Parallaktische Montierung Ib von Carl Zeiss Jena

(gekauft im Jahre 1978 für 1450,00 DDR-Mark im Zeiss-Industrieladen in Berlin)

Umbau auf Schrittmotorsteuerung mit Goto

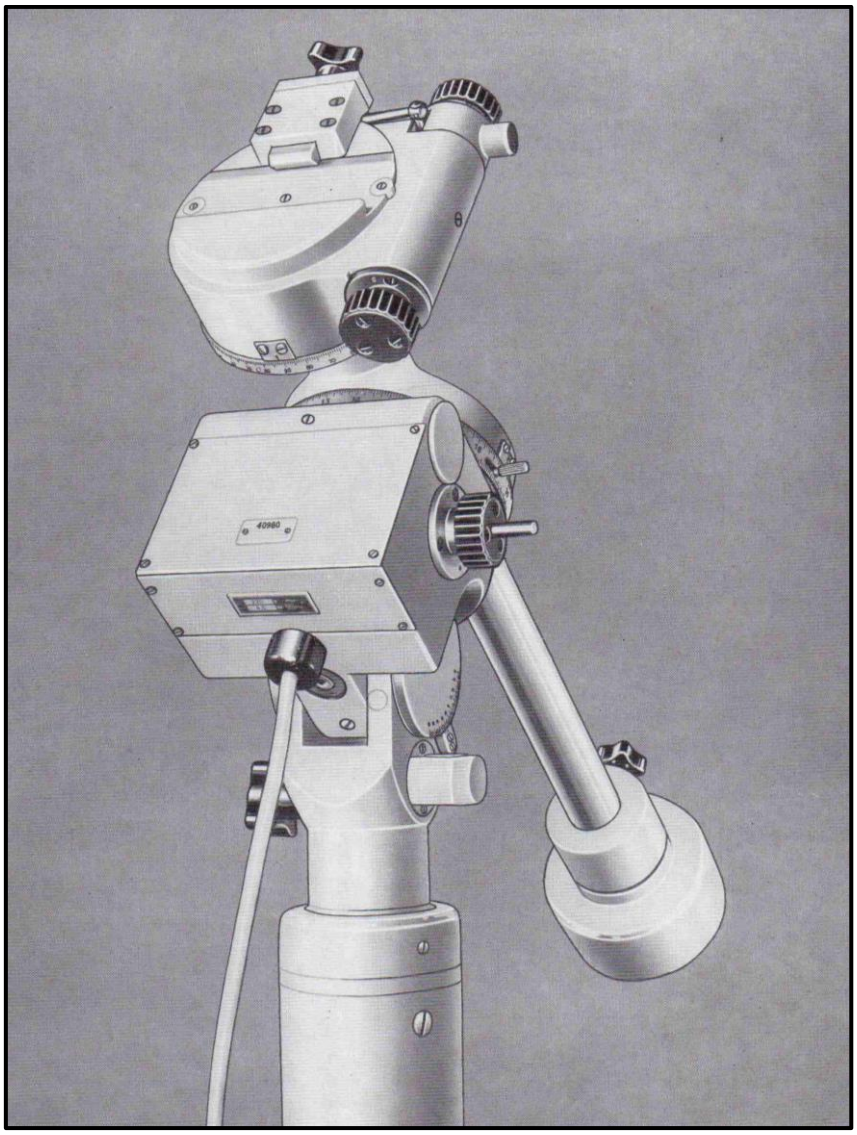


Bild vom Urzustand (Zeiss-Katalog)

Im Jahre 1960 wurde die Montierung Ib von Carl Zeiss Jena neu auf den Markt gebracht. Die Tragkraft beträgt 20kg und ist damit vergleichbar mit der der weitverbreiteten EQ6.

Allerdings ist nur ein einfacher Synchronmotorantrieb für Netzspannung (damals noch 220V) für die Rektaszensionsachse eingebaut, um die Himmelsdrehung auszugleichen. Die Deklinationsachse kann nur manuell betätigt werden.

Die Aufgabe bestand also darin, für diese robuste Montierung eine moderne Steuerung mit Antrieb in beiden Achsen zu entwickeln.

