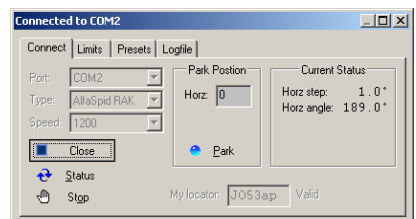
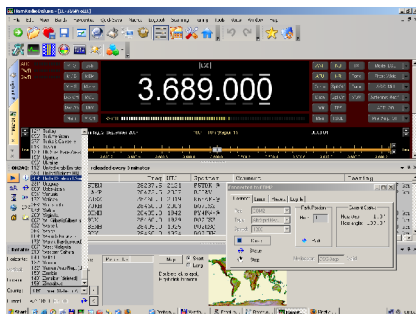


DIGITALE STEUERUNG EINER ANALOGEN ANTENNENROTORS ÜBER RS232

(Im AlfaSpid- Datenprotokoll)

Viele Funkamateure nutzen noch ältere (analoge) Antennenrotoren und haben gelegentlich auch den Wunsch, einen (ohnehin im Shack vorhandenen) Computer zur Antennensteuerung zu nutzen. Insbesondere dann, wenn auch noch eine AFU-Software (wie z.B. das kostenlose und sehr leistungsfähige HamradioDeluxe) zur Verfügung steht. Dieses Programmpaket beinhaltet (neben Logbuch usw) auch ein DX-Cluster und ein Antennensteuerungsmodul, welches gängige digitale Rotoren (Yaesu, AlfaSpid usw) direkt ansprechen kann.



Rotorsteuerung bei HamradioDeluxe

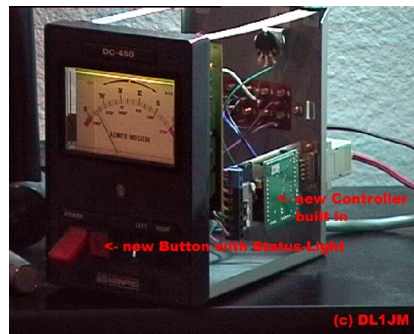
Software ist schnell installiert und eine elektronische Schaltung meist auch relativ schnell zusammengebaut, aber der Aufwand, einen guten soliden Antennenrotor gegen ein digitales Modell auszutauschen ist doch recht hoch und hält damit so manchen OM von Moderni-

sierungs- und Digitalisierungsmaßnahmen ab.

Hier kommt nun die nachfolgende Eigenbaulösung ins Spiel:

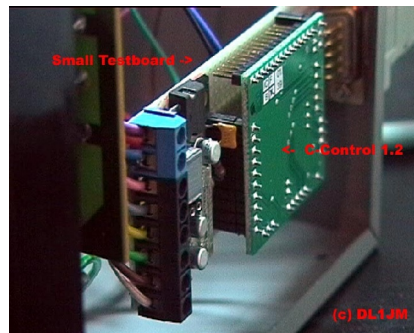
Verfügt man über einen alten analogen Rotor mit Steuergerät (wie z.B. mein alter Kenpro DC-450 o.ä.), steht der Digitalisierung nix mehr im Wege.

Einzigste Voraussetzung ist, dass der Rotor ein DC-Signal liefern muss, welches proportional zur Antennenposition sein sollte (eingebautes Potentiometer im Rortorgehäuse). Die Steuerung sollte über Tasten und/oder Relais (links/rechts) möglich sein.



Mein alter Kenpro- Rotor modifiziert

Die nachfolgende, kleine digitale Schaltung wird ins Rotorsteuergerät eingebaut und einerseits an das o.a. DC-Positionssignal und andererseits an die manuelle Steuerung (Tasten/Relais) angeschlossen.



Das kleine digitale RS232-Steuergerät

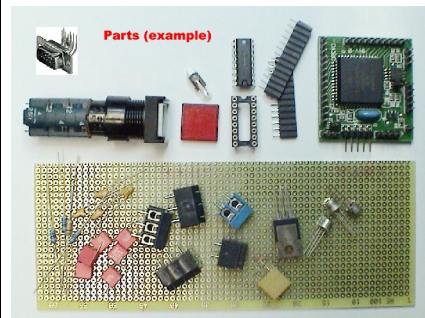
Über ein serielles Kabel wird dann der kleine Controller mit dem PC verbunden.

