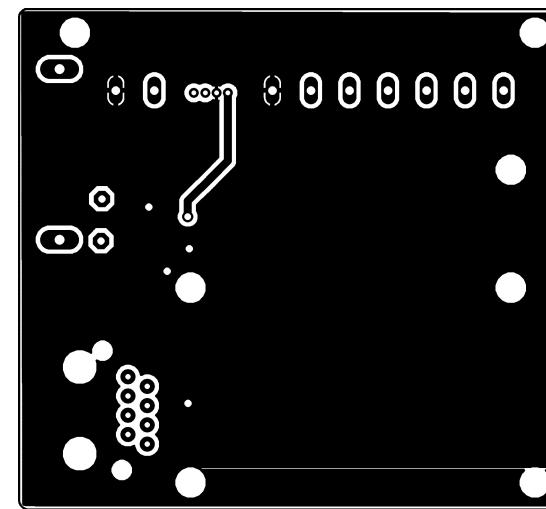


# RGB Shield copper layers

bottom layer



top layer



scale 1:1

## RGB Shield - partlist

Part	Value	Package	Library	Position (mm)	Ori entati on	Farnel I #
C2	220uF	PANASONIC_F	rcl	(4.826 26.416)	MRO	2079275
F1		SH22_5	fuse	(2.413 46.482)	R270	1597013
PAD1		2_15/1_0	wirepad	(7.9883 40.6146)	R0	
PAD2		2_15/1_0	wirepad	(7.8867 35.0647)	R0	
R1	10k	1206	SparkFun	(21.5138 57.7342)	MRO	2078962
U\$1	BUCK_LED_DRI VER	BUCK_LED_DRI VER	tian	(59.055 36.83)	MRO	*led-tech
U\$2	BUCK_LED_DRI VER	BUCK_LED_DRI VER	tian	(59.055 20.955)	MRO	*led-tech
U\$3	BUCK_LED_DRI VER	BUCK_LED_DRI VER	tian	(59.055 5.08)	MRO	*led-tech
U\$5	1PORT_ETHERNET	1PORT_ETHERNET	strukt	(7.62 13.335)	MRO	1284345
X1		W237-132	con-wago-508	(12.3825 54.2925)	R180	1866444
X3		W237-7P	con-wago-508	(45.72 54.2925)	R180	1866449
X4	S4B-ZR	S4B-ZR	con-jst	(22.3647 54.61)	R180	9491830

JST-ZH crimping contacts: Farnel I # 3357533

JST-ZH plugs: Farnel I # 3357569

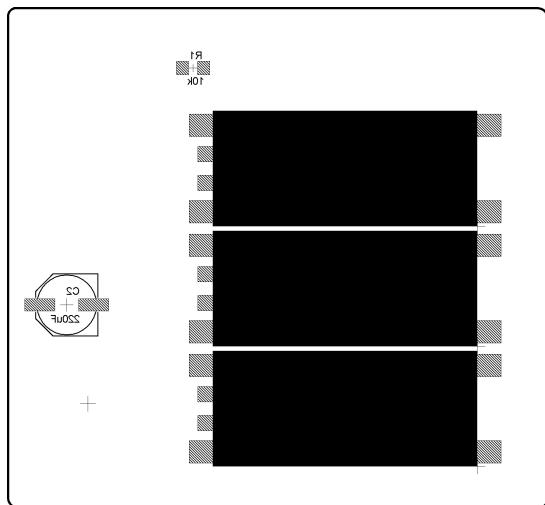
Weldmuel ler-Plugs 2 contacts: 1866531

Weldmuel ler-Plugs 3 contacts: 1866532

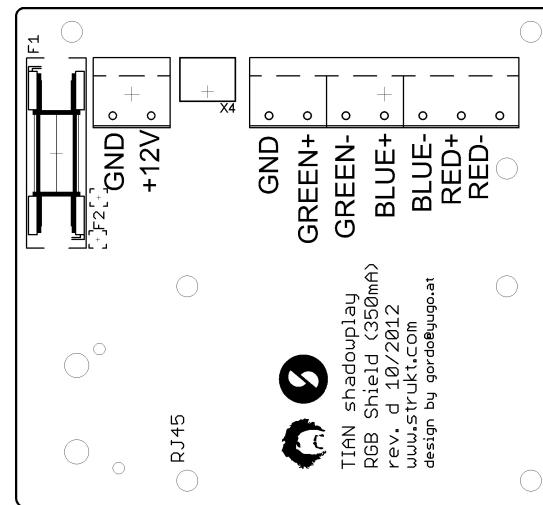
\* ([http://www.led-tech.de/de/LED-Control-List/KonstantstromquelIen/BUCK-Konstantstromquelle--350mA,-30V--LT-1550\\_118\\_119.html](http://www.led-tech.de/de/LED-Control-List/KonstantstromquelIen/BUCK-Konstantstromquelle--350mA,-30V--LT-1550_118_119.html))

