



PC-Aktion auf FS20-Befehl - FS20-PC-Empfänger

Der kleine, am USB-Port betriebene FS20-PC-Empfänger eröffnet beliebigen FS20-Sendern die Welt der direkten, drahtlosen Steuerung von PC-Programmen. Über ein Plug-in der kostenlos erhältlichen Open-Source-Hausautomatisierungs-Software „EventGhost“ können nahezu beliebige Aktionen am PC ausgelöst werden.

FS20 steuert WinAmp & Co.

Mit dem FS20-System ist in puncto Hausautomatisierung bereits vieles realisierbar – insbesondere durch den Einsatz

der Makrosteuerung (FS20 MST1) oder der Funk-Haussteuerungs-Systeme (FHZ). Auch eine Kopplung mit dem KeyMatic-System ist z. B. durch den KM300-FS20 möglich.

Will man allerdings über FS20-Komponenten den vielfach schon fest zum Bestand der heimischen Haustechnik zählenden HTPC (Home Theater Personal Computer) steuern und z. B. Home-Cinema-Anwendungen oder Haus-Automatisierungsaufgaben außerhalb der ELV-Systeme verwenden, stößt man mit der bisher vorhandenen Technik an gewisse Grenzen. Wer recht geschickt etwa in PHP programmieren kann, schafft es vielleicht, den PC direkt zur Mitarbeit zu bewegen, aber das ist ja lange nicht jedermanns Sache.

Eine einfache, von jedem PC-Nutzer ohne Programmierarbeit beherrschbare Lösung musste her, deshalb fiel die Wahl auf die kostenlose Open-Source-Software „EventGhost“. Diese schafft auf einfache Weise eine Brücke zwischen externen Eingabegeräten (wie z. B. Funk- oder IR-Fernbedienungen) und beliebigen PC-Programmen. Über externe Geräte können Makros ausgelöst werden, die bestimmte Aktionen starten und dadurch einen Mediaplayer (z. B. WinAmp) bedienen, E-Mails versenden, den Rechner ausschalten usw.

Den Möglichkeiten, auch zur Einbindung anderer externer

Technische Daten: FS20 PCE

Empfangbare Sender:	alle FS20-Sender (z. B. Funkfernbedienung, Bewegungsmelder, Licht-, Erschütterungs-, Regensensor, Infrarot-Umsetzer ...)
Empfangsfrequenz:	868,35 MHz
Empfangsreichweite:	bis zu 100 m (Freifeld)
Empfangsanzeige:	rote LED
Software/Schnittstelle:	EventGhost (Open Source)/USB-HID
Hardware-Schnittstelle:	USB mit Stecker Typ A
Spannungsversorgung:	USB powered
Stromaufnahme:	<50 mA
USB-Kabellänge:	1 m
Abmessungen (B x H x T):	40 x 16 x 70 mm

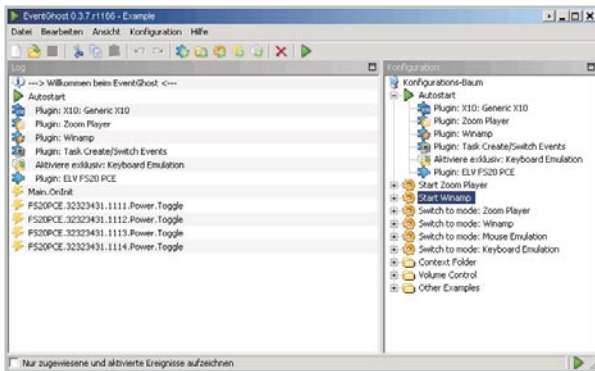


Bild 1: Links im Eventlog erscheinen die vier zuzuordnenden Tasten der Fernbedienung mit ihren Parametern. Sie sollen hier beispielsweise dem Programm „WinAmp“ zugeordnet werden

Steuerungssysteme (z. B. über die EIA232-Schnittstelle), sind hier kaum Grenzen gesetzt.

Einzige Bedingung – es musste ein neues Plug-in für das FS20-System zu „EventGhost“ beigesteuert werden, und der PC musste die Daten der FS20-Sender empfangen können. Letzteres wird durch den hier vorgestellten FS20-PC-Empfänger realisiert.

