

# Lernfähiger RC5-Fernbedienungs-Empfänger

## Preiswerter IR-Lautstärkестeller

Von Jason Vincent-Newson

[jasreb@bigfoot.com](mailto:jasreb@bigfoot.com)

Der hier vorgestellte IR-Fernbedienungsempfänger kann in Verbindung mit jeder RC5-Fernbedienung verwendet werden, um mittels eines Motorpotis die Lautstärke eines hochwertigen (Röhren-)Vorverstärkers einzustellen.



Eine Besonderheit dieses Empfängers besteht darin, dass er so programmiert werden kann, dass jedes beliebige Tastenpaar auf einer RC5-kompatiblen Fernbedienung für die zu steuernde Funktion verwendet werden kann. Man kann dem Empfänger auf sehr einfache Weise beibringen, auf welche Tasten

des Fernbedienungssenders er reagieren soll. Durch Setzen eines Jumpers geht die Empfängerschaltung nach dem Einschalten in den Programmiermodus, der durch eine blinkende LED angezeigt wird. Wenn man jetzt auf dem Handsender die

beiden Tasten betätigt, die zum Fernbedienen der Lautstärke verwendet werden sollen, merkt sich der Empfänger diese Informationen (Systemadressen und Befehle) und speichert sie in einem nichtflüchtigen Speicher.

Auf gleiche Weise ist auch beliebig oft eine Neuprogrammierung möglich, zum Beispiel, wenn ein anderer Handsender verwendet werden soll.

Den Ausgang der Empfängerschaltung bildet ein Motortreiber mit integrierten Power-Opamps, der den 5-V-Motor eines Motorpotis direkt ansteuern kann. Natürlich können die Ausgänge des Motortreibers auch zwei 5-V-Relais (mit Freilaufdioden!) oder Optotriacs

bzw. Solid-state-Relais ansteuern, so dass sich der Fernbedienungsempfänger auch für andere Anwendungen eignet. Die Platine verfügt über eine 10-polige Anschlussleiste, auf der sämtliche Anschlüsse zusammengefasst sind, so dass man Pla-

tine und andere Elemente auch voneinander räumlich trennen kann.

## Die Schaltung

Wie **Bild 1** zeigt, kommt man dank der Verwendung eines Mikrocontrollers für die gesamte Schaltung mit nur drei integrierten Schaltkreisen aus: Neben dem Atmel-Mikrocontroller (IC1) ist das der schon erwähnte Motortreiber (IC2) und ein IR-Empfängerbaustein (IC3).

Der IR-Empfänger sorgt auch schon für eine Demodulation des empfangenen IR-Signals, so dass an seinem Ausgang bereits ein Digitalsignal mit 5-V-Pegel anliegt, das vom Mikrocontroller direkt verarbeitet werden kann. Für diesen Baustein lassen sich vier verschiedene Typen verwenden, die aber unterschiedliche Anschlussbelegungen aufweisen, die im Schaltbild dargestellt sind.

Der für IC1 verwendete Atmel AVR AT90S2343 ist ein kleines 8-Pin-Controllerchen, das 2 K Flash-Programmspeicher und 128 Byte nichtflüchtigen EEPROM-Datenspeicher enthält. Ein interner Oszillator gibt 1 MHz Taktrate vor, womit dank des effizienten Controller-Konzepts eine Rechenleistung von fast 1 MIPS erreicht wird. In der Schaltung des IR-Empfängers werden alle fünf I/O-Pins verwendet:

- PB3 dient als RC5-Dateneingang (active Low)
- PB4 fragt den PROG-Jumper ab (PROG-Modus bei Low aktiv).
- PB1 und PB2 sind die UP/DOWN-Ausgänge, die den L272M-Motortreiber und die UP/DOWN-Anzeige-LEDs steuern).
- PB0 ist gleichzeitig Enable-Ausgang für den Motortreiber (bei Low) und Anzeige-Treiber für die PROG-Modus-LED (bei High).

Der L272-Motortreiber funktioniert wie folgt:

Die beiden invertierenden Eingänge der Opamps sind miteinander verbunden und liegen an einem Spannungsteiler mit R4 und R11, der die Spannung an diesen Eingängen auf etwa 2,5 V legt, wenn der Enable-Ausgang (PB0) des Mikrocontrollers Low ist.