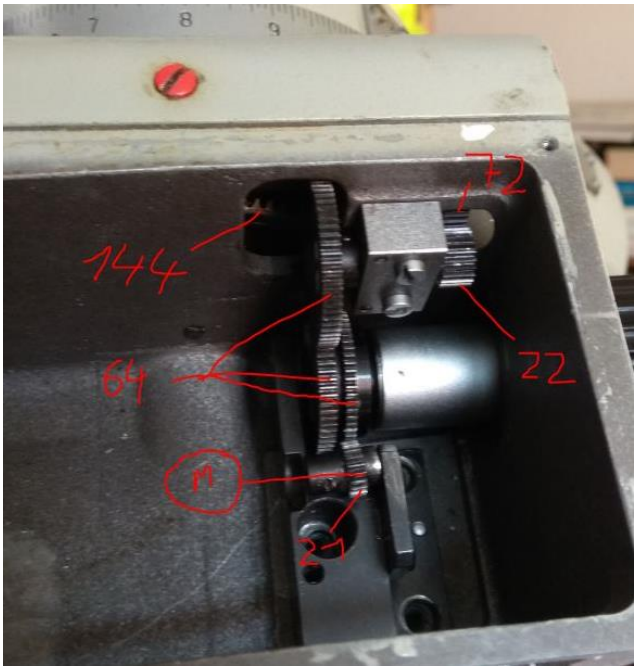
1. Analyse des verbauten Getriebes im Rektaszensionsantrieb

Um einen Schrittmotor in das Gehäuse und an das vorhandene Getriebe anzupassen, wurde der vorhanden Synchronmotor ausgebaut. Das kleine Ritzel des Motors und die Motoraufnahme sollen ohne Veränderungen weiterverwendet werden. Damit ist ein späterer Rückbau auf den Originalzustand möglich.

Auf der Motorwelle (M) sitzt ein kleines Ritzel mit 21 Zähnen. Danach folgen 3 Zahnräder mit z=64. Zwei davon bilden die Rutschkupplung, eines ist zur Drehrichtungsumkehr und hat am anderen Wellenende ein Ritzel mit z=22. Dieses treibt ein Zahnrad mit z=72, das auf der Schneckenwelle sitzt. Das Schneckenrad selbst hat 144 Zähne.

Die Gesamtuntersetzung beträgt also 21/64*22/72/144 = 1:1436,2597

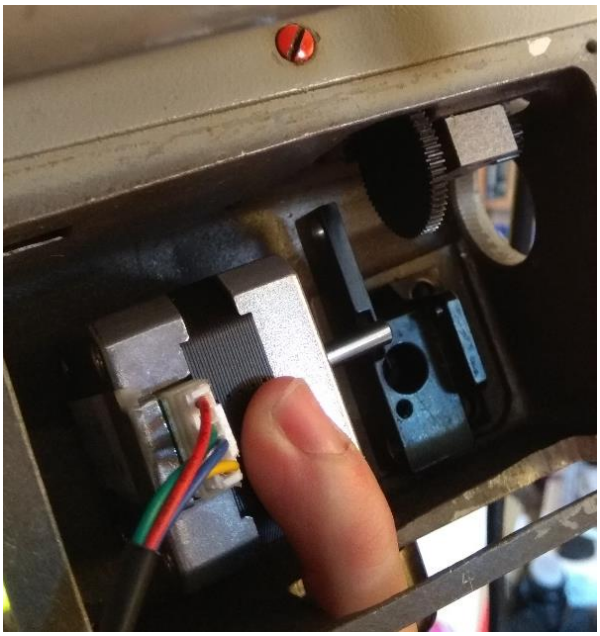
Für normale Nachführgeschwindigkeit muß sich die Motorwelle mit 1 U/min drehen.



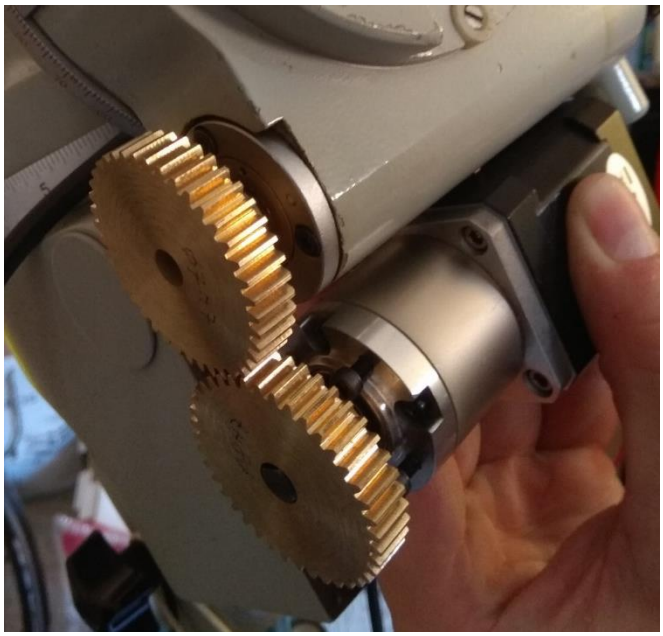
Für die Deklinationsachse gibt es kein Getriebe, dort ist das Handrad direkt auf der Schneckenachse montiert und das Schneckenrad hat wieder 144 Zähne. Die Untersetzung beträgt also 1:144.