# RGB Shield - mounting

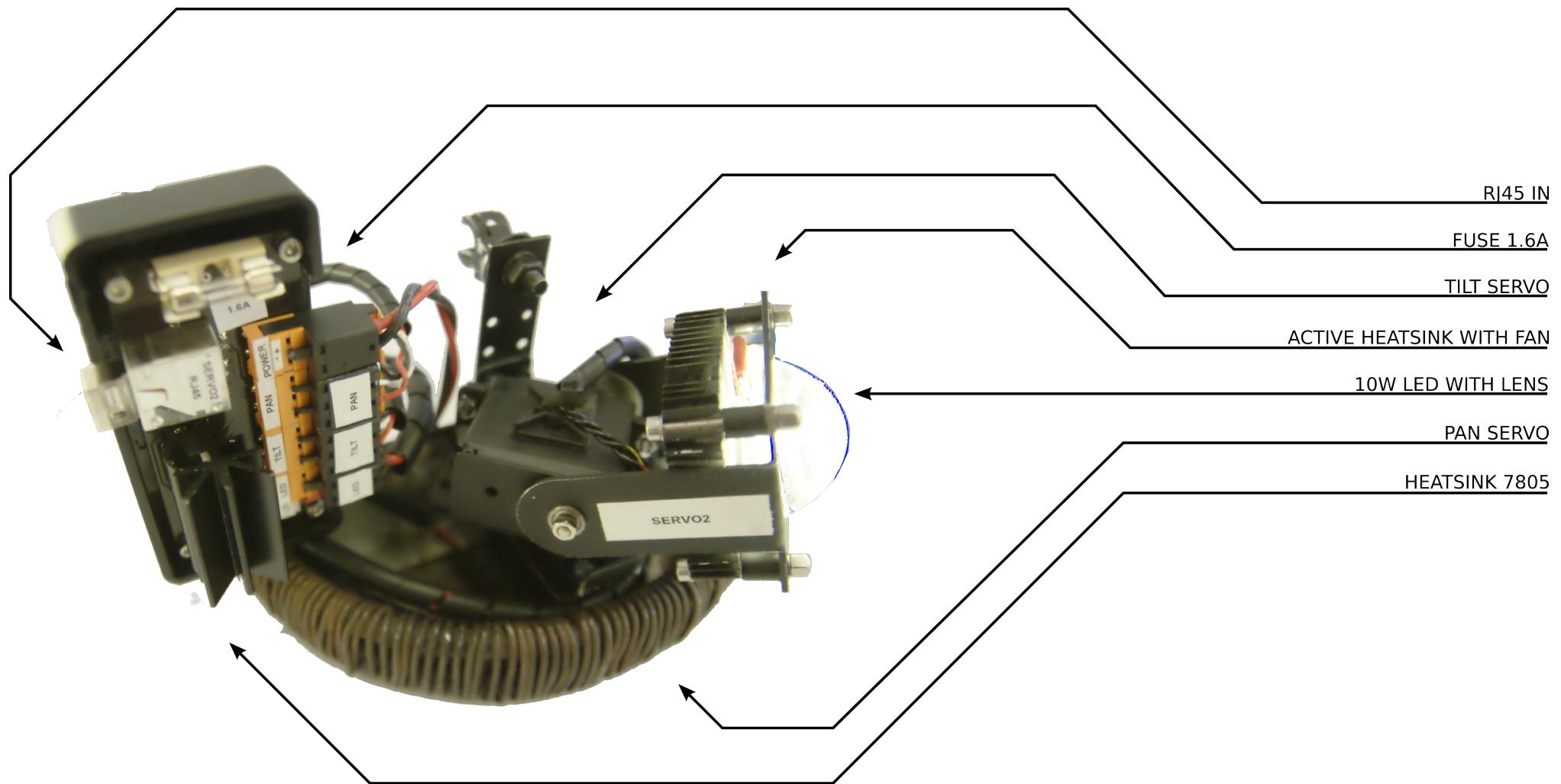
bottom layer



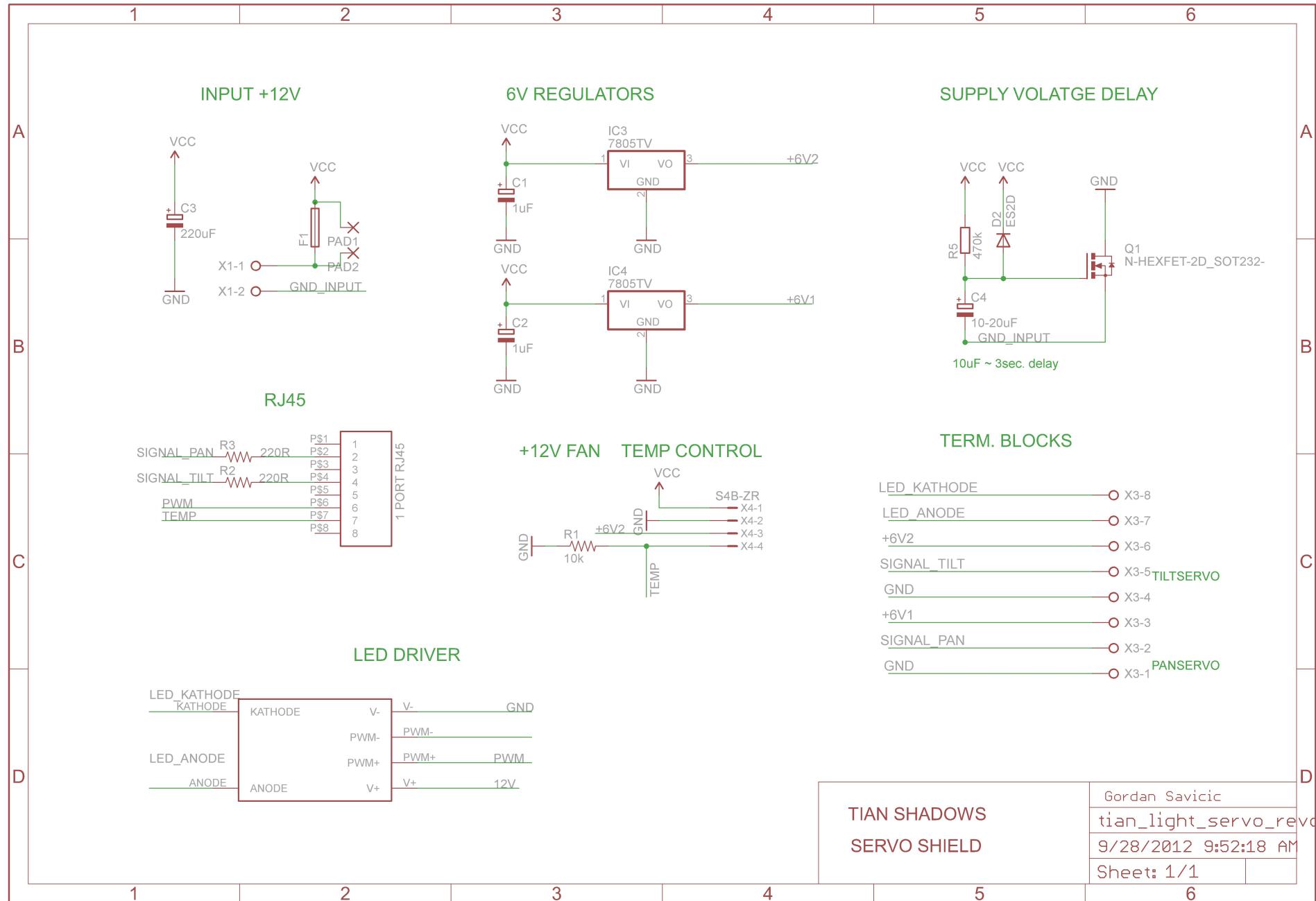
top layer



# Servo Shield

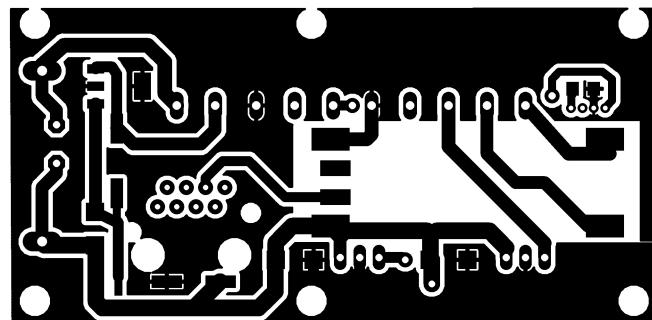


# Servo Shield Schematics

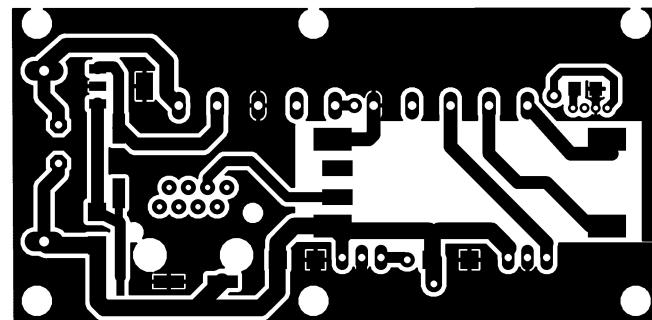


# Servo Shield copper layers

bottom layer



top layer



scale 1:1

# Servo Shield - partlist

Part	Value	Package	Library	Position (mil)	Ori entati on	Farnell Ordernumber
C1	1uF	SMC_C	resistor	(2287.5 325)	MR180	1135064
C2	1uF	SMC_C	resistor	(1487.5 325)	MR180	1135064
C3	220uF	PANASONIC_F	rcl	(962.5 212.5)	MR180	2079275
C4	10-20uF	PANASONIC_G	rcl	(562.5 837.5)	MR270	2079256
D2	ES2D	SMB	diode	(450 487.5)	MR90	2061435
F1		SH22_5	fuse	(175 862.5)	R270	1597013
IC3	7805TV	T0220V	linear	(2687.5 437.5)	R0	9664505
IC4	7805TV	T0220V	linear	(1825 437.5)	R0	9664505
PAD1		2_15/1_0	wirepad	(250 825)	R0	
PAD2		2_15/1_0	wirepad	(250 1025)	R0	
Q1	N-HEXFET-2D_SOT232-	SOT223	transistor-power	(575 1225)	MR90	2061415
R1	10k	1206	SparkFun	(2987.5 1212.5)	MRO	
R2	220R	1206	SparkFun	(1175 862.5)	R180	9233920
R3	220R	1206	SparkFun	(887.5 887.5)	R180	9233920
R5	470k	R1218	rcl	(512.5 237.5)	MRO	9240179RL
U\$1	BUCK_LED_DRI VER	BUCK_LED_DRI VER	tian	(3075 425)	MRO	*led-tech
U\$5	1PORT_ETHERNET	1PORT_ETHERNET	strukt	(925 450)	MR270	1284345
X1		W237-132	con-wago-508	(975 1150)	R0	1866444
X3		W237-8P	con-wago-508	(1987.5 1150)	R0	1866451
X4	S4B-ZR	S4B-ZR	con-j st	(3012.5 1112.5)	R180	9491830