In diesem Modus werden die Aus-

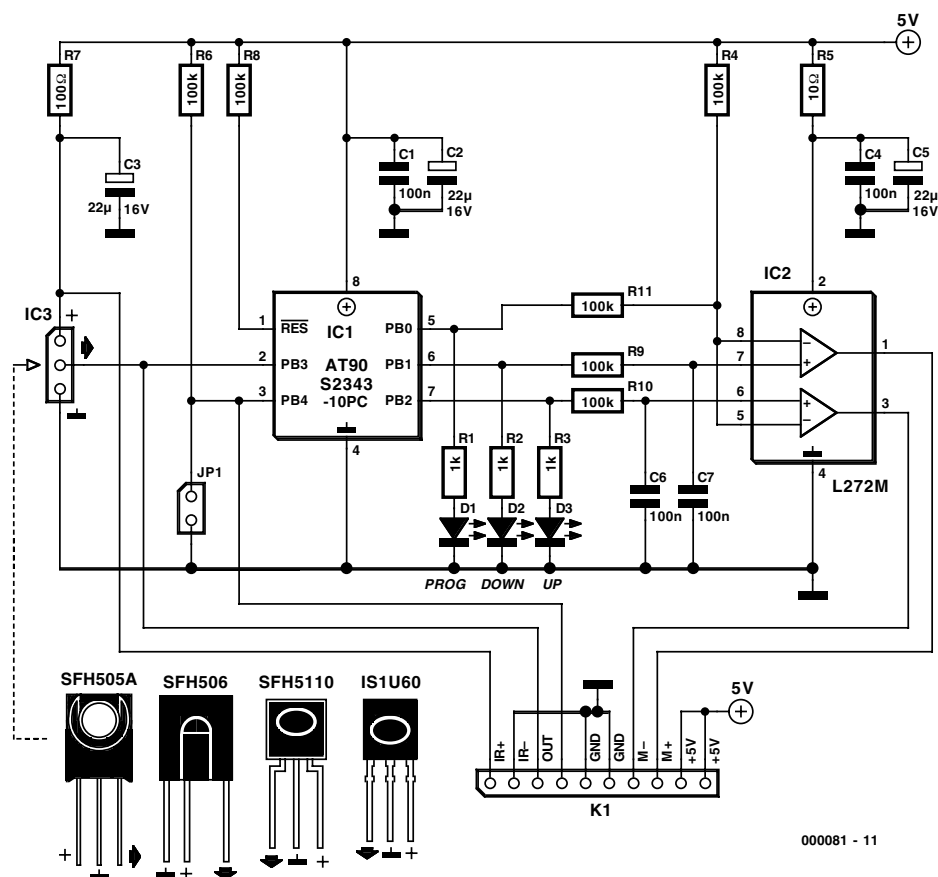


Bild 1. Schaltplan des "lernfähigen" RC5-Empfängers. Der L272M-Motortreiber kann 5-V-Gleichstrommotoren, aber auch 5-V-Relais ansteuern.

gänge des Motortreibers in Abhängigkeit von den Pegeln an den Mikrocontrollerausgängen PB1 und PB2 Up-down-gesteuert. Ist Pin 7 High und Pin 6 Low, ist der eine Ausgang des L272M ebenfalls High und der andere Low. Ändern sich die Pegel an den Eingangspins (6 High, 7 Low), so ändert sich die Polarität an den Ausgangspins ebenfalls und damit auch die Drehrichtung des angesteuerten Motors.

Ist der PB0-Ausgang des Mikrocontrollers HIGH, wird der Motortreiber deaktiviert, weil die Eingangspins (8 und 5) auf +5 V gelegt werden, so dass keiner der beiden Opamp-Ausgänge mehr positiv werden kann. Die 10-Ω- und 100-Ω-Widerstände dienen zusammen mit den Kondensatoren zum Abblocken der 5-V-Anschlüsse, um Störsignale von den ICs fern zu halten. Die Ruhestromaufnahme der Schaltung ist vernachlässigbar, die Betriebsstromaufnahme hängt dementsprechend fast nur von der Stromaufnahme des angeschlossenen Motors ab.