2. Konstruktion der Zusatzteile für Motoreinbau in beiden Achsen

Da bereits Schrittmotore vorhanden waren, wurden im „Trockentest“ die Positionen an der bereits rückgebauten Montierung festgelegt. Für die Deklinationsachse wurde ein Planetengetriebe mit Untersetzung 1:13,73356 verwendet.



RA-Motor



DE-Motor

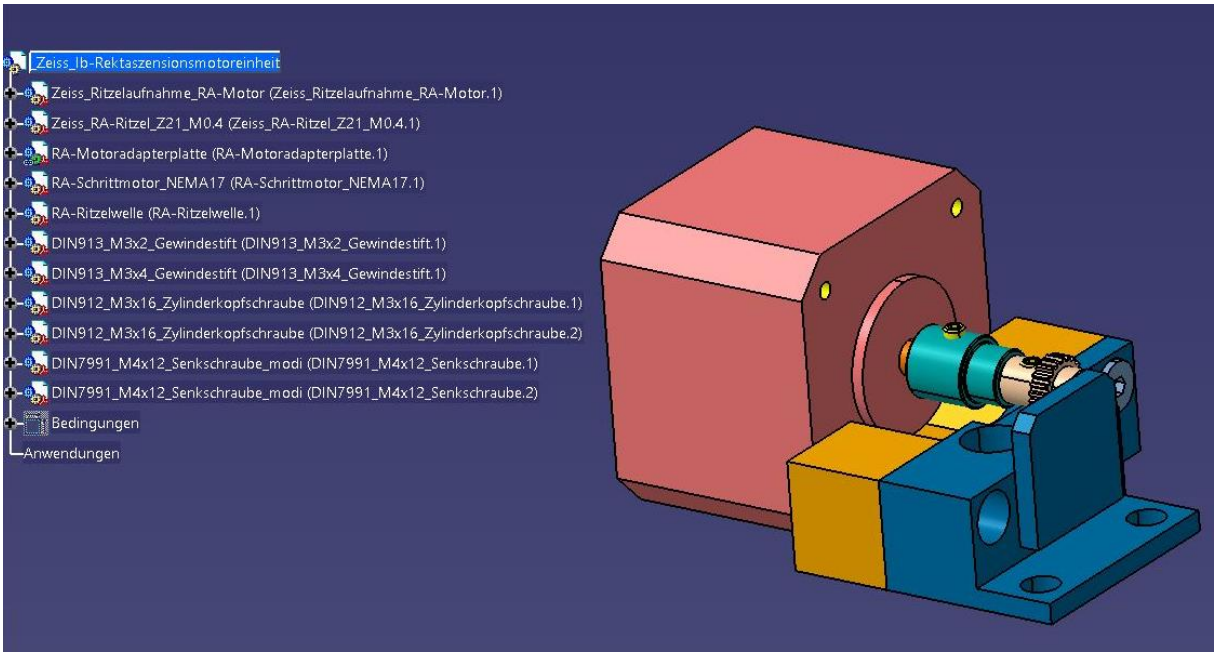
Folgende Teile werden gebraucht:

- RA-Adapterplatte, um einen NEMA17-Schrittmotor statt des Synchronmotors einzubauen
- RA-Ritzelwelle, um das Originalritzel mit z=22 auf die Motorwelle des NEMA-Motors zu befestigen
- DE-Übertragungszahnräder
- DE-Motoraufnahmeplatten
- div. Kleinteile (Schrauben, Muttern, Scheiben)