Heatsink for 7805: Farnell # 1213434

2x Weidmüller Contacts (2pin): Farnell # 1866531

2x Weidmüller Contacts (3pin): Farnell # 1866532

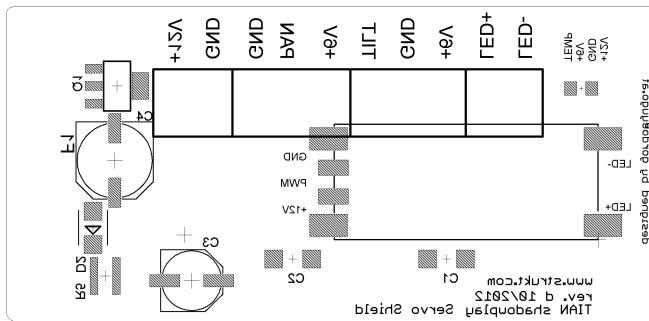
4x Crimping Contacts JST ZH: Farnell # 3357533

1x JST ZH Casing: Farnell # 3357569

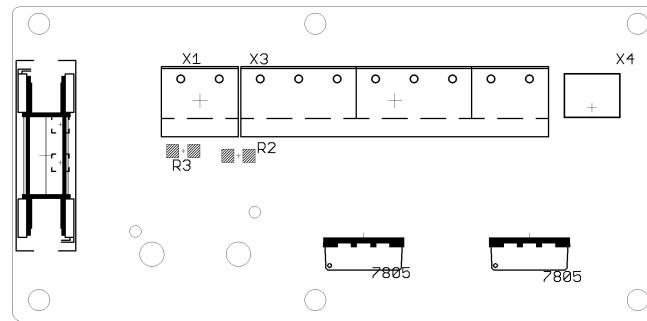
\* [http://www.led-tech.de/de/LED-Controlling/Konstantstromquelle/BUCK-Konstantstromquelle--1000mA,-30V--LT-1553\\_118\\_119.html](http://www.led-tech.de/de/LED-Controlling/Konstantstromquelle/BUCK-Konstantstromquelle--1000mA,-30V--LT-1553_118_119.html)

