

MYC Interface für RC5A3 Rotorsteuerung von Create

Author DK1RI, Version V04.1 20180118

This project can be found in <https://www.github.com/dk1ri> also.

Einleitung

Im geöffneten Gerät sind personengefährdende Spannungen direkt zugänglich. Das Öffnen des Geräts und der Einbau des Interfaces ist nur Personen mit ausreichender fachlicher Qualifikation gestattet.

Der Einbau des Interfaces und der Umbau der Originalschaltung erfordert das Verständnis der Schaltung des Rotorsteuergerätes.

Es gibt unterschiedliche Varianten der Elektronik der Originalsteuerung. Das vorliegende Interface gilt nur für eine Version aus den 90er Jahren. Weitere Details im Text.

Der Änderungsaufwand an der Originalschaltung sollte so gering als möglich sein; allerdings sind doch einige Eingriffe in die Originalschaltung nötig.

Das Interface soll in das Originalgehäuse passen.

Das Steuerung erfolgt über I2C Bus, USB oder seriell (TTL Pegel, 19k2, 8N2) mit MYC Protokoll. Defaultmäßig sind alle Schnittstellen aktiv. Mit dem Initialisierungsbefehl können diese aber deaktiviert werden. Der Initialisierungsbefehl funktioniert aber immer.

Beschreibung und Bedienungshinweise

Die Eagle-Dateien stehen bei [1].

Die Stromversorgung (12V) liefert das Steuergerät. Benötigt werden maximal 80mA zusätzlich.

Die Steuerung macht ein AVR Mikrocontroller.

Wenn die Motorsteuerung des Interfaces aktiv ist, schaltet ein Relais um von Durchleitung von Originalsteuerung zum Motor auf die Interfacesteuerung zum Motor.

Dazu muss nach dem Einschalten ein Aktivierungsbefehl (&H0101) gegeben werden. Dann funktionieren die CW/CCW und Preseteinstellung der Originalsteuerung nicht mehr.

Die Werte des Interface können aber immer abgefragt werden, Die Kompassanzeige funktioniert auch unverändert immer.

Die Geschwindigkeitssteuerung kann das Interface nicht machen: das Potentiometer zur Einstellung liegt in einem galvanisch getrennten Zweig der Stromversorgung.

Das Signal der Anschlag-LED, die ebenfalls galvanisch getrennt ist, wird durch einen in Reihe zur Anzeige LED geschalteten Optokoppler übertragen und verwendet.

Alle Richtungseinstellungen erfolgen relativ zu Nord; 0 bzw. 360 Grad im Norden. Werte von 0 bis 359 sind erlaubt. Ohne weitere Einstellungen (Defaultwert) sind die angezeigten Werte der Antennenrichtung dann richtig, wenn die Antenne beim Rotoranschlag nach Süden zeigt. Dies entspricht der Kompassanzeige, wenn keine Änderungen vorgenommen wurden. Andere Richtungen sind möglich. Mit dem Befehl &H08 wird die Richtung eingegeben, in die die Antenne bei Rotoranschlag (nicht die Richtung des Rotoranschlags!!) steht (default: 180 Grad).

Die numerische ID des Interfaces ist auf 1 gesetzt; der Name auf „Device 1“.

Diese Werte können verändert werden.

Im EEPROM sind folgende Werte gespeichert und bleiben auch nach dem Abschalten erhalten: Presetrichtung, die Kalibrierungswerte, Antennenrichtung am Anschlag, Device ID, Device Name, aktive Interfaces und I2C Adresse

Das Interface ist fest mit der Stromversorgung des Rotors verbunden.

Die Relais sind hardwaremäßig so verschaltet, dass beide Motorwicklungen des Rotors nie gleichzeitig Spannung bekommen.

Der Rotor kann nur um 360 Grad gedreht werden. Ohne Eichung kann dieser Wert etwas größer oder kleiner sein.

Wird dabei der Endschalter ausgelöst, schaltet der Motor sofort ab.

Ist die Preset-Antennenrichtung bei Presetbetrieb 5 Grad oder weniger am Rotoranschlag, wird eine Preset-Antennenrichtung in 5 Grad Abstand verwendet. Falls der Rotor trotzdem über die Grenze hinausfährt, wird der Rotor dann sofort abgeschaltet und der Presetbetrieb wird beendet.

Die Stromversorgung des Originalgeräts

Es gibt offenbar unterschiedliche Versionen der Original-Steuerung. Bei meinem Gerät stimmt auch Schaltung und Schaltplan nicht überein.