Er empfängt die FS20-Sendebefehle sämtlicher FS20-Sender (ohne Geräteanzahlbeschränkung und ohne Anlernprozedur am FS20 PCE), stellt im Gegensatz zu einer Zentralen-Lösung dar (treiberlos aufgrund der HID-Plug-and-Play-Realisierung) und erfordert keinerlei Konfiguration und Bedienung am Gerät selbst.

Der PC-Empfänger wird direkt am USB-Port betrieben. Er empfängt alle FS20-Signale und übermittelt Hauscode, Adresscode und Sendebefehl (und weitere Parameter, wenn empfangen) an den PC.

Die Windows-Software „EventGhost“ [1] verarbeitet die empfangenen Befehle (Events) und löst entsprechend der einfach realisierbaren Zuordnung zwischen Auslöser und Reaktion die angestrebte Aktion aus. Möglich ist hier fast alles, und dank des offenen Systems können jederzeit eigene Programmlösungen eingebunden und so das Programm noch universeller gemacht werden. „EventGhost“ ist eine Open-Source-Software, die von einer internationalen Community weiterentwickelt und genutzt wird. Das Programm ist (einschließlich Source-Code) völlig kostenlos erhältlich. Wer die zukünftige Weiterentwicklung, die Dokumentation und das Forum durch eine freiwillige Spende unterstützen möchte (Donationware), findet dazu auf der Projekt-Homepage einen Link.

Um die „EventGhost“-FS20-Unterstützung zu vervollständigen, wird im folgenden „ELVjournal“ 6/09 ein passender, ebenfalls sehr preiswerter FS20-PC-Sender vorgestellt werden. Auch für diesen wird ein eigenes Plug-in direkt in „EventGhost“ integriert werden.

Übrigens – passend zu den möglichen Multimedia-Steuerungs-aufgaben via „EventGhost“ gibt es jetzt auch eine neue FS20-Fernbedienung (siehe Titelbild), die FS20 S16R, deren Tasten ursprünglich für die direkte Nutzung am ebenfalls in diesem „ELVjournal“ vorgestellten FS20-Zwischendecken-Radio FS20 ZDR bedruckt sind. Die Bedruckung ermöglicht aber auch eine hervorragende Nutzung für andere Multimedia-Anwendungen.

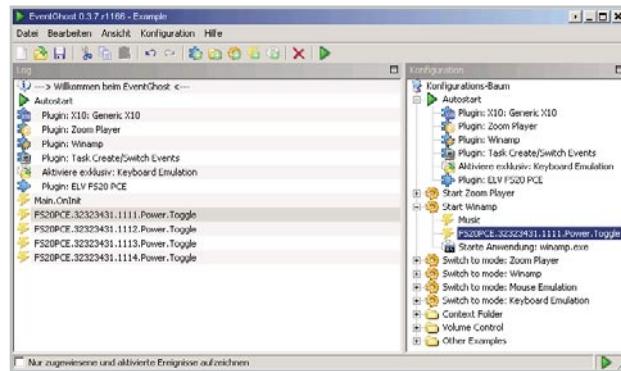


Bild 2: Per Drag & Drop erfolgt hier von links nach rechts die Zuordnung der FS20-Befehle zu den gewünschten Aktionen

Installation des FS20 PCE und Programmbedienung

Die eigentliche Installation ist sehr schnell erledigt, da, wie gesagt, keine Treiber-Installation anfällt, der FS20 PCE ist ein USB-HID-Gerät. Demzufolge ist er einfach mit einem USB-Port zu verbinden und nach kurzem Warten hat der PC das Gerät automatisch erkannt. Ist dies erfolgt, lädt man zunächst „EventGhost“ von [1] herunter und installiert das Programm.

Nach dem Start ist das Konfigurationsmenü anzuwählen und hier die Option „**PlugIn hinzufügen**“ zu wählen. Unter „**Fernbedienungsempfänger**“ findet man „**FS20 PCE**“ – nach Auswählen des Gerätes und Bestätigung über „**OK**“ ist der FS20-Fernbedienempfänger in die Konfiguration aufgenommen.

