



Heimautomatisierung mit FS20 und freier Software

Tim Weber
9. November 2007

<https://scytale.name/files/talks/fs20/>

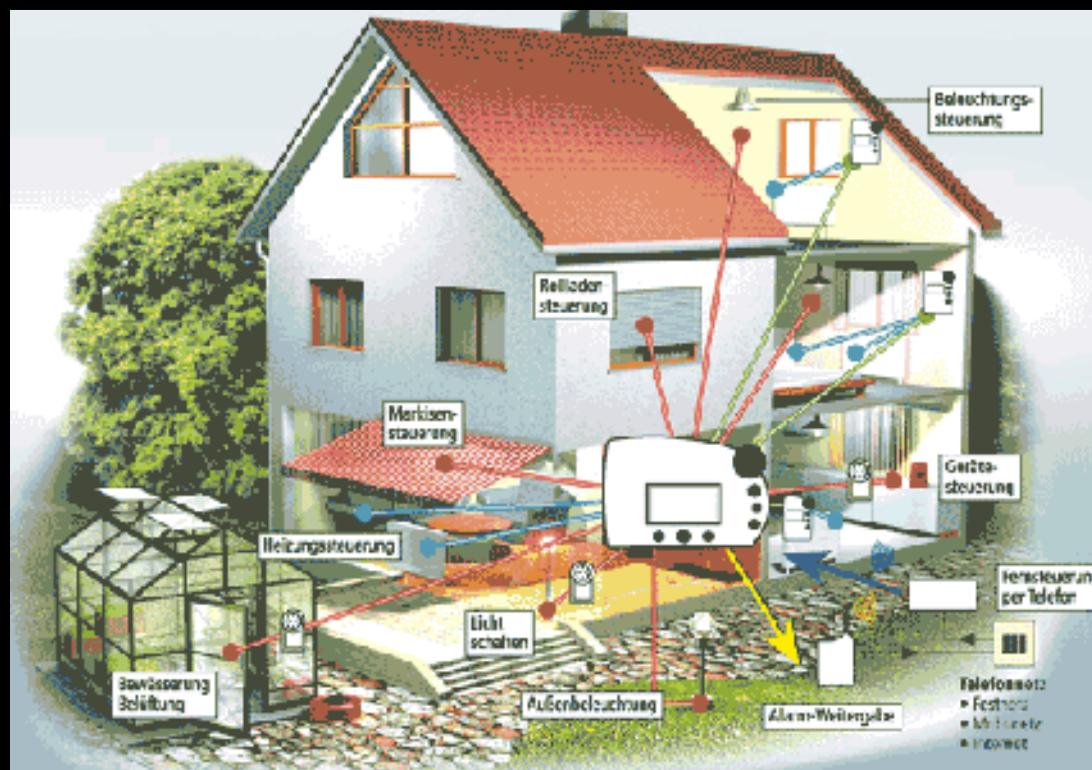


Übersicht

- Was ist FS20?
- Grundlagen des Systems
- Bedienung und Komponenten
- Was geht ganz ohne Rechner?
- Was geht mit Rechner?
- Verfügbare Software
- Beispielszenarien

Was ist FS20?

- (nicht veröffentlichter) Standard für Funksschaltsysteme, Nachfolger von FS10
- 868.35 MHz (ISM/SRD, max. 35mW, DC <1%)





Grundlagen des Systems

- Geräte sind entweder Sender oder Empfänger
- Kommunikation unidirektional, keine Bestätigung
- Übertragung unverschlüsselt, kein wirklicher Schutz vor Abhören oder Schaltung durch Dritte
- Erhältlich von ELV (eigentlicher Hersteller?) und Conrad sowie bei diversen Onlineshops; Preisvergleich lohnt sich



Adressierung

- nur Ziffern von 1-4 zulässig
- Hauscode: achtstellig, muss bei allen Geräten gleich sein (z.B. 12342342), 65.536 Mögl.
- Adressgruppe und Einzeladresse, je zweistellig (z.B. 11 21) identifizieren ein Gerät
 - Empfehlung: Adressgruppe definiert Ort (Wohnzimmer, Arbeitszimmer etc.), Einzeladresse den Gerätetyp (Deckenlampe, Rolladen)
 - Die Adressen, die mit 44 anfangen oder enden, sind reserviert für Gruppen (nächste Folie)