In meiner Originalschaltung erfolgt die Versorgung der Steuerschaltung über zwei Trafowicklungen und ergibt nach der Gleichrichtung (gemessen) bei -13 und + 13V als Maximum und -8 und +8V als Minimum mit 50Hz. Wenn der Motor der Kompassanzeige anläuft, sinkt die + oder - Spannung (abhängig von Vor- oder Rücklauf) für mehrere 50Hz Perioden auf bis zu 5V ab.

Auch anders als der Schaltplan speist die positive und negative Versorgungsspannung das Richtungspotentiometer, das die Position des Rotors angibt, über 220Ohm Widerstände.

Zur Verbesserung wurden folgende Änderungen vorgenommen:

Die beiden Lampen zur Beleuchtung der Kompassanzeige werden direkt mit Wechselspannung gespeist; es wird die Wicklung verwendet, die die – Spannung speist. Neuere Versionen der Rotorsteuerung machen das ähnlich.

Zu den Kondensatoren der Gleichrichter wurden je 220uF parallel geschaltet. Die beiden Kondensatoren sind „fliegend“ an der äußeren Lampenfassung angelötet. Dort ist auch der Anschluss der Stromversorgung des Interfaces. Das Gehäuse muss für diesen Umbau komplett zerlegt werden.

Die Spannung am Rotorpotentiometer wurde mit VR1 und VR2 auf dem Interface auf +-2,5V stabilisiert. Diese Begrenzung wirkt auch auf die Originalschaltung, hat aber keinen Einfluss. Bei Verwendung des Interfaces sollte die Rotorsteuerung mit „MAN“ eingeschaltet werden. Das verringert den Stromverbrauch.

Nochmal der Hinweis: Es kann auch Varianten der Original-Steuerung geben. Der Umbau und Einbau des Interfaces muss dann anders erfolgen.

Potentiometerspannung

Wie oben beschrieben, schwankt die Spannung am Rotorpotentiometer sehr stark. Beide Spannungen schwanken symmetrisch um den Nullpunkt. Die Werte der Spannung ändern sich auch mit dem Schalten der Relais.

Die Position des Rotors wird in der Originalschaltung mit einer Brückenschaltung gemessen. Der Brumm der Potentiometerspannung hat daher kaum Einfluss.

Das Interface verwendet nicht die Brückenspannung sondern macht nur Messungen am Abgriff des Potentiometers über einen geeigneten Spannungsteiler. Der Widerstand R_y kann aus 3 Widerständen zusammengesetzt werden und muss 145kOhm (zB 143 + 2.2kOhm) betragen

Mit den gewählten Widerständen liegt (bei meiner Schaltung!) die Spannung am AD-Wandlereingang zwischen 0V und 1,1 V. Der AD Wandler verwendet die interne Referenz von 1,1V Der Eingang der AD Wandlers ist mit Dioden geschützt.

Das Rotor-Potentiometers ist zwar linear, die Funktion Rotorwinkel - Spannung am AD Eingang hat aber wegen des variablen Innenwiderstandes des Rotor-Potentiometers eine geringe Nichtlinearität. Die wird durch die Firmware korrigiert.

360 Grad Eichung

Die Eichung legt die beiden Spannungswerte fest, die sich vor und nach einer 360 Grad Drehung des Rotors ergeben. Als Defaultwerte sind zunächst die Werte eingetragen, die mein Interface zeigt. Zunächst wird der Rotor in Position manuell wenige Grad vor den Linksanschlag gefahren. Dies ist normalerweise die 180 Grad Stellung. Mit dem Befehl &H0B wird der zugehörige Spannungswert zwischengespeichert.

Danach wird der Rotor um genau 360 Grad gedreht. Die Kompassanzeige sollte wieder bei der gleichen Anzeige stehen. Auch diese Stellung soll wenige Grad vor dem Anschlag liegen (überprüfen!).

Danach wird mit dem Befehl &H0C wieder die zweite Spannung des Positonspotentiometers gemessen und beide Spannungswerte im EEPROM gespeichert. Damit ist die Steuerung geeicht und auch die Fehler der Spannungsteilerwiderstände kompensiert.

Besser als die Ablesung der Antennenrichtung mit der Kompassanzeige ist die Überprüfung der wahren Antennenrichtung. Dadurch entfällt der Fehler, den die Kompassanzeige liefert.

Die Befehle &H0B und &H0C stehen bei der normalen Anwendung im MYC System nicht zur Verfügung.

