

[Vorbemerkungen](#)

[Hardware](#)

[Verkabelung](#)

[Software](#)

[Code – Arduino](#)

[Code – PHP](#)

[Code – HomeMatic](#)

[Erklärungen und Probleme](#)

[Bonus: Benachrichtigungen aufs Smartphone](#)

Vorbemerkungen

Als blutiger Anfänger in Sachen Hausautomatisierung habe ich mir jüngst im Rahmen einer mittelschweren Wohnraum-Renovierung Funk-Rollomotoren der Firma Somfy aufschwätzen lassen. Grundsätzlich sind die Teile bestimmt total hochwertig und absolut zuverlässig, doch jetzt, wo ich über weitere Automatisierungsschritte nachdenke, kommen schnell die Nachteile ans Tageslicht: Somfy-Geräte sprechen nicht mit denen von anderen Herstellern. Und viel schlimmer noch: Die Somfy-Netzwerklösung Tahoma ist unverschämte teuer und lenkt dazu noch allen Traffic über einen Somfy-Internetdienst. Nicht mit mir!

Da ich mich, vor allem aufgrund von Heizung und Thermostaten, jetzt insgesamt für HomeMatic entschieden habe, stehe ich – wie sicher schon der Eine oder Andere vor mir – vor dem Problem: Wie kriege ich meine Somfy-Geräte dazu, mit meiner HomeMatic-Zentrale zu sprechen? Die Lösung war eigentlich gar nicht mal so kompliziert – es brauchte nur einen Arduino, ein paar Optokoppler und eine Somfy-Fernbedienung – sowie Einiges an Hilfestellung von den kompetenten Jungs im [arduino.cc-Forum](#). Danke noch einmal an dieser Stelle!

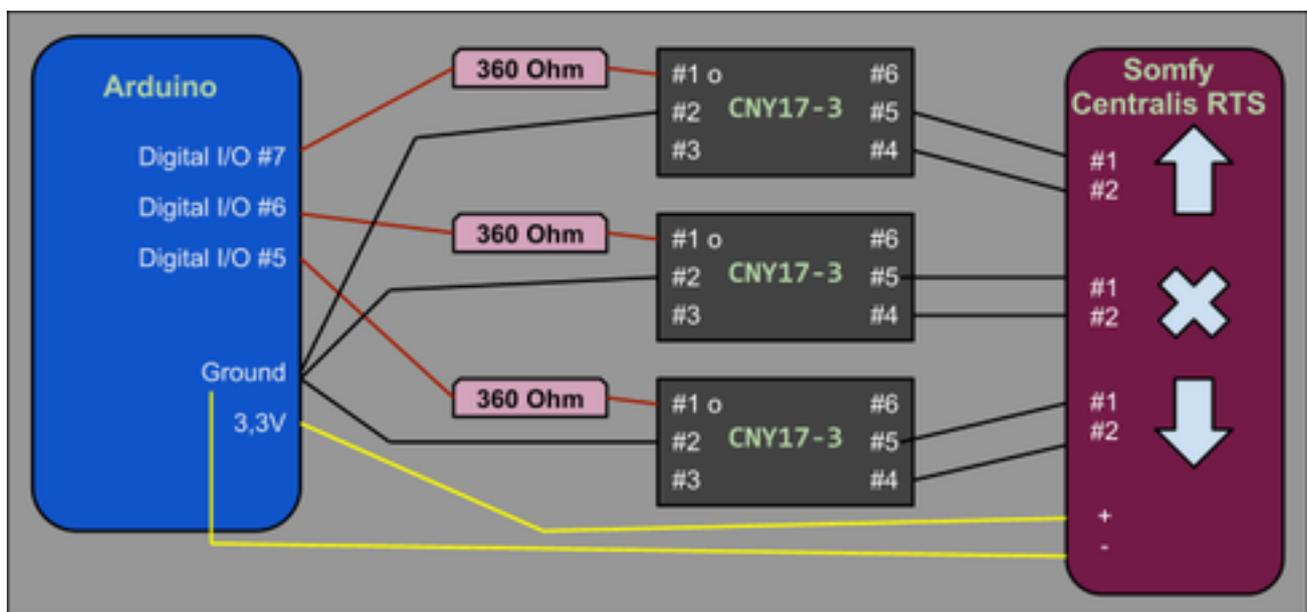
Der Aufbau meiner Netzwerk-Rollo-Fernsteuerung ist simpel: Der Arduino hängt an der Somfy-Fernbedienung und löst diese aus, sobald er entsprechende Befehle aus dem Netzwerk erhält. Da die Fernbedienung mit 3,3 Volt läuft, der Arduino aber auf den I/O-Pins 5 Volt ausgibt, müssen die Tasten per Optokoppler angesteuert werden (Elektronik-Pros werden mir bei dem "müssen" vermutlich widersprechen, aber für mein beschränktes Verständnis von der Materie trifft es durchaus zu).