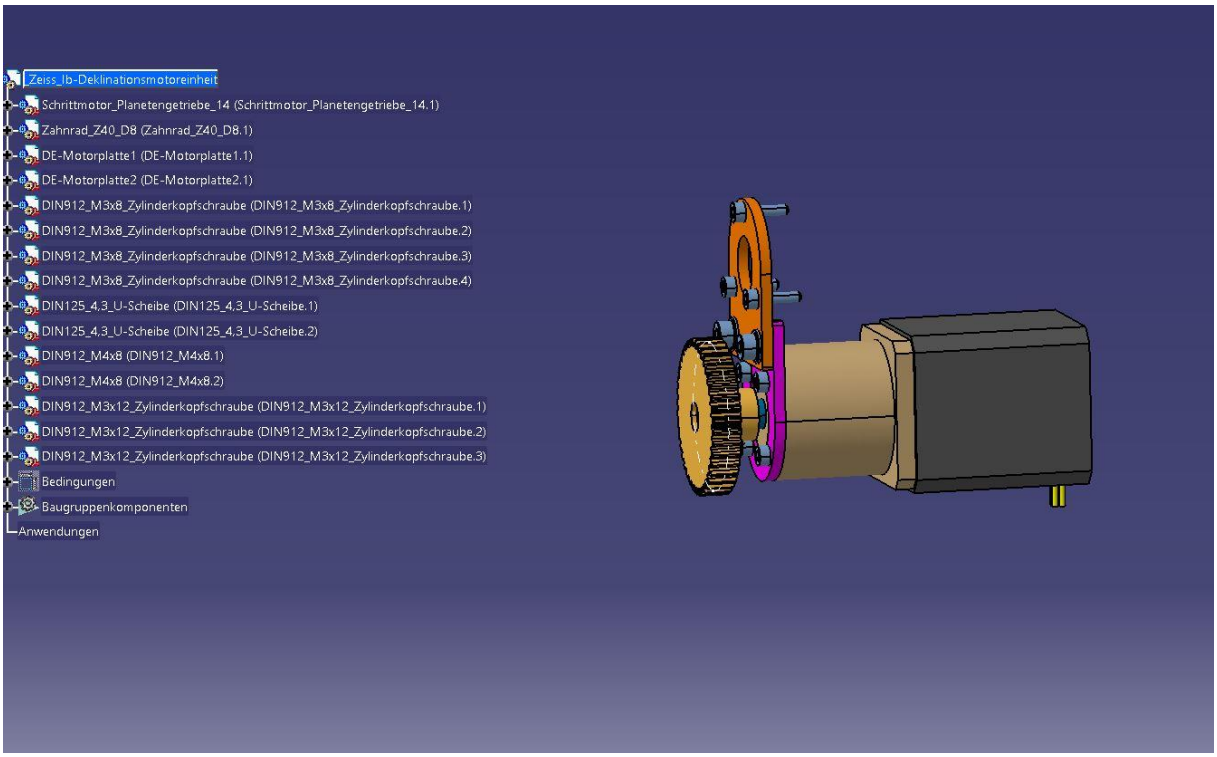
Konstruiert wurde mit der 3D-Konstruktionssoftware Catia V5.

Eine Mitarbeiter einer örtlichen Firma stellte diese Teile nach den gelieferten CAD-Daten her.