Genauigkeit

Der AD Wandler selbst hat eine Genauigkeit von ca. 0,4 Grad. Da die 360 Grad aber nicht einer vollen Drehung des Potentiometer entsprechen und auch die Temperaturtoleranzen zu berücksichtigen sind, ist die erreichbare Genauigkeit aber geringer.

Außerdem gibt es einen systembedingten Linearitätsfehler (<1 %) durch die Art der Messschaltung, da der schaltungstechnische Ersatzwiderstand des Rotor-Potentiometers nicht konstant ist. Durch die hohen Werte von $R_9 - R_{11}$, R_y wird der Effekt verringert. Dieser Fehler wird auch weitestgehend durch die Firmware kompensiert. Bei Verwendung anderer Widerstände muss diese Korrektur neu berechnet werden. Wer das rechnen will, kann zum Beispiel das LTSpice (TM) Modell und die Tabelle verwenden [5].

Der größte Fehler dürfte aber durch die Nichtlinearität des Potentiometers entstehen. Die Genauigkeit des Interface ist also nicht viel besser als die des Originalgeräts.

Befehle

Zu Details zum MYC Protokoll und zur Bedienung siehe [3] und [4] (aktuell).

Folgende Befehle werden von der I2C / RS232 / USB Schnittstelle akzeptiert; dies ist eine Kopie aus dem Bascom Programm:

Announce0:
'Befehl &H00
'basic annouement wird gelesen
'basic announcement is read
Data "0;m;DK1RI;RC5 rotator control;V05.1;1;145;1;24:1-1"
,

Announce1:
'Befehl &H01 0|1
'Motorsteuerung aus /an schalten
'control off / on
Data "1;os,Device off on;1;0,off;1,on"
,

Announce2:
'Befehl &H02
'liest Antennenposition 0... 359
'read antenna position
Data "2;ap,antenna position;1;360;lin:degree"
,

Announce3:
'Befehl &H03 0|1
'Manual / Preset switch
Data "3;os;1;0,manual;1,preset"
,

Announce4:
'Befehl &H04 0 to 359
'preset Antennerichtung; kann motor im preset mode starten
'preset, real antenna direction may starts motor if preset mode
Data "4;op,Preset Position;1;360;lin;degree"
,

Announce5:
'Befehl &H05
'manuell startet den Motor ccw
'manual start motor ccw
Data "5;ou,Motor ccw;1;0,idle;1,ccw"
,

Announce6:
'Befehl &H06
'manuell stoppt den Motor
'manual stops motor
'Data "6;ou,Motor stop;1;0,idle;1,stop"
,

Announce7:
'Befehl &H07
'manuell startet den Motor cw
'manual start motor cw
Data "7;ou,Motor cw;1;0,idle;1,cw"
,

Announce8:
'Befehl &H08 0 to 359
'schreibt Antenna_limit_direction 0 ... 359

```

'write Antenna_limit_direction
Data "8;op,Antenna_limit_direction;1;360;lin;degree"
,

Announce9:
'Befehl &H09
'liest Antenna_limit_direction
'read Antenna_limit_direction
Data "9;as8"
,

Announce10:
'Befehl &H0A
'liest Werte
'read values
Data "10;aa,Values 1;b,controlon;b,Preset;b,Motor_cw;b,Motor_ccw;b,not_Limit;b,preset_limits"
,

Announce11:
'Befehl &H0B
'liest Werte
'read values
Data "11;aa,values 2;w,preset value;w,Ccw_correction;w,Cw_correction,w,dir_rotor;w,preset_rotor"
,

Announce12:
'Befehl &H0C
'liest die Spannung an der ccw Grenze, Eichung (noch vor hardware Limit)
'read voltage for ccw limit, calibration (before hardware limit)
Data "12;ou;1;0,idle;1,CCW calibration"
,

Announce13:
'Befehl &H0D
'liest die Spannung an der cw Grenze, Eichung (noch vor hardware Limit)
'read voltage for cw limit, calibration (before hardware limit)
Data "13;ou;1;0,idle;1,CW calibration"
,

Announce14:
'Befehl &H0E
'liest limit
'read limit
Data "14;aa,limit value;a,Limit"
,

Announce15:
'Befehl &H0F
'Motorsteuerung aus /an status lesen
'read on / off control
'Data "1;as,as1"
,

Announce16:
'Befehl &HF0<n><m>
'liest announcements
'read n announcement lines
Data "240;ln,ANNOUNCEMENTS;145;24"