Die Hardware

Hier eine Übersicht über die verbauten Geräte und Teile:

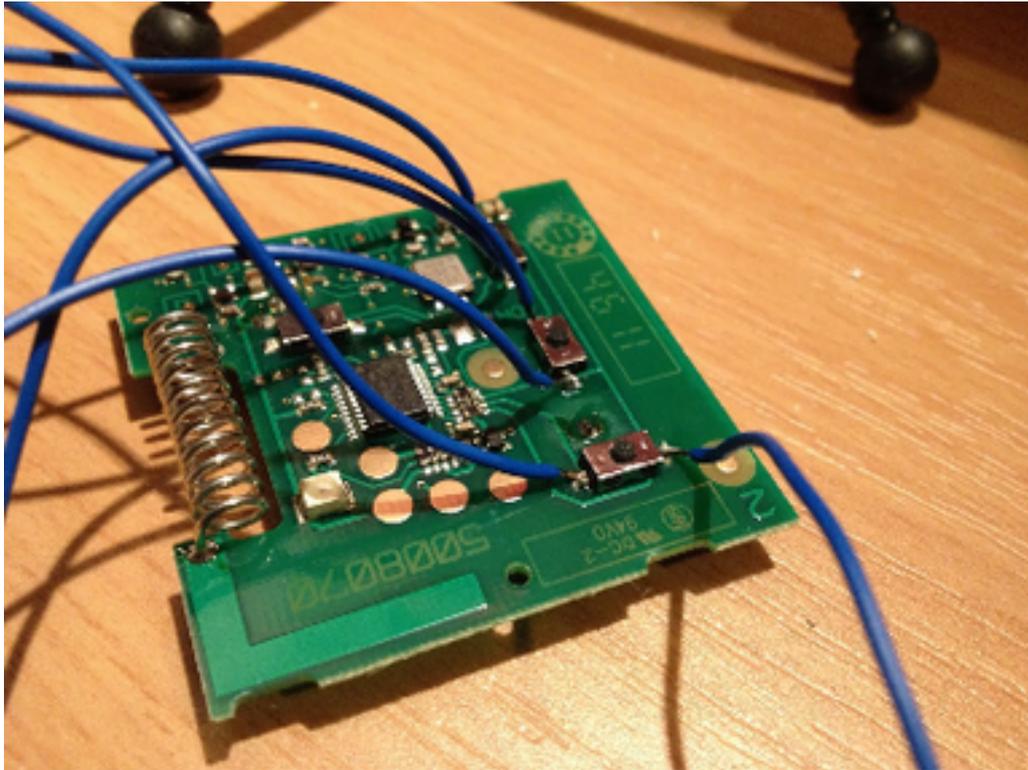
- Arduino mit Ethernet Shield
- [Somfy Centralis RTS](#) (ein "Eine Fernbedienung für alle Rollos"-Funk-Wandschalter; denkbar wären aber auch Mehrkanal-Fernbedienungen)
- 3 * Optokoppler [CNY17-3 von Conrad](#) – je einen pro Fernbedienungstaste (auf, ab, stop)
- 3 * Widerstand à 360 Ohm

Die Verkabelung



Verkabelungsschema;

Anmerkung zur Verkabelung: Die gelbe Verbindung der Arduino-Stromversorgung mit der Somfy-Fernbedienung ist in meinem Fall eher theoretischer Natur ... ausprobiert habe ich sie bisher noch nicht – mein Setup läuft noch mit 3V-Knopfzelle auf der Somfy-Seite.



Der Somfy-Funkwandschalter ohne Gehäuse und mit "angezapften" Bedienknöpfen

Die Software

Grundsätzlich ließe sich auf dem Arduino ein kleiner Webserver betreiben, der Befehle entgegen nimmt und daraufhin die Rollos ansteuert. Da ich aber ohnehin eine Netzwerkfestplatte mitsamt PHP-fähigem Webserver im Netz hängen habe und diese mich die Beschränkungen des Arduino-Webrowsers umgehen lässt, erschien mir eine getrennte Umsetzung sinnvoller. Der Arduino wartet also lediglich auf Befehle, die per UDP-Pakete aus meinem Heimnetz kommen, und löst dann entsprechend die Fernbedienung aus. Komplizierte TCP-Verbindungen mit Handshakes oder Ähnlichem schießen für meine Zwecke weit übers Ziel hinaus – sind aber durchaus denk- und umsetzbar.

