

# RC5

(Dekodierung mit  
PIC-Mikrocontroller)

Autor:

Buchgeher Stefan

Letzte Bearbeitung:

29. September 2004

# Inhaltsverzeichnis

|           |  |           |
|-----------|--|-----------|
| <b>1.</b> | <b>RC5-PROTOKOLL</b> .....                             | <b>3</b>  |
| <b>2.</b> | <b>HARDWARE</b> .....                                  | <b>4</b>  |
| <b>3.</b> | <b>SOFTWARE</b> .....                                  | <b>5</b>  |
| 3.1.      | Das Dekodierverfahren .....                            | 5         |
| 3.2.      | Benötigte Register, Konstanten und Portdefinition..... | 7         |
| 3.3.      | Initialisierung (Unterprogramm „INIT“).....            | 8         |
| 3.4.      | Unterprogramme für die RC5-Dekodierung .....           | 9         |
| 3.4.1.    | Unterprogramm RC5ROUTINE.....                          | 9         |
| 3.4.2.    | Unterprogramm RC5CHECKTELEGRAMM .....                  | 12        |
| 3.4.3.    | Unterprogramm RC5AKTION.....                           | 15        |
| <b>4.</b> | <b>DEMONSTRATIONSBEISPIEL</b> .....                    | <b>16</b> |
| 4.1.      | Hardware.....  | 16        |
| 4.2.      | Software.....  | 17        |
| 4.3.      | Anmerkungen zur Software .....                         | 25        |
| <b>5.</b> | <b>QUELLEN</b> .....                                   | <b>26</b> |

# 1. RC5-Protokoll

Der RC5-Code wurde von Philips entwickelt und ist in Europa ein sehr weit verbreiteter Standard zur Infrarot-Datenübertragung. Er wird sehr oft bei Infrarot-Fernbedienungen eingesetzt.

Im Ruhezustand, also wenn keine Taste gedrückt ist, ist der Pegel low. Bei einem Tastendruck wird ein 24.889ms langer Code gesendet. Dieser Code wird so oft wiederholt, solange die Taste gedrückt bleibt, wobei dazwischen je eine Pause von 88.889ms erfolgt (siehe Bild 1.1 Oben).

Der RC5-Code ist *biphase* kodiert. Das bedeutet dass ein Bit aus zwei alternierenden Halbbits zusammengesetzt wird. Anders ausgedrückt bedeutet dies, dass eine logische „1“ aus einem Low-High-Übergang und eine logische „0“ aus einem High-Low-Übergang besteht. (siehe Bild 1.1 Mitte und Unten).

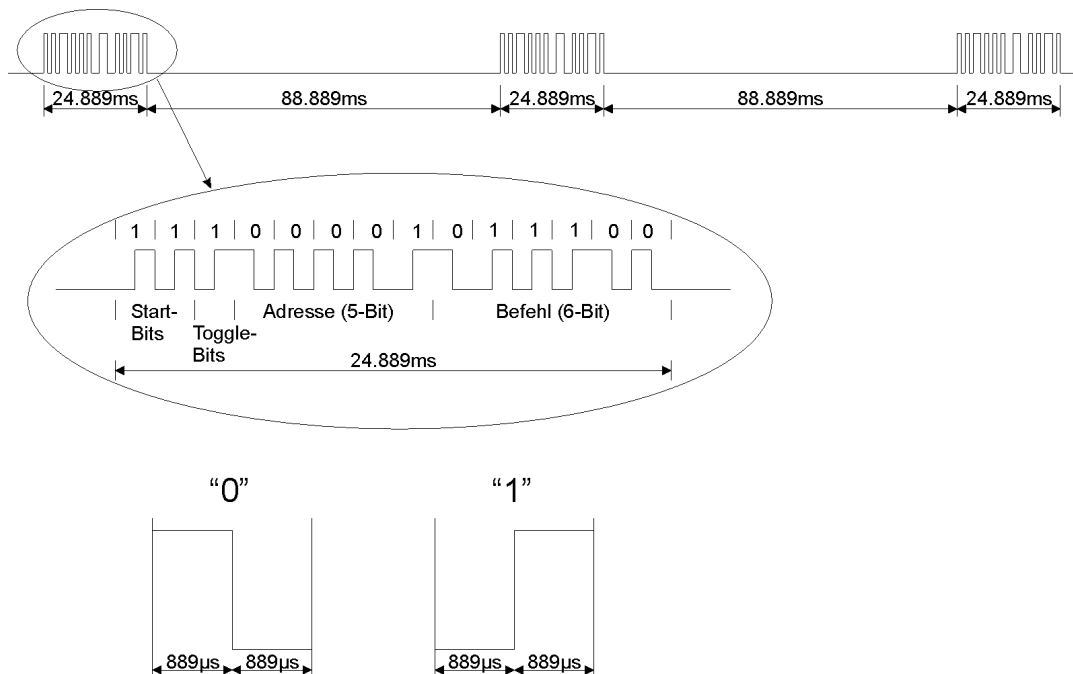


Bild 1.1: RC5-Protokollaufbau

Der ausgestrahlte Code besteht aus einem 14 Bit langen Datenwort (siehe Bild 1.1 Mitte) und ist wie folgt aufgebaut.

- 2 Start-Bits
- 1 Toggle-Bit
- 5 Systemadressen-Bits
- 6 Befehls-Bits

Die beiden Startbits dienen zur Einstellung der so genannten *auto gain control (AGC)* im Empfänger IC.

Das Toggle-Bit ändert bei jedem Tastendruck seinen Zustand. Damit kann zwischen einem dauerhaften Druck der gleichen Taste und einem mehrmaligen Drücken der gleichen Taste unterschieden werden.

Die Systemadressen-Bits geben an, welches Gerät (TV1, Videorekorder, etc.) von der Fernbedienung angesprochen werden soll. Tabelle 1.1 zeigt die Zuordnung zwischen Gerät und Systemadresse laut Philips. Die Systemadresse ist in dieser Tabelle dezimalkodiert.

| Adr. | Gerät  | Adr. | Gerät   |
|------|--|------|---|
| 0    | TV1 (Fernsehempfänger 1)                           | 16   | PREAMP1 (Audio-Vorverstärker 1)               |
| 1    | TV2 (Fernsehempfänger 2)                           | 17   | TUNER (Radio-Tuner)                           |
| 2    | TXT (Video-/Teletext)                              | 18   | REC1 (Kassettenrekorder 1)                    |
| 3    | Erweiterung für TV1 und TV2                        | 19   | PREAMP2 (Audio-Vorverstärker 2)               |
| 4    | LV1 (Laser Video Player 1)                         | 20   | CD (CD-Player)                                |
| 5    | VCR1 (Videorekorder 1)                             | 21   | PHONO (HiFi-Anlagensteuerung)                 |
| 6    | VCR2 (Videorekorder 2)                             | 22   | SAT (Satellitenempfänger)                     |
| 7    | Frei   | 23   | REC2 (Kassettenrekorder 2) / Digital-Rekorder |
| 8    | SAT1 (Satellitenreceiver 1)                        | 24   | Frei  |
| 9    | Erweiterung für VCR1 und VCR2                      | 25   | Frei  |
| 10   | SAT2 (Satellitenreceiver 2)                        | 26   | CD-R (Beschreibbare CD)                       |
| 11   | Frei   | 27   | Frei  |
| 12   | LV2/CDLV ((Laser Video Player 2 / CD-Video-Player) | 28   | Frei  |
| 13   | Frei   | 29   | Lichtsteuerungen                              |
| 14   | CDPH (CD-Photo-Player)                             | 30   | Frei  |
| 15   | Frei   | 31   | Frei  |

Tabelle 1.1: Systemadressen

Die Adressen 7 und 13 sind für Selbstbaugeräte und für Experimentierzwecke vorgesehen.

Die Befehls-Bits geben an, welche Taste auf der Fernbedienung gedrückt wurde. Tabelle 1.2 zeigt die Befehle welche bei alle Adressen gelten.

| Befehl | Taste auf der Fernbedienung | Befehl | Taste auf der Fernbedienung |
|--------|-----------------------------|--------|-----------------------------|
| 0      | 0                           | 18     | Brightness +                |
| 1      | 1                           | 19     | Brightness -                |
| 2      | 2                           | 20     | Color saturation +          |
| 3      | 3                           | 21     | Color saturation -          |
| 4      | 4                           | 22     | Bass +                      |
| 5      | 5                           | 23     | Bass -                      |
| 6      | 6                           | 24     | Treble +                    |
| 7      | 7                           | 25     | Treble -                    |
| 8      | 8                           | 26     | Balance right               |
| 9      | 9                           | 27     | Balance left                |
| 16     | Volume +                    | 63     | System select               |
| 17     | Volume -                    |        |                             |

Tabelle 1.2: Gemeinsame Befehle aller Adressen

## 2. Hardware

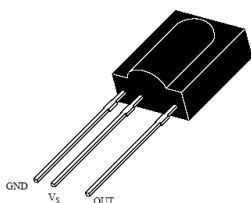


Bild 2.1: IR-Empfangsmodul

Die Hauptkomponente zur RC5-Dekodierung ist ein bei Conrad erhältliches IR-Empfangsmodul vom Typ TSOP1736 (Bestell-Nr.: 171069). Dieses IR-Empfangsmodul beinhaltet alle für die Demodulation notwendigen Funktionsgruppen. Der Ausgang dieses ICs liefert den im Abschnitt 1 beschriebene Code, mit dem Unterschied dass dieser Datenstrom invertiert ist. Im Ruhezustand

ist der Ausgang nicht Low (so wie im Abschnitt 1 beschrieben), sondern High.

Das folgende Bild zeigt das Blockschaltbild des für die Demodulation verwendeten IR-Empfänger TSOP1736.

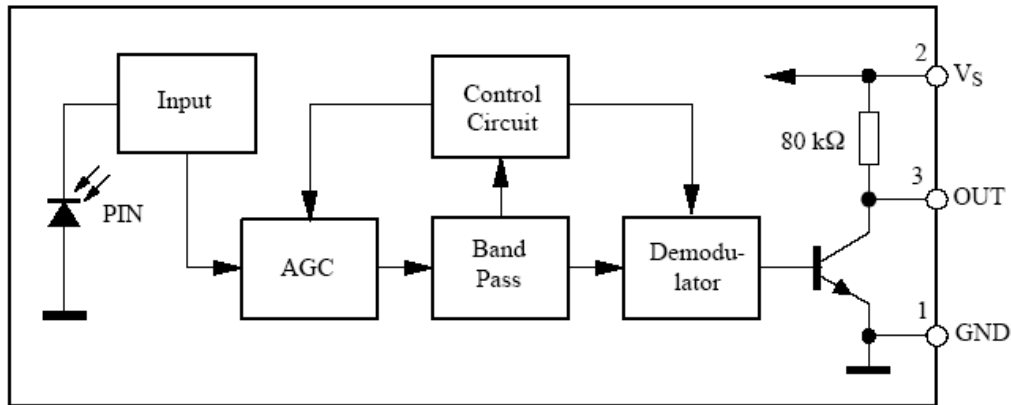


Bild 2.2: Blockschaltbild des IR-Empfänger TSOP1736

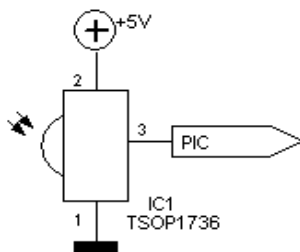


Bild 2.3: Beschaltung des IR-Empfänger TSOP1736

Für die Dekodierung (z.B. mit einem PIC-Mikrocontroller) des vom IR-Empfänger empfangenen Datenstroms ist neben dem IR-Empfänger selbst kein weiteres Bauteil notwendig. Die Pins 1 und 2 des IR-Empfängers sind mit einer Betriebsspannung, welcher laut Datenblatt zwischen 4.5V und 5.5V liegen soll, zu versorgen.

