

HTML

Objektleiste

↓  
1/TV 4/PLAY  
Sevent/Typ/ Befehl

- TV
- Steckdose
- Hifi

Objekte/subklasse

- |             |             |          |
|-------------|-------------|----------|
| • TV        | • Steckdose | • Hifi   |
| - (Status)  | - Ein       | - Lauter |
| - Play      | - Aus       | - Leiser |
| - Lauter    | - Alle Aus  | - Mute   |
| - Leiser    |             | - Kanal  |
| - Kanal 0-9 |             |          |

Klassen

IR

Funk

Methode

IR SEND(MEC(32/x84))

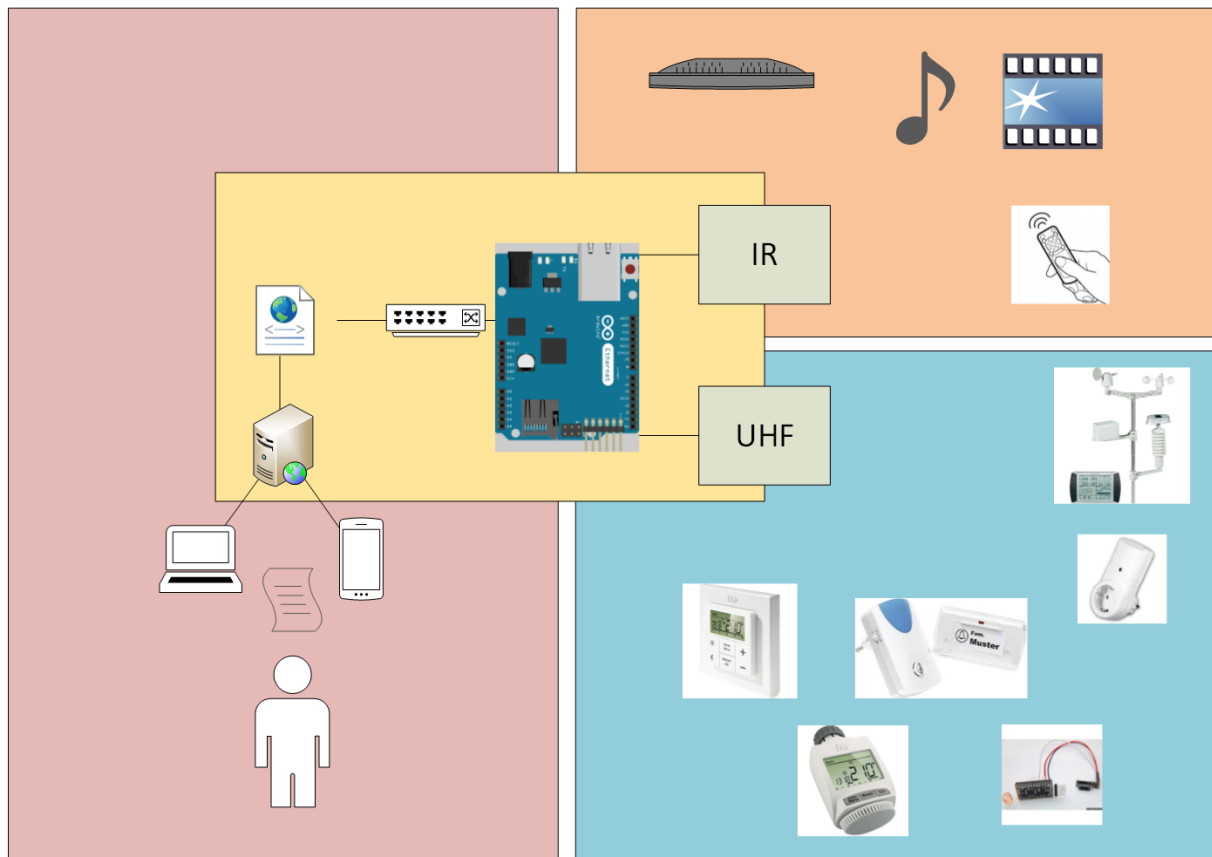
# Arduino Projekt

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 // https://www.dpunkt.de/leseproben/3692/5%20Universalfernbedienung.pdf
3 #include <SPI.h>
4 #include <Ethernet.h>
5 #include <IRremote.h>
6 class InfraredProxy
7 {
8     IRsend _infrared_sender;
9     void read_line(EthernetClient& client, char* buffer, const int buffer_length)
10    {
11        int buffer_pos = 0;
12        while (client.available() && (buffer_pos < buffer_length - 1))
13        {
14            const char c = client.read();
15            if (c == '\n')
16                break;
17            if (c != '\r')
18                buffer[buffer_pos++] = c;
19        }
20        buffer[buffer_pos] = '\0';
21    }
22    bool send_ir_data(const char* protocol, const int bits, const long value)
23    {
24        bool result = true;
25        if (!strcasecmp(protocol, "NEC"))
26            _infrared_sender.sendNEC(value, bits);
27        else if (!strcasecmp(protocol, "SONY"))
28            _infrared_sender.sendSony(value, bits);
29        else if (!strcasecmp(protocol, "RC5"))
30            _infrared_sender.sendRC5(value, bits);
31        else if (!strcasecmp(protocol, "RC6"))
32            _infrared_sender.sendRC6(value, bits);
33        else
34            result = false;
35        return result;
36    }
37    bool handle_command(char* line)
38    {
39        strsep(&line, " ");
40        char* path = strsep(&line, " ");
41        char* args[3];
42        for (char** ap = args; (*ap = strsep(&path, "/")) != NULL;)
43            if (**ap != '\0')
44                if (++ap >= &args[3])
45                    break;
46        const int bits = atoi(args[1]);
47        const long value = atol(args[2]);
48        return send_ir_data(args[0], bits, value);
49    }
50 public:
51     void receive_from_server(EthernetServer server)
52     {
53         const int MAXLINE = 256;
54         char line[MAXLINE];
55         EthernetClient client = server.available();
56         if (client)
57         {
```

```

59     while (client.connected())
60     {
61         if (client.available())
62         {
63             read_line(client, line, MAXLINE);
64             Serial.println(line);
65             if (line[0] == 'G' && line[1] == 'E' && line[2] == 'T')
66                 handle_command(line);
67             if (!strcmp(line, ""))
68             {
69                 client.println("HTTP/1.1 200 OK\n");
70                 break;
71             }
72         }
73     }
74     delay(1);
75     client.stop();
76 }
77 };
78 //— ENDE DER DEKLARATION —
79 const unsigned int PROXY_PORT = 80;
80 const unsigned int BAUD_RATE = 115200;
81 byte mac[] = {
82     0x90, 0xA2, 0xDA, 0x0E, 0xDB, 0xAE }; // MAC Arduino Ethernet (David)
83 EthernetServer server(PROXY_PORT);
84 InfraredProxy ir_proxy;
85 void setup()
86 {
87     // Open serial communications and wait for port to open:
88     Serial.begin(115200);
89     // start the Ethernet connection and the server:
90     Ethernet.begin(mac);
91     server.begin();
92     Serial.print("server is at ");
93     Serial.println(Ethernet.localIP());
94 }
95 void loop()
96 {
97     ir_proxy.receive_from_server(server);
98 }

```