Gruppenadressierung

- Funktionsgruppe: 44xx („4411: Deckenlampen“)
- lokaler Master: xx44 („2144: Wohnzimmer“)
- globaler Master: 4444 („ganzes Haus“)
- Jedem Empfänger können bis zu vier Adressen zugewiesen werden:
 - Einzeladresse (2111)
 - Funktionsgruppe (4411, Deckenlampen)
 - lokaler Master (2144, Wohnzimmer)
 - globaler Master (4444, ganzes Haus)



Bedienung von Sendern

- Sender haben immer mindestens 4 Tasten: je zwei für ein Gerät (aus und an), meist auch „togglen“ von 4 Geräten möglich
- beim gedrückt halten erfolgt Dimmbefehl
- Programmiermodus (Hauscode, zu schaltende Geräte, Timer, Senderspezifisches) durch Halten von Tastenkombinationen
- Beispiele: Aufputz-Taster, Handfernbedienung, Bewegungsmelder, Mastersteckdose, Türkontakt, Klingelschalter, Zeitschaltuhr



Bedienung von Empfängern

- Empfänger besitzen meist eine Taste
 - manuelles „togglen“
 - gedrückt halten für Lernmodus
- im Lernmodus wird Hauscode sowie Adresse nach dem nächsten empfangenen Befehl gesetzt
- Beispiele: Unterputz-Empfänger, Steckdose, Tongeber (Gong)



Was geht ohne einen Rechner?

- „normales“ Ein- und Ausschalten
- Dimmen (je nach Empfänger und Gerät)
- mit Zeitschaltuhr: Wecken, Urlaubsschaltung
- mit Bewegungsmelder: Licht, Alarm
- mit Mastersteckdose und Gong: „Waschmaschine-fertig-Signal“
- telefonischer Notruf bei Einbruch
- Heizungssteuerung (FHT) mit Fenster- und Temperatursensoren



Scheiß auf ohne Rechner, was geht mit?

- Quasi alles.
- an USB anschließbarer Transciever „FHZ“
- mitgelieferte Windows-Software für Zeit- und Klickisteuerung (Grundriss als Hintergrundbild?)
- für Unixe Perl-Tool „FHEM“ mit Kommandozeile und verschiedenen Weboberflächen, beliebig in eigene Software einbindbar



Hard- und Software-Verwirrung

- PC-Zentrale FHZ in mehreren Ausführungen:
 - FHZ1000: FS20 (Schaltssystem), HMS (Sensoren), FHT80b (Heizung)
 - FHZ1300: zusätzlich FHT8b (Heizung), KS300 (Wetterstation)
 - FHZ1350: zusätzlich AW50 (Alarmwähler)
- auch mit WLAN-PCMCIA-Karte erhältlich
- FHZ1000 teilweise durch Software kastriert (kein HMS und FHT), ich empfehle Bestellung bei CESA mit „Standardsoftware“



Konfiguration von FHEM

- (Windows-Software interessiert mich nicht)

```
attr global logfile /home/scy/fhem-%Y-%m.log
```

```
attr global statefile /home/scy/fhem.save
```

```
attr global verbose 4
```

```
attr global port 7072
```

```
attr global modpath /opt/fhem
```