**Achtung:** Der hier verwendete IR-Empfänger (IC1, TSOP1736) invertiert den empfangenen Datenstrom. Im Ruhezustand ist der Ausgang nicht Low (so wie im Abschnitt 1 beschrieben), sondern High.

## 3. Software

### 3.1. Das Dekodierverfahren

Die Aufgabe der PIC-Software besteht bei der RC5-Dekodierung aus mehreren Teilaufgaben:

1. In kurzen Zeitabständen (alle 250µs) das vom IR-Empfangsmodul erzeugte Signal abtasten.<sup>1</sup> Aus diesen Abtastungen den Telegrammbeginn und die im Abschnitts 1 beschriebenen Low- und High-Bits ermitteln
2. Die empfangenen Low- und High-Bits zwischenspeichern.
3. Nachdem alle zu einem Telegramm gehörenden Bits eingelesen wurden, dieses empfangene Telegramm mit dem zuvor empfangenen Telegramm vergleichen.

<sup>1</sup> Die 250µs-Zeitabstände müssen für die RC5-Dekodierung sehr exakt sein, so dass als Taktquelle für den PIC-Mikrocontroller nur ein Quarz mit einer entsprechenden Quarzfrequenz verwendet werden kann (z.B. ein 4,096MHz-Quarz). Mit einem geeigneten Vorteiler (im PIC) lässt sich so eine einfache und relativ genaue 250µs-Zeitbasis erzeugen.

Nur wenn nacheinander mehrere gleiche Telegramme empfangen wurden, wird das soeben empfangene Telegramm als gültig angesehen.<sup>2</sup> Durch diese Maßnahme, dass mehrere gleiche Telegramme empfangen werden müssen, werden andere IR-Protokolle ignoriert. D.h. die Software „reagiert“ nur auf Fernbedienungen welche das RC5-Protokoll aussenden.

4. Aus einem als gültig erkannten Telegramm die Adresse und den Befehl ermitteln und in die Register RC5ADRESSE bzw. RC5BEFEHL sichern. Weiters das Toggle-Bit auslesen und im Register RC5STATUS sichern, und ein Flag namens RC5TELGUELTIG ebenfalls im Register RC5STATUS setzen. Das gesetzte Flag RC5TELGUELTIG zeigt an, dass ein gültiges RC5-Telegramm empfangen und dekodiert wurde, und dieses nun für die anderen Unterprogramme zur Verfügung steht.
5. Je nach empfangenem Befehl oder empfangener Adresse z.B. einen Zähler erhöhen oder vermindern, oder einen Schaltausgang setzen oder rücksetzen, etc. Diese Aufgabe richtet sich nach der Aufgabenstellung bzw. Anwendung.

Die Aufgaben 1. und 2. werden von einem Unterprogramm namens RC5ROUTINE (siehe Abschnitt 3.4.1.) zyklisch alle 250µs ausgeführt. Als Zeitbasis für den 250µs-Takt dient das Timer-0-Interrupt. Der Aufruf des Unterprogramms RC5ROUTINE erfolgt hier aber nicht von der ISR (Interrupt-Service-Routine), sondern vom Hauptprogramm. Die ISR setzt nur alle 250µs ein Flag zur Kennzeichnung, dass das Hauptprogramm das Unterprogramm RC5ROUTINE aufrufen soll.

Für die Aufgaben 3. und 4. ist das Unterprogramm RC5CHECKTELEGRAMM (siehe Abschnitt 3.4.2.) zuständig. Dieses Unterprogramm wird erst dann ausgeführt, wenn das Unterprogramm RC5ROUTINE das Flag RC5CHECKTEL im Register RC5STATUS setzt, also wenn ein komplettes RC5-Telegramm vom Unterprogramm RC5ROUTINE dekodiert wurde.

Die 5. Aufgabe kann von einem beliebigen Unterprogramm ausgeführt werden. Hier, bei der Demo lautet dieses Unterprogramm RC5AKTION.

Diese drei Unterprogramme (RC5ROUTINE, RC5CHECKTELEGRAMM und RC5AKTION) werden vom Hauptprogramm aufgerufen, wenn die jeweiligen Bits (= Botschaftsflags) gesetzt sind. Das Hauptprogramm prüft dazu ständig nacheinander die Flags FLAG250USEK, RC5CHECKTEL und RC5NEUETASTE. Ist das Flag FLAG250USEK gesetzt ruft das Hauptprogramm das Unterprogramm RC5ROUTINE auf. Das (Botschafts)-Flag FLAG250USEK wird von einer Timer-ISR alle 250µs gesetzt. Ist das Flag RC5CHECKTEL gesetzt, ruft das Hauptprogramm das Unterprogramm RC5CHECKTELEGRAMM auf. Ist das Flag RC5NEUETASTE gesetzt, ruft das Hauptprogramm das Unterprogramm RC5AKTION auf. Der Name dieses Unterprogramms (RC5AKTION) kann beliebig gewählt werden.

---

<sup>2</sup> Dies ist möglich, da ja das RC5-Telegramm einer gedrückten Taste mehrmals (ca. alle 114ms gesendet wird, solange diese Taste gedrückt bleibt) gesendet wird

## 3.2. Benötigte Register, Konstanten und Portdefinition

### Register:

Für die RC5-Dekodierung sind neben einigen internen Register (SFR, **Spezielle Funktions-Register**) noch einige eigene Register notwendig:

- **RC5STATUS:** Statusregister der RC5-Dekodierung. Dieses Register beinhaltet folgende Bits:

| Bit | Flagname       | Funktion  |
|-----|----------------|---|
| 0   | RC5START       | Hilfsflag zur Erkennung eines Telegrammstarts   |
| 1   | RC5STARTFLANKE | Hilfsflag zur Erkennung eines Telegrammstarts   |
| 2   | RC5CHECKTEL    | gesetzt, wenn ein vollständiges RC5-Telegramm dekodiert wurde                                 |
| 3   | RC5TELGUELTIG  | gesetzt, wenn mehrmals das selbe Telegramm empfangen wurde                                    |
| 4   | RC5TOGGLEALT   | Toggle-Bit des „alten“ Telegramms   |
| 5   | RC5TOGGLENEU   | Toggle-Bit  |
| 6   | RC5NEUETASTE   | gesetzt, wenn eine neue Taste gedrückt wurde und mehrmals das selbe Telegramm empfangen wurde |

Tabelle 3.1: Zusammensetzung von RC5STATUS

- **RC5ZAEHLER1, RC5ZAEHLER2:** Zählregister (siehe Abschnitt 3.4.1)
- **RC5ZAEHLERGUELTTEL:** Dieses Zählregister gibt an, wie viele gleiche RC5-Telegramme noch empfangen werden müssen, damit der empfangene RC5-Code als gültig anerkannt wird
- **RC5TELEGRAMM0A, RC5TELEGRAMM0B:** Hilfsregister (diese dienen zur Ermittlung des Telegrammstarts.)
- **RC5TELEGRAMM1, RC5TELEGRAMM2:** Beinhalten das empfangene RC5-Telegramm
- **RC5TELEGRAMM1ALT, RC5TELEGRAMM2ALT:** Dienen zur Ermittlung eines gültigen RC5-Telegramms (der ausgesendete RC5-Code muss von der Dekodiersoftware zumindest zweimal gleich empfangen werden).
- **RC5ADRESSE:** Beinhalten die dekodierte Systemadresse (also die Systemadresse der verwendeten Fernbedienung; siehe Tabelle 1.1)
- **RC5BEFEHL:** Beinhalten den dekodierten Befehl (also den Wert der auf der Fernbedienung gedrückten Taste.)

### Konstanten:

- **KONSTRC5ZAEHLERSTART:** Diese Konstante dient zum Laden von RC5ZAEHLER1, unmittelbar nach einem Telegrammbeginn
- **KONSTRC5ZAEHLER:** Diese Konstante dient zum Nachladen von RC5ZAEHLER1
- **KONSTRC5GUELTTEL:** Diese Konstante gibt ab, wie viele gleiche RC5-Telegramme empfangen werden müssen, damit dieser als gültig angesehen wird.

Es kann erforderlich sein, dass bei einer Anwendung die PIC-Taktfrequenz erhöht werden muss. In diesem Fall kann es notwendig sein, die Konstanten KONSTRC5ZAEHLERSTART und KONSTRC5ZAEHLER anzupassen.

**Portdefinition:**

Im Allgemeinen wird bei jeder Anwendung der Eingangspin für die RC5-Dekodierung an einem anderen Portpin verwendet. Damit dies in der Software nur an einer Stelle berücksichtigt werden muss befindet sich in der Software eine Portdefinition für den RC5-Eingang. Diese besteht aus den folgenden 3 Parametern:

- RC5INPORT: Dieser Parameter gibt den Port an (z.B. Port A, Port B,...)
- RC5INTRIS: Dieser Parameter ist für die Initialisierung des Verwendeten Ports zuständig. Für die RC5-Dekodierung muss der verwendete Portpin als Eingang definiert werden
- RC5IN: Dieser Parameter gibt den Portpin des Verwendeten Ports an (z.B. 0, 1, 2, usw.)

Achtung: Wird für RC5INPORT der Port B verwendet, so muss für RC5INTRIS das zum Port B zugehörige TRIS-Register definiert werden. Eine mögliche Portdefinition für einen PIC-Mikrocontroller der PIC16Fxx-Familie ist:

```
RC5INPORT    equ    PORTB
RC5INTRIS    equ    TRISB
RC5IN        equ    0
```

Bei Verwendung eines PIC-Mikrocontrollers aus der PIC12Fxx-Familie lautet diese Portdefinition wie folgt:

```
RC5INPORT    equ    GPIO
RC5INTRIS    equ    TRISIO
RC5IN        equ    3
```

### 3.3. Initialisierung (Unterprogramm „INIT“)

Dieses Unterprogramm dient zur Initialisierung des Mikrocontrollers. Der Portpin an dem der RC5-Empfänger angeschlossen ist muss als Eingang definiert werden.

Da das Unterprogramm RC5ROUTINE zyklisch (alle 250µs) aufgerufen wird, ist eine entsprechende Zeitbasis notwendig. Diese wird mit Hilfe eines Timer-Interrupt erzeugt. Für die Definition der Zeitbasis ist hier das mikrocontrollerinterne Funktions-Register OPTREG (in der Registerbank 1) zuständig. Damit bei einer PIC-Taktfrequenz von 4,096MHz eine Zeitbasis von 250µs erzeugt wird, muss das Register OPTREG mit dem binären Wert b'00001000' geladen werden. Das Zählregister für dies Zeitbasis (Funktions-Register TMR0, in Registerseite 0) muss gelöscht werden.

Weiters müssen für die RC5-Dekodierung einige Register vorbelegt (initialisiert) werden.

