

Datenversand über handelsübliche RC - Fernbedienungen

Sommer 2009

Erstellt von
Niels Keller

Inhalt

1. EINFÜHRUNG	2
1.1. WORUM GEHT ES?.....	2
1.2. WARUM KEINE 433 MHZ-MODULE?.....	2
1.3. WAS BRAUCHT MAN?	2
1.4. IST SO ETWAS ZULÄSSIG?	2
2. DAS ZUSAMMENSTELLEN DER HARDWARE.....	3
SENDER UND EMPFÄNGER	3
ANSCHLUSS DER SENDER UND EMPFÄNGER AN DIE UP	3
GRUNDBESCHALTUNG EINES ATMEGAS32.....	4
3. DER QUELLCODE.....	4
DAS WICHTIGSTE IN KÜRZE.....	4
WAS MACHT DIESER CODE?	9
4. PRAXISBEISPIELE.....	10

1. Einführung

1.1. Worum geht es?

Ziel ist es mit möglichst einfachen Mitteln eine serielle Kommunikation zwischen zwei nicht miteinander verbundenen Systemen zu erstellen. Die Kommunikation soll nicht schnell sein, sondern zuverlässig. Nicht schnell bedeutet hier 1 Byte pro Sekunde. Die Abstände zwischen Sender und Empfänger sollen im Bereich zwischen zwei und zehn Metern liegen.

1.2. Warum keine 433 MHz-Module?

Hatte es mir in den Kopf gesetzt das mit einfachen Mitteln zu lösen. Ja, ist keine Begründung. Ich gebe es zu: Ich habe es nicht gepackt, die RFM-12 Module zum Laufen zu bringen. Wollte aber dennoch nicht darauf verzichten, Daten aus einem isolierten (Beton) Raum zu erhalten. Daher mit dem nächstgelegenen Gegenstand der Funken kann ausprobiert – und das war eben eine Fernbedienung aus einem Modellauto.

1.3. Was braucht man?

- Ein Spielzeugauto mit Fernbedienung, z. B. aus dem Müllermarkt oder einem Online-Auktionshaus. Die Fernbedienung sollte entweder im 27 MHz oder im 40 MHz Band senden und empfangen können. Das Problem mit den Empfängern ist leider, dass die sowohl im 27 Mhz als auch im 40 MHz – Band empfangen und das auch noch selbst entscheiden. Bidirektional daher nicht möglich...
- Zwei Atmega. Habe zwei Atmega32 genommen. Geht aber auch mit zwei 8ern.
- Für die PC-Kommunikation einen Pegelwandler.
- Kleinteile, wie Kabel und Batterien.

1.4. Ist so etwas zulässig?

Das wüsste ich auch gerne. Falls es jemand weiß, soll er das bitte mitteilen.

2. Das Zusammenstellen der Hardware

Sender und Empfänger

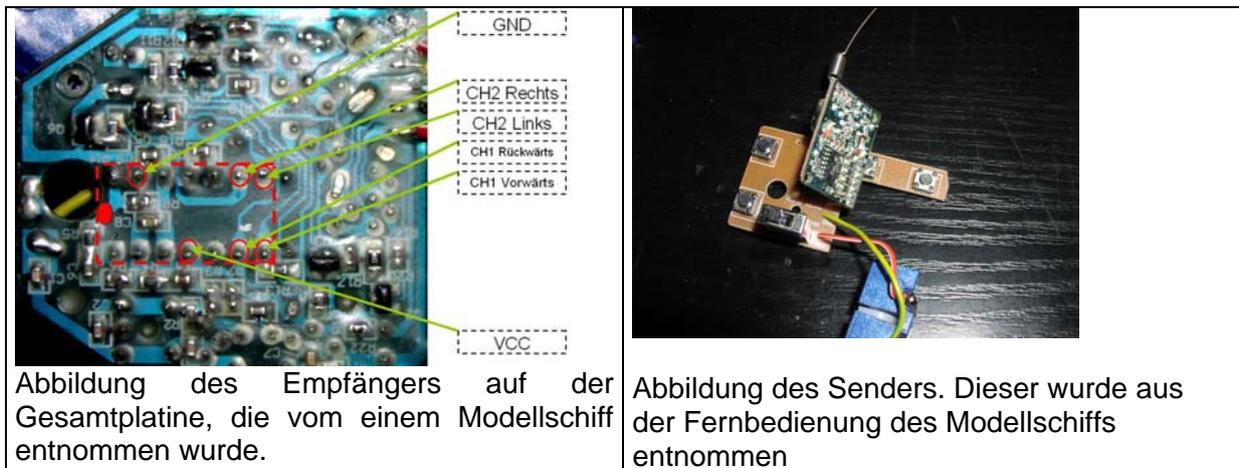
Hat man sich ein solches RC-Fahrzeug gekauft, hat man i.d.R. einen Controller vom Typ