# Servo Shield - mounting

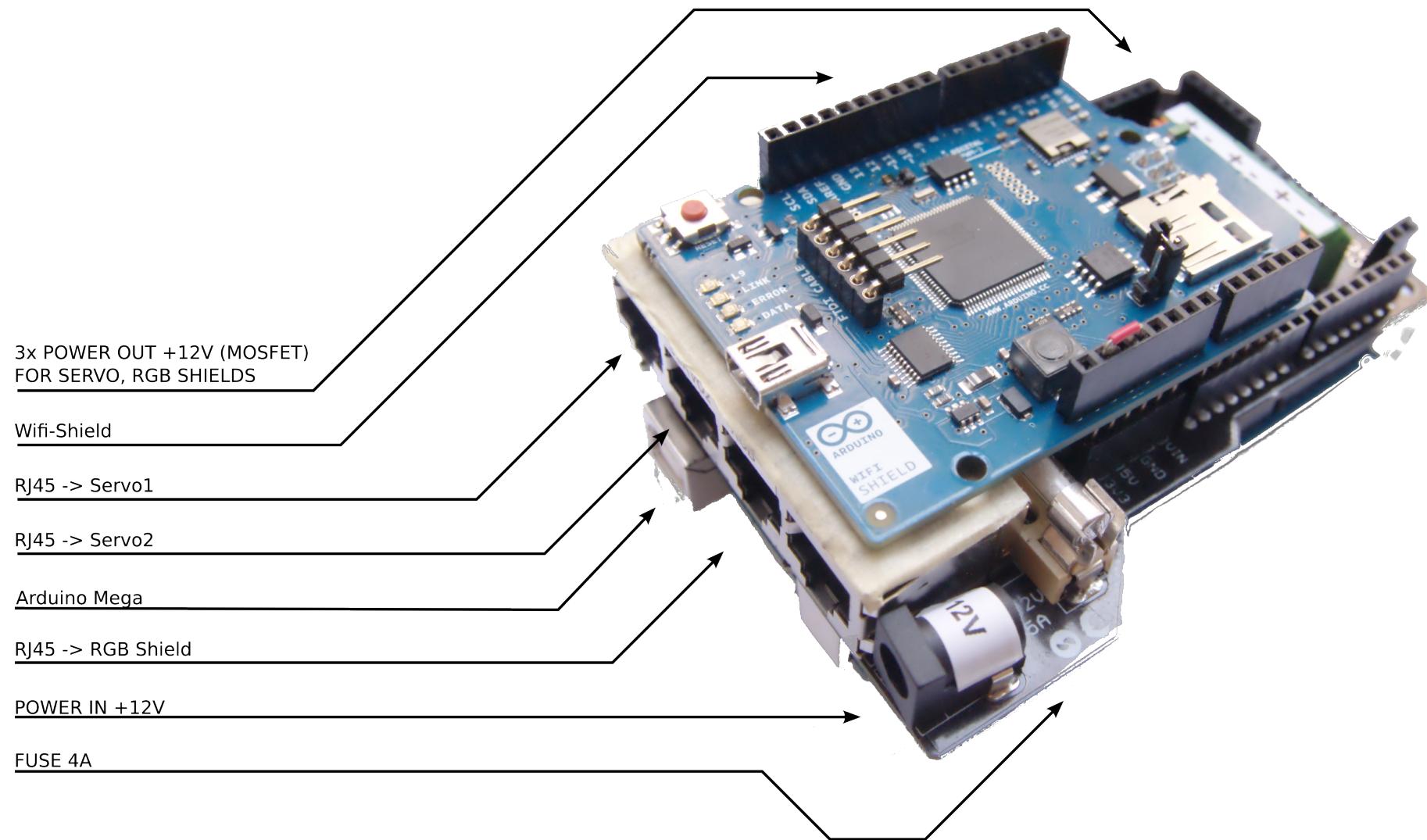
bottom layer



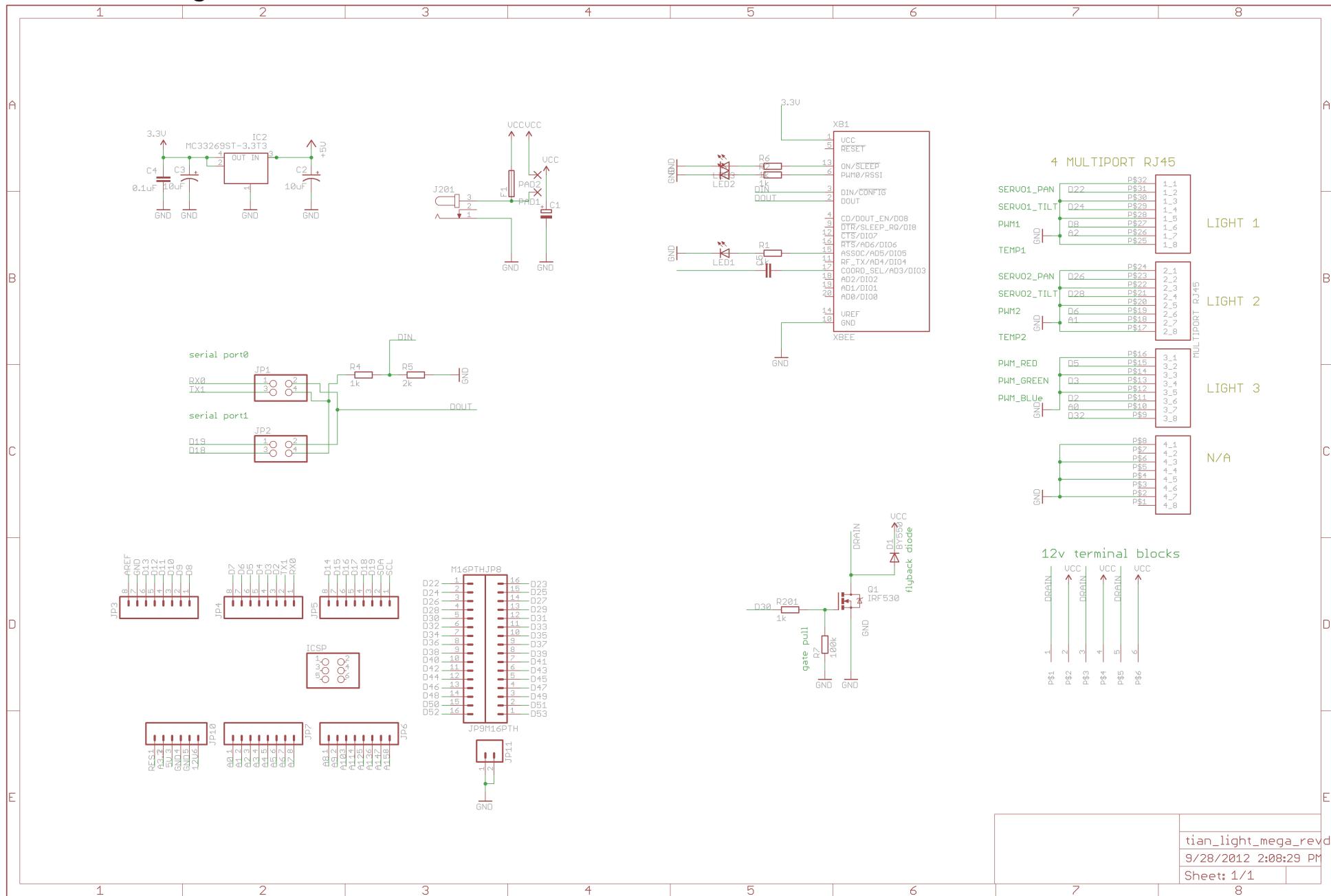
top layer



# Arduino Mega - Tian shield

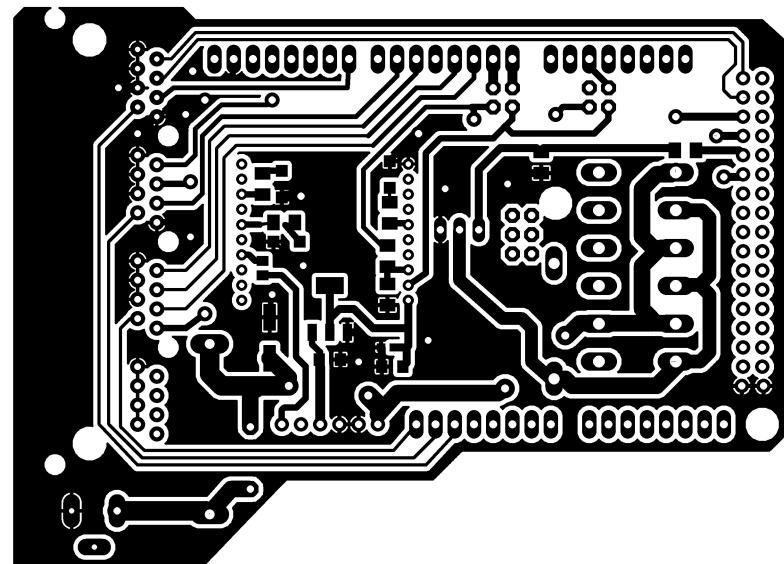


# Arduino Mega - Tian shield - schematics

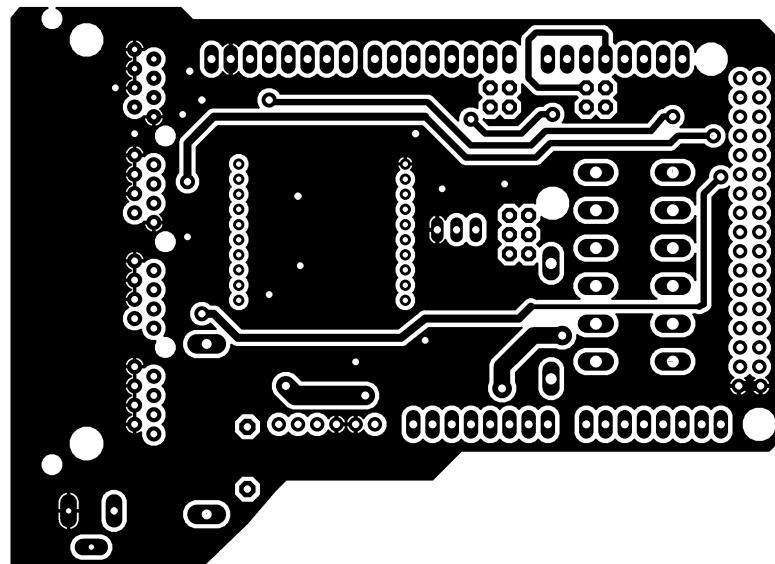


# Arduino Mega - Tian shield - copper layers

bottom layer



top layer



scale 1:1

# Arduino Mega - Tian shield - partlist

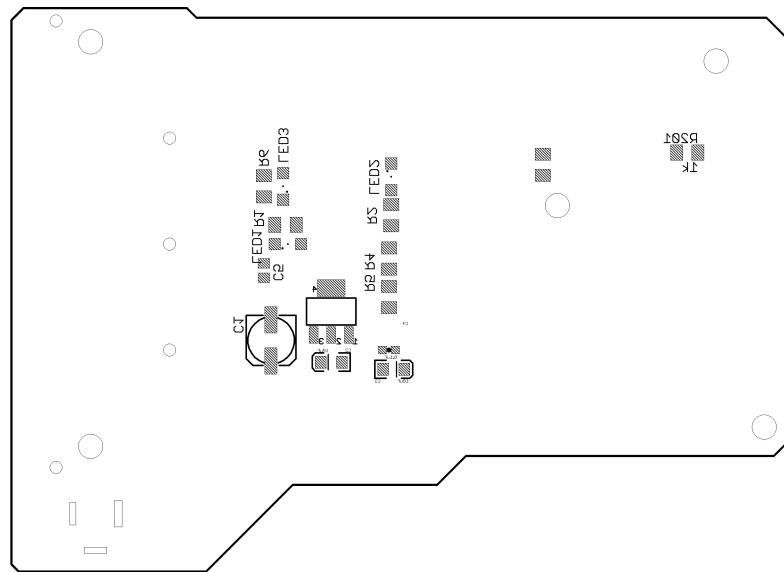
Part	Value	Package	Library	Position (mm)	Orientation	Farnell Ordernumber
C1		153CLV-0605	rcl	(20. 6375 14. 605)	MR270	1870656
C2	10uF	EIA3216	SparkFun	(28. 575 11. 7475)	MRO	1190107
C3	10uF	EIA3216	SparkFun	(36. 83 10. 795)	MR180	1190107
C4	0. 1uF	0603-CAP	SparkFun	(36. 195 13. 335)	MRO	1740614
C5		C0805	rcl	(19. 685 23. 8125)	MR90	1740614
D1	BY550	D027-15	diode	(58. 2295 16. 8275)	R270	4525929
F1		SH22_5	fuse	(12. 7 2. 54)	R90	1597013
IC2	MC33269ST-3. 3T3	SOT223	linear	(28. 575 18. 415)	MRO	1652295
ICSP		2X03	pinehead	(53. 975 28. 2575)	R270	1593421
J201		SPC4077	con-jack	(-2. 54 -8. 255)	R0	1654842
JP1		2X02	pinehead	(51. 435 46. 355)	R180	1593421
JP2		2X02	pinehead	(64. 135 46. 355)	R180	1593421
JP3		1X08_LONGPADS	SparkFun	(31. 115 51. 435)	R180	sparkfun PRT-10007
JP4		1X08_LONGPADS	SparkFun	(52. 705 51. 435)	R180	sparkfun PRT-10007
JP5		1X08_LONGPADS	SparkFun	(57. 785 51. 435)	R0	sparkfun PRT-10007
JP6		1X08_LONGPADS	SparkFun	(62. 865 3. 175)	R0	sparkfun PRT-10007
JP7		1X08_LONGPADS	SparkFun	(40. 005 3. 175)	R0	sparkfun PRT-10007
JP8	M16PTH	1X16	SparkFun	(85. 725 48. 895)	R270	sparkfun PRT-10007
JP9	M16PTH	1X16	SparkFun	(83. 185 48. 895)	R270	sparkfun PRT-10007
JP10		1X06	SparkFun	(22. 225 3. 175)	R0	sparkfun PRT-10007
JP11		1X02_LOCK	SparkFun	(83. 185 8. 255)	R0	sparkfun PRT-10007
LED1		SML1206	led	(22. 86 27. 305)	MR180	1226420
LED2		SML1206	led	(36. 5125 36. 195)	MR270	1226420
LED3		SML1206	led	(22. 225 34. 925)	MR90	1226420
PAD1	2, 15/1, 0	wirepad		(18. 0975 -5. 3975)	R0	
PAD2	2, 15/1, 0	wirepad		(18. 0975 2. 8575)	R0	
PORT201	4PORT_ETHERNET	4PORT_ETHERNET	strukt	(-3. 175 27. 305)	R270	1346693
Q1	IRF530	T0220BV	transistor-power	(45. 72 26. 3525)	R180	1218513
R1	1k	M1206	rcl	(22. 5425 29. 845)	MRO	2008380

## Arduino Mega - Tian shield - partlist

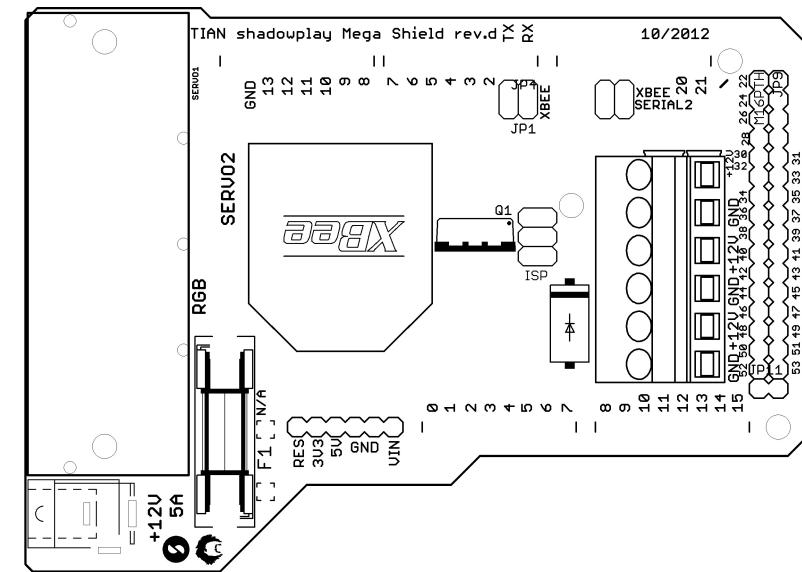
R2	1k	M1206	rcl	(36. 5125 31. 115)	MR270	2008380