Als nächste Schritte sind dann nacheinander die gewünschten FS20-Befehle zu senden, z. B. die Tasten der Fernbedienung zu betätigen, die etwa „WinAmp“ ansteuern sollen. Die empfangenen Befehle erscheinen links im Eventlog (Abbildung 1). Diese sind dann der gewünschten Aktion per Drag & Drop zuzuordnen, Abbildung 2 zeigt dies exemplarisch. Die „EventGhost“-Dokumentation [2] beschreibt alle Details dieser Zuordnungen ausführlich.

Die gewünschten Aktionen werden einfach über „Makro hinzufügen“ ausgewählt, wie es Abbildung 3 zeigt, danach erfolgt wieder die beschriebene Zuordnung des FS20-Befehls, Abbildung 4 zeigt dies für den langen Tastendruck auf der FS20-Fernbedienung zum Ausschalten des Rechners. Je nach gewünschter Funktion müssen zuvor zusätzliche Plug-ins hinzugefügt werden.

Diese Vorgänge kann man beliebig oft mit weiteren FS20-Befehlen wiederholen. Das FS20-PCE-Plug-in kennt alle 32 verfügbaren FS20-Befehle (Tabelle 1), die neben Hauscode und Adresse im Klartext im Eventlog erscheinen. Wer selbst in die Programmierung einsteigen möchte, findet die komplette Befehlsspezifikation der USB-HID-Telegramme als PDF-Dokument im „EventGhost“-Plug-in-Verzeichnis. Zusätzliche Plug-ins lassen sich sehr einfach selbst hinzufügen und beschränken sich tatsächlich nur auf das Kopieren einer einzigen Datei (die immer „__init__.py“ heißt – inkl. der Unterstriche) in ein neu anzulegendes Unterverzeichnis im Installationsverzeichnis „C:\Programme\EventGhost\plugins“. Die ausführliche Dokumentation des Programms inkl. umfangreichem Wiki auf der Projekt-Webseite gibt, wie gesagt, eine gute Hilfestellung.



Bild 3: Über „Makro hinzufügen“ können weitere Makros hinzugefügt werden. Hier beispielsweise der Befehl „Rechner ausschalten“, mit dem per FS20 der PC heruntergefahren werden kann

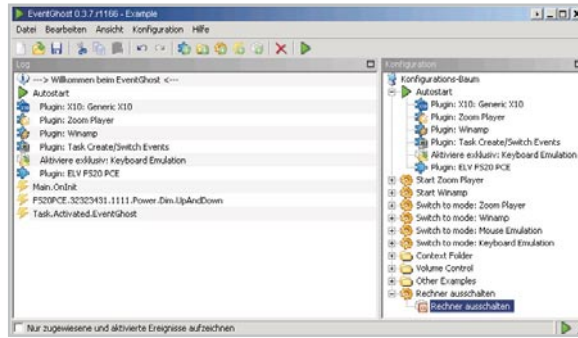
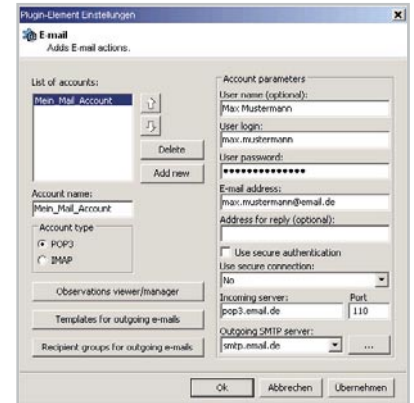


Bild 4: Hier soll ein langer Tastendruck den Rechner ausschalten. Links der zuzuordnende Befehl, rechts die zugehörige Aktion

Eine der wohl interessantesten Aktionen ist das Aussenden einer E-Mail (Abbildung 5). Die etwas knifflige Aufgabe ist aber dank der großen „EventGhost“-Community bereits gelöst – siehe den Forumsbeitrag und den zugehörigen Thread in [3]. Die dort im ersten Post verlinkte Datei „__init__.py“ muss mit rechtem Mausklick ins „EventGhost“-Installationsverzeichnis unter „plugins“ in ein neu anzulegendes Unterverzeichnis „E-mail“ gespeichert werden. Dann sind beliebige FS20-Befehle auch mit einer genau passenden E-Mail-Aussendung verknüpfbar.