```

'
Announce17:
'Befehl &HFC
'Liest letzten Fehler
'read last error
Data "252;aa,LAST ERROR;20,last_error"
'

'
Announce18:
'Befehl &HFD
'Geraet aktiv Antwort
'Life signal
Data "253;aa,MYC INFO;b,ACTIVE"
'

'
Announce19:
'Befehl &HFE :
'eigene Individualisierung schreiben
'write individualization
Data "254;ka,INDIVIDUALIZATION;20,NAME,Device 1;b,NUMBER,1;a,I2C,1;b,ADRESS,2,{0 to 127};a,RS232,1;a,USB,1"
'

'
Announce20:
'Befehl &HFF :
'eigene Individualisierung lesen
'read individualization
Data "255;la,INDIVIDUALIZATION;20,NAME,Device 1;b,NUMBER,1;a,I2C,1;b,ADRESS,2,{0 to 127};a,RS232,1;b,BAUDRATE,0,{19200};3,NUMBER_OF_BITS,8N1;a,USB,1"
'

'
Announce21:
' - no operate command is done before Control_on is set
Data "R !\$3 !\$5 !\$7 IF \$15=0"
'

'
Announce22:
' - motor at limit will switch off , no need for logic device to send switch off command
Data "R \$6 IF \$14=1"
'

'
Announce23:
'No Preset If hardwarelimit
Data "R !\$3 IF \$14=1"
'

I2C

Die Default Adresse ist 2 .

Mit dem Befehl &HFE03<n> kann die Adresse in n (1 ... 128) geändert werden.

Pullup Widerstände für den I2C Bus (R1/R2) können bei Bedarf bestückt werden. Der Gesamtwiderstand am Bus sollte zwischen 1 und 10 kOhm liegen.

Wenn Geräte am I2C Bus nur 3.3V vertragen (zB der Raspberry), dürfen die Pullup Widerstände nicht bestückt werden.

Fehlermeldungen

Der Befehl &HFC liefert den letzten Fehler im Format:
aktuelle Befehlsnummer - Fehler - Befehlsnummer beim Auftritt des Fehlers
Dazu werden die empfangenen Befehle von 0 bis 255 umlaufend gezählt.

Reset

Ist der Reset Jumper JP1 beim Anlegen der Versorgungsspannung überbrückt, werden wieder die Defaultwerte eingelesen. Dies ist hilfreich, wenn die aktuelle I2C Adresse verloren gegangen ist.

Watchdog

Die Befehlseingabe und Ausführung muss in weniger als 1 Sekunde beendet sein. Danach werden die bereits empfangenen Daten gelöscht. Dies soll falsche Eingaben vermeiden. Mit dem &HFC "letzten Fehler" Befehl kann man Eingabefehler sehen.

Bei einem Lesebefehl müssen die Daten innerhalb von 10 Sekunden vom I2C Master abgeholt werden – wenn die I2C Schnittstelle gerade verwendet wird. Danach werden die Daten gelöscht.

Diese Zeit kann mit dem Wert Tx_factor im Bascom Programm geändert werden. Neue Befehle können erst eingegeben werden, wenn alle Daten abgeholt wurden. Wird die RS232 / USB Schnittstelle verwendet, werden die Daten sofort ausgegeben.

Es gibt einen kompletten Reset, wenn die Hauptschleife länger als 2 Sekunde dauert, zum Beispiel, wenn die I2C Schnittstelle nicht korrekt arbeitet.

Software

Die Steuerung übernimmt ein AVR Mikrocontroller ATMEGA168 oder größer
Die Software wurde in BASCOM geschrieben [2]

Programmierung des Prozessors

Zur Programmierung des Prozessors ist ein 6poliger ISP Stecker vorhanden.

Die Fuses müssen möglicherweise programmiert werden (sh Bascom Programm) !! Prozessortyp und Frequenz müssen ggf angepasst werden. Beim Programmieren sollte der Jumper JP3 entfernt werden.

USB Schnittstelle

Das Interface kann mit der USB Platine UM2102 von ELV bestückt werden. Die USB Platine wird auf der Unterseite der Interfaces verlötet: der USB Stecker zeigt nach innen. Die 4 inneren Pins des Verbinders ST2 sind mit dem 4 poligen Verbinder JP4 auf dem Interface zu verbinden. USB Platine und Interface müssen voneinander isoliert werden. Dazu wird der Modul über eine Stiftleiste verbunden. Der Modul sollte am Schluss verlötet werden. [USB Befestigung](#)

Die 5V Stromversorgung erfolgt dann über USB Der Spannungsregler IC2, C2 und C3 entfallen.