Code – Arduino

```
#include <SPI.h>
#include <Ethernet.h>
#include <EthernetUdp.h>

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
byte ip[] = { 192, 168, 1, 4 };
unsigned int localPort = 8888;
unsigned long lastMillis = 0;
unsigned long nowMillis = 0;
long block = 4000;
int dir[] = { 6, 5, 7 };
char packetBuffer[UDP_TX_PACKET_MAX_SIZE];
EthernetUDP Udp;

void setup() {
  Ethernet.begin(mac, ip);
  Udp.begin(localPort);
  pinMode(dir[0], OUTPUT);
  pinMode(dir[1], OUTPUT);
  pinMode(dir[2], OUTPUT);
}

void sendCommand(int d) {
  if (lastMillis == 0 || (nowMillis - lastMillis) > block) {
    lastMillis = nowMillis;
    digitalWrite(dir[d], HIGH);
    delay(500);
    digitalWrite(dir[d], LOW);
    delay(100);
  }
}

void loop() {
  nowMillis = millis();
  int packetSize = Udp.parsePacket();
  if(packetSize == 4) {
    Udp.read(packetBuffer,UDP_TX_PACKET_MAX_SIZE);
    String clientMsg = packetBuffer;
    if (clientMsg.indexOf('>') == 1) {
```

```

        if (clientMsg.substring(0,1) == "1") {
            String d = clientMsg.substring(2,3);
            int c = d.toInt();
            if (c >= 0 && c <= 2) {
                sendCommand(c);
            }
        }
    }
}

```

*// Der Arduino erwartet ein drei Zeichen langes UDP-Paket, mit
// Geräte-ID, '>'-Zeichen und Richtungs-ID.*

*// Die Geräte-ID wird momentan nicht benutzt, soll aber schon
// einmal für zukünftige Ausbauten vorsorgen.*

// '>' ist lediglich ein Trenner.

*// Gültige Richtungs-IDs sind: 0 (stop), 1 (runter), 2 (hoch) */*

```

    delay(10);
}

```

Code – PHP

```

<?php
/* Vorab wird noch geprüft, ob GET-Parameter vorhanden sind, diese
das richtige Format haben usw. ... */

$fp = fsockopen("udp://192.168.1.4", 8888, $errno, $errstr, 15);

if (!$fp) {
    echo "$errstr ($errno)<br />\n";
} else {
    $out = "1>".substr($_GET['action'],2,1)."\n";
    fwrite($fp, $out);
    fclose($fp);
}
/* HTML usw.... */
?>

```

Code – HomeMatic

Aufseiten der HomeMatic-CCU braucht es lediglich einen passenden Auslöser, dem folgende Skriptzeile (natürlich mit entsprechend angepasster URL und jeweils den Richtungen zugeordneten GET-Parametern) angehängt ist:

```
string stdout;  
string stderr;  
system.Exec("wget -O /dev/null -q http://WEBSERVER/home/index.php?  
action=1%3E1")
```

Das Trennzeichen '>' sollte dabei natürlich für die Url entsprechend kodiert werden (man könnte es, mit kleineren Anpassungen an den Skripten, natürlich auch weglassen oder durch ein anderes Zeichen ersetzen).

Name	Beschreibung	Bedingung (Wenn...)	Aktivität (Dann..., Sonst...)	Aktion
Rollos Wohnzimmer (runter)		Zeit: Täglich nachts beginnend am 24.12.2012 auslösen zu Zeitpunkten	Skript: ... verzögert um 1 Stunden ausführen	<input type="checkbox"/> system intern

Bedingung: Wenn...

Zeitsteuerung ▼ Täglich nachts beginnend am 24.12.2012 auslösen zu Zeitpunkten ▼

UND ▼

ODER ▼

Aktivität: Dann... Vor dem Ausführen alle laufenden Verzögerungen für diese Aktivitäten beenden (z.B. Retriggern).

Skript ▼ string stdout; string stderr; system.Exec("wget -O /dev/null... verzögert um ▼ 1

Stunden ▼

Aktivität: Sonst... Vor dem Ausführen alle laufenden Verzögerungen für diese Aktivitäten beenden (z.B. Retriggern).

So sieht mein HomeMatic-Auslöser für das Herunterfahren der Jalousien im Wohnzimmer eine Stunde nach Sonnenuntergang aus.