In meiner technischen Lösung kann der neue Controller jederzeit mit neuer Software (update) versehen werden, ohne dass das Gerät nachträglich nochmal geöffnet werden müsste (eingebautes Programmier-Interface).



Eine zusätzliche beleuchtete Taste zeigt die Datenkommunikation an und ermöglicht verschiedene Modi (Terminal-Mode, autom. PC-Steuerung und Soft-update)

Man benötigt nur wenige Bauteile (Kosten unter 50,- Euro), wobei das Kernstück ein C-Control der Fa. Conrad ist.

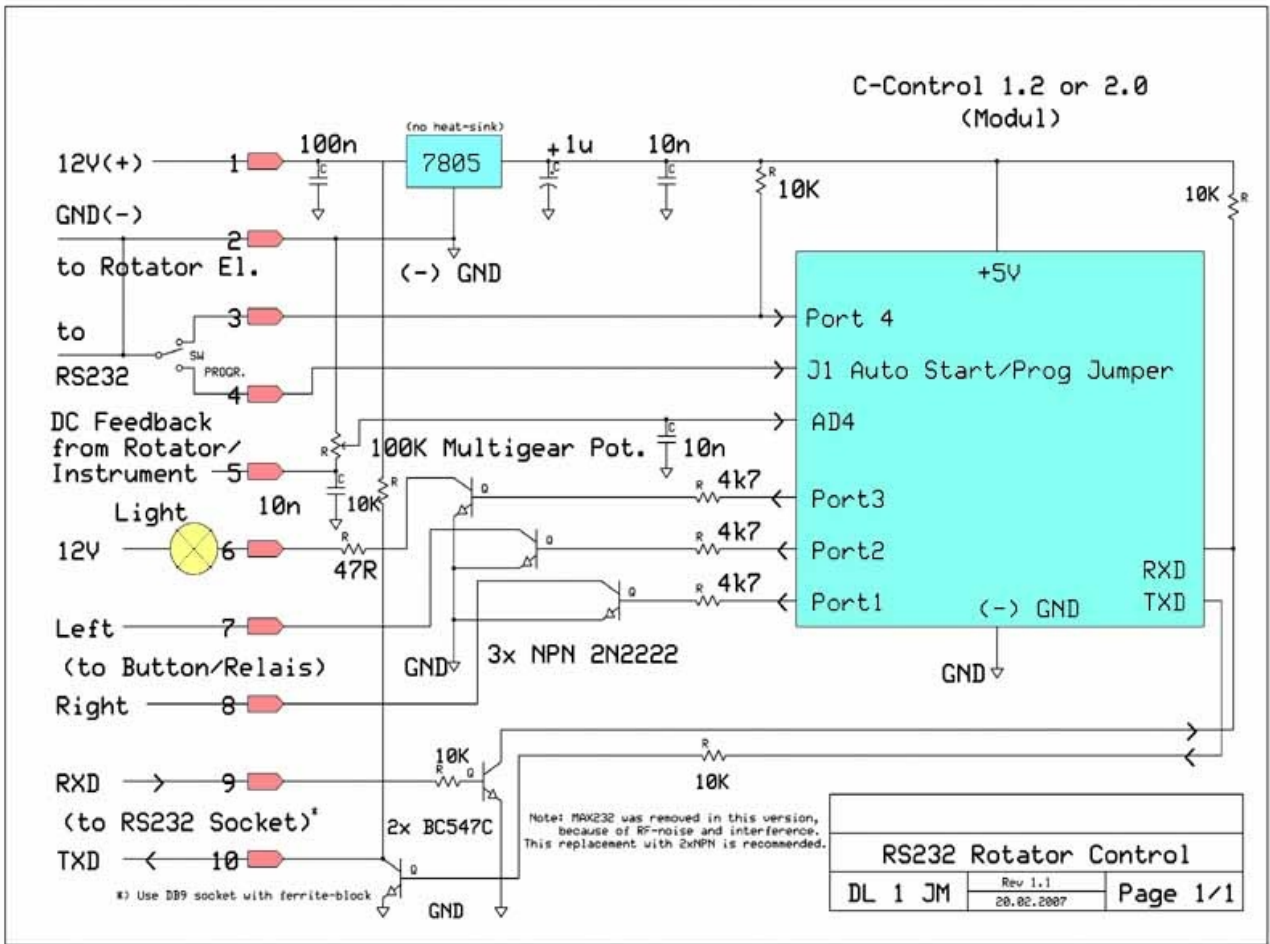


Wenige Bauteile sind nur erforderlich

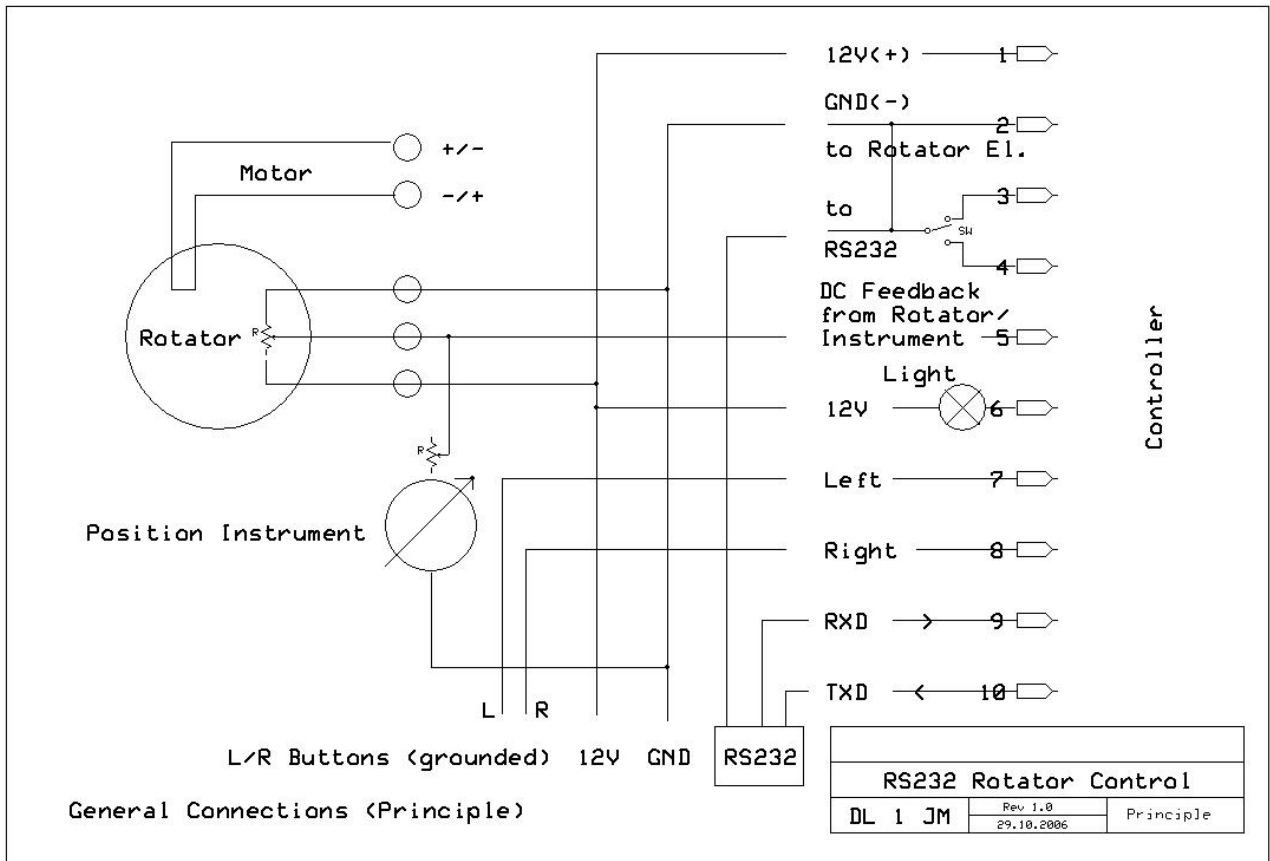
Die kleine Schaltung wird zweckmässigerweise auf einem Löttraster aufgebaut.