R4	1k	M1206	rcl	(36. 195 25. 4)	MR270	2008380
R5	2k	M1206	rcl	(36. 195 20. 32)	MR270	1670231RL
R6	1k	M1206	rcl	(19. 685 34. 925)	MR270	2008380
R7	100k	M1206	rcl	(56. 515 37. 7825)	MR90	1697458
R201	1k	M1206	rcl	(75. 565 39. 37)	MRO	2008380
U\$3	MFKDSP6-5. 08	PHONEI X	con-phoeni x-MFKDSP	(78. 74 8. 89)	R90	1793053
XB1	XBEE	XBEE	maxstream	(27. 94 28. 575)	R180	1885919

# Arduino Mega Shield - mounting

bottom layer



top layer



# Arduino Mega Shield - sourcecode

```
#include <Servo.h>
#include <SPI.h>
#include <WiFi.h>
#include <math.h>

Servo Servo1_pan; // create servo object to control a servo
Servo Servo1_tilt;

Servo Servo2_pan; // create servo object to control a servo
Servo Servo2_tilt;

String currentLine = "";
String tweet = "";

// PINS
const int mosfet12 = 30;
const int Light1PWM = 8;
const int Light2PWM = 6;
const int RedPWM = 5;
const int GreenPWM = 2;
const int BluePWM = 3;

// MAXIMUM ratings
const int ServoPanLeftmax = 25;
const int ServoPanRightmax = 160;
const int ServoTiltUpmax = 165;
const int ServoTiltDownmax = 101;

const int ServoTiltCenter = (ServoTiltUpmax + ServoTiltDownmax) /2;
const int ServoPanCenter = (ServoPanLeftmax + ServoPanRightmax) /2;

float radius = 68;

float step1 = 0;
float stepc1 = 0;
float step2 = PI;
float stepc2 = 0;
float rgbStep = 0.0;
float rgbSpeed = 0.0;
float rStep=0.0;
float gStep=0.0;
float bStep=0.0;

float sinus1=0;
float sinus2=PI/4;

float speed1=0.00205f;
float speedc1=0.00304f;
float speed2=0.00205f;
float speedc2=0.000304f;
```

# Arduino Mega Shield - sourcecode

```
float dlinspeed=0.0002f;
float ddlinspeed=0.00001f;

int oldX1=0;
int oldX2=0;
int oldY1=0;
int oldY2=0;
bool ean firstPositon=true;
int sequence=-2;
int sTimer=0;
float lin1=0;
float lin2=0.5;
float dlin1=0.01;
float dlin2=0.01;

int rrr=0;
int ggg=0;
int bbb=0;

int MAXMAX_BRIGHT = 255;
int MINMAX_BRIGHT = 0;
int MAX_BRIGHT = MAXMAX_BRIGHT;

bool ean hasChecked=false;
int darkTimer=0;
int darkTimer2=0;
int lastBright1=0;
int lastBright2=0;

int maxservo = 165;
int minservo = 15;
int red, green, blue;

int pausi=0;
int servo_delay=0;
bool ean servo_attached=false;

int LightBrightness = 0;

int server_timeout = 0;

// WIFI STUFF
char ssid[] = "ff20d";
char pass[] = "xxxxxxxxxxxxxxxxxxxxxx";

// server intian is 10.10.0.31
//char nameserver[] = "10.10.0.91";
char nameserver[] = "10.10.0.91";
int keyIndex = 0;
```