RA-Einheit

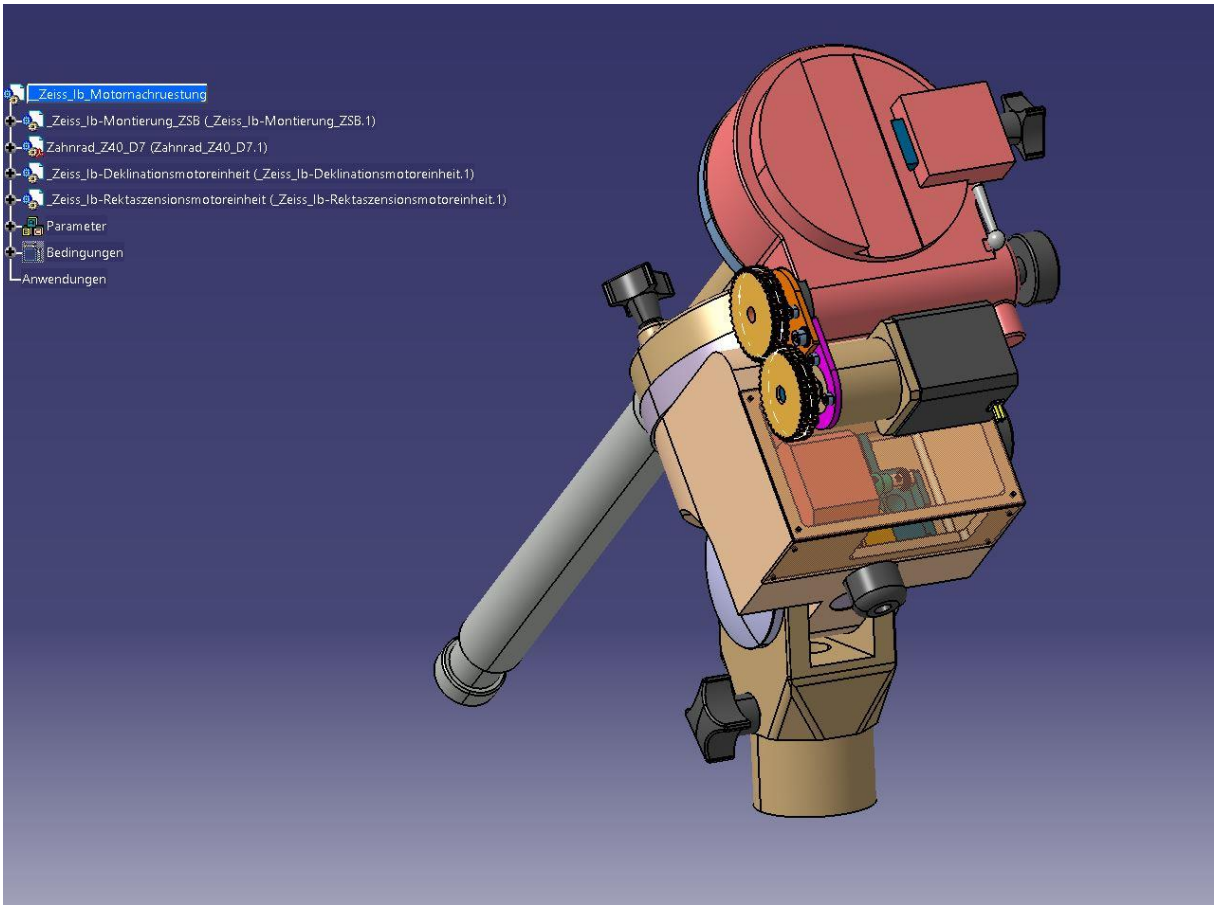


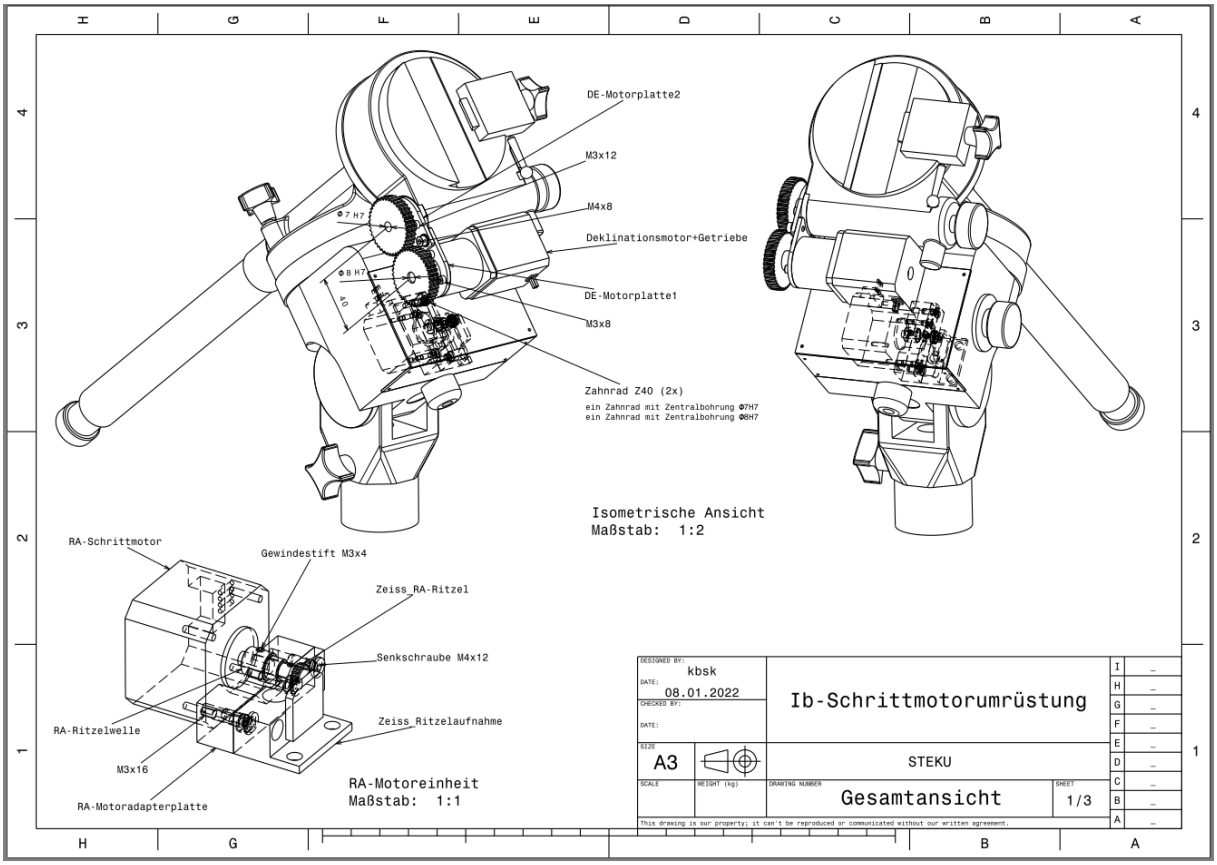
DE-Einheit