## Bauhinweise

Für den Aufbau der Schaltung wurde eine kleine, einseitige Platine entwickelt (**Bild 2**). Zumindest für den Mikrocontroller IC1 sollte man eine Fassung vorsehen, IC2 kann man

## AT90S Mikrocontroller selbst programmiert

Wer einen für Atmel-AT90S-Mikrocontroller geeigneten Programmer hat, der kann die Software für den RC5-Empfänger gratis vom Elektor-Website herunterladen und den Controller IC1 damit selbst programmieren. Bei <http://www.elektor.de> findet man unter "Gratis-Downloads" bei den Downloads für Dezember 2000 die Datei # **000081-11.exe**, die sowohl den Sourcecode (Bascom AVR Basic) als auch die Hex-Datei für die Programmierung des Mikrocontrollers enthält. Wer den Controller nicht selbst programmieren kann oder keine Downloadmöglichkeit hat, der findet in der Stückliste auch die Bestellnummern für den programmierten Controller und die Softwarediskette.

auch direkt auf die Platine löten. Für die Anschlüsse kann man anstelle von Stiftleisten natürlich auch einzelne Lötstifte (Lötnägel) verwenden. Die Stiftleiste bietet aber auch die Möglichkeit, die Platine als steckbare Baugruppe in eine passende Buchsenleiste einzustecken.

## Anwendung

Wenn beim Anlegen der Betriebsspannung an die Platine der PROG-Jumper gesteckt ist (d.h., wenn PB4 beim Einschalten an Masse liegt), startet die Schaltung im Programmiermodus (PROG-Modus):

### PROG-Modus

Die rote (PROG) LED leuchtet und die Down-LED (D2) blinkt, um anzuzeigen, dass auf dem Fernbedienungs-Handsender die Taste gedrückt werden muss, die der Volume-Down-Funktion zugeordnet sein soll. Die RC5-Fernbedienung sendet beim Drücken der Taste den der Taste zugeordneten Befehlscode und die Fernbedienungs-Systemadresse. Sobald diese beiden Codes empfangen wurden, hört die Down-LED zu blinken auf, dafür beginnt jetzt die Up-LED zu blinken, um anzuzeigen, dass die Taste für UP auf dem Fernbedienungs-Handsender gedrückt werden muss. Sobald nun der dieser Taste zugeordnete Code empfangen wurde, geht die rote PROG-LED aus, der Fernbedienungsempfänger wechselt gleichzeitig vom PROG-Modus in den NORMAL-Modus.

### NORMAL-Modus

In diesem Modus wartet der Fernbedienungsempfänger fortwährend das vom IR-Sensor empfangene Signal aus und wartet darauf, dass er einen der zuvor im PROG-Modus "gelernten" Befehlskodes erkennt. Sobald der Mikrocontroller einen der beiden bekannten Codes identifiziert, steuert er den Motortreiber entsprechend dem empfangenen Befehl (Up oder Down). Die Ansteuerung hält so lange an, wie die entsprechende Taste des Fernbedienungs-Handsenders gedrückt wird. Lässt man die Taste los, wird kein Code mehr empfangen, und die Motortreiberausgänge gehen auf Null (Masse).

Beim Einschalten im Normal-Modus (wenn der PROG-Jumper nicht gesetzt ist) kontrolliert der Mikrocontroller, ob in seinem nichtflüchtigen Datenspeicher Codes gespeichert sind. Wenn keine relevanten Daten festgestellt werden (Empfänger ist noch nicht programmiert oder hat die programmierten Daten verloren), geht der Empfänger automatisch in den PROG-Modus.

(000081-1e)

## Stückliste

### Widerstände:

R1,R2,R3 = 1 k  
R4,R6,R8...R11 = 100 k  
R5 = 10 Ω  
R7 = 100 Ω

### Kondensatoren:

C1,C4,C6,C7 = 100 n  
C2,C3,C5 = 22 µ/16V stehend

### Halbleiter:

D1,D2,D3 = Low-current LED  
IC1 = AT90S2343-10PC, programmiert\* **000081-41** (siehe Text)  
IC2 = L272M

IC3 = IR-Empfänger/Demodulator mit 36 kHz Mittenfrequenz, IS1U60 (Sharp) oder SFH506-36, SFH5110-36, SFH5110 (Siemens)

### Außerdem:

JPI = 2-polige Stiftleiste mit Jumper.  
K1 = 10-polige SIL-Stiftleiste  
Platine\* **000081-1**.  
Diskette\* mit Sourcecode und Hex-Datei **000081-11**

\*Programmierte Controller, Platinen und Disketten siehe Serviceside in der Heftmitte und Website [www.elektor.de](http://www.elektor.de)