RX-2-G 80714285

TX-2-G 80421285

Die Bezeichnung am Ende variiert. Der eine RX ist der Empfänger der TX der Sender.

Die Chips haben eine 3.3 Volt oder eine 5 Volt TT-Logik. Sie sehen wie folgt aus:



Anschluss der Sender und Empfänger an die UP

Es handelt sich um 2 Kanal Sender und Empfänger. Jeder Kanal ist bei dem Sender mit zwei Tastern versehen – diese sind Links, Rechts, Vorwärts, Rückwärts beim Fahrzeug zugeordnet.

Drückt man am Sender den Druckknopf, so wird am Empfänger ein HighPegel am jeweiligen Kanal ausgegeben. Vgl. hierzu die o.s. Abbildung.

Ersetzt man den Taster durch einen npn-Transistor oder einem Optokoppler, so kann man die Zustände auch durch eine UP steuern lassen.

Jeder Knopf wird durch einen npn-Transistor oder einem Optokoppler vom Typ 4N28 gebrückt. Die Basis wird dann mit dem entsprechenden Port des Atmegas verbunden.

Falls es jemand nachbauen möchte, hier die Belegungen, wie sie im Quelltext gemacht wurden:

Hoch	Alias	Portd.7
Runter	Alias	Portc.0
Rechts	Alias	Portb.0
Links	Alias	Portb.1

Anmerkung dazu: Es gibt Sender, bei denen gibt es drei Zustände für einen Kanal. Dafür gibt es einen speziellen Quelltext, der hier nicht mit dabei ist. Aber nicht alle RC-Fahrzeuge verwenden diesen Kanal. Manche Chips haben ihn und andere nicht. Er ist häufig bei

Modellschiffen zu finden (aber das ist eher ein Gefühl als...). Manchmal werden diese Zustände auch gar nicht an den Bedienelementen verwendet und sind am Chip gebrückt. Da muss man mal genauer schauen. Die Empfänger können diesen Zustand nach meiner Erfahrung aber alle empfangen.

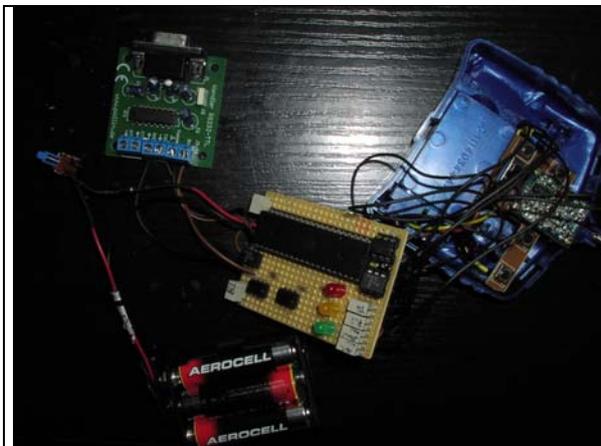


Abbildung des Senders.

Man erkennt:

- Batterien mit 4,5V Gesamtspannung
- Pegelwandler von Pollin
- Punktrasterplatine mit 3 LEDs, 4 Optokopplern, einem Atmega32
- Modifizierte Fernbedienung

Der Empfänger sieht genauso aus. Beim Empfänger wurden die Massen mit dem Atmega zusammengelegt. Und die aus der Abbildung erkennbaren Pins an die entsprechenden Pins gelegt. Die Pins nach Quelltext sind:

I_runter	Alias	Pind.3
I_hoch	Alias	Pind.4
I_rechts	Alias	Pind.6
I_links	Alias	Pind.5

Grundbeschaltung eines Atmegas32

Für Einsteiger sei gesagt, dass beim oben abgebildeten Atmega folgende Versorgungen angeschlossen wurden.

