

CDE Rotor Controller

written by Glen Popiel - KW5GP

Ergaenzung Ralf DL1KRK, Michael DL5KBG , nutzt Programmteile von Erik Bartmann.

### **Anforderung für Antenne G07:**

Mast mit 4 Antennen je um 90° versetzt

Antennensteuerung zeigt alle 4 Richtungen als LED Kreis mit 40 zweifarb-leds an,

Die aktivierte Antenne wird rot dargestellt. Die anderen 3 Positionen bleiben grün der Rest der LED bleibt.

Aus Antennenumschaltung für die 4 Antennen ändert auch die Position der roten Lampe (aktive Antenne) Anzeige also abhängig von Ansteuerung durch Poti.

LED Bargraphanzeige: 1x Gleichspannung in Abhängigkeit der Antennenstellung des Rotors/Poti von 0 bis 5V .

Von Bargraph auf Dot hardwaremässig umgeschaltet. Dot mode results when pin 9 is pulled at least 200mV below V+ or left open circuit.

4 Stück LEDtreiber LM3914, jeder bedient 10LEDs, 2Farb, Die LEDs werden von einem Relais von rot nach grün umgeschaltet.

Es soll je Quadrant nur eine LED leuchten.

### **4x Schaltausgang für die Umschaltung Rot/Grün der jeweiligen Antenne**

Antennentaster: 4 Eingänge zur Antennenwahl, wenn nicht mehr genügend Pinne vorhanden sind geht aus ein analoger Eingang und die Taster in einer Widerstandmatrix

Im Display sollte die Richtung als Gradzahl passen zur ausgewählten Antenne stehen. Ebenso die Antennennummer.

Anzahl der genutzten PINS auf über 22 > Arduino Mega VERWENDUNG,

3 Relais steuern die HF an die aktive Antenne siehe Skizze

USB zum PC

LCD Display mit 2x18 oder besser 2x20 Zeichen, 2 Leitungen für I<sup>2</sup>C Bus notwendig

Antenne soll nur maximal 90° drehen, um schnelle QSOs zu ermöglichen

Ablaufsteuerung ändern/ ergänzen

von 360° Drehung auf 90° mit Mitte bei 180°, dh. 45° nach links 45 grad nach rechts

G07- Antennenrotor auch keine Bremse

\* Bei manueller Steuerung ist es sicher etwas umständlich wenn man den Rotor nur um 90° drehen kann,

\*zumal wir zur zeit noch unterschiedliche Antennengruppen benutzen.

Motor wird mit 2 Geschwindigkeiten über Spannungssteuerung angetrieben.

\*Das Poti hat keine 360°, es wird bei unseren Rotoren über zwei Zahnräder betätigt.

Rotorpoti zum Auslesen der Antennenstellung. 10Bit Auflösung reichen!

Relais/Optokoppler für Drehen im Uhrzeigersinn

Relais/Optokoppler für Drehen entgegengesetzt

Motor wird mit 2 Geschwindigkeiten über Spannungssteuerung angetrieben.

Antennenrotor besitzt ein 450° bzw 360° Potentiometer und keine Endlagenschalter

Rotorpoti zum Auslesen der Antennenstellung. 10Bit Auflösung reichen!

Relais/Optokoppler für Drehen im Uhrzeigersinn

Relais/Optokoppler für Drehen entgegengesetzt

Relais/Optokoppler für Geschwindigkeitsumschaltung

Antennenumschaltung Relais 1

Antennenumschaltung Relais 2 + 3!

=> Ant1: alle aus - Ant.2: 2/3 an + 1 aus - Ant3: 2/3 aus + 1 an - Ant4: alle an  
Hier ist noch eine Diskrepanz, Skizze von Michael wahrscheinlich richtig (klaus)  
Buchse PTT: zum Sperren der Antennenwahlschalter (NICHT des Rotors) beim Senden  
\*Das Drehen des Rotors beim Senden ist kein Problem. Bei betätigter PTT muss das Umschalten der Antennenrelais blockiert werden.  
Optokoppler statt einfache Ansteuerungstransistoren  
Taster Rotor: selbsterklärend

Fragen aus Programm von KW5GP:

warum ist der zusätzliche ADS1153 (?) Verwendet? 16Bit auflösung, Arduino bietet 8bit Auflösung für AD Wandlung, Entlastung des Programmablaufs?.