Der folgende Programmausschnitt zeigt eine mögliche Initialisierungsroutine für einen PIC12F629. Die schwarz hervorgehobenen Stellen sind für die RC5-Dekodierung notwendig.

```
INIT          clrf    TMR0                ;Timer0 auf 0 voreinstellen
              clrf    GPIO
              movlw  0x07                ;Alle Comparatoreingaenge
```

```

movwf CMCON           ; auf digital I/O umschalten

bsf  STAT,RP0        ;Registerseite 1

movlw b'00001000'    ;kein Vorteiler von TMR0
movwf OPTREG

clrf  TRISIO         ;Port als Ausgang definieren
bsf  RC5INTRIS,RC5IN ;RC5IN als Eingang definieren

bcf  STAT,RP0        ;Registerseite 0

clrf  GPIO           ;Port loeschen

clrf  RC5STATUS      ;RC5-Statusregister initialisieren
; (loeschen)

clrf  RC5TELEGRAMM0A ;RC5TELEGRAMM0A initialisieren (loeschen)
clrf  RC5TELEGRAMM0B ;RC5TELEGRAMM0B initialisieren (loeschen)
clrf  RC5TELEGRAMM1ALT ;RC5TELEGRAMM1ALT initialisieren
; (loeschen)
clrf  RC5TELEGRAMM2ALT ;RC5TELEGRAMM2ALT initialisieren
; (loeschen)
clrf  RC5DEMOZAEHLER ;RC5DEMOZAEHLER initialisieren (loeschen)

movlw KONSTRC5GUELTTEL ;RC5ZAEHLERGUELTTEL initialisieren (mit
movwf RC5ZAEHLERGUELTTEL; der Konstanten KONSTRC5GUELTTEL)

return

```

## 3.4. Unterprogramme für die RC5-Dekodierung

Wie im Abschnitt 3.1 schon beschrieben sind für die RC5-Dekodierung die folgenden 3 Unterprogramme notwendig:

- *RC5ROUTINE*
- *RC5CHECKTELEGRAMM*
- *RC5AKTION*<sup>3</sup>

### 3.4.1. Unterprogramm RC5ROUTINE

Das Unterprogramm RC5ROUTINE ist für die RC5-Dekodierung die wichtigste Komponente.

Diese Routine wird ca. alle 250µs aufgerufen

#### **Aufgabe:**

Die Aufgabe des Unterprogramms RC5ROUTINE besteht darin, den am RC5-Eingangspin (RC5IN) anstehenden Datenstrom zyklisch (alle 250µs) abzutasten. Aus diesen Abtastungen den Telegrammbeginn und die laut RC5-Protokoll definierten Low- und High-Bits ermitteln. Die so ermittelten Low- und High-Bits in den Registern RC5TELEGRAMM1 und RC5TELEGRAMM2 zwischenspeichern, so dass daraus der

<sup>3</sup> Dieses Unterprogramm ist für die eigentliche Dekodierung nicht von Bedeutung. Hier erfolgen nur die Aktionen, welche z.B. beim drücken der Taste „0“, oder „Lautstärke +“ etc. erfolgen sollen. Der Name dieses Unterprogramms kann daher auch beliebig geändert werden, oder sogar entfallen, wenn diese Tätigkeiten in einem anderen Unterprogramm ausgeführt werden.

übertragene Befehl und die übertragene Systemadresse ermittelt werden kann. (Anm. Das Ermitteln der Systemadresse und des Befehls ist nicht die Aufgabe dieses Unterprogramms. Diese Tätigkeit wird u.a. vom Unterprogramm RC5CHECKTELEGRAMM übernommen. Siehe auch Abschnitt 3.4.2.)

### Vorgehensweise:

Das folgende Bild zeigt das Flussdiagramm dieses nicht allzu komplexen Unterprogramms.

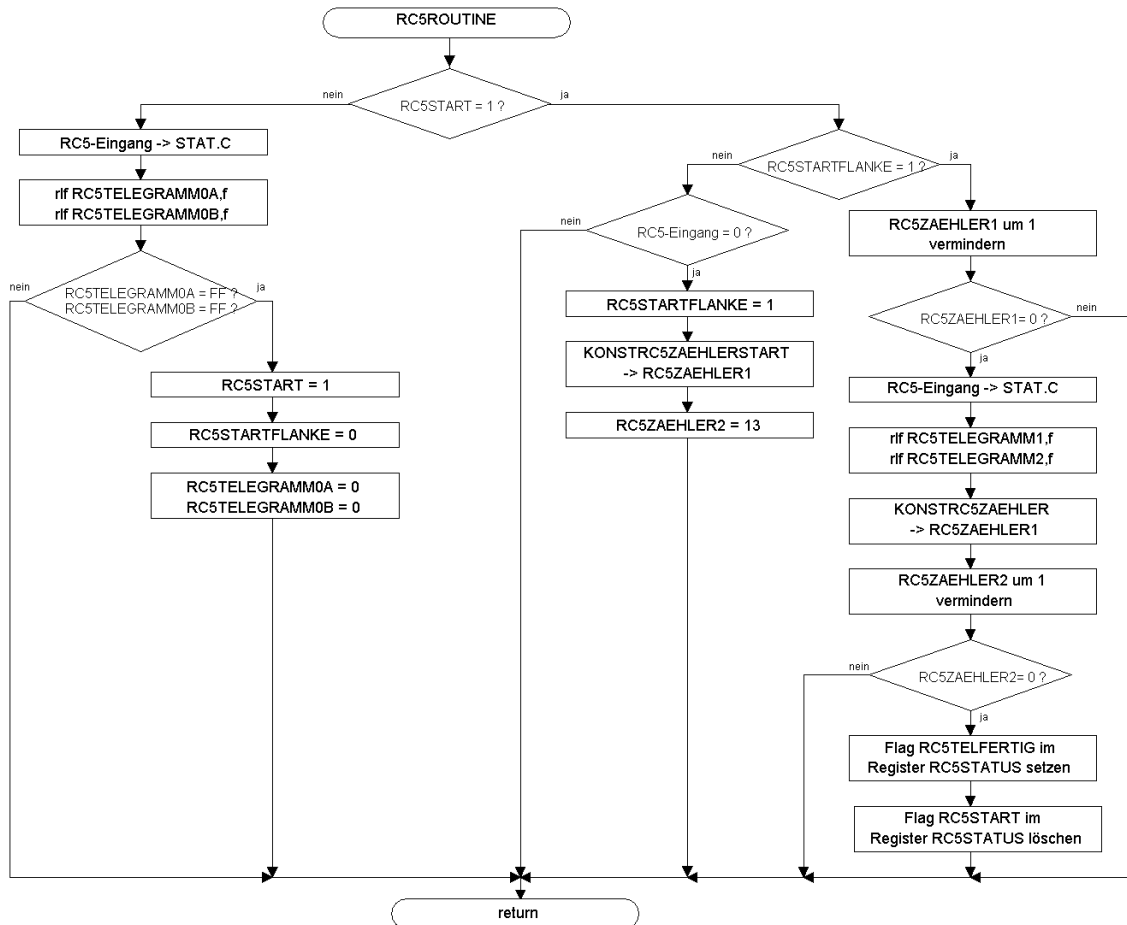


Bild 3.1: Flussdiagramm (RC5ROUTINE)

Zuerst muss auf einen Telegrammbeginn gewartet werden. Für diesen Zweck dienen die beiden Flags RC5START und RC5STARTFLANKE im Register RC5STATUS. Diese Flags werden bei der Initialisierung des Mikrocontrollers gelöscht (siehe Abschnitt 3.3), sind also zu Beginn gelöscht.

Ein Telegrammbeginn wird folgendermaßen erkannt: solange das Flag RC5START nicht gesetzt ist wird der abgetastete RC5-Eingang an die niederwertigste Stelle des Registers RC5TELEGRAMM0A geschoben. Alle anderen Bits dieses Registers werden um je eine Position weiter geschoben. Das höchstwertige Bit vom Register RC5TELEGRAMM0A wird an die niederwertigste Stelle des Registers RC5TELEGRAMM0B geschoben. Alle anderen Bits dieses Registers werden auch hier um je eine Position weiter geschoben. Anschließend wird geprüft, ob beide Register den Wert FFh beinhalten. Beinhalten beide Register (RC5TELEGRAMM0A und RC5TELEGRAMM0B) den Wert FFh, so wird zurzeit kein RC5-Code übertragen und es kann auf einen Telegrammbeginn gewartet werden. Das Flag RC5START wird gesetzt, das Flag RC5STARTFLANKE wird gelöscht, und die Register RC5TELEGRAMM0A und RC5TELEGRAMM0B werden ebenfalls gelöscht. Und nun wird solange gewartet

bis am RC5-Eingang ein Low anliegt. Ist dies der Fall so erfolgt ein Telegrammbeginn. Das Flag RC5STARTFLANKE wird zu diesem Zweck gesetzt und die beiden Zählregister RC5ZAEHLER1 und RC5ZAEHLER2 geladen.

Da das RC5-Telegramm aus 14 Bits besteht und das erste Startbit soeben erkannt wurde muss die Software nur mehr 13 Bits dekodieren. Daher wird das Register RC5ZAEHLER2 mit dem Wert 13 geladen. Dieses Zählregister gibt somit an wie viele Bits des RC5-Telegramms noch zu dekodieren sind.

Der Telegrammbeginn wurde nun erkannt. Nun müssen die einzelnen Bits dekodiert werden. Ein Bit nach dem RC5-Protokoll besteht aus einer 889µs langen Low-Phase und einer 889µs langen High-Phase (bei einer „0“, bei einer „1“ umgekehrt). Bei jedem weiteren Aufruf dieses Unterprogramms wird nun das Zählregister RC5ZAEHLER1 um 1 vermindert. Beinhaltet es danach den Wert 0, so wird der abgetastete RC5-Eingang an die niederwertigste Stelle des Registers RC5TELEGRAMM1 geschoben. Alle anderen Bits dieses Registers werden um je eine Position weiter geschoben. Das höchstwertige Bit vom Register RC5TELEGRAMM1 wird an die niederwertigste Stelle des Registers RC5TELEGRAMM2 geschoben. Alle anderen Bits dieses Registers werden auch hier um je eine Position weiter geschoben. (Anders ausgedrückt: der Zustand an RC5-Eingang wird dem RC5-Telegramm hinzugefügt). Anschließend wird der Zähler RC5ZAEHLER1 mit der Konstanten KONSTRC5ZAEHLER neu geladen und der Zähler RC5ZAEHLER2 um 1 vermindert. Besitzt der Zähler RC5ZAEHLER2 nun den Wert 0, so wurde ein komplettes RC5-Telegramm eingelesen und in den Registern RC5TELEGRAMM1 und RC5TELEGRAMM2 gesichert. Zur Kennzeichnung, dass ein komplettes RC5-Telegramm empfangen wurde, muss das Botschaftsflag RC5TELFERIG im Register RC5STATUS gesetzt werden, während das Flag RC5START (ebenfalls im Register RC5STATUS) zurückgesetzt werden muss, da ja nun auf einen neuen Telegrammbeginn gewartet werden muss.

