Systemtechnik

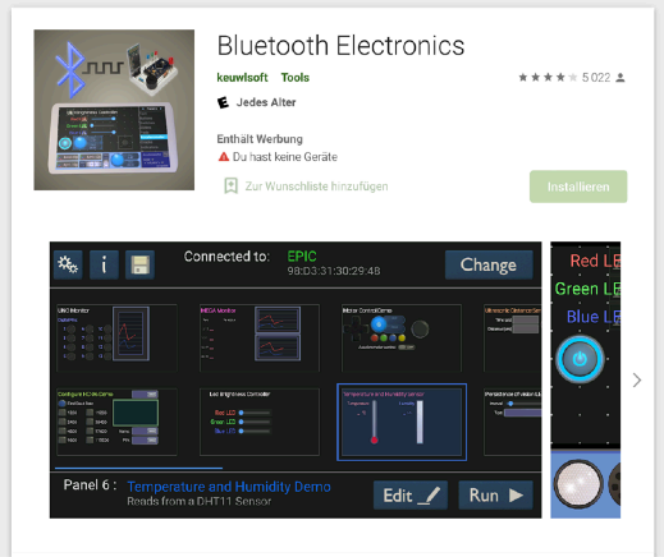
Hardwarenahe Programmierung
Protokoll 3: Bluetooth

Verfasser:

- **Jonas Ruhland**
- **Clemens Rumpfhuber**

Übungsleiter: DI Mag. Ferdinand Hell

Bluetooth mit Arduino



Zu Beginn der Stunde haben wir die Android App: „Bluetooth Electronics“ installiert.

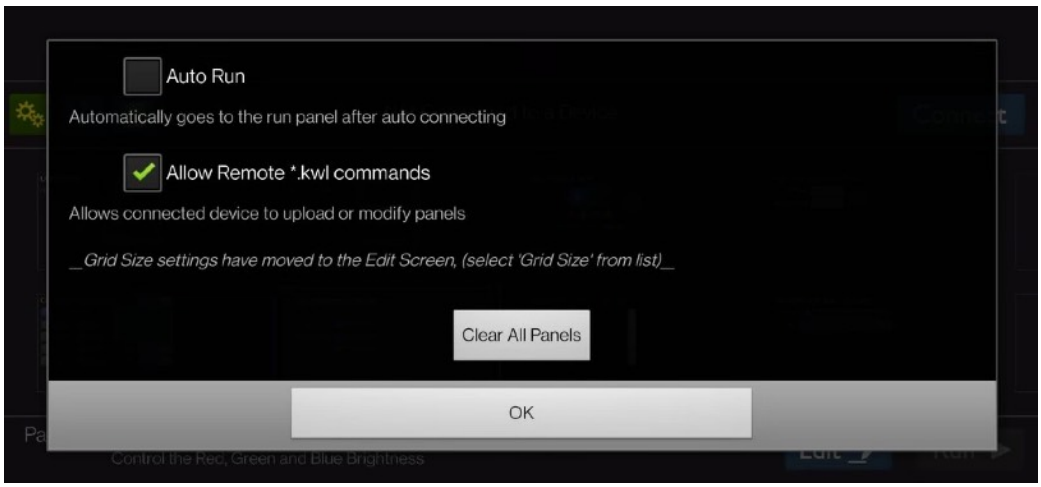
https://play.google.com/store/apps/details?id=com.keuwl.arduinobluetooth

Mit dieser App kann man eine GUI auf dem Handy mittels Steuerelementen zusammenbauen. Anschließend kann ein Arduino Code exportiert werden. Diesen kopiert man sich in die Arduino Entwicklungsumgebung und wird dann angepasst.

Wichtig ist, dass der Arduino entweder nativ einen Bluetooth Adapter hat oder via Bus angeschlossen ist.