# Arduino Mega Shield - sourcecode

```
bool can_wifi_shield = true;
bool can_wifi_enable = true;
bool can_isConnected=false;

int status = WL_IDLE_STATUS;
IPAddress server(10, 10, 0, 31);
//IPAddress server(10, 10, 0, 91);
WiFiClient client;

void setup() {
    currentLine.reserve(256);
    tweet.reserve(100);
    Serial.begin(19200);
    // set the digital pin as output:
    pinMode(mosfet12, OUTPUT);
    pinMode(Light1PWM, OUTPUT);
    pinMode(Light2PWM, OUTPUT);
    pinMode(BluePWM, OUTPUT);
    pinMode(RedPWM, OUTPUT);
    pinMode(GreenPWM, OUTPUT);
    set_brightness(5);

    digitalWrite(mosfet12, LOW);

    if (wifi_enable == true) {
        // check for the presence of the shield:
        if (WiFi.status() == WL_NO_SHIELD) {
            Serial.println("WiFi shield not present");
            wifi_shield = false;
        }
        if (wifi_shield == true) {
            // attempt to connect to WiFi network:
            while (status != WL_CONNECTED) {
                Serial.print("Attempting to connect to SSID: ");
                Serial.println(ssid);
                // Connect to WPA/WPA2 network. Change this line if using open or WEP network:
                status = WiFi.begin(ssid, pass);
                // wait 10 seconds for connection:
                delay(9000);
            }
            Serial.println("Connected to WiFi");
            printWifiStatus();
        }
        //Serial.println("\nStarting connection to server...");
        // if you get a connection, report back via serial:
        if (client.connect(nameserver, 80)){
            isConnected=true;
            sequence=-1;
            Serial.println("connected");
            // Make a HTTP request:
            client.println("GET / HTTP/1.1");
        }
    }
}
```

# Arduino Mega Shield - sourcecode

```
client.println("Host: local host");
client.println("Connection: close");
client.println();
}
}

void make_request() {
//Serial.println("\nStarting connection to server...");
if (client.connect(server, 80)) {
Serial.println("connected");
client.println("GET / HTTP/1.1");
client.println("Host: local host");
client.println("Connection: close");
client.println();
server_timeout=0;
isConnected=true;
} else{
isConnected=false;
server_timeout++;
if (server_timeout>5){
sequence=0;
server_timeout=0;
}
Serial.println("timeout");
}
}

int getIntFromString(String s){
char ca[s.length()+1];
s.toCharArray(ca, (s.length()+1));
return atoi(ca);
}

void server_check() {
make_request();
while (client.available()) {
char inChar = client.read();
// add incoming byte to end of line:
//Serial.write(inChar);
currentLine += inChar;

// if you get a newline, clear the line:
if (inChar == '\n') {
currentLine = "";
}
if ( currentLine.endsWith("</html >")) {
String content = currentLine.substring(currentLine.indexOf("<body>")+6, currentLine.indexOf("</body>"));
int kommalndex1 = content.indexOf(",");
int kommalndex2 = content.indexOf(", ", kommalndex1+1);
int kommalndex3 = content.indexOf(", ", kommalndex2+1);
int kommalndex4 = content.indexOf(", ", kommalndex3+1);
```

# Arduino Mega Shield - sourcecode

```
if(kommalndex1!=-1){
    String s = content.substring(0, kommalndex1);
    String br = content.substring(kommalndex1+1, kommalndex2);
    String rr = content.substring(kommalndex2+1, kommalndex3);
    String gg = content.substring(kommalndex3+1, kommalndex4);
    String bb = content.substring(kommalndex4+1);
    sequence=getIntFromString(s);
    MAX_BRIGHT=getIntFromString(br);
    rrr=getIntFromString(rr);
    ggg=getIntFromString(gg);
    bbb=getIntFromString(bb);
    //Serial.println();
    //Serial.print("sequence: ");
    Serial.println(sequence);
    //Serial.print("brightness: ");
    //Serial.println(MAX_BRIGHT);
    //Serial.print("red: ");
    //Serial.println(rr);
    //Serial.print("green: ");
    //Serial.println(gg);
    //Serial.print("blue: ");
    //Serial.println(bb);
}
}

void printWifiStatus() {
    Serial.print("SSID: ");
    Serial.println(WiFi.SSID());
}

// print your WiFi shield's IP address:
IPAddress ip = WiFi.localIP();
Serial.print("IP Address: ");
Serial.println(ip);

long rssi = WiFi.RSSI();
Serial.print("signal strength (RSSI):");
Serial.print(rssi);
Serial.println(" dBm");
}

void set_brightness(int brightness) {
    int new_brightness = map(brightness, 0, 100, 255, 0);
    analogWrite(BluePWM, new_brightness);
    analogWrite(RedPWM, new_brightness);
    analogWrite(GreenPWM, new_brightness);
    analogWrite(Light1PWM, new_brightness);
    analogWrite(Light2PWM, new_brightness);
    return;
}
```

# Arduino Mega Shield - sourcecode

```
void rgbled(int r1, int g1, int b1){  
    //b1=(b1*3)/5;  
    g1=(g1*5)/6;  
    //r1=(r1*2)/5;  
    int r2 = map(r1, 0, 255, MAXMAX_BRIGHT, MAXMAX_BRIGHT-MAX_BRIGHT);  
    int g2 = map(g1, 0, 255, MAXMAX_BRIGHT, MAXMAX_BRIGHT-MAX_BRIGHT);  
    int b2 = map(b1, 0, 255, MAXMAX_BRIGHT, MAXMAX_BRIGHT-MAX_BRIGHT);  
  
    analogWrite(GreenPWM, g2);  
    analogWrite(BluePWM, b2);  
    analogWrite(RedPWM, r2);  
}  
  
void bright1(int brightness){  
    if(brightness<15) brightness=15;  
    lastBright1=brightness;  
    int new_brightness = map(brightness, 0, 255, MAXMAX_BRIGHT, MAXMAX_BRIGHT-MAX_BRIGHT);  
    analogWrite(Light1PWM, new_brightness);  
}  
void bright2(int brightness){  
    if(brightness<15) brightness=15;  
    lastBright2=brightness;  
    int new_brightness = map(brightness, 0, 255, MAXMAX_BRIGHT, MAXMAX_BRIGHT-MAX_BRIGHT);  
    analogWrite(Light2PWM, new_brightness);  
}  
  
void powerServoOn(bool enable){  
    digitalWrite(mosfet12, HIGH);  
    // delay the activation of the servos  
    if(servo_delay>5000 && servo_attached == false) {  
        //Serial.println("switching mosfet on, 5sec. delay for servo.attach");  
        Servo1_pan.write(90);  
        Servo1_tilt.write(110);  
        Servo2_pan.write(90);  
        Servo2_tilt.write(110);  
        Servo1_pan.attach(22);  
        Servo1_tilt.attach(24);  
        Servo2_pan.attach(26);  
        Servo2_tilt.attach(28);  
        servo_attached=true;  
        servo_delay=0;  
    }  
    else {  
        servo_delay++;  
        if (servo_delay > 10000){  
            servo_delay=0;  
            //Serial.println("resetting counter");  
        }  
    }  
}
```

