

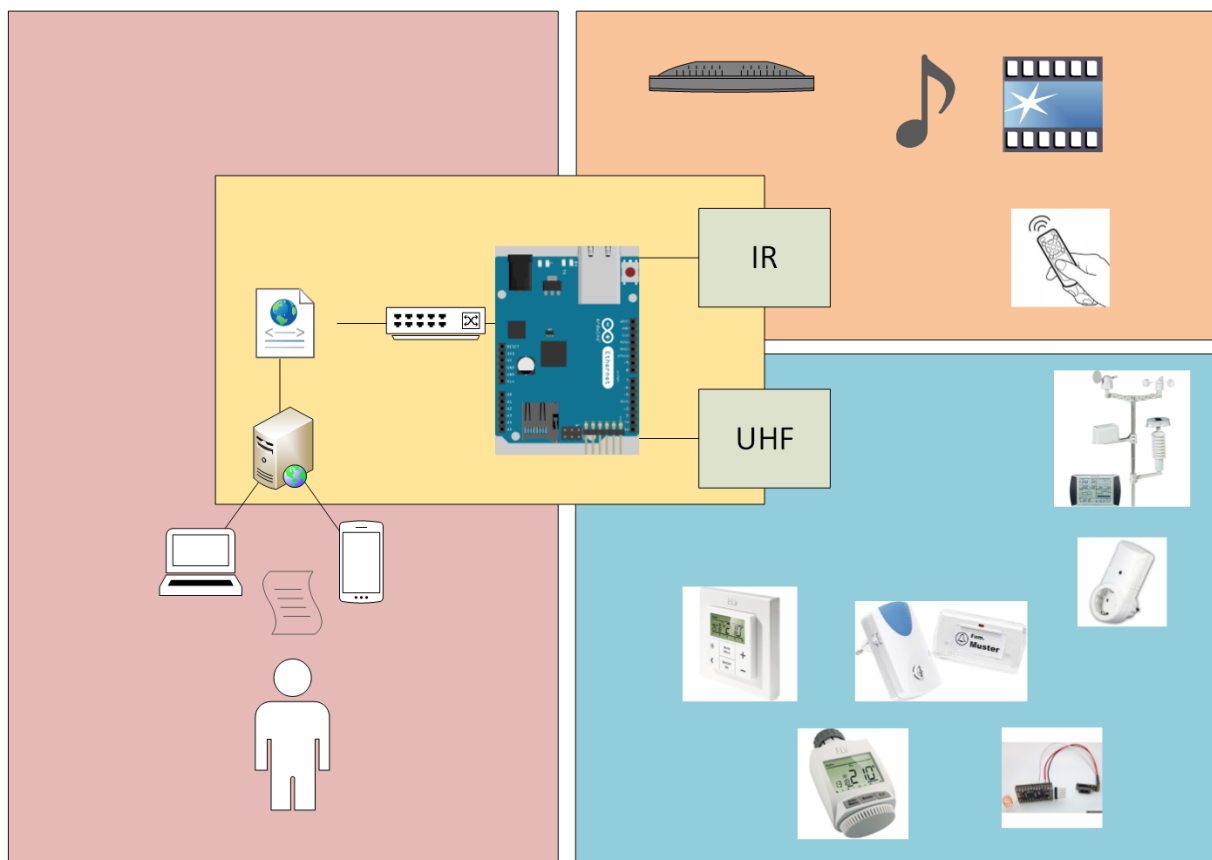
Arduino Projekt

David Arenz & Matthias Lehmann

Ziel des Projektes:

Schaffung einer universellen Plattform zur Hausautomatisierung

1. Einbinden verschiedener Steuerungssystemen aus dem Konsumerbereich
 - Verwirklichung eines Universeller Infrarot Sender/Empfängers
 - TV, Hifi, Lampen usw.
 - Beliebige Fernbedienungen als Bedienelemente
 - Einbindung von UHF Funktransceivern (433/866 MHz)
 - Funksteckdosen, Funkdimmer
 - Empfang von Wetterdaten
2. Ansteuerung verschiedener Systeme bündeln
 - API ähnliche Befehle
 - Abarbeiten von Befehlsketten
 - Ggf. Überwachung und Regelung von Parametern
3. HMI Schnittstelle per Webserver



1 Aufgabe 1

Listing 1: ../code/InfraredProxy/InfraredProxy.ino

```
1  #include <SPI.h>
   #include <Ethernet.h>
3  #include <IRremote.h>

5  class InfraredProxy {
   IRsend _infrared_sender;

7
   void read_line(EthernetClient& client, char* buffer, const int buffer_length) {
9       int buffer_pos = 0;
       while (client.available() && (buffer_pos < buffer_length - 1)) {
11          const char c = client.read();
           if (c == '\n')
13              break;
           if (c != '\r')
15              buffer[buffer_pos++] = c;
       }
17       buffer[buffer_pos] = '\0';
   }

19
   bool send_ir_data(const char* protocol, const int bits, const long value) {
21       bool result = true;
       if (!strcasecmp(protocol, "NEC"))
23           _infrared_sender.sendNEC(value, bits);
       else if (!strcasecmp(protocol, "SONY"))
25           _infrared_sender.sendSony(value, bits);
       else if (!strcasecmp(protocol, "RC5"))
27           _infrared_sender.sendRC5(value, bits);
       else if (!strcasecmp(protocol, "RC6"))
29           _infrared_sender.sendRC6(value, bits);
       else
31           result = false;
       return result;
33   }

35   bool handle_command(char* line) {
       strsep(&line, " ");
37       char* path = strsep(&line, " ");

39       char* args[3];
       for (char** ap = args; (*ap = strsep(&path, "/")) != NULL;)
41         if (**ap != '\0')
           if (++ap >= &args[3])
43             break;
       const int bits = atoi(args[1]);
45       const long value = atol(args[2]);
       return send_ir_data(args[0], bits, value);
47   }

49   public:

51   void receive_from_server(EthernetServer server) {
       const int MAXLINE = 256;
53       char line[MAXLINE];
       EthernetClient client = server.available();
55       if (client) {
           while (client.connected()) {
57               if (client.available()) {
```

```

        read_line(client, line, MAXLINE); Serial.println(line);
59     if (line[0] == 'G' && line[1] == 'E' && line[2] == 'T')
        handle_command(line);
61     if (!strcmp(line, "")) {
        client.println("HTTP/1.1 200 OK\n");
63     break;
    }
65 }
}
67 delay(1);
client.stop();
69 }
}
71 };

73 //—— ENDE DER DEKLARATION ——

75 const unsigned int PROXY_PORT = 80;
77 const unsigned int BAUD_RATE = 115200;
byte mac[] = { 0x90, 0xA2, 0xDA, 0x0E, 0xDB, 0xAE }; // MAC Arduino Ethernet (David)
79 EthernetServer server(PROXY_PORT);
InfraredProxy ir_proxy;

81 void setup() {
83 // Open serial communications and wait for port to open:
    Serial.begin(115200);
85 // start the Ethernet connection and the server:
    Ethernet.begin(mac);
87 server.begin();
    Serial.print("server is at ");
89 Serial.println(Ethernet.localIP());
}

91 void loop() {
93     ir_proxy.receive_from_server(server);
}

```