Anstelle des USB Moduls kann auch eine serielle Schnittstelle (TTL Pegel, nicht invertierend, 19k2, 8N1) angeschlossen werden.

SMD

Die Leiterplatte ist teilweise mit SMD bestückt. Bei den nötigen Bauteilen sind das aber nur relativ

großen Kondensatoren (1206) .

Stromversorgung

Stromaufnahme ist ca. 120mA max. Die Stromversorgung übernimmt das Rotor Steuergerät, bzw der USB Modul (nur 5V)

Leiterplattenbestückung, Gehäuseeinbau

Offenbar gibt es mehrere Varianten der Elektronik der Originalsteuerung. Weitere Details siehe oben „Die Stromversorgung des Originalgeräts“.

Das folgende gilt für meine Version:

Die Dioden und Transistoren sind unkritisch.

Die SMD Kondensatoren können auch kleiner sein, 4,7u oder 10u reichen auch.

Der Prozessor ist ein ATmega168 oder größer.

Der Widerstände Ry muss 145kOhm betragen und kann aus bis zu 3 Widerständen zusammengesetzt werden: zum Beispiel 143kOhm und 2.2kOhm in Reihe. Das ist ausreichend genau.

Die Interfaceplatine wird in das Gehäuse der Rotorsteuerung eingebaut. ***Dazu ist das Gerät vom Stromnetz zu trennen !! Es sind unbedingt entsprechende VDE Sicherheitsrichtlinien zu beachten (die ein Funkamateure kennen sollte).***

Die Platine wird in der Schale des Deckels mit drei Schrauben befestigt. Da das Interface dann sehr dicht an der nicht isolierten Netzsicherung liegt, ist diese unbedingt zu isolieren !!!

Der Umbau der Spannungsversorgung ist oben beschrieben. Die beiden Kondensatoren sind „fliegend“ an der äußeren Lampenfassung angelötet. Dort ist auch der Anschluss der Stromversorgung des Interfaces.

Der I2C / USB Anschluss wird als Kabelanschluss durch die Öffnung unterhalb der Sicherung geführt. Als Buchse für I2C verwende ich eine Kabelbuchse für einen 3-poligen 3,5mm Klinkenstecker.

Anschlüsse

SL2, Power, I2C

1 schwarz	GND, siehe oben
2 braun	12V, siehe oben
3 rot	I2C GND
4 orange	I2C SCL
5 gelb	I2C SDA

SL3 Poti

1 schwarz	Poti - Spannung, mit Anschluss 4 verbinden
2 braun	Poti Abgriff, mit Anschluss 5 verbinden
3 rot	Poti + Spannung, mit Anschluss 6 verbinden
4 orange	Limit Anode des Kopplers, Kabel am Anschluss 7 (weiß/violett) auftrennen und SL3/orange mit Anschluss 7 verbinden

5 gelb Limit Kathode des Koppler, mit abgetrenntem Draht von Anschluss 7 (weiß / violett) verbinden

SL5 Motor

1 schwarz Mit Mittelanschluss des CW CCW Schalters verbinden
2 braun cw, Anschluss 3 (braun) abtrennen und mit braunen Draht verbinden
3 rot cw mit Anschluss 3 verbinden
4 orange ccw, Anschluss 2 (rot) abtrennen und (orangen Draht) mit Anschluss 2 verbinden
5 gelb ccw mit abgetrenntem Draht von Anschluss 2 (rot) verbinden

Versionen

Diese Beschreibung gilt für die
Leiterplattenversion 3.3
Bascom Version V05.1

Copyright

Die Ideen in diesem Dokument unterliegen der GPL (Gnu Public Licence V2) soweit keine früheren, anderen Rechte betroffen sind.

Die Verwendung der Unterlagen erfolgt auf eigene Geafahr; es wird keinerlei Garantie übernommen.

The ideas of this document can be used under GPL (Gnu Public License V2) as long as no earlier other rights are affected.

The usage of this document is on own risk, there is no warranty.

Referenzen

- [[1] dk1ri.de/dhw/rotorsteuerung_create_eagle.zip
- [2] dk1ri.de/dhw/rotorsteuerung_create.zip
- [3] dk1ri.de/myc/MYC.pdf
- [4] dk1ri.de/myc/Description.pdf (englisch)
- [5] dk1ri.de/dhw/rotorsteuerung_widerstandsberechnung.zip
- [6] dk1ri.de/myc/Definitions.pdf (englisch)