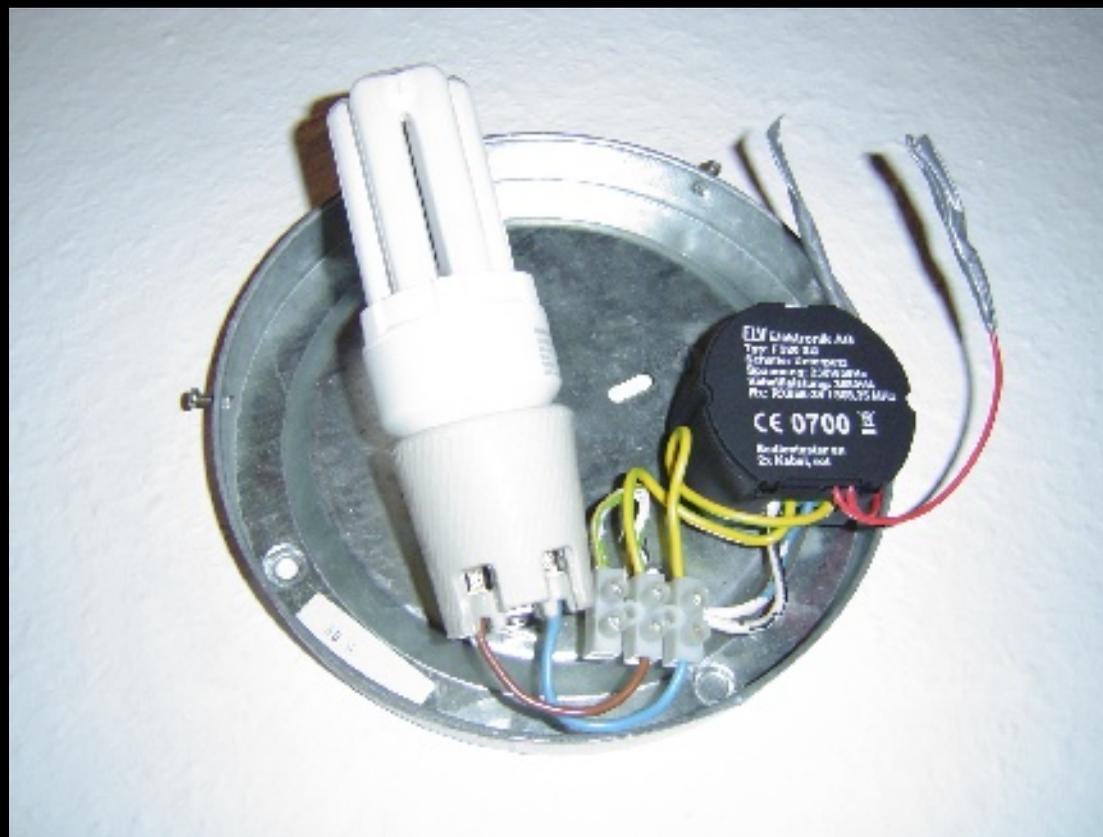
```
# define NAME TYP OPTIONEN
```

```
# FTDI-USB2Serial-Converter-Treiber benötigt
```

```
define FHZ FHZ /dev/ttyUSB0
```

1. Schritt: Küchenlampe an- und ausschalten (I)

- Deckenlampe, verbunden mit Unterputzschalter „FS20 SU“, der aber aus Platzmangel in die Lampe eingebaut wurde
- Wandschalter trennt jetzt gesamte Konstruktion vom Strom anstatt die Lampe zu schalten





1. Schritt: Küchenlampe an- und ausschalten (II)

- Adressierung in FHEM nicht mit 4 (Adresse) oder 8 (Hauscode) Ziffern von 1-4, sondern mit 2 oder 4 Hex-Ziffern, Umrechnungstabelle:

11	12	13	14	21	22	23	24	31	32	33	34	41	42	43	44
0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f

```
# define NAME TYP HAUSCODE ADRESSE
```

```
define kitLamp FS20 1b6d 14 # HC 12342342, Adr. 1221
```



1. Schritt: Küchenlampe an- und ausschalten (III)

- FHEM läuft als Daemon, Befehle müssen mit Clientscript unter Portangabe gesendet werden
- Wrapper-Script bietet sich an (z.B. unter „/usr/local/bin/fhem“ speichern):

```
#!/bin/sh
```

```
/opt/fhem/fhem.pl 7072 "$*"
```

- und jetzt:

```
fhem set kitLamp on
```



2. Schritt: Scanner bei Bedarf anschalten, danach wieder aus

- stinknormale Steckdose „FS20 ST-2“ und saned-Scanserver, dann statt direktem Einbinden in inetd folgendes Wrapper-Script einbinden:

```
#!/bin/sh
```

```
fhem set scanner on; sleep 10
```

```
saned
```

```
sleep 5; fhem set scanner off
```