Pinnummer	Bezeichnung	Anschluss an
10	VCC	an Pluspol der 4,5Volt
11,31	GND	an Minuspol der 4,5 Volt
32,30	AREF, AVCC	an Pluspol der 4,5 Volt

3. Der Quellcode

Ich muss mich direkt entschuldigen, dass der Quellcode nicht kommentiert ist. Aber ich vergesse das im Eifer des Gefechts immer wieder. Aber die Bascom Sprache ist ja fast selbsterklärend.

Das Wichtigste in Kürze

Der Quelltext ist für Sender und Empfänger in einem. Zum Wechsel beim Beschreiben des Atmegas muss folgende Zeile ändern:

```
'0=Senden
'1=Empfangen
Funktion=1
```

Der Quelltext darf gerne geändert und verbessert werden.

```

$regfile = "m32def.dat"           ' specify the used micro
$crystal = 1000000                ' used crystal frequency
$hwstack = 100                    ' default use 32 for the hardware
stack                               stack
$swstack = 100                    ' default use 10 for the SW stack
$framesize = 60

$baud = 4800

Wait 1

Enable Interrupts
Declare Sub Zuruecksetzen
Declare Sub Zahlen
Declare Function Tonerzeugung(byval Textinhalt As String) As Integer

Dim I As Integer
Dim Zahler As Integer
Dim Zahler2 As Integer
Dim Funktion As Integer
Dim Zwischenspeicher As Integer
Dim Startzahler As Byte , Startzahlerx As Eram Byte
Dim Pulses As Word , Periods As Word
Dim Eingabe As String * 260

Dim Akt_wert1 As Word
Dim Akt_wert2 As Word
Dim Akt_wert3 As Word
Dim Akt_wert4 As Word
Dim Akt_wert5 As Word
Dim Eingabetextlaenge As Integer

Dim Wartezeit As Integer
Dim Automatik As Bit
Dim Aktueller_status As Byte
Dim Alter_status As Byte

Dim Init As Bit
Dim Inhalt(80) As Byte
Dim Schwelle_u As Integer
Dim Schwelle_o As Integer
Dim Zeichen As Byte
Dim A As Byte , S As String * 2

Config Pina.1 = Input
Config Pind.3 = Input
Config Pind.4 = Input
Config Pind.5 = Input
Config Pind.6 = Input

Config Portb.2 = Output
Config Portb.5 = Output
Config Portb.6 = Output
Config Portb.7 = Output
Config Porta.0 = Output

Config Portc.0 = Output
Config Portd.7 = Output
Config Portb.0 = Output
Config Portb.1 = Output

Led_err Alias Portb.5
Led_tx Alias Portb.6
Led_rx Alias Portb.7

Hoch Alias Portd.7
Runter Alias Portc.0
Rechts Alias Portb.0
Links Alias Portb.1

I_runter Alias Pind.3
I_hoch Alias Pind.4
I_rechts Alias Pind.6
I_links Alias Pind.5

Pc_kommunikation Alias Portb.2