\* Den einzigen Vorteil des ADS sehe ich in der Auflösung, allerdings sind die 10bit des Arduino für uns wahrscheinlich ausreichend. Wenn wir den I<sup>2</sup>C-Bus auch nicht mehr brauchen (das Display kann auch direkt angeschlossen werden), halte ich es für eine größere Entlastung.  
Warum wird im Programm Poti kalibriert bei jedem Programmstart und speichert es in den Eeprom.  
Erklärungsversuche:

Kalibriert weil ggf. neuer Poti eingebaut ist.

Weil Poti nur geringe Genauigkeit +/-20% in der Linearität besitzt und dies zu ungenauer Ausrichtung führen würde. Kann man die gespeicherten Werte auslesen?

Poti hat zwar 360° Weg, aber nur 340° Kohlebelag, (aus: Datenblatt 360°Poti Conrad) im Programmbeschreibung im Buch berücksichtigt, aber im Programmablauf nicht gefunden...es vergessen?-weiter suchen

\*Durch die "Kalibrierung" blicke ich auch nicht ganz durch. Ich vermute der Spannungswert für 0 und 360° an den Endlagen wird ins Eeprom geschrieben.

\*Damit wird dann eine Wertetabelle für jedes Grad im Eeprom generiert, die dann auf der Anzeige in Grad dargestellt werden. Das damit die Unlinearitäten

\*des Potentiometers kompensiert glaube ich nicht. Dazu braucht man mehr als nur zwei Referenzpunkte.

\* Ein Ansatz wären vielleicht zwei Referenzfahrten, bei denen durch den Verlauf Zeit und Spannung die Winkel den Spannungen zugeordnet werden, die

\*man dann im Eeprom abspeichert. Das setzt aber einen gleichmäßig drehenden Rotor voraus.

Weil Poti nur geringe Genauigkeit +/-20% in der Linearität besitzt und dies zu ungenauer Ausrichtung führen würde. Kann man die gespeicherten Werte auslesen ?

Modellaufbau sinnvoll um die Fragen vorab zu klären

Beibehaltung der AD- Wandlung, um Arduino zu entlasten und die Eepromspeicherung zu behalten?

**Später berücksichtigen und weiteres Material:**

**in beiden Rotoren ist das gleiche Poti verbaut und drehen wahrscheinlich auch den gleichen Winkelbereich.**

**Ralf, 2019\_02\_22:**

Sie werden nur durch verschiedene Übersetzungen an den Drehbereich der Rotoren angepasst.

Zum Einmessen reicht es sicher ein Programmteil zu schreiben, der nur einmal durchlaufen wird und dabei die Endlagen und mehrere Winkelpositionen im Eeprom speichert.

Der dann in einen "SET-Modus" ausführen wird und durch eine bestimmte Tastenkombination beim Einschalten aktiviert wird. Das muss dann sicher nur nach einem Tausch des Potis wiederholt werden.

Mit dieser Methode sollte das Hauptprogramm für beide Rotoren gleich sein.

Es muss nur die Erkennung der Endlagen angepasst werden.

Bei dem großen die Abfrage der Endschalter und bei dem kleinen der Spannungswerte an den Endlagen.

Zur Sicherheit noch eine Interrupt-Routine dazu, die auslöst wenn sich bei laufendem Motor die Rotorstellung nicht ändert und den Motor abstellt.

Falls das Logprogramm eine Gradzahl überträgt, muss der Arduino entscheiden, welche Antenne geschaltet und wie weit gedreht werden soll. Dafür sollte er auch die Öffnungswinkel der Antennen kennen und berücksichtigen.

hier mal ein paar Infos zu den von uns eingesetzten Rotoren.

	2m und GHz	70cm
Rotor	Yaesu G-2700sdx	Yaesu G-400RC
Drehwinkel	450Grad	360Grad
Motor	DC	AC
Bremse	keine	keine
Endlagen	Schalter	fährt auf Block
Poti	500Ohm	500Ohm

Für 70cm müssen wir uns über kurz oder lang auch einen großen Rotor zulegen. Mit vier Antennengruppen ist es doch schon etwas grenzwertig.