- natürlich vorher Scanner definieren:

```
define scanner FS20 1b6d 02
```



3. Schritt: Drucker bei Aufträgen anschalten,
danach wieder aus

- „FS20 ST-2“ und CUPS-Druckerserver sowie dieses Script (für Drucker namens FS-3700+):

```
while true; do
  if [[ "$(lpq -P FS-3700+ | tail -n +2 | head -n 1)" \
    != 'no entries' ]]; then
    fhem set printer on-for-timer 320 &>/dev/null
    sleep 25
  fi
  sleep 5
done
```



4. Schritt: Schwer abzuschaltendes Gerät (I)

- „FS20 ST-2“ an Schreibtisch-Steckdosenleiste für PC, Monitor, USB-Platte etc.
- Einschalten soll via Fernbedienung möglich sein, Ausschalten aber nur, wenn man die Aus-Taste länger gedrückt hält („herunterdimmen“)



4. Schritt: Schwer abzuschaltendes Gerät (II)

- Lösung: entsprechendem Tastenpaar nicht die Adresse der Steckdose zuweisen, sondern eine freie Adresse, z.B. 4221 (FHEM: d4), während die Steckdose z.B. 1114 (FHEM: 03) hat, dann folgende Konfiguration:

```
define desk          FS20 1b6d 03
define metaDesk     FS20 1b6d d4
define deskOnMeta   notify metaDesk:(on|off|dimup) set desk on
define deskOffMeta  notify metaDesk:dimdown          set desk off
```



5. Schritt: Gute-Nacht-Timer

- Dimmfunktion des Küchenlampen-Aufputz-Taster momentan nicht genutzt (da nicht dimmbar); soll sich stattdessen bei „hochdimmen“ für 5 Minuten anschalten, danach aus

```
define kLTimer notify kitLamp:dimUp \  
    define kLTimerRun at +00:05:00 set kitLamp off
```



6. Schritt: Temperatur und Luftfeuchte erfassen

- Sensoren „HMS100T“ bzw. „HMS100TF“ besitzen keinen Hauscode, sondern zufällige ID (65.000 Kombinationen), diese muss erstmal aus dem FHEM-Log gefischt werden („unknown device XY, please define“), dann z.B.

```
define innen HMS e157
define aussen HMS 0b02
define tempLog notify (innen|ausen) \
    "/usr/local/bin/fhem-rrd.sh "@ "%""
# @: Gerätename, %: Event (z.B. "measured-temp: 21.7")
```

Weitere Ideen

- FS20-Wardriving (FHZ empfängt (und sendet!) alle Signale, egal welcher Hauscode)
- intelligente Alarmanlage (Bewegungsmelder löst nur dann Alarm aus, wenn nicht Bluetooth-ID des Hausherrn im Raum erkannt wird)
- Mails, SMS, VoIP-Anrufe bei Temperaturproblemen im Server- oder Kühlschrank oder bei Wasser auf dem Bad-Fußboden (Waschmaschine!)
- Module für eigene Basteleien gibt's!



Warnhinweise

- Das FS20-Protokoll ist so abhör- und fälschungssicher wie Buschtrommeln, also nichts Sicherheitskritisches anschließen (Mixer, Server, Stromversorgung Raptorengehege)!
- Das 868-MHz-Band hat eine „<1%“-duty-cycle-Begrenzung, im Schnitt afaik 30s zwischen Befehlen, eignet sich also nicht für „lass die Küchenlampe blinken“... ;-)
- Wie eingangs erwähnt: niemand garantiert dir, dass ein abgesendeter Befehl ankam!



Uniformierte Ressourcenlokatoren

- ELV: <http://www.elv.de/>
- Conrad: <http://www.conrad.de/fs20/>
- FHEM: <http://www.koeniglich.de/fhem/>
- guter (günstiger) Shop: <http://www.cesonet.de/>
- dieser Vortrag:
<https://scytale.name/files/talks/fs20/>