```

```

'0=Senden
'1=Empfangen
Funktion = 1

' zahlen wie oft das System bereits gestartet wurde
Startzahler = Startzahlerx                                     'Lesen aus dem Eeprom
Startzahler = Startzahler + 1
Startzahlerx = Startzahler                                   'Schreiben in den Eeprom

For I = 1 To 6
'Funktionstest der LEDs
Toggle Led_err
Toggle Led_tx
Toggle Led_rx
Waitms 150
Next I

Pc_kommunikation = 1
Waitms 1000
Config Serialin = Buffered , Size = 254

Wartezeit = 250

If Funktion = 1 Then
'print "Entwickelt von Niels Keller."
'print "-----"
'print "Gebe ab nun Daten aus ..."
Led_err = 1
Led_tx = 0
Led_rx = 0
Eingabe = ""
Aktueller_status = 0
Alter_status = 10
Zahler = 1
Zahler2 = 0
Do

If I_hoch = 1 And I_rechts = 1 Then
Aktueller_status = 1
If Alter_status = Aktueller_status Then
Else
'hier berechnen und ausgeben
Print " " ; Chr(zahler2);
If Zahler2 = 13 Then
'Line feed
Print " " ; Chr(10);
End If

Zahler2 = 0
Zahler = 1
Eingabe = ""
Alter_status = 1
End If
Else

If I_hoch = 1 Or I_rechts = 1 Then
If I_hoch = 1 Then
Aktueller_status = 2
If Alter_status = Aktueller_status Then
Else

If Zahler = 1 Then
Zahler2 = Zahler2 + 1
End If

If Zahler = 2 Then
Zahler2 = Zahler2 + 2
End If

If Zahler = 3 Then
Zahler2 = Zahler2 + 4
End If

If Zahler = 4 Then
Zahler2 = Zahler2 + 8
End If

If Zahler = 5 Then
Zahler2 = Zahler2 + 16
End If

If Zahler = 6 Then
Zahler2 = Zahler2 + 32
End If

```

```

If Zahler = 7 Then
Zahler2 = Zahler2 + 64
End If

If Zahler = 8 Then
Zahler2 = Zahler2 + 128
End If

Incr Zahler
Eingabe = "1"
Alter_status = 2
Toggle Led_rx
End If
End If
If I_rechts = 1 Then
Aktueller_status = 3
If Alter_status = Aktueller_status Then
Else

If Zahler = 1 Then
Zahler2 = Zahler2 + 1
End If

If Zahler = 2 Then
Zahler2 = Zahler2 + 2
End If

If Zahler = 3 Then
Zahler2 = Zahler2 + 4
End If

If Zahler = 4 Then
Zahler2 = Zahler2 + 8
End If

If Zahler = 5 Then
Zahler2 = Zahler2 + 16
End If

If Zahler = 6 Then
Zahler2 = Zahler2 + 32
End If

If Zahler = 7 Then
Zahler2 = Zahler2 + 64
End If

If Zahler = 8 Then
Zahler2 = Zahler2 + 128
End If

Incr Zahler
Eingabe = "1"
Alter_status = 3
Toggle Led_rx
End If
End If
End If

If I_runter = 1 Or I_links = 1 Then
If I_runter = 1 Then
Aktueller_status = 4
If Alter_status = Aktueller_status Then
Else
Incr Zahler
Eingabe = "0"
Alter_status = 4
Toggle Led_rx
End If
End If
If I_links = 1 Then
Aktueller_status = 5
If Alter_status = Aktueller_status Then
Else
Incr Zahler
Eingabe = "0"
Alter_status = 5
Toggle Led_rx
End If
End If
End If
End If

'Warten um ihn weniger scharf zu machen
Waitms 20
Led_tx = 0
Led_rx = 0