Zum Kapitel „Bedienung“ gehört auch die Signalisierungsfunktion der LED am FS20 PCE. Solange eine ordnungsgemäße USB-Verbindung besteht, leuchtet sie. Bei einem FS20-Befehlsempfang verlischt sie kurz und signalisiert somit den Empfang.

Bild 5: Das E-Mail-Plug-in für „EventGhost“, eine Fleißarbeit aus der „EventGhost“-Community



Schaltungsbeschreibung

Dank des hochintegrierten Mikrocontrollers IC 1 ist die in Abbildung 6 gezeigte Schaltung des FS20 PCE sehr übersichtlich. Bei dem 8-Bit-Mikrocontroller handelt es sich um einen C8051F326 der Firma SiliconLabs, in dem die vollständige Hardware zur Realisierung einer USB-Schnittstelle bereits integriert ist. Für die Kommunikation mit einem PC benötigt der C8051F326 damit keine weiteren Bauelemente am USB. Am Pin 26 P0.3 des Mikrocontrollers ist über einen Vorwiderstand die rote LED D 1 als Empfangs- und Aktivitätsanzeige angeschlossen.

Das 868,35-MHz Funk-Empfangsmodul gibt sein digitales Ausgangssignal über Pin 1 P0.0 an den Mikrocontroller aus, der dieses kontinuierlich auf FS20-Telegramme hin auswertet. Die Spannungsversorgung der Schaltung erfolgt über die USB-Schnittstelle. Dabei stellt der PC auf der VBUS-Leitung (USB-Power) eine Spannung von 5 V zur Verfügung, die über den in IC 1 integrierten Spannungsregler auf 3,3 V reduziert wird. Diese Spannung dient nur zum Betrieb des Mikrocontrollers.

Da das Empfangsmodul für eine hohe Empfangsreichweite eine genaue 3-V-Spannung benötigt, wird dieses über den Linearregler IC 2 versorgt. C 1, C 3, C 6, C 7 und C 8 dienen zur Pufferung bzw. als Störunterdrückung.

Nachbau

Der Nachbau des FS20 PCE gestaltet sich sehr einfach, da alle SMD-Bauteile bereits ab Werk bestückt sind und die weitere Bestückung sich auf nur wenige Bauteile beschränkt.

Wir beginnen mit der Bestückung der roten LED, die zunächst polrichtig (der längere Anschluss ist die Anode, er gehört in

Tabelle 1: Mögliche FS20 PCE-Events

Nr.	Event-Bezeichnung	Bedeutung im FS20-System
00	Do.CommandOff	Aus
01	Do.CommandDim6%	Einschalten auf Helligkeitsstufe 1 (6,25 %)
02	Do.CommandDim13%	Einschalten auf Helligkeitsstufe 2 (12,5 %)
03	Do.CommandDim19%	Einschalten auf Helligkeitsstufe 3 (18,75 %)
04	Do.CommandDim25%	Einschalten auf Helligkeitsstufe 4 (25 %)
05	Do.CommandDim31%	Einschalten auf Helligkeitsstufe 5 (31,25 %)
06	Do.CommandDim38%	Einschalten auf Helligkeitsstufe 6 (37,5 %)
07	Do.CommandDim44%	Einschalten auf Helligkeitsstufe 7 (43,75 %)
08	Do.CommandDim50%	Einschalten auf Helligkeitsstufe 8 (50 %)
09	Do.CommandDim56%	Einschalten auf Helligkeitsstufe 9 (56,25 %)
10	Do.CommandDim63%	Einschalten auf Helligkeitsstufe 10 (62,5 %)
11	Do.CommandDim69%	Einschalten auf Helligkeitsstufe 11 (68,75 %)
12	Do.CommandDim75%	Einschalten auf Helligkeitsstufe 12 (75 %)
13	Do.CommandDim81%	Einschalten auf Helligkeitsstufe 13 (81,25 %)
14	Do.CommandDim88%	Einschalten auf Helligkeitsstufe 14 (87,5 %)
15	Do.CommandDim94%	Einschalten auf Helligkeitsstufe 15 (93,75 %)
16	Do.CommandOn	Einschalten auf Helligkeitsstufe 16 (100 %)
17	Do.CommandPreviousValue	Auf letztem Helligkeitswert einschalten
18	Do.CommandToggle	Wechsel zwischen „Aus“ und „An, alter Wert“
19	Do.CommandDimUp	Eine Helligkeitsstufe heller
20	Do.CommandDimDown	Eine Helligkeitsstufe dunkler
21	Do.CommandDimUpAndDown	Heraufdimmen bis Max., Pause, Herabdimmen bis Min. usw.
22	Program.Time	Timer-Programmierung (Start/Ende, „dim-ontime“)
23	Program.SendStatus	Reserviert für bidirektionale Komponenten und Anlernen
24	Do.CommandOff Do.CommandPreviousValue	Aus für Timer-Zeit, danach alte Helligkeit
25	Do.CommandOn Do.CommandOff	An (100 %) für Timer-Zeit, danach Aus
26	Do.CommandPreviousValue Do.CommandOff	An (alte Helligkeit) für Timer-Zeit, danach Aus
27	Program.Reset	Auf Auslieferungszustand zurücksetzen
28	Program.DimUpTime	Timer-Programmierung Zeit für Heraufdimmen
29	Program.DimDownTime	Timer-Programmierung Zeit für Herabdimmen
30	Do.CommandOn Do.CommandPreviousState	An (100 %) für Timer-Zeit, danach alter Zustand
31	Do.CommandPreviousValue Do.CommandPreviousState	An (alte Helligkeit) für Timer-Zeit, danach alter Zustand