# Arduino Mega Shield - sourcecode

```
}

void powerServoOff(){
  //Serial.println("detaching servos, turning off mosfet");
  servo_attached=false;
  servo_delay=0;
  Servo1_pan.detach();
  Servo1_tilt.detach();
  Servo2_pan.detach();
  Servo2_tilt.detach();
  digitalWrite(mosfet12, LOW);
}

void writePan(Servo servo, int svalue){
  if(svalue < ServoPanLeftmax)svalue=ServoPanLeftmax;
  if(svalue > ServoPanRightmax)svalue=ServoPanRightmax;
  servo.write(svalue);
}

void writeTilt(Servo servo, int svalue){
  if(svalue < ServoTiltDownmax)svalue=ServoTiltDownmax;
  if(svalue > ServoTiltUpmax)svalue=ServoTiltUpmax;
  servo.write(svalue);
}

void draw_circle(Servo servo_pan, Servo servo_tilt, int r, int i)
{
  int x = float(ServoPanCenter) + float(r) * cos(i);
  int y = float(ServoTiltCenter) - float(r) * sin(i);
  writePan(servo_pan, x);
  writeTilt(servo_tilt, y);

  return;
}

// move the servo a given amount
void moveServo(Servo servo, int delta) {
  int previousValue = servo.read();
  int newValue = previousValue + delta;
  if (newValue > maxservo || newValue < minservo) {
    return;
  }
  servo.write(newValue);
  Serial.println(newValue);
}

void color (unsigned int red, unsigned int green, unsigned int blue) {
  analogWrite(RedPWM, 255-red);
  analogWrite(BluePWM, 255-blue);
  analogWrite(GreenPWM, 255-green);

  return;
}
```

# Arduino Mega Shield - sourcecode

```
void rgbBlink(float blink1, float blink2){  
    rgbStep+=blink1;  
    rgbSpeed+=blink2;  
    int rr=float(rrr)*(sin(rgbSpeed)+1)/2;  
    int gg=float(ggg)*(sin(rgbSpeed)+1)/2;  
    int bb=float(bbb)*(sin(rgbSpeed)+1)/2;  
    rgbled(rr, gg, bb);  
}  
  
void blueBlink(float blink1, float blink2){  
    rgbStep+=blink1;  
    rgbSpeed+=blink2;  
    int rr=float(0)*(sin(rgbSpeed)+1)/2;  
    int gg=float(30)*(sin(rgbSpeed)+1)/2;  
    int bb=float(255)*(sin(rgbSpeed)+1)/2;  
    rgbled(rr, gg, bb);  
}  
  
void rgbColorBlink(float blink1, float blink2){  
    rgbStep+=blink1;  
    rgbSpeed+=blink2;  
    int rr = 127+127.0f*sin(rgbStep);  
    int gg = 127+127.0f*cos(rgbStep);  
    int bb = 127+127.0f*sin(rgbStep+(PI/2));  
    rr=float(rr)*(sin(rgbSpeed)+1)/2;  
    gg=float(gg)*(sin(rgbSpeed)+1)/2;  
    bb=float(bb)*(sin(rgbSpeed)+1)/2;  
    rgbled(rr, gg, bb);  
}  
  
void gBlink(float blink1){  
    gStep+=blink1;  
    int gg = 127+127.0f*cos(gStep);  
    rgbled(0, gg, 0);  
}  
void rBlink(float blink1){  
    gStep+=blink1;  
    int rr = 127+127.0f*cos(gStep);  
    rgbled(rr, 0, 0);  
}  
  
void movement0(){  
    step1+=speed1;  
    step2+=speed2;  
    stepc1+=speedc1;  
    stepc2+=speedc2;  
    int x1 = float(ServoPanCenter) + (radius*0.8) * cos(step1);  
    int y1 = float(ServoTiltCenter-20) - (radius*0.14) * sin(stepc1);  
    int x2 = float(ServoPanCenter) + (radius*0.76) * cos(-step2);  
    int y2 = float(ServoTiltCenter-20) - (radius*0.16) * sin(-stepc2);  
}
```

# Arduino Mega Shield - sourcecode

```
wri tePan(Servo1_pan, x1);
wri tePan(Servo2_pan, x2);
wri teTilt(Servo1_tilt, y1);
wri teTilt(Servo2_tilt, y2);
}

void movement2(){
    step1+=speed1/1.35;
    step2+=speed2/1.35;

    float cosStep1=cos(step1);
    float cosStep2=cos(step2);
    float sinStep1=sin(step1);
    float sinStep2=sin(step2);
    float cosQ1=cosStep1*cosStep1;
    float cosQ2=cosStep2*cosStep2;

    int x1 = float(ServoPanCenter) + radius * cosStep1;
    int y1 = float(ServoTiltCenter-15) - (radius*0.02) * sinStep1;
    int x2 = float(ServoPanCenter) + radius * cosStep2;
    int y2 = float(ServoTiltCenter-25) - (radius*0.02) * sinStep2;

    wri tePan(Servo1_pan, x1);
    wri teTilt(Servo1_tilt, y1);
    wri tePan(Servo2_pan, x2);
    wri teTilt(Servo2_tilt, y2);

    //movement1 is coupled with light
    lastBright1 = 255 - 255.0 * cosQ1;
    if(sinStep1>=0)lastBright1=0;
    bright1(lastBright1);
    lastBright2 = 255 - 255.0 * cosQ2;
    if(sinStep2>=0)lastBright2=0;
    bright2(lastBright2);
}

void movement1(){
    step1+=speed1/3.55;
    step2+=speed2/3.55;

    float cosStep1=cos(step1);
    float cosStep2=cos(step2);
    float sinStep1=sin(step1);
    float sinStep2=sin(step2);
    float cosQ1=cosStep1*cosStep1;
    float cosQ2=cosStep2*cosStep2;

    int x1 = float(ServoPanCenter) + radius * cosStep1;
    int y1 = float(ServoTiltCenter-15) - (radius*0.02) * sinStep1;
    int x2 = float(ServoPanCenter) + radius * cosStep2;
    int y2 = float(ServoTiltCenter-25) - (radius*0.02) * sinStep2;

    wri tePan(Servo1_pan, x1);
```

# Arduino Mega Shield - sourcecode

```
writeTilt(Servo1_tilt, y1);
writePan(Servo2_pan, x2);
writeTilt(Servo2_tilt, y2);

//movement1 is coupled with light
lastBright1 = 255 - 255.0 * cos01;
if(signStep1>=0) lastBright1=0;
bright1(lastBright1);
lastBright2 = 255 - 255.0 * cos02;
if(signStep2>=0) lastBright2=0;
bright2(lastBright2);
}

void moveSlowServo(Servo currentServo, boolean panServo, int previous, int next, int servoTime){
    if(next - previous < 0) {
        for (int i=previous; i>next; i--) {
            delay(servoTime);
            //Serial.println(i);
            if (panServo == 1) {
                writePan(currentServo, i);
            }
            else {
                writeTilt(currentServo, i);
            }
        }
    }
    else if (next - previous > 0) {
        for (int i=previous; i<next; i++) {
            delay(servoTime);
            if (panServo == 1) {
                writePan(currentServo, i);
            }
            else {
                writeTilt(currentServo, i);
            }
        }
    }
}

void randomServo(){
    int x1 = random(ServoPanLeftmax, ServoPanRightmax);
    int x2 = random(ServoPanLeftmax, ServoPanRightmax);
    int y1 = random(ServoTiltDownmax, ServoTiltUpmax);
    int y2 = random(ServoTiltDownmax, ServoTiltUpmax);

    int previousx1 = Servo1_pan.read();
    int previousy1 = Servo1_tilt.read();
    int previousx2 = Servo2_pan.read();
    int previousy2 = Servo2_tilt.read();

    moveSlowServo(Servo1_pan, 1, previousx1, x1, 25);
    moveSlowServo(Servo2_pan, 1, previousx2, x2, 25);
}
```