Die Schaltbilder und das BASIC-Listing sind den nachfolgenden Seiten zu entnehmen.

Nun viel Erfolg beim Nachbau: DL1JM



Das Schaltbild der C-Control-Rotorsteuerung



Die Anschaltung der digitalen Steuerung beim Kenpro DC-450 (als Beispiel)

```

*****
C-Control/Integer-BASIC Rotator-Control Ver.1.4 (c)2006 by DL1JM
This program can be distributed freely as it is. The copyright
may not be removed. This program will be used on Your own
risk! DL1JM is not responsible for any damages or accidents as
result of the use of this software. This program was checked
carefully, but may include some bugs. There is no guarantee for
operation. Needed: Serial Connection 9600bd N,8,1,no
handshake for programming and for terminal-mode and 1200bd
N,8,1,no handshake for Hamradio-Deluxe COM2 Connections:
Port 1,2,3,4,AD4,RXD,TXD,IJ,5V,GND. Not needed:
Programming adaptor, because DL1JM-interface includes it.
Programming and different modes depend on interface-switch
J1/Port4 This Program was designed for old/slow C-control
M1.x.For newer versions, timings for blinking light etc. maybe
changed Rotator DC-feedback should be carefully adjusted by
interface-potentiometer. The AD4-input should never ever exceed
5V!!! The max. value at max. rotation should be always 5V at
AD4, this is equal to &HFF (-255). -180 degrees "0" position
result in a readout of(200) at Adj.-mode. Adjustment should
be made very carefully and the potentiometer should be below
mid-position at the beginning! The accuracy of 8-bit AD-
converter and rotator is approx. +/-2 deg.
*****
--- Programoperations
DEFINE LOOP BYTE
DEFINE POSH BYTE
DEFINE POSZ BYTE
DEFINE POSE BYTE
DEFINE B01 BYTE
DEFINE B02 BYTE
DEFINE B03 BYTE
DEFINE B04 BYTE
DEFINE B05 BYTE
DEFINE B06 BYTE
DEFINE B07 BYTE
DEFINE B08 BYTE
DEFINE B09 BYTE
DEFINE B10 BYTE
DEFINE B11 BYTE
DEFINE B12 BYTE
DEFINE B13 BYTE
DEFINE MOV BYTE
DEFINE ROT BYTE
DEFINE BLNK BYTE
DEFINE DEGREE WORD
DEFINE TIM BYTE
DEFINE POSITION WORD
DEFINE RIGHT PORT[1]
DEFINE LEFT PORT[2]
DEFINE LIGHT PORT[3]
DEFINE SWITCH PORT[4]
DEFINE STATUS AD[4]

#MODUS
BAUD R9600
LIGHT=0
LEFT=0

RIGHT=0
BLNK=20
GOSUB BLINKLIGHT
IF SWITCH<=0 THEN GOTO REMOTE
PRINT
PRINT"***TERMINALMODE***"

#BEGIN
PRINT
PRINT "Actual Position:";
GOSUB GETPOSITION
PRINT POSITION;"DEG";,"New Direction, 777=Adjust,
888=PARK, 999=Exit? ";
INPUT DEGREE
PRINT DEGREE
IF DEGREE=777 THEN GOTO ADJUST