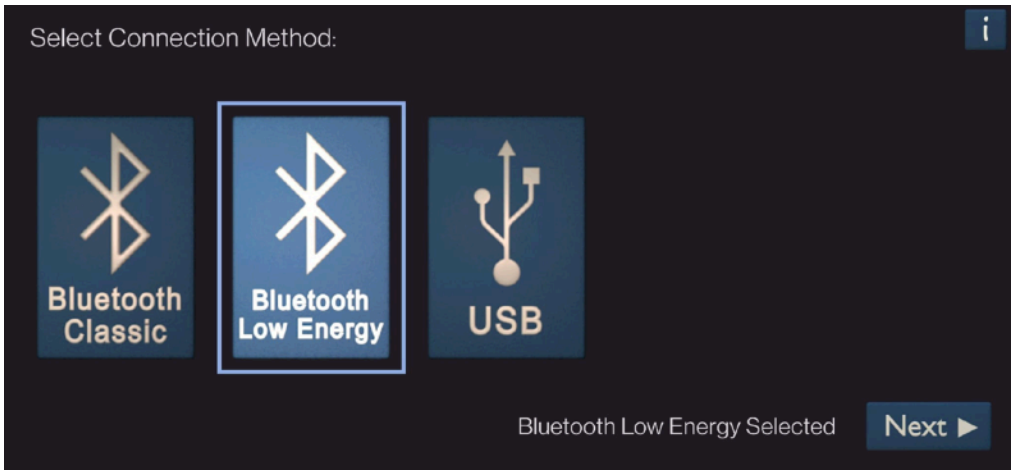
App konfigurieren

Beim ersten Mal starten der Konfiguration, haben wir „Aloe Remote .*kwl commands“ aktiviert. Dies ist unter den Einstellungen (Zahnräder, siehe links im Hintergrund) durchzuführen.

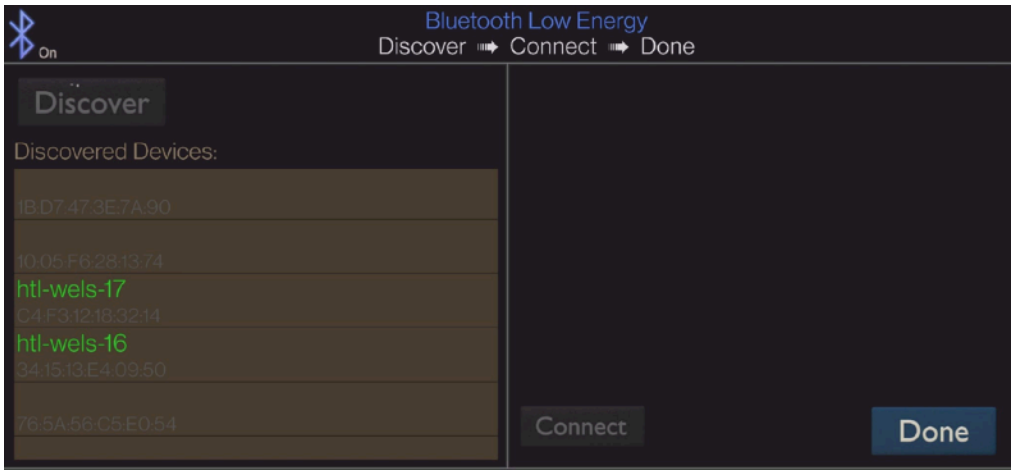


Verbindung via Bluetooth

Unter Bluetooth gibt es zwei verschiedene Versionen. „Bluetooth Classic“ ist ein alter Standard. Wir jedoch verwenden „Bluetooth Low Energy“, um uns mit dem Bluetooth-Modul am Arduino zu verbinden.