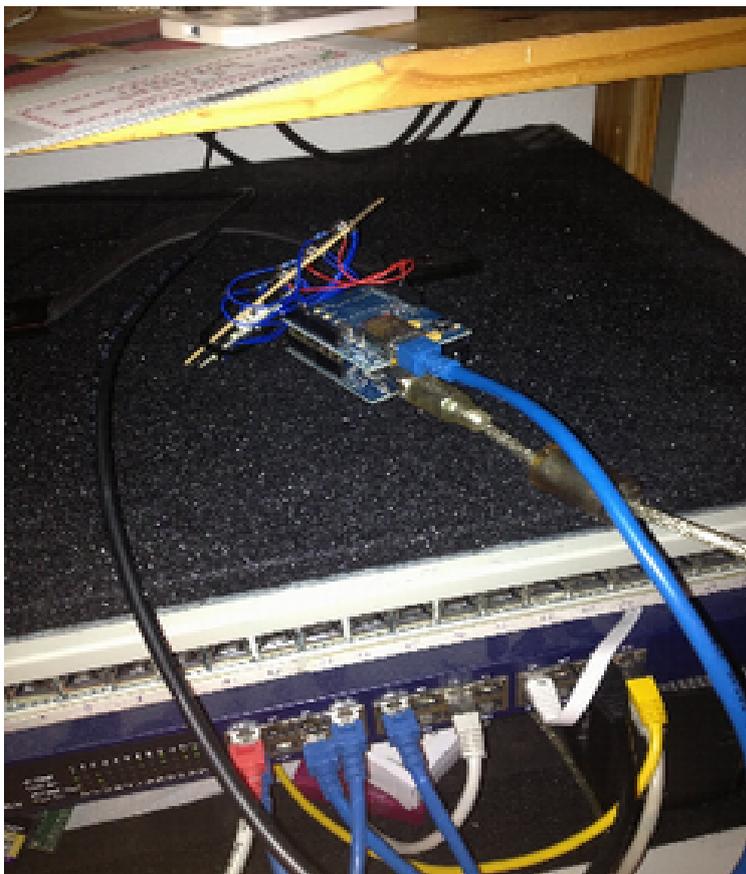
Sobald ich mir noch einen Türöffnungssensor besorgt habe, werde ich die Steuerung an dieser Stelle noch etwas verfeinern: Die Rollos sollen abhängig von Sonnenaufgang und Sonnenuntergang hoch beziehungsweise runterfahren, jedoch nur, falls die Balkontüre geschlossen ist (nicht, dass mich mein eigenes Haus [2001-mäßig aussperrt](#)).

Erklärungen und Probleme

Wie ihr vielleicht bemerkt habt, ist im Arduino-Code eine Zeitsperre eingebaut. Die umgeht das Problem, dass PHP per `fsockopen()` aus einem mir unbekanntem Grund denselben Befehl immer zweimal abschickt – von zwei unterschiedlichen Ports aus. Falls mir das einer erklären kann, wäre ich äußerst dankbar!

Außerdem enthält das Code-Konzept eine per GET-Parameter an den Webserver übermittelte Geräte-ID, die letztendlich nicht weiter benutzt wird – sie soll nur für zukünftige Ausbauten vorsorgen.

Tipp: Wollt Ihr das Ganze nachbauen und es funktioniert trotz gut aussehender Lötstellen nicht (idealerweise wird für den Aufbau ohnehin erst einmal ein Steckbrett verwendet), versucht einfach mal, die Wandschalter-Knöpfe andersherum an Pin 4 und 5 der Optokopplers zu hängen – der darin enthaltene Transistor ist nämlich verpolungsempfindlich (an dieser Stelle geht mein Dank noch einmal ans arduino.cc-Forum!).



Hier verrichtet die Konstruktion ihren Dienst.

Bonus: Benachrichtigungen aufs Smartphone

Zur Kontrolle des HomeMatic-Skripts sowie für den Fall, dass die Rollos aufgrund einer offen stehenden Balkontür einmal nicht herunter gelassen werden können, lasse ich mir Benachrichtigungen aufs Handy schicken. Das Zauberwort dafür heißt Pushover und ist ein Onlinedienst für Push-Nachrichten, der eine API für eigene Skripte und Programme zur Verfügung stellt. Dazu braucht es nur ein entsprechend vorbereitetes Benutzerkonto bei pushover.net sowie etwas curl-Magie. Den Aufruf dazu bringe ich, in ein PHP-Skript verpackt, im HomeMatic-Auslöser beziehungsweise dessen „ELSE“-Pfad unter – so  zumindest der Plan für die nahe Zukunft. Im Moment fehlt mir noch der Öffnungssensor, das PHP-Skript jedoch funktioniert einwandfrei.

Diese Handvoll PHP-Zeilen pusht mir Nachrichten über das Fehlschlagen meiner Rollo-Aktionen aufs Handy:

```
<?php
$url = 'https://api.pushover.net/1/messages.xml';
$fields = array(
    'token' => 'hXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXrc',
    'user' => 'EBXXXXXXXXXXXXXXXXXXXXXXXXXXXXTw',
    'device' => 'meinFon',
    'title' => 'HomeMatic-Meldung',
    'message' => 'Fehler: Jalousie konnte
                nicht heruntergefahren werden.'
);
$fields_string = http_build_query($fields);

$ch = curl_init();

curl_setopt($ch,CURLOPT_URL, $url);
curl_setopt($ch,CURLOPT_POST, count($fields));
curl_setopt($ch,CURLOPT_POSTFIELDS, $fields_string);
$result = curl_exec($ch);

curl_close($ch);
?>
```