### Hier das Unterprogramm:

```
RC5ROUTINE  btfsc RC5STATUS,RC5START ;Flag RC5START gesetzt?
            goto  RC5WEITER1      ; ja -> Weiter beim Label RC5WEITER1

            bcf   STAT,C          ; nein: dem Telegramm0 mit Hilfe des
            ; Carry
            btfsc RC5INPORT,RC5IN ; den aktuellen Zustand vom RC5-Eingang
            bsf   STAT,C          ; hinzufuegen
            rlf   RC5TELEGRAMM0A,f
            rlf   RC5TELEGRAMM0B,f

            movf  RC5TELEGRAMM0B,w
            sublw b'11111111'

            btfss STAT,Z          ; RC5TELEGRAMM0B = FF ?
            goto  RC5ROUTINEFERTIG ; nein: UP verlassen

            movf  RC5TELEGRAMM0A,w ; ja: RC5TELEGRAMM0A = FF ?
            sublw b'11111111'

            btfss STAT,Z
            goto  RC5ROUTINEFERTIG ; nein: UP verlassen

            bsf   RC5STATUS,RC5START ; ja: Flag RC5START setzen
            bcf   RC5STATUS,RC5STARTFLANKE ; Flag RC5STARTFLANKE loeschen
            clrf  RC5TELEGRAMM0A ; Register RC5TELEGRAMM0A und
            clrf  RC5TELEGRAMM0B ; Register RC5TELEGRAMM0B loeschen
            goto  RC5ROUTINEFERTIG ; UP verlassen
```

## RC5 (Dekodierung mit PIC-Mikrocontroller)

```
RC5WEITER1  btfsc RC5STATUS,RC5STARTFLANKE ;Flag RC5STARTFLANKE gesetzt?
             goto  RC5WEITER2           ; ja -> Weiter beim Label RC5WEITER2

             btfsc RC5INPORT,RC5IN      ; nein: RC5-Eingang = 0?
             goto  RC5ROUTINEFERTIG    ; nein: UP verlassen

             bsf   RC5STATUS,RC5STARTFLANKE ; ja: Flag RC5STARTFLANKE setzen
             movlw KONSTRC5ZAEHLERSTART ; RC5ZAEHLER1 = KONSTRC5ZAEHLERSTART
             movwf RC5ZAEHLER1
             movlw .13                   ; RC5ZAEHLER2 = 13
             movwf RC5ZAEHLER2
             goto  RC5ROUTINEFERTIG    ; UP verlassen

RC5WEITER2  decfsz RC5ZAEHLER1,f        ;RC5ZAEHLER1 um 1 vermindern und auf 0
             ; pruefen
             goto  RC5ROUTINEFERTIG    ;wenn RC5ZAEHLER1 ungleich 0 -> UP
             ; verlassen

             bcf   STAT,C               ;RC5ZAEHLER1 = 0: dem RC5-Telegramm mit
             ; Hilfe
             btfsc RC5INPORT,RC5IN      ; des Carry den aktuellen Zustand von
             bsf   STAT,C               ; RC5-Eingang hinzufuegen
             rlf   RC5TELEGRAMM1,f
             rlf   RC5TELEGRAMM2,f

             movlw KONSTRC5ZAEHLER      ; RC5ZAEHLER1 = KONSTRC5ZAEHLER
             movwf RC5ZAEHLER1

             decfsz RC5ZAEHLER2,f        ; RC5ZAEHLER2 um 1 vermindern und auf 0
             ; pruefen
             goto  RC5ROUTINEFERTIG    ; wenn RC5ZAEHLER2 ungleich 0 -> UP
             ; verlassen

             bsf   RC5STATUS,RC5CHECKTEL ; RC5ZAEHLER2 = 0: Flag RC5CHECKTEL
             bcf   RC5STATUS,RC5START ; setzen und Flag RC5START loeschen

RC5ROUTINEFERTIG
             Return
```

### Anzahl der Zyklen:

Min: 10 Zyklen (ohne Aufruf)

Max: 21 Zyklen (ohne Aufruf)

## 3.4.2. Unterprogramm RC5CHECKTELEGRAMM

### Aufgaben und Vorgehensweise:

- Neues Telegramm (RC5TELEGRAMMxALT) mit dem neuen Telegramm (RC5TELEGRAMMx) vergleichen.
- Bei Gleichheit:
  - Das Zählregister (RC5ZAEHLERGUELTTEL) für die gleichen Telegramme um 1 vermindern.
  - Ist dieses Zählregister nun 0 (d.h. Es wurden mehrere gleiche RC5-Telegramme empfangen) die RC5-Adresse aus den folgenden Bits zusammensetzen:
    - Bit 6 von RC5TELEGRAMM1 -> Bit 0 der RC5-Adresse
    - Bit 7 von RC5TELEGRAMM1 -> Bit 1 der RC5-Adresse
    - Bit 0 von RC5TELEGRAMM2 -> Bit 2 der RC5-Adresse

- Bit 1 von RC5TELEGRAMM2 -> Bit 3 der RC5-Adresse
  - Bit 2 von RC5TELEGRAMM2 -> Bit 4 der RC5-Adresse
- und den RC5-Befehl aus den 6 niederwertigsten Bits von RC5TELEGRAMM1 zusammensetzen
- Das Flag RC5TOGGLENEU nach RC5TOGGLEALT kopieren und Bit 3 von RC5TELEGRAMM2 nach RC5TOGGLENEU kopieren.
- Zählerregister (RC5ZAEHLERGUELTTEL) für die Ermittlung der gültigen Telegramme neu mit der Konstanten KONSTRC5GUELTTEL laden
- Botschaftsflag "Gültiges RC5-Telegramm" setzen
- Prüfen, ob das neue Togglebit (RC5TOGGLENEU) mit dem alten Togglebit (RC5TOGGLEALT) übereinstimmt, dazu die benachbarten Flags RC5TOGGLEALT (Bit 4) und RC5TOGGLENEU (Bit 5) Exklusiv-Oder-Verknüpfen und nach RC5TELEGRAMM1 kopieren  
(Anm.: Das Register RC5TELEGRAMM1 wird hier als temporäres Hilfsregister verwendet, da der Inhalt für die RC5-Dekodierung hier nicht mehr benötigt wird)
- Bei Ungleichheit:
  - Die Register RC5TELEGRAMMx in die Register RC5TELEGRAMM1ALT kopieren und den Zähler RC5ZAEHLERGUELTTEL mit der Konstante KONSTRC5GUELTTEL neu laden.
- Anforderungsflag RC5CHECKTEL (im Register RC5STATUS) löschen

Das folgende Flussdiagramm soll das soeben beschriebene Unterprogramm verdeutlichen.

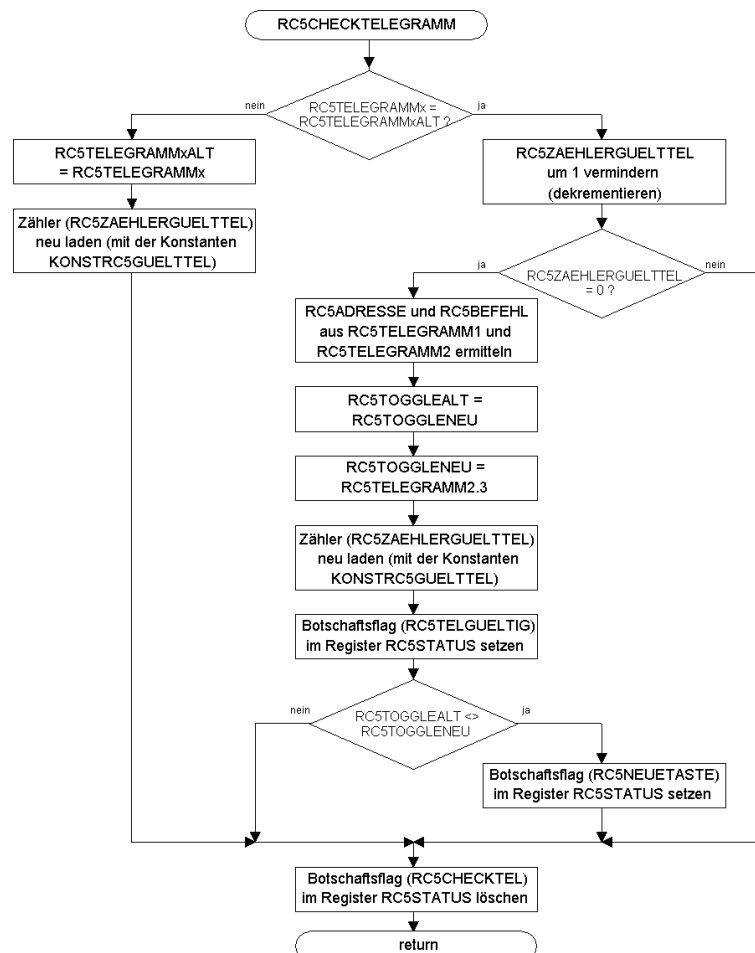


Bild 3.2: Flussdiagramm (RC5CHECKTELEGRAMM)

**Hier das Unterprogramm:**

RC5CHECKTELEGRAMM

```

movf RC5TELEGRAMM1ALT,w ;RC5TELEGRAMM1ALT mit RC5TELEGRAMM1
                           ;vergleichen

subwf RC5TELEGRAMM1,w

btfss STAT,Z
goto RC5UNGLEICH ;bei Ungleichheit springe zum Label
                 ;RC5UNGLEICH

movf RC5TELEGRAMM2ALT,w ;bei Gleichheit RC5TELEGRAMM2ALT mit
subwf RC5TELEGRAMM2,w ; RC5TELEGRAMM2 vergleichen

btfss STAT,Z
goto RC5UNGLEICH ;bei Ungleichheit springe zum Label
                 ;RC5UNGLEICH

decfsz RC5ZAEHLERGUELTTEL,f ;bei Gleichheit von RC5TELEGRAMM1ALT
                           ; mit RC5TELEGRAMM1 und RC5TELEGRAMM2ALT
                           ; mit RC5TELEGRAMM2 das Zaehregister
                           ; fuer die gleichen Telegramme
                           ; (RC5ZAEHLERGUELTTEL) um 1 vermindern.
goto RC5NEUESTELFERTIG ; Zaehregister RC5ZAEHLERGUELTTEL = 0
                           ; nein: springe zum Label
                           ; RC5NEUESTELFERTIG
                           ; ja: mehrere (durch die Konstante
                           ; KONSTRC5GUELTTEL definierte) gleiche
                           ; und gueltige RC5-Telegramme wurden
                           ; empfangen

clrf RC5ADRESSE ; nun die RC5-Adresse zusammensetzen

btfsc RC5TELEGRAMM1,6
bsf RC5ADRESSE,0
btfsc RC5TELEGRAMM1,7
bsf RC5ADRESSE,1
btfsc RC5TELEGRAMM2,0
bsf RC5ADRESSE,2
btfsc RC5TELEGRAMM2,1
bsf RC5ADRESSE,3
btfsc RC5TELEGRAMM2,2
bsf RC5ADRESSE,4

movf RC5TELEGRAMM1,w ; ... und den RC5-Befehl aus den 6
andlw b'00111111' ; niederwertigseten Bits von
movwf RC5BEFEHL ; RC5TELEGRAMM1 zusammensetzen

bcf RC5STATUS,RC5TOGGLEALT ; RC5TOGGLENEU -> RC5TOGGLEALT
btfsc RC5STATUS,RC5TOGGLENEU
bsf RC5STATUS,RC5TOGGLEALT

bcf RC5STATUS,RC5TOGGLENEU ; RC5TELEGRAMM2,3 ->
; RC5TOGGLENEU

btfsc RC5TELEGRAMM2,3
bsf RC5STATUS,RC5TOGGLENEU

movlw KONSTRC5GUELTTEL ; Zaehregister fuer die Ermittlung
movwf RC5ZAEHLERGUELTTEL ; gueltiger Telegramme neu aus der
                           ; Konstante laden

bsf RC5STATUS,RC5TELGUELTIG ; Botschaftsflag "Gueltiges
                           ; RC5-Telegramm" setzen

;Pruefen, ob das neue Togglebit (RC5TOGGLENEU) mit dem alten