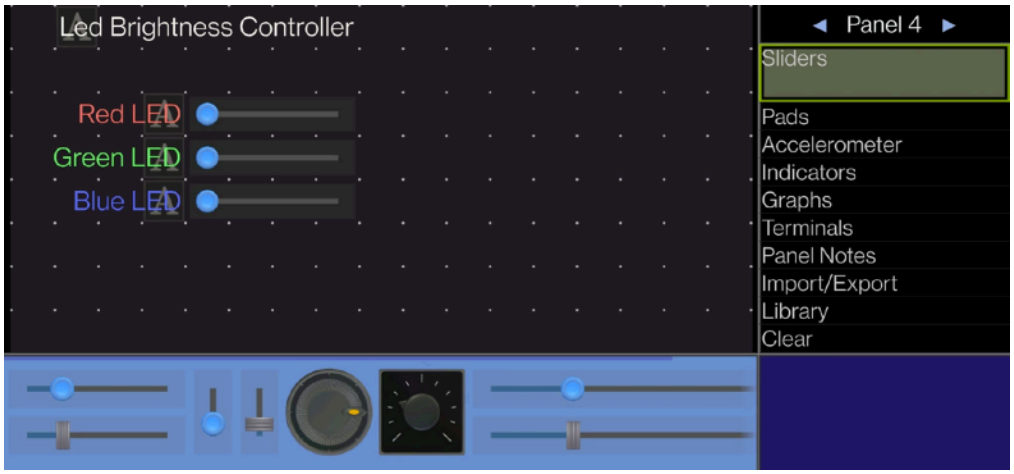
In der nächsten Ansicht, sieht man alle Bluetoothgeräte in der Nähe. Dort muss sich das richtige Geräte auswählen und auf „Connect“ und anschließend „Done“ klicken.

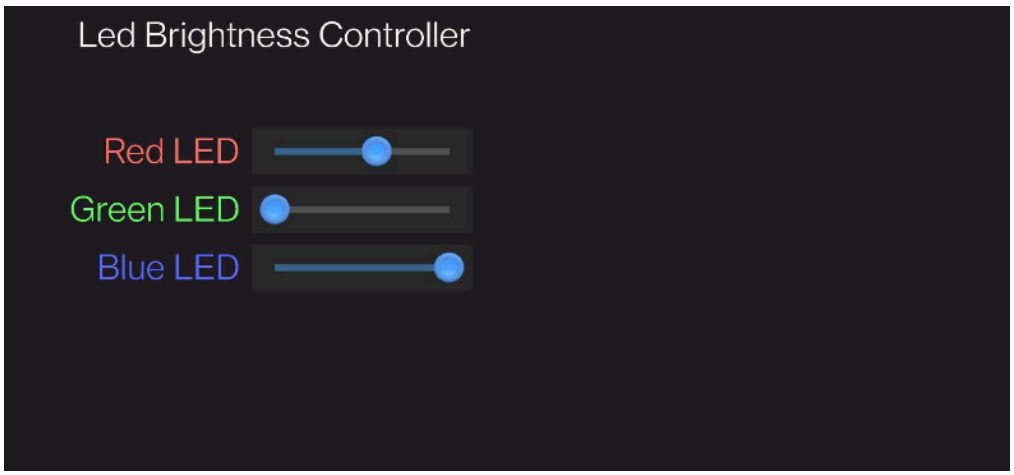


Anschließend ist man im Dashboard der Applikation, wo man alle erstellte Panel sieht.

Praktische Anwendung

LED Brightness Controller





Als erstes haben wir eine Vorlage vom Entwickler übernommen und adaptiert.