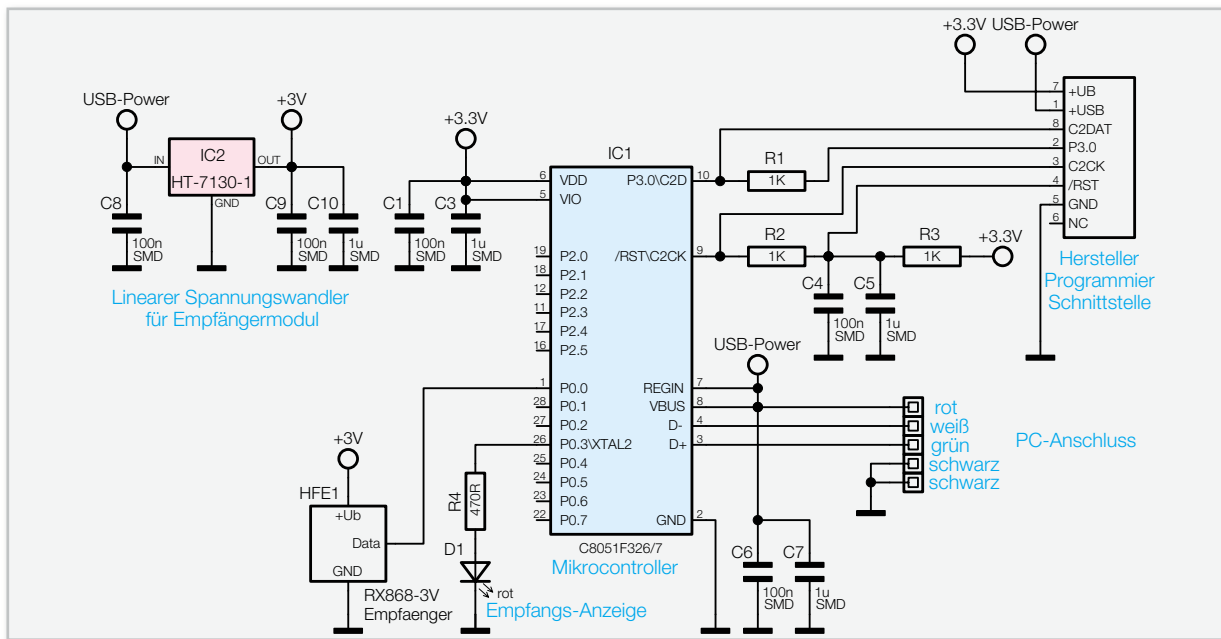


Bild 6: Die Schaltung des FS20 PCE beschränkt sich dank im Mikrocontroller integrierter USB-Schnittstelle auf wenige Elemente

die mit „+“ markierte Bohrung) und mit einem Abstand von 9 mm zwischen Platine und LED-Spitze einzulöten ist.

Anschließend sind drei Lötstifte in die Platine einzulöten, auf die das Funk-Empfangsmodul aufgelötet wird.