```

## RC5 (Dekodierung mit PIC-Mikrocontroller)

```
;Togglebit (RC5TOGGLEALT) uebereinstimmt
rlf  RC5STATUS,w      ;Die benachbarten Flags RC5TOGGLEALT (Bit
xorwf RC5STATUS,w    ; 4) und RC5TOGGLENEU (Bit 5) Exklusiv
; Oder-verknuepfen und nach RC5TELEGRAMM1
movwf RC5TELEGRAMM1  ; kopieren (Anm.: Das Register
; RC5TELEGRAMM1 wird hier als temporaeres
; Hilfsregister verwendet, da der Inhalt
; fuer die RC5-Dekodierung hier nicht
; mehr benoetigt wird)
btfsc RC5TELEGRAMM1,RC5TOGGLENEU ;Ist das Ergebnis der Exklusiv-
; Oder 1, so
bsf  RC5STATUS,RC5NEUETASTE ; sind die beiden Toggleflags
; ungleich. Es wurde also eine neue Taste
; gedrueckt.
goto RC5NEUESTELFERTIG

;Bei Ungleichheit von RC5TELEGRAMM1ALT mit RC5TELEGRAMM1 oder von
;RC5TELEGRAMM2ALT mit RC5TELEGRAMM2
RC5UNGLEICH movf  RC5TELEGRAMM1,w  ;RC5TELEGRAMM1ALT = RC5TELEGRAMM1
movwf RC5TELEGRAMM1ALT
movf  RC5TELEGRAMM2,w  ;RC5TELEGRAMM2ALT = RC5TELEGRAMM2
movwf RC5TELEGRAMM2ALT

movlw KONSTRC5GUELTTEL ;Zaehler RC5ZAEHLERGUELTTEL mit der
movwf RC5ZAEHLERGUELTTEL ;Konstante KONSTRC5GUELTTEL neu laden
goto RC5NEUESTELFERTIG

RC5NEUESTELFERTIG
bcf  RC5STATUS,RC5CHECKTEL ;Anforderungsflag RC5CHECKTEL loeschen
return
```

### Anzahl der Zyklen:

Min: 14 Zyklen (ohne Aufruf)

Max: 43 Zyklen (ohne Aufruf)

## 3.4.3. Unterprogramm RC5AKTION

Hier erfolgen nun die für die Aufgabenstellung gewünschten Aktionen auf einen dekodierten RC5-Code.

In diesem Unterprogramm kann z.B. abhängig von einem bestimmten RC5-Code (also von einer bestimmten gedrückten Taste auf einer Fernbedienung welche den RC5-Code ausstrahlt) ein Zählregister erhöht oder vermindert werden, oder ein bestimmtes Flag oder eine I/O-Pin kann nun gesetzt oder rückgesetzt werden.

Hier, zur Erläuterung das Unterprogramm RC5AKTION des Demonstrationsbeispiels von Abschnitt 4:

### Aufgabe und Vorgehensweise des Unterprogramms „RC5AKTION“ im Demo-beispiels:

Entspricht der empfangene RC5-Befehl der Konstanten KONSTRC5DEMOPLUS, (die Konstante KONSTRC5DEMOPLUS entspricht der Taste "Lautstärke erhöhen"), den Zähler RC5DEMOZAEHLER um 1 erhöhen. Entspricht der empfangene RC5-Befehl der Konstanten KONSTRC5DEMOMINUS, (die Konstante KONSTRC5DEMOMINUS entspricht der Taste "Lautstärke vermindern"), den Zähler RC5DEMOZAEHLER um 1 vermindern. Anschließend nur die 3 niederwertigsten Bits vom Zähler RC5DEMOZAEHLER am Port GPIO ausgeben.

Zum Schluss das Anforderungsflag RC5NEUETASTE wieder löschen

**Hier das Unterprogramm:**

```

RC5AKTION    movlw  KONSTRC5DEMOPLUS ;Entspricht der empfangene RC5-Befehl der
              subwf  RC5BEFEHL,w      ; Konstanten KONSTRC5DEMOPLUS, (die
                                      ; Konstante KONSTRC5DEMOPLUS entspricht
                                      ; der Taste "Lautstaerke erhoehen") ...

              btfsc  STAT,Z
              incf  RC5DEMOZAEHLER,f  ;... den Zaehler RC5DEMOZAEHLER um 1
                                      ; erhoehen

              movlw  KONSTRC5DEMOMINUS ;Entspricht der empfangene RC5-Befehl der
              subwf  RC5BEFEHL,w      ; Konstanten KONSTRC5DEMOMINUS. (Die
                                      ; Konstante KONSTRC5DEMOMINUS entspricht
                                      ; der Taste "Lautstaerke vermindern") ...

              btfsc  STAT,Z
              decf  RC5DEMOZAEHLER,f  ;... den Zaehler RC5DEMOZAEHLER um 1
                                      ; vermindern

              movf  RC5DEMOZAEHLER,w  ;Nur die 3 niederwertigsten Bits vom
                                      ;Zaehler

              andlw  b'00000111'      ; RC5DEMOZAEHLER am Port GPIO ausgeben
              movwf  GPIO

              bcf   RC5STATUS,RC5NEUETASTE ;Anforderungsflag loeschen
              return
    
```

**Anzahl der Zyklen:**  
14 Zyklen (ohne Aufruf)

## 4. Demonstrationsbeispiel

Das folgende Beispiel dient nur zur Demonstration der RC5-Auswertesoftware. Sie zeigt eine mögliche Einbindung der im Abschnitt 3 besprochenen Unterprogramme.

### 4.1. Hardware

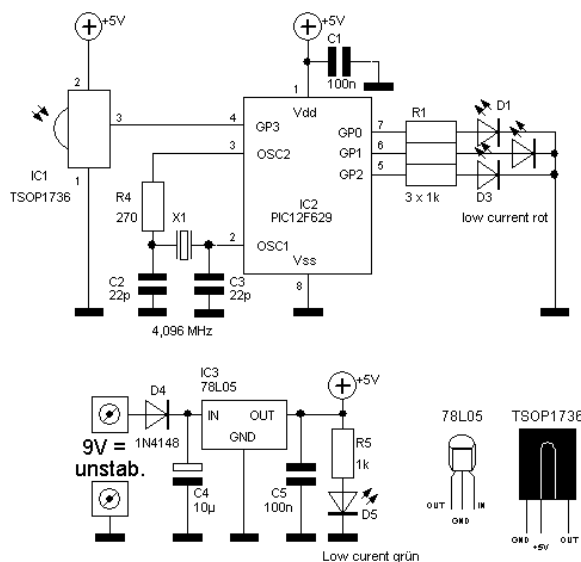


Bild 4.1: Schaltung zur Demonstration der RC5-Dekodierung

Bei diesem Demonstrationsbeispiel übernimmt ein Mikrocontroller vom Typ PIC12F629 die Dekodierung des vom IR-Empfänger (IC1, TSOP1736) demodulierten RC5-Datenstroms.

Die Leuchtdioden D1 bis D3 mit ihren Vorwiderständen R1 bis R3 zeigen den Wert eines 3-Bit-Zählers an, also eines Zählers welcher von 0 bis 7 zählt. Bei einem Druck auf die Taste „Lautstärke +“ auf der Fernbedienung, wird dieser Zähler um 1 erhöht, wobei nach einem Zählerstand von 7 wieder bei 0 begonnen wird. Bei einem Druck auf die Taste „Lautstärke -“ wird der Zähler um 1 vermindert. (Hier erfolgt nach einem Zählerstand von 0 der Wert 7).

Für die Takterzeugung wird eine Standard-Schaltung bestehend aus einem 4,096-MHz-Quarz (X1), zwei Kondensatoren (C2 und C3) und einen Widerstand R4.

Als „Reset-Quelle“ wird der interne Reset verwendet, sodass Pin 4 (MCLR/GP3) als zusätzlicher Portpin genutzt werden kann. Dieser kann jedoch nur als Eingangspin verwendet werden. Hier wird er als RC5-Eingang verwendet.

Der Kondensator C1 dient zur Entkoppelung der Betriebsspannung für den Mikrocontroller IC2. Für diesen Koppelkondensator sollte ein Keramiktyp verwendet werden. Dieser muss möglichst nahe beim Mikrocontroller angebracht werden.

Die Stromversorgung ist ebenfalls sehr einfach. Ein Festspannungsregler (IC3) vom Typ 78L05 übernimmt mit den Kondensatoren C4 und C5 die Spannungsregelung. Als Spannungsquelle dient beispielsweise ein unstabiliertes 9-V-Steckernetzteil. Die Leuchtdiode D5 mit dem Vorwiderstand R5 dient zur Spannungskontrolle. Für die Leuchtdiode D5 sollte eine Low-Current-Version verwendet werden. Die Diode D4 dient hier als Verpolungsschutz.

## 4.2. Software

```

;*****
;** Demonstrationsbeispiel zur RC5-Dekodierung **
;** **
;** Entwickler: Buchgeher Stefan **
;** Entwicklungsbeginn der Software: 3. Mai 2004 **
;** Funktionsfaehig seit: 4. Mai 2004 **
;** Letzte Bearbeitung: 25. Juni 2004 **
;*****

List p=PIC12F629

;***** Register (in Registerseite 0) *****
TMR0          equ    1          ;Timer0-Register
STAT          equ    3          ;Statusregister
GPIO          equ    5          ;Port-Register
INTCON        equ    0B        ;Interrupt-Register
CMCON         equ    1F        ;Komparator-Register

;***** Register (in Registerseite 1) *****
OPTREG        equ    1          ;OPTION-Register
TRISIO        equ    5          ;Port-Richtungsregister

;***** Eigene Register (in Registerbank 0) *****
ISR_STAT_TEMP equ    20        ;Zwischenspeicher des Statusregister der ISR
ISR_w_TEMP    equ    21        ;Zwischenspeicher des Arbeitsregister der ISR
FLAGISRHP     equ    22        ;beinhaltet Botschaftsflags ISR -> HP

```

## RC5 (Dekodierung mit PIC-Mikrocontroller)

```

RC5STATUS          equ    23          ;RC5-Statusregister
RC5ZAEHLER1       equ    24          ;Zaehlregister (siehe Unterprogramm RC5ROUTINE)
RC5ZAEHLER2       equ    25          ;Zaehlregister (siehe Unterprogramm RC5ROUTINE)
RC5ZAEHLERGUELTTEL equ    26          ;Dieses Zaehlregister gibt an, wie viele gleiche
; RC5-Telegramme noch empfangen werden muessen,
; damit der empfangene RC5-Code als gueltig
; anerkannte wird
RC5TELEGRAMMOA    equ    27          ;Hilfsregister (dieses dient zur Ermittlung des
; Telegrammstarts.)
RC5TELEGRAMMOB    equ    28          ;wie RC5TELEGRAMMOA
RC5TELEGRAMM1     equ    29          ;Beinhaltet das empfangene RC5-Telegramm
RC5TELEGRAMM2     equ    2A          ;wie RC5TELEGRAMM1
RC5TELEGRAMM1ALT  equ    2B          ;Dient zur Ermittlung eines gueltigen RC5-Tele-
; gramms (der ausgesendete RC5-Code muss von der
; Dekodiersoftware zumindest zweimal gleich
; empfangen werden).
RC5TELEGRAMM2ALT  equ    2C          ;wie RC5TELEGRAMM1ALT
RC5ADRESSE        equ    2D          ;Beinhaltet die dekodierte Systemadresse
; (also die Systemadresse der verwendeten
; Fernbedienung)
RC5BEFEHL         equ    2E          ;Beinhaltet den dekodierten Befehl (also den
; Wert der auf der Fernbedienung gedruckten
; Taste.)
RC5DEMOZAEHLER   equ    2F          ;Zaehlregister zur Demonstration

;***** Bits in Registern der Registerbank 0 *****
;Register STAT
C                  equ    0          ;Carrybit im Statuswort-Register
Z                  equ    2          ;Zerobit im Statuswort-Register
RP0                equ    5          ;Seitenauswahlbit im Statuswort-Register

;Register INTCON
TOIF               equ    2          ;TMR0-Interruptflag im INTCON-Register

;***** Bits in den eigenen Registern der Registerbank 0 *****
;Register FLAGISRHP
FLAG250USEK       equ    0

;Register RC5STATUS
RC5START          equ    0          ;Hilfsflag zur Erkennung eines Telegrammstarts
RC5STARTFLANKE   equ    1          ;Hilfsflag zur Erkennung eines Telegrammstarts
RC5CHECKTEL      equ    2          ;gesetzt, wenn ein vollstaendiges RC5-Telegramm
; dekodiert wurde
RC5TELGUELTIG    equ    3          ;gesetzt, wenn mehrmals das selbe Telegramm
; empfangen wurde
RC5TOGGLEALT     equ    4          ;Toggle-Bit des "alten" Telegramms
RC5TOGGLENEU     equ    5          ;Toggle-Bit
RC5NEUETASTE     equ    6          ;gesetzt, wenn eine neue Taste gedruickt wurde
; und mehrmals das selbe Telegramm empfangen
; wurde

;***** Portbelegung *****
;Port A
RC5INPORT        equ    GPIO
RC5INTRIS        equ    TRISIO
RC5IN            equ    3

;***** Konstanten *****
KONSTRC5ZAEHLERSTART equ    .6      ;Diese Konstante dient zum Laden von
; RC5ZAEHLER1, unmittelbar nach einem
; Telegrammbeginn
KONSTRC5ZAEHLER    equ    .7      ;Diese Konstante dient zum Nachladen von
; RC5ZAEHLER1
KONSTRC5GUELTTEL   equ    .2      ;Anzahl der gueltig empfangenen RC5-Telegramme
KONSTRC5DEMOPLUS   equ    .16     ;entspricht der Taste "Lautstaerke erhoehen"
KONSTRC5DEMOMINUS equ    .17     ;entspricht der Taste "Lautstaerke vermindern"

;***** Ziele der Registeroperationen *****
w                  equ    0
f                  equ    1