```

```

Loop
End If

If Funktion = 0 Then
Do
Led_err = 1
Call Zuruecksetzen
If Automatik = 0 Then
Led_rx = 1
Wartezeit = 125
Zeichen = Ischarwaiting()
If Zeichen = 1 Then
Zeichen = Waitkey()
Eingabe = Chr(zeichen)
'Preamble
Hoch = 1
Rechts = 1
Waitms Wartezeit
Call Zuruecksetzen
'Ende der Preamble
Zahler = Tonerzeugung(eingabe)
Led_tx = 0
End If
End If
Loop
End If

Function Tonerzeugung(byval Textinhalt As String) As Integer
Local Lange As Integer
Dim Zahler3 As Integer
Dim Langenzahler As Integer
Dim Zustand2 As Bit
Dim Signaldauer As Integer 'Angabe in Zyklen
Dim Frequenz As Integer 'Angabe in "Pseudo" Hertz(1/s)
Dim Aktuelleszeichens As String * 1
Dim Aktuellescharzeichen As String * 1
Dim Aktuelleszeichen As Integer
Dim Binarstring As String * 8
Dim Binarstring2 As String * 8
Dim Binarstring3 As String * 10

Lange = Len(textinhalt)
Langenzahler = 1

While Langenzahler <= Lange

'Senden der Preamble
Hoch = 1
Rechts = 1
Waitms Wartezeit
Call Zuruecksetzen
'Ende der Preamble
Alter_status = 0

'Umrechnen des Stringwerts eines Buchstabens in den ASCII-Wert
Aktuelleszeichens = Mid(textinhalt , Langenzahler , 1 )
Zahler2 = 0
While Zahler2 < 256
Aktuellescharzeichen = Chr(zahler2)
If Aktuelleszeichens = Aktuellescharzeichen Then
Aktuelleszeichen = Zahler2
Zahler2 = 255
End If
Incr Zahler2
Wend

'Umwandeln in einen "binären" String mit der Bitreihenfolge 0 bis 7 (links bis rechts)
Binarstring2 = Bin(aktuelleszeichen)
Binarstring2 = Right(binarstring2 , 8)
Binarstring = ""
Zahler = 1
While Zahler < 9
Zahler2 = 9 - Zahler
Binarstring = Binarstring + Mid(binarstring2 , Zahler2 , 1 )
Incr Zahler
Wend

Zahler3 = 1
While Zahler3 < 9
If Mid(binarstring , Zahler3 , 1 ) = "1" Then
If Alter_status = 1 Or Alter_status = 4 Or Alter_status = 3 Or Alter_status = 0 Then

```

```

Rechts = 1
Alter_status = 2
Else
If Alter_status = 2 Or Alter_status = 4 Or Alter_status = 3 Then
Hoch = 1
Alter_status = 1
End If
End If
Else
If Alter_status = 3 Or Alter_status = 1 Or Alter_status = 2 Or Alter_status = 0 Then
Links = 1

Alter_status = 4
Else
If Alter_status = 4 Or Alter_status = 2 Or Alter_status = 1 Then
Runter = 1
Alter_status = 3
End If
End If
End If
Waitms Wartezeit
Call Zuruecksetzen
Toggle Led_tx

Incr Zahler3
Wend

'Senden der Preamble
Hoch = 1
Rechts = 1
Waitms Wartezeit
Call Zuruecksetzen
Alter_status = 0
'Ende der Preamble

Incr Langenzahler
Wend

Tonerzeugung = Lange
End Function

Sub Zuruecksetzen
Hoch = 0
Runter = 0
Rechts = 0
Links = 0
End Sub

```

Was macht dieser Code?

Ach wenn ich das noch wüsste ;-)

Ist eigentlich gar nicht schwer. Der Sender wartet auf eine Eingabe, die er auf seiner seriellen Schnittstelle erwartet. Kommt ein Zeichen, so wandelt er dieses in eine Signalfolge um. Kommen mehrere Zeichen, so puffert er diese in seinem Speicher.

Wie funktioniert die Codierung?

Ein High-Bit wird durch: Hoch oder Rechts
ein Low-Bit durch: Runter oder Links
kodierte.

Kommen **zwei Bit mit gleichem Inhalt** hintereinander, so muss ein **Wechsel** erfolgen. Dadurch ist das Format **nicht zeitkritisch** in der Abtastung.

Die Bytebegrenzung erfolgt durch das gleichzeitige Senden von Hoch und Rechts.

Der Empfänger liest dann in dieser die Zeichen aus und dekodiert daraus wieder alle Zeichen. Nach dem Dekodieren eines Zeichens wird dieses über dessen serielle Schnittstelle ausgegeben.

4. Praxisbeispiele

Verwende das Gerät um Einstellungen an verschiedene Arten von uPs zu versenden und von diesen auszulesen. Habe hier bereits vor einiger Zeit ein kleines (ist wörtlich zu nehmen) Video eingestellt: <http://www.mikrocontroller.net/attachment/49352/RC-Funk.wmv>

Man kann die Übertragung z.B. unter Verwendung von Hyperterminal zwischen zwei PCs einsetzen. Also Daten von PC zu PC, oder von PC auf uP, oder von uP auf PC, oder von uP auf uP zu übertragen.