IF DEGREE>887 THEN GOTO PARK
GOTO ROTATE

#PARK
IF DEGREE>888 THEN GOTO REMARK
DEGREE=0
GOTO ROTATE

#REMARK
PRINT
PRINT"***RESTART * Close Terminalprog.now!*"
GOTO MODUS

#ADJUST
PRINT "Value should be adjusted with <<200>> at -180
Deg.Position.Now: ";STATUS
IF STATUS<=200 THEN GOTO ADJUST
PRINT"***SUCCESS ***"
GOTO BEGN

#REMOTE
BAUD &HE4 for 600bd
BAUD R1200

#MAINLOOP
LEFT=0
RIGHT=0
LIGHT=0
MOV=0
IF SWITCH=0 THEN GOTO MODUS
IF RXD=0 THEN GOTO MAINLOOP
GOSUB READCOM
IF B01<&H57 THEN GOTO MAINLOOP
IF B12=&H0F AND B13=&H20 THEN GOTO STOP
IF B12=&H2F AND B13=&H20 THEN GOTO SET
IF B12=&H1F AND B13=&H20 THEN GOSUB STATUS
GOTO MAINLOOP

#READCOM
TOGLIGHT
GET B01
IF B01<&H57 THEN GOTO ENDREAD
GET B02

GET B03
GET B04
GET B05
GET B06
GET B07
GET B08
GET B09
GET B10
GET B11
GET B12
GET B13

#ENDREAD
TOGLIGHT
TOGLIGHT
PAUSE 10
NEXT LOOP
RETURN

#STOP
LEFT=0
RIGHT=0
GOSUB STATUS
GOTO MAINLOOP

#STATUS
GOSUB GETPOSITION
POSITION=POSITION+360
PUT &H57
POSH=POSITION/100
PUT POSH-&H30
POSITION=POSITION-POSH*100
POSZ=POSITION/10
PUT POSZ-&H30
POSE=POSITION-POSZ*10
PUT POSE-&H30
PUT &H0
PUT &H0
PUT &H0
PUT &H0
PUT &H0
PUT &H0
PUT &H0
RETURN

#SET
LIGHT=0
DEGREE=(B02-48)*100+(B03-48)*10+(B04-48)
DEGREE=DEGREE-360

#ROTATE
DEGREE=DEGREE*20/36
IF DEGREE>99 THEN DEGREE=DEGREE-100 ELSE
DEGREE=DEGREE+100

IF DEGREE<51 AND STATUS<150 THEN
DEGREE=DEGREE+200

#RECHECK
IF (DEGREE=STATUS OR DEGREE=1+STATUS OR
DEGREE=STATUS+1) THEN GOTO MAINLOOP
IF DEGREE<STATUS THEN GOTO LEFT
IF MOV=2 THEN GOSUB DELAYREVERSE
RIGHT=1
MOV=1
GOTO ROTATION

#LEFT
IF MOV=1 THEN GOSUB DELAYREVERSE
LEFT=1
MOV=2

#ROTATION
ROT=STATUS+1
SECOND=0
PAUSE 20

#CHECKIT
TOGLIGHT
IF RXD=0 THEN GOTO COMPARE
GOSUB READCOM
IF B01<&H57 THEN GOTO COMPARE
IF B12=&H0F AND B13=&H20 THEN GOTO STOP
IF B12=&H2F AND B13=&H20 THEN GOTO SET
IF B12=&H1F AND B13=&H20 THEN GOSUB STATUS

#COMPARE
GOSUB CHECKMOVE
IF ROT=0 THEN GOTO MAINLOOP
IF DEGREE>STATUS AND MOV=1 THEN GOTO CHECKIT
IF DEGREE<STATUS AND MOV=2 THEN GOTO CHECKIT
LEFT=0
RIGHT=0
GOTO RECHECK

#CHECKMOVE
IF SECOND<2 THEN RETURN
IF ROT=STATUS THEN ROT=0 ELSE ROT=STATUS
SECOND=0
RETURN

#DELAYREVERSE
RIGHT=0
LEFT=0
LIGHT=0
PAUSE 30
RETURN

#GETPOSITION
POSITION=STATUS*36/20
IF POSITION>179 THEN POSITION=POSITION-180 ELSE
POSITION=POSITION+180
RETURN
END

```

Das Software-Listing des kleinen BASIC-Programms (kann auch per E-Mail von DL1JM angefordert werden)