Beim automatisch genierten Code haben wir eine kleine Änderung bei der Ausgabe der GUI durchgeführt. Um die Strings in der „setup()“-Funktion, muss F(„<string>“) geschrieben werden. Dadurch wird der String im Flash-Speicher abgelegt und nicht in den RAM.

```
int update_interval=100; // time interval in ms for updating panel indicators
unsigned long last_time=0; // time of last update
char data_in; // data received from serial link
int slider_value; // Received Slider Values
String text; // String for text elements

#define LED_R 3
#define LED_G 5
#define LED_B 6

void setup() {

  Serial.begin(115200); //Change baud rate as required!
```

```
  pinMode(LED_R, OUTPUT);
  pinMode(LED_G, OUTPUT);
  pinMode(LED_B, OUTPUT);


////////////// Build panel in app

Serial.println(F("*.kwl"));
delay(2);
Serial.println(F("clear_panel()"));
delay(2);
Serial.print(F("set_grid_size(17,8)"));
delay(2);
Serial.println(F(""));
delay(2);
Serial.print(F("add_text(3,2,xlarge,"));
delay(2);
Serial.print(F("R,Red LED,250,115,11"));
delay(2);
Serial.println(F("0,0)"));
delay(2);
Serial.print(F("add_text(3,3,xlarge,"));
delay(2);
Serial.print(F("R,Green LED,110,245,"));
delay(2);
Serial.println(F("110,0)"));
delay(2);
Serial.print(F("add_text(3,4,xlarge,"));
delay(2);
Serial.print(F("R,Blue LED,90,110,24"));
delay(2);
Serial.println(F("5,0)"));
delay(2);
Serial.print(F("add_text(1,0,xlarge,"));
delay(2);
Serial.print(F("L,Led Brightness Con"));
delay(2);
Serial.print(F("troller,250,245,240,"));
delay(2);
Serial.println(F("0)"));
delay(2);
Serial.print(F("add_slider(4,3,1,0,2"));
delay(2);
Serial.println(F("55,0,G,A,0)"));
delay(2);
Serial.print(F("add_slider(4,4,1,0,2"));
delay(2);
Serial.println(F("55,0,B,A,0)"));
delay(2);
Serial.print(F("add_slider(4,2,1,0,2"));
delay(2);
Serial.println(F("55,0,R,A,0)"));
delay(2);
Serial.print(F("set_panel_notes(LED "));
delay(2);
Serial.print(F("Brightness,Control t"));
delay(2);
Serial.print(F("he Red, Green and Bl"));
delay(2);
Serial.print(F("ue Brightness,with S"));
delay(2);
Serial.print(F("liders that adjust t"));
delay(2);
Serial.print(F("he PWM.,Requires: Ar"));
delay(2);
Serial.print(F("udino, HC-06, LEDs, "));
```

```
delay(2);
Serial.println(F("Resistors)"));
delay(2);
Serial.println(F("run()"));
delay(2);
Serial.println(F("*"));
delay(2);

}

void loop() {

  //////////////   Receive and Process Data

  if (Serial.available()){
    data_in=Serial.read();  //Get next character

    if(data_in=='G'){ //  Slider
      slider_value=Serial.parseInt();
      //<--- Perhaps do something with slider_value here
      analogWrite(LED_G, slider_value);
    }

    if(data_in=='B'){ //  Slider
      slider_value=Serial.parseInt();
      //<--- Perhaps do something with slider_value here
      analogWrite(LED_B, slider_value);
    }

    if(data_in=='R'){ //  Slider
      slider_value=Serial.parseInt();
      //<--- Perhaps do something with slider_value here
      analogWrite(LED_R, slider_value);
    }

  }

  //////////////  Send Data to Android device

  unsigned long t=millis();
  if ((t-last_time)>update_interval){
    last_time=t;

    // Update Text Element
    text="abc"; // <--- Set text to send here
    Serial.print("0"+text+"");
delay(2);

    // Update Text Element
    text="abc"; // <--- Set text to send here
    Serial.print("0"+text+"");
delay(2);

    // Update Text Element
    text="abc"; // <--- Set text to send here
    Serial.print("0"+text+"");
delay(2);

    // Update Text Element
    text="abc"; // <--- Set text to send here
    Serial.print("0"+text+"");
delay(2);

  }

}
```

DC Motor Controller

In dieser Übung haben wir einen DC Motor via Smartphone ferngesteuert. Dabei konnte die Geschwindigkeit und Richtung des Motors vom Smartphone bedient werden.