```

## RC5 (Dekodierung mit PIC-Mikrocontroller)

```

;***** Konfigurations *****
__config b'00000110000010'
;
; +----- Bit 13-12 (BG1:BG0): Bandgap-Calibration
; ||||| 0 0 : Lowest bandgap voltage
; ||||| -> 1 : Highest bandgap voltage
;
; +++----- Bit 11-9: Reserve
;
; +----- Bit 8 (CPD): Data Code Protection (Data Memory)
; ||||| 0 : CPD on (= enabled)
; ||||| -> 1 : CPD off (= disabled)
;
; +----- Bit 7 (CP): Code Protection (Programm Memory)
; ||||| 0 : CP on (= enabled)
; ||||| -> 1 : CP off (= disabled)
;
; +----- Bit 6 (BODEN): Brown-Out Detection
; ||||| -> 0 : BODEN on (= disabled)
; ||||| 1 : BODEN off (= enabled)
;
; +----- Bit 5 (MCLR)
; ||||| -> 0 : GP3 = I/O
; ||||| 1 : GP3 = MCLR
;
; +----- Bit 4 (PWRT): Power Up Timer
; ||||| -> 0 : PWRT on (= enabled)
; ||||| 1 : PWRT off (= disabled)
;
; +----- Bit 3 (WDT): Watchdog Timer
; |||| -> 0 : WDT off (= disabled)
; |||| 1 : WDT on (= enabled)
;
; +----- Bit 2-0 (FOSC2:FOSC0): Oszillator Selectio
; 000 : LP Oszillator (GP4=CLKOUT, GP5=CLKIN)
; 001 : XT Oszillator (GP4=CLKOUT, GP5=CLKIN)
; -> 010 : HS Oszillator (GP4=CLKOUT, GP5=CLKIN)
; 011 : EC Oszillator (GP4=I/O, GP5=CLKIN)
; 100 : INTOSC Oszillator (GP4=I/O, GP5=I/O)
; 101 : INTOSC Oszillator (GP4=CLKOUT, GP5=I/O)
; 110 : RC Oszillator (GP4=I/O, GP5=RC)
; 111 : RC Oszillator (GP4=CLKOUT, GP5=RC)

;***** Einsprungadressen *****
ORG 0x000
goto Beginn
ORG 0x004
goto ISR

;***** Tabellen *****
; keine Tabellen

;***** ISR - Timer0 *****

;*****
; ** Interrupt Service Routine: **
; ** **
; ** Aufruf: **
; ** alle 250 us **
; ** **
; ** Aufgaben: **
; ** + w-Register (=Arbeitsregister) und Status-Register zwischenspeichern (PUSH) **
; ** + Zeitbasis fuer 250us erzeugen (Botschaftsflag FLAG250USEK im Register **
; ** FLAGISRHP setzen **
; ** + Das Timer-Interrupt-Flag T0IF wieder loeschen **
; ** + w-Register (=Arbeitsregister) und Statusregister wiederherstellen (POP). **
; ** **
; ** Anzahl der Zyklen: **
; ** 11 Zyklen (ohne Aufruf) **
;*****
ISR
PUSH movwf ISR_w_TEMP ;w-Register retten
swapf STAT,w ;Statusregister
movwf ISR_STAT_TEMP ; retten

;Beginn der eigentlichen ISR-Routine
bsf FLAGISRHP,FLAG250USEK;Botschaftsflag setzen
;Ende der eigentlichen ISR-Routine

ISRfertig bcf INTCON,T0IF ;T0-Interruptflag loeschen

POP swapf ISR_STAT_TEMP,w ;Status-Register
movwf STAT ; und
swapf ISR_w_TEMP,f ; w-Register
swapf ISR_w_TEMP,w ; wieder herstellen

```

## RC5 (Dekodierung mit PIC-Mikrocontroller)

```

retfie

;***** Unterprogramme *****
;*****
;** Initialisierung des Prozessor: **
;** + Comparatoreingange auf digitale I/O-Pins umschalten **
;** + TMR0-ISR soll alle 250us aufgerufen werden, daher keinen Vorteiler verwenden **
;** + Port: als Ausgaenge definieren und loeschen **
;** Der IR-Eingang wird separat mit dem Befehl bsf RC5INTRIS,RC5IN als Eingang **
;** definiert **
;** + Register fuer die RC5-Dekodierung initialisieren **
;*****
INIT      clrf      TMR0          ;Timer0 auf 0 voreinstellen

          clrf      GPIO
          movlw    0x07          ;Alle Comparatoreingange
          movwf   CMCON        ; auf digital I/O umschalten

          bsf      STAT,RP0     ;Registerseite 1

          movlw   b'00001000'   ;kein Vorteiler von TMR0
          movwf   OPTREG

          clrf    TRISIO        ;Port als Ausgang definieren
          bsf    RC5INTRIS,RC5IN ;RC5IN als Eingang definieren

          bcf    STAT,RP0     ;Registerseite 0

          clrf    GPIO        ;Port loeschen

          clrf    RC5STATUS    ;RC5-Statusregister initialisieren (loeschen)
          clrf    RC5TELEGRAMM0A ;RC5TELEGRAMM0A initialisieren (loeschen)
          clrf    RC5TELEGRAMM0B ;RC5TELEGRAMM0B initialisieren (loeschen)
          clrf    RC5TELEGRAMM1ALT ;RC5TELEGRAMM1ALT initialisieren (loeschen)
          clrf    RC5TELEGRAMM2ALT ;RC5TELEGRAMM2ALT initialisieren (loeschen)
          clrf    RC5DEMOZAEHLER ;RC5DEMOZAEHLER initialisieren (loeschen)

          movlw   KONSTRC5GUELTTEL ;RC5ZAEHLERGUELTTEL initialisieren (mit der
          movwf   RC5ZAEHLERGUELTTEL ; Konstanten KONSTRC5GUELTTEL)

          return

;***** RC5 Routinen *****
;*****
;** RC5ROUTINE: **
;** ** **
;** Das Unterprogramm RC5ROUTINE ist für die RC5-Dekodierung die wichtigste Komponente. **
;** ** **
;** Diese Routine wird ca. alle 250 us aufgerufen **
;** ** **
;** Aufgabe: **
;** Die Aufgabe des Unterprogramms RC5ROUTINE besteht darin, den am RC5-Eingangspin **
;** (RC5IN) anstehenden Datenstrom zyklisch (alle 250us) abzutasten. Aus diesen **
;** Abtastungen den Telegrammbeginn und die laut RC5-Protokoll definierten Low- und High- **
;** Bits ermitteln. Die so ermittelten Low- und High-Bits in den Registern RC5TELEGRAMM1 **
;** und RC5TELEGRAMM2 zwischenspeichern, so dass daraus der uebertragene Befehl und die **
;** uebertragene Systemadresse ermittelt werden kann. (Anm. Das Ermitteln der System- **
;** adresse und des Befehls ist nicht die Aufgabe diese Unterprogramms. Diese Taetigkeit **
;** wird u.a. vom Unterprogramm RC5CHECKTELEGRAMM uebernommen.) **
;** ** **
;** Vorgehensweise: **
;** Zuerst muss auf einen Telegrammbeginn gewartet werden. Für diesen Zweck dienen die **
;** beiden Flags RC5START und RC5STARTFLANKE im Register RC5STATUS. Diese Flags werden **
;** bei der Initialisierung des Mikrocontrollers geloescht, sind also zu Beginn ge- **
;** loescht. **
;** Ein Telegrammbeginn wird folgendermassen erkannt: solange das Flag RC5START nicht **
;** gesetzt ist wird der abgetastete RC5-Eingang an die niederwertigste Stelle des **
;** Registers RC5TELEGRAMM0A geschoben. Alle anderen Bits dieses Registers werden um je **
;** eine Position weiter geschoben. Das hoechstwertige Bit vom Register RC5TELEGRAMM0A **
;** wird an die niederwertigste Stelle des Registers RC5TELEGRAMM0B geschoben. Alle **
;** anderen Bits dieses Registers werden auch hier um je eine Position weiter geschoben. **
;** Anschliessend wird geprueft, ob beide Register den Wert FFh beinhalten. Beinhalten **
;** beide Register (RC5TELEGRAMM0A und RC5TELEGRAMM0B) den Wert FFh, so wird zurzeit **
;** kein RC5-Code uebertragen und es kann auf einen Telegrammbeginn gewartet werden. Das **