Zuletzt lötet man die einzelnen Adern des USB-Kabels auf die entsprechend gekennzeichneten Platinenanschlüsse und fixiert das USB-Kabel mit einem Kabelbinder auf der Platine. Abbildung 7 zeigt die so weit fertiggestellte Platine mit der exakten Kabelmontage. Diese ist nun mit der LED voran (diese muss genau in die zugehörige Öffnung fassen) in die obere Gehäusehälfte des bearbeiteten Gehäuses zu legen, die untere Gehäusehälfte aufzulegen und beide Hälften mit vier Schrauben zu verbinden.

Es folgt ein erster Funktionstest durch Anschließen an einen PC-USB-Port. Dabei muss die LED, nachdem Windows den FS20 PCE erfolgreich installiert hat, dauerhaft leuchten. Empfängt das Gerät ein FS20-Signal, verlischt die LED kurz. Nach dem erfolgreich verlaufenen Funktionstest kann der FS20 PCE in Betrieb genommen werden. **ELV**

Internet:

[1] <http://www.eventghost.org>

[2] http://www.eventghost.org/docs/user_docs.html

[3] <http://www.eventghost.org/forum/viewtopic.php?f=9&t=1168>

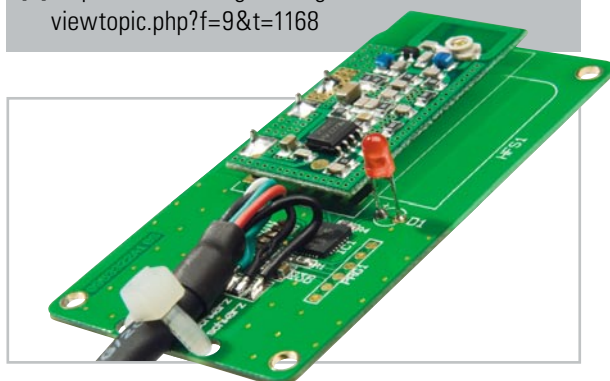
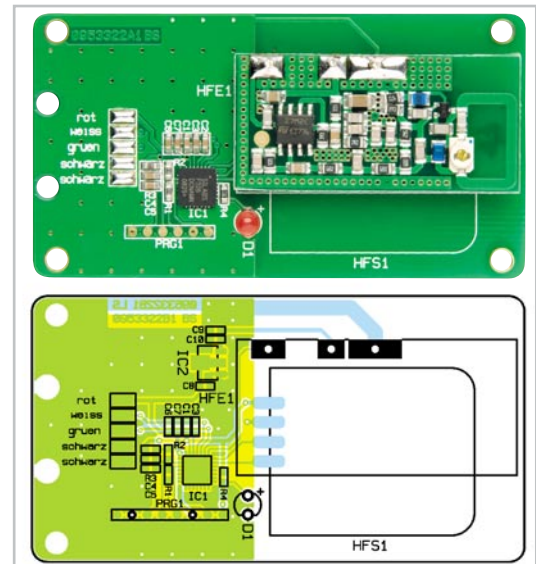


Bild 7: Die fertig aufgebaute Platine des FS20 PCE. Hier sieht man sehr gut die Details zum Einbau von LED und Funkmodul sowie zur Montage des USB-Kabels

Ansicht der fertig bestückten Platine des FS20 PCE mit zugehörigem Bestückungsplan



Stückliste: FS20 PCE

Widerstände:

470 Ω /SMD/0603	R4
1 k Ω /SMD/0603	R1-R3

Kondensatoren:

100 nF/SMD/0603	C1, C4, C6, C8, C9
1 μ F/SMD/0603	C3, C5, C7, C10

Halbleiter:

ELV09888/SMD	IC1
HT7130/SMD	IC2
LED, 3 mm, Rot	D1

Sonstiges:

Empfangsmodul RX868-3V, 868 MHz	HFE1
3 Stiftleisten, 1 x 1-polig, gerade, print	HFE1
1 Gehäuse STRAPU Typ 2741, komplett, bearbeitet und bedruckt, Schwarz	
1 Kabelbinder, 90 mm	
1 USB-Kabel mit Typ A Stecker (einseitig), USB 2.0, 1 m, Schwarz	