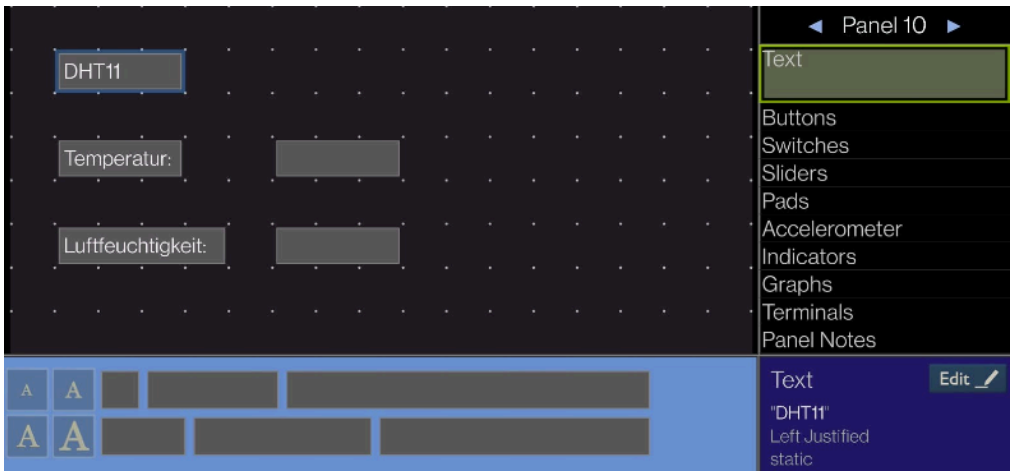
Auch hier haben wir wieder die die Stringausgaben angepasst und den Code für die DC Motor programmiert.

```
#define PWM_MOT_1 3
#define PWM_MOT_2 5
#define PWM_FWD 6
#define PWM_BWD 4

int update_interval=100; // time interval in ms for updating panel indicators
unsigned long last_time=0; // time of last update
char data_in; // data received from serial link
int slider_value; // Received Slider Values

void setup() {

  Serial.begin(115200);

  pinMode(PWM_MOT_1, OUTPUT);
  pinMode(PWM_MOT_2, OUTPUT);
  pinMode(PWM_FWD, OUTPUT);
  pinMode(PWM_BWD, OUTPUT);

////////////// Build panel in app

Serial.println(F("*.kwl"));
delay(10);
Serial.println(F("clear_panel()"));
delay(10);
Serial.print(F("set_grid_size(17,8)"));
delay(10);
Serial.println(F(""));
delay(10);
Serial.print(F("add_text_box(1,1,4,L"));
delay(10);
Serial.print(F(",Richtung,245,240,24"));
delay(10);
Serial.println(F("5,)"));
delay(10);
Serial.print(F("add_text_box(6,1,4,L"));
delay(10);
Serial.print(F(",Geschwindigkeit,245"));
```

```
delay(10);
Serial.println(F(",240,245,)"));
delay(10);
Serial.print(F("add_switch(1,2,3,C,c"));
delay(10);
Serial.println(F(",0,0)"));
delay(10);
Serial.print(F("add_slider(6,2,7,0,2"));
delay(10);
Serial.println(F("55,0,A,A,0)"));
delay(10);
Serial.print(F("set_panel_notes(-,,,"));
delay(10);
Serial.println(F(")"));
delay(10);
Serial.println(F("run()"));
delay(10);
Serial.println(F("*"));
delay(10);

}

void loop() {

  //////////////   Receive and Process Data

  if (Serial.available()){
    data_in=Serial.read();  //Get next character

    if(data_in=='C'){ //Switch On
      //<--- Insert code for switch on here
      Serial.println("direction forward");
      digitalWrite(PWM_FWD, HIGH);
      digitalWrite(PWM_BWD, LOW);
    }
    if(data_in=='c'){ // Switch Off
      //<--- Insert code for when switch turned off here
      Serial.println("direction backwards");
      digitalWrite(PWM_FWD, HIGH);
      digitalWrite(PWM_BWD, LOW);
    }

    if(data_in=='A'){ //  Slider
      slider_value=Serial.parseInt();
      //<--- Perhaps do something with slider_value here
      Serial.println("move");
      Serial.println(slider_value);
      if (slider_value != 0) {
        analogWrite(PWM_MOT_1, slider_value);
      }
    }
  }
  //////////////  Send Data to Android device

  unsigned long t=millis();
  if ((t-last_time)>update_interval){
    last_time=t;

  }

}
```