```

## RC5 (Dekodierung mit PIC-Mikrocontroller)

```

**      Flag RC5START wird gesetzt, das Flag RC5STARTFLANKE wird geloescht, und die Register **
**      RC5TELEGRAMM0A und RC5TELEGRAMM0B werden ebenfalls geloescht. Und Nun wird solange **
**      gewartet bis am RC5-Eingang ein Low anliegt. Ist dies der Fall so erfolgt ein Tele- **
**      grammbeginn. Das Flag RC5STARTFLANKE wird zu diesem Zweck gesetzt und die beiden **
**      Zaehregister RC5ZAEHLER1 und RC5ZAEHLER2. **
**      Da das RC5-Telegramm aus 14 Bits besteht und das erste Startbit soeben erkannt wurde **
**      muss die Software nur mehr 13 Bits dekodieren. Daher wird das Register RC5ZAEHLER2 **
**      mit dem Wert 13 geladen. Dieses Zaehregister gibt somit an wie viele Bits des RC5- **
**      Telegramms noch zu dekodieren sind. **
**
**      Der Telegrammbeginn wurde nun erkannt. Nun muessen die einzelnen Bits dekodiert **
**      werden. Ein Bit nach dem RC5-Protokoll besteht aus einer 889us langen Low-Phase und **
**      einer 889us langen High-Phase (bei einer "0", bei einer "1" umgekehrt). Bei jedem **
**      weiteren Aufruf dieses Unterprogramms wird nun das Zaehregister RC5ZAEHLER1 um 1 **
**      vermindert. Beinhaltet es danach den Wert 0, so wird der abgetastete RC5-Eingang an **
**      die niederwertigste Stelle des Registers RC5TELEGRAMM1 geschoben. Alle anderen Bits **
**      dieses Registers werden um je eine Position weiter geschoben. Das hoechstwertige Bit **
**      vom Register RC5TELEGRAMM1 wird an die niederwertigste Stelle des Registers **
**      RC5TELEGRAMM2 geschoben. Alle anderen Bits dieses Registers werden auch hier um je **
**      eine Position weiter geschoben. (Anders ausgedrueckt: der Zustand an RC5-Eingang **
**      wird dem RC5-Telegramm hinzugefuegt). Anschliessend wird der Zaehler RC5ZAEHLER1 mit **
**      der Konstanten KONSTRC5ZAEHLER neu geladen und der Zaehler RC5ZAEHLER2 um 1 ver- **
**      mindert. Besitzt der Zaehler RC5ZAEHLER2 nun den Wert 0, so wurde ein komplettes **
**      RC5-Telegramm eingelesen und in den Registern RC5TELEGRAMM1 und RC5TELEGRAMM2 ge- **
**      sichert. Zur Kennzeichnung, dass ein komplettes RC5-Telegramm empfangen wurde, muss **
**      das Botschaftsflag RC5TELFERIG im Register RC5STATUS gesetzt werden, waehrend das **
**      Flag RC5START (ebenfalls im Register RC5STATUS) zurueckgesetzt werden muss, da ja **
**      nun auf einen neuen Telegrammbeginn gewartet werden muss. **
**
**      Anzahl der Zyklen: **
**      Min: 11 Zyklen (ohne Aufruf) **
**      Max: 25 Zyklen (ohne Aufruf) **
*****
RC5ROUTINE      btfsf   RC5STATUS,RC5START      ;Flag RC5START gesetzt?
                goto    RC5WEITER1              ; ja -> Weiter beim Label RC5WEITER1

                bcf     STAT,C                  ; nein: dem Telegramm0 mit Hilfe des Carry
                btfsf   RC5INPORT,RC5IN        ; den aktuellen Zustand vom RC5-Eingang
                bsf     STAT,C                  ; hinzufuegen
                rlf     RC5TELEGRAMM0A,f
                rlf     RC5TELEGRAMM0B,f

                movf    RC5TELEGRAMM0B,w
                sublw   b'11111111'

                btfsf   STAT,Z                  ; RC5TELEGRAMM0B = FF ?
                goto    RC5ROUTINEFERTIG        ; nein: UP verlassen

                movf    RC5TELEGRAMM0A,w
                sublw   b'11111111'

                btfsf   STAT,Z                  ;
                goto    RC5ROUTINEFERTIG        ; nein: UP verlassen

                bsf     RC5STATUS,RC5START      ; ja: Flag RC5START setzen
                bcf     RC5STATUS,RC5STARTFLANKE ; Flag RC5STARTFLANKE loeschen
                clrf    RC5TELEGRAMM0A        ; Register RC5TELEGRAMM0A und
                clrf    RC5TELEGRAMM0B        ; Register RC5TELEGRAMM0B loeschen
                goto    RC5ROUTINEFERTIG        ; UP verlassen

RC5WEITER1      btfsf   RC5STATUS,RC5STARTFLANKE ;Flag RC5STARTFLANKE gesetzt?
                goto    RC5WEITER2              ; ja -> Weiter beim Label RC5WEITER2

                btfsf   RC5INPORT,RC5IN        ; nein: RC5-Eingang = 0?
                goto    RC5ROUTINEFERTIG        ; nein: UP verlassen

                bsf     RC5STATUS,RC5STARTFLANKE ; ja: Flag RC5STARTFLANKE setzen
                movlw   KONSTRC5ZAEHLERSTART ; RC5ZAEHLER1 = KONSTRC5ZAEHLERSTART
                movwf   RC5ZAEHLER1
                movlw   .13                    ; RC5ZAEHLER2 = 13
                movwf   RC5ZAEHLER2
                goto    RC5ROUTINEFERTIG        ; UP verlassen

RC5WEITER2      decfsz  RC5ZAEHLER1,f          ;RC5ZAEHLER1 um 1 vermindern und auf 0 pruefen
                goto    RC5ROUTINEFERTIG        ;wenn RC5ZAEHLER1 ungleich 0 -> UP verlassen

                bcf     STAT,C                  ;RC5ZAEHLER1 = 0: dem RC5-Telegramm mit Hilfe
                btfsf   RC5INPORT,RC5IN        ; des Carry den aktuellen Zustand von
                bsf     STAT,C                  ; RC5-Eingang hinzufuegen

```

## RC5 (Dekodierung mit PIC-Mikrocontroller)

```

rlf    RC5TELEGRAMM1,f
rlf    RC5TELEGRAMM2,f

movlw  KONSTRC5ZAEHLER      ; RC5ZAEHLER1 = KONSTRC5ZAEHLER
movwf  RC5ZAEHLER1

decfsz RC5ZAEHLER2,f      ; RC5ZAEHLER2 um 1 vermindern und auf 0 pruefen
goto   RC5ROUTINEFERTIG   ; wenn RC5ZAEHLER2 ungleich 0 -> UP verlassen

bsf    RC5STATUS,RC5CHECKTEL ; RC5ZAEHLER2 = 0: Flag RC5CHECKTEL setzen
bcf    RC5STATUS,RC5START   ; und Flag RC5START loeschen

```

```

RC5ROUTINEFERTIG
return

```

```

;*****
;** RC5CHECKTELEGRAMM: **
;** **
;** Aufgaben und Vorgehensweise: **
;** + Neues Telegramm (RC5TELEGRAMMxALT) mit dem neuen Telegramm (RC5TELEGRAMMx) ver- **
;** gleichen **
;** + Bei Gleichheit: **
;** + Das Zaehlregister fuer die gleichen Telegramme (RC5ZAEHLERGUELTTEL) um 1 ver- **
;** mindern. **
;** + Ist dieses Zaehlregister nun 0 (d.h. Es wurden mehrere gleiche RC5-Telegramme **
;** empfangen) die RC5-Adresse aus den folgenden Bits zusammensetzen: **
;** Bit 6 von RC5TELEGRAMM1 -> Bit 0 der RC5-Adresse **
;** Bit 7 von RC5TELEGRAMM1 -> Bit 1 der RC5-Adresse **
;** Bit 0 von RC5TELEGRAMM2 -> Bit 2 der RC5-Adresse **
;** Bit 1 von RC5TELEGRAMM2 -> Bit 3 der RC5-Adresse **
;** Bit 2 von RC5TELEGRAMM2 -> Bit 4 der RC5-Adresse **
;** und den RC5-Befehl aus den 6 niederwertigsten Bits von RC5TELEGRAMM1 zusammen- **
;** setzen **
;** + Das Flag RC5TOGGLENEU nach RC5TOGGLEALT kopieren und Bit 3 von RC5TELEGRAMM2 **
;** nach RC5TOGGLENEU kopieren. **
;** + Zaehlregister fuer die Ermittlung der gueltigen Telegramme (RC5ZAEHLERGUELTTEL) **
;** neu mit der Konstanten KONSTRC5GUELTTEL laden **
;** + Botschaftsflag "Gueltiges RC5-Telegramm" setzen **
;** + Pruefen, ob das neue Togglebit (RC5TOGGLENEU) mit dem alten Togglebit **
;** (RC5TOGGLEALT) uebereinstimmt, dazu die benachbarten Flags RC5TOGGLEALT (Bit 4) **
;** und RC5TOGGLENEU (Bit 5) Exklusiv-Oder-verknuepfen und nach RC5TELEGRAMM1 **
;** kopieren (Anm.: Das Register RC5TELEGRAMM1 wird hier als temporaeres Hilfs- **
;** register verwendet, da der Inhalt fuer die RC5-Dekodierung hier nicht mehr **
;** benoetigt wird) **
;** + Bei Ungleichheit: **
;** + Die Register RC5TELEGRAMMx in die Register RC5TELEGRAMM1ALT kopieren und den **
;** Zaehler RC5ZAEHLERGUELTTEL mit der Konstante KONSTRC5GUELTTEL neu laden. **
;** + Anforderungsflag RC5CHECKTEL (im Register RC5STATUS) loeschen **
;** **
;** Anzahl der Zyklen: **
;** Min: 14 Zyklen (ohne Aufruf) **
;** Max: 43 Zyklen (ohne Aufruf) **
;*****

```

```

RC5CHECKTELEGRAMM
movf   RC5TELEGRAMM1ALT,w   ;RC5TELEGRAMM1ALT mit RC5TELEGRAMM1 vergleichen
subwf  RC5TELEGRAMM1,w

btfss  STAT,Z
goto   RC5UNGLEICH         ;bei Ungleichheit springe zum Label RC5UNGLEICH

movf   RC5TELEGRAMM2ALT,w   ;bei Gleichheit RC5TELEGRAMM2ALT mit
subwf  RC5TELEGRAMM2,w     ; RC5TELEGRAMM2 vergleichen

btfss  STAT,Z
goto   RC5UNGLEICH         ;bei Ungleichheit springe zum Label RC5UNGLEICH

decfsz RC5ZAEHLERGUELTTEL,f ;bei Gleichheit von RC5TELEGRAMM1ALT mit
; RC5TELEGRAMM1 und RC5TELEGRAMM2ALT mit
; RC5TELEGRAMM2 das Zaehlregister fuer die
; gleichen Telegramme (RC5ZAEHLERGUELTTEL) um
; 1 vermindern.
goto   RC5NEUESTELFERTIG   ; Zaehlregister RC5ZAEHLERGUELTTEL = 0
; nein: springe zum Label RC5NEUESTELFERTIG

; ja: mehrere (durch die Konstante
; KONSTRC5GUELTTEL definierte) gleiche und
; gueltige RC5-Telegramme wurden empfangen
clrf   RC5ADRESSE         ; nun die RC5-Adresse zusammensetzen ...