# Arduino Mega Shield - sourcecode

```
moveSlowServo(Servo1_tilt, 0, previousy1, y1, 25);
moveSlowServo(Servo2_tilt, 0, previousy2, y2, 25);
//writePan(Servo1_pan, x1);
//writePan(Servo2_pan, x2);
//writeTilt(Servo1_tilt, y1);
//writeTilt(Servo2_tilt, y2);
}

void checkDarkness(){
if(lastBrightness+lastBrightness<16){
    if(!hasChecked){
        //darkTimer=0;
        darkTimer2=0;
        server_check();
        hasChecked=true;
    }else{
        darkTimer2++;
        int limit=15000;
        if(sequence==0) limit=15500;
        if(darkTimer2>limit){
            darkTimer2=0;
            hasChecked=false;
        }
    }
}else{
    darkTimer++;
    hasChecked=false;
    if(darkTimer>20000){
        darkTimer=0;
        server_check();
    }
}
}

void loop()
{

//set_brightness(LightBrightness);
//if(isConnected){
    if(sequence>=0){
        checkDarkness();
    }
//}
//}

if(sequence==2){
    digitalWrite(mosfet12, HIGH);
    digitalWrite(0, 0.0043);
    sTimer++;
    if(sTimer>8000){
        sTimer=0;
        //sequence=1;
    }
}
```

# Arduino Mega Shield - sourcecode

```
MAX_BRI GHT=128;
server_check();

}else if(sequence==1){
  digitalWrite(mosfet12, HIGH);
  gBlink(0.0043);
  sTimer++;
  if(sTimer>8000){
    sTimer=0;
    //sequence=1;
    server_check();
  }
}else if(sequence==0){
  powerServoOff();
  rgblEd(0, 0, 0);
}else if(sequence==4){
  powerServoOn();
  movement1();
  siNus1+=0.0012;
  siNus2-=0.0023;

  rgblink(0.0042, 0.0005);
}else if(sequence==2){
  powerServoOn(0);
  siNus1+=0.0012;
  siNus2-=0.0023;
  movement2();
  rgblEd(random(256), random(256), random(256));
  rgblink(0.0142, 0.0087);
  //rgblEd(0, 20, 255);

}else if(sequence==1){
  powerServoOn(0);

  dlinSpeed+=dliSpeed;
  if(dlinSpeed<0.0001)dliSpeed=0.00003;
  if(dlinSpeed>0.001)dliSpeed=-0.00003;

  if(pausi==0){
    linN1+=dlinN1;
    if(linN1>1)dlinN1=-dlinSpeed;
    if(linN1<-0.05)dlinN1=dlinSpeed;
    linN2+=dlinN2;
    if(linN2>1)dlinN2=-dlinSpeed;
    if(linN2<-0.05){
      dlinN2=dlinSpeed;
      pausi++;
    }
  }else{
    pausi++;
    if(pausi>10){
      randomServo();
    }
  }
}
```

# Arduino Mega Shield - sourcecode

```
        pausi =0;
    }

float hlin1=hlin1;
float hlin2=hlin2;
if(hlin1<0){
    hlin1=0;
}
if(hlin2<0){
    hlin2=0;
}
lastBright1 = 255.0 * hlin1;
lastBright2 = 255.0 * hlin2;
bright1(lastBright1);

int lastBright3 = 255-lastBright1;
bright2(lastBright1);

lastBright1=lastBright1/2;

blueBlink(0.0142, 0.000727);
//rgbBlink(0.0142, 0.0057);

}else if(sequence==3){
powerServoOn(0);
lin1+=dlin1;
if(lin1>1)dlin1=-0.0001;
if(lin1<0)dlin1=0.0001;
lin2+=dlin2;
if(lin2>1)dlin2=-0.0001;
if(lin2<0)dlin2=0.0001;

lastBright1 = 255 - 255.0 * lin1;
lastBright2 = 255 - 255.0 * lin2;
bright1(lastBright1);

int lastBright3 = 255-lastBright1;
bright2(lastBright1);

lastBright1=lastBright1/2;

rgbBlink(0.0142, 0.0027);

//rgbled(lastBright1, lastBright1, lastBright1);
movement0();
}
```

# Arduino Mega Shield - sourcecode

```
if (Serial.available() >0) {  
    byte incoming = Serial.read();  
    // turn off 12V power for shields  
    if (incoming == 'q') {  
        digitalWrite(mosfet12, LOW);  
        Serial.println("mosfet off");  
    }  
    // turn on  
    else if (incoming == 'o') {  
        digitalWrite(mosfet12, HIGH);  
    }  
  
    else if(incoming == '0'){  
        sequence=0;  
    }else if(incoming == '1'){  
        sequence=1;  
    }else if(incoming == '2'){  
        sequence=2;  
    }else if(incoming == '3'){  
        sequence=3;  
    }else if(incoming == '4'){  
        sequence=4;  
    }  
  
    //set speed  
    else if (incoming == '5'){  
        speed1-=0.00013;  
        speed2-=0.00013;  
    }  
    else if (incoming == '6'){  
  
        speed1+=0.00013;  
        speed2+=0.00013;  
        Serial.println(speed1);  
        Serial.println(speed2);  
    }else if(incoming == '7'){  
        MAX_BRIGHT-=5;  
        if(MAX_BRIGHT<MINMAX_BRIGHT)MAX_BRIGHT=MINMAX_BRIGHT;  
    }else if(incoming == '8'){  
        MAX_BRIGHT+=5;  
        if(MAX_BRIGHT>MAXMAX_BRIGHT)MAX_BRIGHT=MAXMAX_BRIGHT;  
    }  
    // motor circles  
    else if (incoming == 'm') {  
        for (int j=0; j<100; j+=20) {  
            set_brightness(j);  
        }  
    }  
    // motors right 1step  
    else if (incoming == 's') {  
        moveServo(Servo2_pan, 1);  
        moveServo(Servo1_pan, 1);  
    }
```

# Arduino Mega Shield - sourcecode

```
}

// motors left 1step
else if (incoming == 'a') {
    moveServo(Servo2_pan, -1);
    moveServo(Servo1_pan, -1);
}
// motors up 1step
else if (incoming == 'w') {
    moveServo(Servo2_tilt, 1);
    moveServo(Servo1_tilt, 1);
}
// motors down 1step
else if (incoming == 'z') {
    moveServo(Servo2_tilt, -1);
    moveServo(Servo1_tilt, -1);
}
// increase brightness
else if (incoming == 'd') {
    LightBrightness +=1;
}
// decrease brightness
else if (incoming == 'c') {
    LightBrightness -=1;
}
// increase brightness faster
else if (incoming == 'f') {
    LightBrightness +=10;
}
// decrease brightness faster
else if (incoming == 'v') {
    LightBrightness -=10;
}
// set full brightness
else if (incoming == 'g') {
    set_brightness(100);
    LightBrightness = 100;
}
// check webclient
else if (incoming == 'k') {
    server_check();
}
// test rgb aka double rainbow
else if (incoming == 'u') {
    for (int i=0; i<255; i++) {
        analogWrite(RedPWM, 255-i);
        analogWrite(BluePWM, 255);
        analogWrite(GreenPWM, 255);
        delay(5);
    }
    for (int i=0; i<255; i++) {
        analogWrite(RedPWM, i);
        analogWrite(BluePWM, 255);
        analogWrite(GreenPWM, 255);
        delay(5);
    }
}
```

# Arduino Mega Shield - sourcecode

```
analogWrite(GreenPWM, 255);
delay(5);
}
for (int i=0; i<255; i++) {
  analogWrite(GreenPWM, 255-i);
  analogWrite(BluePWM, 255);
  analogWrite(RedPWM, 255);
  delay(5);
}
for (int i=0; i<255; i++) {
  analogWrite(BluePWM, i);
  analogWrite(BluePWM, 255);
  analogWrite(RedPWM, 255);
  delay(5);
}
for (int i=0; i<255; i++) {
  analogWrite(BluePWM, 255-i);
  analogWrite(RedPWM, 255);
  analogWrite(GreenPWM, 255);
  delay(5);
}
for (int i=0; i<255; i++) {
  analogWrite(BluePWM, i);
  analogWrite(RedPWM, 255);
  analogWrite(GreenPWM, 255);
  delay(5);
}
}
```