Anzeige eines DHT11

In dieser Übung haben wir das Smartphone als Anzeige für einen DHT11 verwendet. Auf der Anzeige wurden Temperatur und Luftfeuchtigkeit ausgegeben.

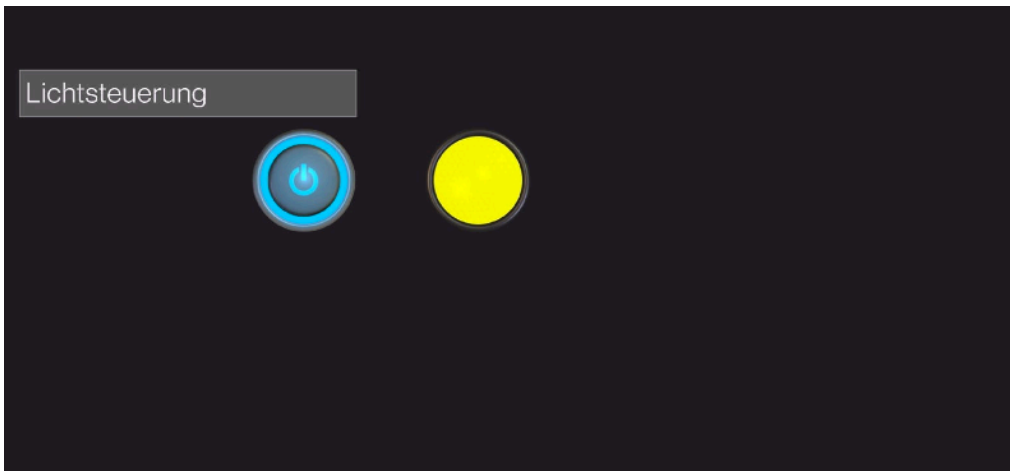



```
#include "DHT.h"

#define DHTPIN A0     // Digital pin connected to the DHT sensor
#define DHTTYPE DHT11   // DHT 11
DHT dht(DHTPIN, DHTTYPE); // create dht

float humidity, temperature = 0;

int update_interval=2000; // update every 1 sec
unsigned long last_time=0; // time of last update
char data_in; // data received from serial link
char text[10]; // String for text elements

void setup() {

  Serial.begin(115200); //Change baud rate as required!

Serial.println(F("*.kwl"));
delay(10);
Serial.println(F("clear_panel()"));
delay(10);
Serial.print(F("set_grid_size(17,8)"));
delay(10);
Serial.println(F(""));
delay(10);
Serial.print(F("add_text_box(1,1,3,L"));
delay(10);
Serial.print(F(",DHT11,245,240,245,)"));
delay(10);
Serial.println(F(""));
delay(10);
Serial.print(F("add_text_box(6,3,3,L"));
delay(10);
Serial.print(F(",,245,240,245,A)"));
delay(10);
Serial.println(F(""));
delay(10);
Serial.print(F("add_text_box(6,5,3,L"));
delay(10);
Serial.print(F(",,245,240,245,B)"));
delay(10);
```

```
Serial.println(F(""));
delay(10);
Serial.print(F("add_text_box(1,3,3,L"));
delay(10);
Serial.print(F(",Temperatur:,245,240"));
delay(10);
Serial.println(F(",245,)"));
delay(10);
Serial.print(F("add_text_box(1,5,4,L"));
delay(10);
Serial.print(F(",Luftfeuchtigkeit:,2"));
delay(10);
Serial.println(F("45,240,245,)"));
delay(10);
Serial.print(F("set_panel_notes(-,,,"));
delay(10);
Serial.println(F(")"));
delay(10);
Serial.println(F("run()"));
delay(10);
Serial.println(F("*"));
delay(10);

}

void loop() {
  if (Serial.available()){
    data_in=Serial.read();  //Get next character

  }

  unsigned long t=millis();
  if ((t-last_time)>update_interval){
    last_time=t;

    // read sensor value
    read_sensor();

    // Update Text Element
    sprintf(text, "*A%d.%02d", (int)temperature, (int)(temperature*100)%100);
    Serial.print(text);
    delay(10);

    // Update Text Element
    sprintf(text, "*B%d.%02d", (int)humidity, (int)(humidity*100)%100);
    Serial.print(text);
    delay(10);

  }

}

void read_sensor() {
  // Reading temperature or humidity takes about 250 milliseconds!
  // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
  float h = dht.readHumidity();
  // Read temperature as Celsius (the default)
  float t = dht.readTemperature();

  if (!(isnan(h) || isnan(t))) {
    humidity = h;
    temperature = t;
  }

}
```