```

## RC5 (Dekodierung mit PIC-Mikrocontroller)

```

btfsc RC5TELEGRAMM1,6
bsf RC5ADRESSE,0
btfsc RC5TELEGRAMM1,7
bsf RC5ADRESSE,1
btfsc RC5TELEGRAMM2,0
bsf RC5ADRESSE,2
btfsc RC5TELEGRAMM2,1
bsf RC5ADRESSE,3
btfsc RC5TELEGRAMM2,2
bsf RC5ADRESSE,4

movf RC5TELEGRAMM1,w ; ... und den RC5-Befehl aus den 6 nieder-
andlw b'00111111' ; wertigsten Bits von RC5TELEGRAMM1
movwf RC5BEFEHL ; zusammensetzen

bcf RC5STATUS,RC5TOGGLEALT ; RC5TOGGLENEU -> RC5TOGGLEALT
btfsc RC5STATUS,RC5TOGGLENEU
bsf RC5STATUS,RC5TOGGLEALT

bcf RC5STATUS,RC5TOGGLENEU ; RC5TELEGRAMM2,3 -> RC5TOGGLENEU
btfsc RC5TELEGRAMM2,3
bsf RC5STATUS,RC5TOGGLENEU

movlw KONSTRC5GUELTTEL ; Zaehregister fuer die Ermittlung
movwf RC5ZAEHLERGUELTTEL ; gueltiger Telegramme neu aus der
; Konstante laden

bsf RC5STATUS,RC5TELGUELTIG ; Botschaftsflag "Gueltiges RC5-Telegramm"
; setzen

;Pruefen, ob das neue Togglebit (RC5TOGGLENEU) mit dem alten Togglebit
; (RC5TOGGLEALT) uebereinstimmt
rlf RC5STATUS,w ;Die benachbarten Flags RC5TOGGLEALT (Bit 4)
xorwf RC5STATUS,w ; und RC5TOGGLENEU (Bit 5) Exklusiv-Oder-
; verknuepfen und nach RC5TELEGRAMM1
movwf RC5TELEGRAMM1 ; kopieren (Anm.: Das Register RC5TELEGRAMM1
; wird hier als temporaeres Hilfsregister
; verwendet, da der Inhalt fuer die RC5-
; Dekodierung hier nicht mehr benoetigt wird)

btfsc RC5TELEGRAMM1,RC5TOGGLENEU ;Ist das Ergebnis der Exklusiv-Oder 1, so
bsf RC5STATUS,RC5NEUETASTE ; sind die beiden Toggleflags ungleich. Es
; wurde also eine neue Taste gedruickt.

goto RC5NEUESTELFERTIG

;Bei Ungleichheit von RC5TELEGRAMM1ALT mit RC5TELEGRAMM1 oder von
;RC5TELEGRAMM2ALT mit RC5TELEGRAMM2
RC5UNGLEICH movf RC5TELEGRAMM1,w ;RC5TELEGRAMM1ALT = RC5TELEGRAMM1
movwf RC5TELEGRAMM1ALT
movf RC5TELEGRAMM2,w ;RC5TELEGRAMM2ALT = RC5TELEGRAMM2
movwf RC5TELEGRAMM2ALT

movlw KONSTRC5GUELTTEL ;Zaehler RC5ZAEHLERGUELTTEL mit der
movwf RC5ZAEHLERGUELTTEL ;Konstante KONSTRC5GUELTTEL neu laden
goto RC5NEUESTELFERTIG

RC5NEUESTELFERTIG
bcf RC5STATUS,RC5CHECKTEL ;Anforderungsflag RC5CHECKTEL loeschen
return

;***** Demospezifisches Unterprogramm *****
;*****
;** RC5AKTION: **
;** **
;** Allgemeines: **
;** Hier erfolgen nun die fuer die Aufgabenstellung gewuenschten Aktionen auf einen **
;** dekodierten RC5-Code. **
;** **
;** Aufgabe und Vorgehensweise: **
;** Entspricht der empfangene RC5-Befehl der Konstanten KONSTRC5DEMOPLUS, (die Konstante **
;** KONSTRC5DEMOPLUS entspricht der Taste "Lautstaerke erhoehen"), den Zaehler **
;** RC5DEMOZAEHLER um 1 erhoehen. Entspricht der empfangene RC5-Befehl der Konstanten **
;** KONSTRC5DEMOMINUS, (die Konstante KONSTRC5DEMOMINUS entspricht der Taste **
;** "Lautstaerke vermindern"), den Zaehler RC5DEMOZAEHLER um 1 vermindern. Anschliessend **
;** nur die 3 niederwertigsten Bits vom Zaehler RC5DEMOZAEHLER am Port GPIO ausgeben. **
;** Zum Schluss das Anforderungsflag RC5NEUETASTE wieder loeschen. **

```

## RC5 (Dekodierung mit PIC-Mikrocontroller)

```

; **
; ** Anzahl der Zyklen:
; ** 14 Zyklen (ohne Aufruf)
; **
; *****
RC5AKTION    movlw   KONSTRC5DEMOPLUS    ;Entspricht der empfangene RC5-Befehl der
              subwf   RC5BEFEHL,w       ; Konstanten KONSTRC5DEMOPLUS, (die Konstante
              ; KONSTRC5DEMOPLUS entspricht der Taste
              ; "Lautstaerke erhoehen") ...

              btfsc   STAT,Z
              incf   RC5DEMOZAEHLER,f   ;... den Zaehler RC5DEMOZAEHLER um 1 erhoehen

              movlw   KONSTRC5DEMOMINUS ;Entspricht der empfangene RC5-Befehl der
              subwf   RC5BEFEHL,w       ; Konstanten KONSTRC5DEMOMINUS. (Die Konstante
              ; KONSTRC5DEMOMINUS entspricht der Taste
              ; "Lautstaerke vermindern") ...

              btfsc   STAT,Z
              decf   RC5DEMOZAEHLER,f   ;... den Zaehler RC5DEMOZAEHLER um 1 vermindern

              movf   RC5DEMOZAEHLER,w   ;Nur die 3 niederwertigsten Bits vom Zaehler
              andlw  b'00000111'       ; RC5DEMOZAEHLER am Port GPIO ausgeben
              movwf  GPIO

              bcf    RC5STATUS,RC5NEUETASTE ;Anforderungsflag loeschen
              return

; ***** Hauptprogramm *****

; *****
; ** Aufgaben des Hauptprogramms:
; ** + Controller initialisieren (Unterprogramm INIT)
; ** + Interrupt freigeben
; ** + Taetigkeiten/Unterprogramme, die alle 250us durchgefuehrt werden muessen:
; ** + Unterprogramm RC5ROUTINE aufrufen
; ** + Wurde ein RC5-Telegramm empfangen (Flag RC5CHECKTEL ist gesetzt) das Unter-
; ** programm RC5NEUESTELEGRAMM abarbeiten. Dieses Unterprogramm vergleicht das empf-
; ** angene RC5-Telegramm mit dem vorhergehene. Erst wenn zwei gleiche RC5-Telegramme
; ** empfangen wurde, wird dies als gueltig anerkannt und das Flag RC5NEUETASTE gesetzt
; ** + Wurden zumindest zwei NEUE gleiche und gueltige RC5-Telegramme empfangem (Flag
; ** RC5NEUETASTE ist gesetzt) das Unterprogramm RC5AKTION abarbeiten. In diesem
; ** Unterprogramm erfolgen nun die gewuenschten Reaktion auf die, auf der Fernbedienung
; ** gedruckten Tasten
; **
; *****
Beginn       call    INIT                ;Controller initialisieren

              movlw  b'10100000'        ;Timer0 freigeben durch Setzen von
              movwf  INTCON              ; GIE und T0IE im Register INTCON

HPSCHLEIFE  ; Hoechste Prioritaet
              btfsc  FLAGISRHP,FLAG250USEK;Wenn Anforderungsflag FLAG250USEK gesetzt
              goto   HPWEITER1          ; weiter bei HPWEITER1

              btfsc  RC5STATUS,RC5CHECKTEL ;Wenn Anforderungsflag RC5TELFERTIG gesetzt
              goto   HPWEITER2          ; weiter bei HPWEITER2

              ; Niedrigste Prioritaet
              btfsc  RC5STATUS,RC5NEUETASTE;Wenn Anforderungsflag RC5NEUETASTE gesetzt
              goto   HPWEITER3          ; weiter bei HPWEITER3

              goto   HPSCHLEIFE

HPWEITER1   ; Taetigkeiten, die alle 250 us durchgefuehrt werden muessen
              ; (Hoechste Prioritaet)
              call   RC5ROUTINE
              bcf    FLAGISRHP,FLAG250USEK;Anforderungsflag fuer die 250 us-Aktivitaeten
              ; zuruecksetzen
              goto   HPSCHLEIFE

HPWEITER2   ; Taetigkeiten die nach Setzten des Anforderungsflag RC5TELFERTIG
              ; ausgefuehrt werden muessen
              call   RC5CHECKTELEGRAMM
              goto   HPSCHLEIFE

HPWEITER3   ; Taetigkeiten die nach Setzten des Anforderungsflag RC5TELGUELTIG
              ; ausgefuehrt werden muessen (Niedrigste Prioritaet)
              call   RC5AKTION
              goto   HPSCHLEIFE

```

end

### 4.3. Anmerkungen zur Software

Die Software besteht im Wesentlichen aus einem kurzen Hauptprogramm, einer kurzen Interrupt-Service-Routine (kurz ISR), einem Unterprogramm zur Initialisierung des Mikrocontroller (siehe Abschnitt 3.3) und die im Abschnitt 3.4 besprochenen Unterprogramme zur RC5-Dekodierung.

Die ISR wird alle 250µs aufgerufen und besitzt „nur“ die Aufgabe eine genaue Zeitbasis für die RC5-Dekodierung zu erzeugen. Zu diesem Zweck setzt die ISR ein Flag namens FLAG250USEK im Register FLAGISRHP.

Die ISR wird, wie schon mehrmals erwähnt, alle 250µs aufgerufen. Diese 250µs ergeben sich folgendermaßen: TMR0 wird mit dem Wert 0 geladen – es dauert also 256 Taktzyklen bis das Register TMR0 wieder den Wert 0 besitzt, der Vorteiler besitzt den Wert 1 (vgl. Unterprogramm INIT, Abschnitt. 3.3). Der Taktzyklus ergibt sich aus dem verwendeten Quarz (X1). Dieser ist bei der PIC-Familie wie folgt definiert:

$$\frac{4}{f_{\text{Quarz}}}$$

Daraus ergibt sich folgender Zusammenhang:

$$ISRAUFRUF [\mu s] = \frac{4 \cdot 256 \cdot 1}{f_{\text{Quarz}} [MHz]} = \frac{4 \cdot 256}{4,096} = 250$$

Also eine ISR-Aufruf alle 250µs.

Das Hauptprogramm befindet sich nach der Initialisierung (Unterprogramm INIT) und der Interruptfreigabe in einer Endlosschleife. Diese Schleife besitzt die Aufgabe ständig die so genannten Botschaftsflags abzufragen. Ist eines dieser Botschaftsflags gesetzt, so muss vom Hauptprogramm eine bestimmte Aufgabe ausgeführt werden. Diese Aufgaben sind in Form von Unterprogrammen vorhanden. Eines dieser Botschaftsflags wird in der Timer-ISR gesetzt. Und zwar wird alle 250µs ein Flag gesetzt.

Hier Zusammenfassend die Tätigkeiten in der Endlosschleife, welche durch die Botschaftsflags ausgelöst werden:

- Tätigkeiten und/oder Unterprogramme, die alle 250µs durchgeführt werden müssen:
  - Unterprogramm RC5ROUTINE aufrufen (siehe Abschnitt 3.4.1)
- Tätigkeit(en) die durch das Flag RC5CHECKTEL ausgelöst werden:
  - Unterprogramm RC5CHECKTELEGRAMM abarbeiten (siehe Abschnitt 3.4.2)
- Tätigkeit(en) die durch das Flag RC5NEUETASTE ausgelöst werden:
  - Unterprogramm RC5AKTION abarbeiten (siehe Abschnitt 3.4.3)

Das Unterprogramm INIT dient zur Initialisierung des Controllers. Hier werden unter anderem die Ports konfiguriert (Port dient als Eingang oder als Ausgang), der oder die Timer eingestellt usw. Dieses Unterprogramm ist vom Controllertyp abhängig und je nach Anwendung mehr oder weniger umfangreich. Siehe auch Abschnitt 3.3 (Initialisierung)

Die Unterprogramme RC5ROUTINE und RC5CHECKTEL sollten nur in Ausnahmefällen geändert werden, während das Unterprogramm RC5AKTION an die jeweilige Aufgabe der Software angepasst werden muss.

## **5. Quellen**

- Buch „Infrarot-Datenübertragung“ (ISBN: 3-89576-115-x, Autor: Frank Wohlrabe)
- ELV-Journal 4/99