Steuerung eines Bewegungssensors mit LED

In dieser Übung haben wir einen Bewegungssensor verwendet, um eine LED einzuschalten, wenn dieser eine Bewegung wahrnimmt. Zusätzlich wurde der Status der LED auf das Smartphone übertragen. Auch ein permanentes Leuchten der LED konnte mithilfe des Handys durchgeführt werden.




```
#define LED 6
#define SENSOR 8

int update_interval=500; // time interval in ms for updating panel indicators
unsigned long last_time=0; // time of last update
char data_in; // data received from serial link
int red,green,blue; // RGB color

unsigned int force_led_on = 0;
unsigned int button_state = 0;

void setup() {

  // set pin modes for led and sensor
  pinMode(LED, OUTPUT);
  pinMode(SENSOR, INPUT);

  Serial.begin(115200); //Change baud rate as required!

////////////// Build panel in app

Serial.println(F("*.kwl"));
delay(10);
Serial.println(F("clear_panel()"));
delay(10);
Serial.print(F("set_grid_size(17,8)"));
delay(10);
Serial.println(F(""));
delay(10);
Serial.print(F("add_text_box(0,1,6,L"));
delay(10);
Serial.print(F(",Lichtsteuerung,245,"));
delay(10);
Serial.println(F("240,245,)"));
delay(10);
Serial.print(F("add_switch(4,2,4,D,d"));
delay(10);
Serial.println(F(",0,0)"));
```

```
delay(10);
Serial.print(F("add_led(7,2,2,L,0,0,"));
delay(10);
Serial.println(F("0)"));
delay(10);
Serial.print(F("set_panel_notes(-,,,"));
delay(10);
Serial.println(F(")"));
delay(10);
Serial.println(F("run()"));
delay(10);
Serial.println(F("*"));
delay(10);

}

void loop() {

  if (Serial.available()){
    data_in=Serial.read();  //Get next character

    if(data_in=='D'){ //Switch On
      led_switch(1);
      force_led_on = 1;
    }
    if(data_in=='d'){ // Switch Off
      led_switch(0);
      force_led_on = 0;
    }

  }

  unsigned long t=millis();
  if ((t-last_time)>update_interval){
    last_time=t;

    // check if led is set on by phone
    if (force_led_on == 0) {

      // read button state
      button_state = digitalRead(SENSOR);

      // check if the pushbutton is pressed. If it is, the buttonState is HIGH:
      if (button_state == HIGH) {
        led_switch(1);
      } else {
        led_switch(0);
      }

    }

  }
}

void led_switch(int state) {
  // arduino led
  analogWrite(LED, state == 1 ? 250 : 0);

  // smartphone led
  red = state == 1 ? 255 : 0;
  green = state == 1 ? 255 : 0;
  blue = 0;
  Serial.print("*LR"+String(red)+"G"+String(green)+"B"+String(blue)+"");
  delay(10);
}
```