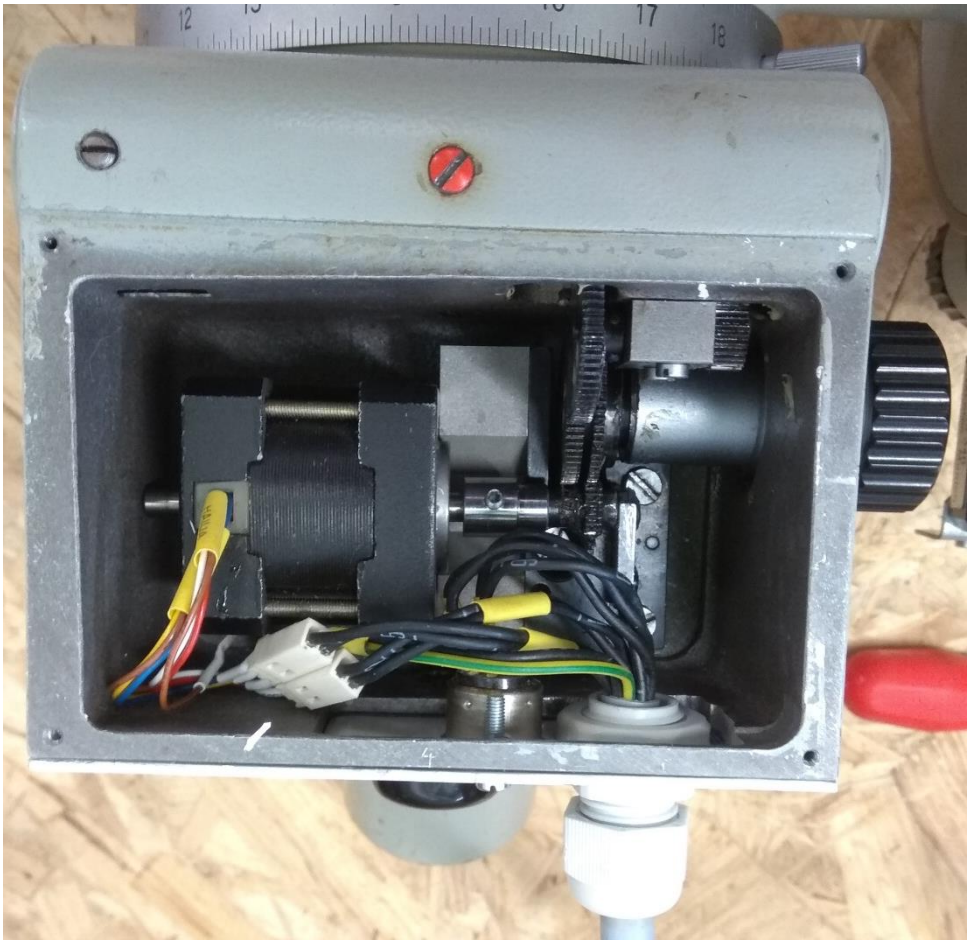
Es wurde darauf geachtet, daß auch der Mechanismus zum Auskuppel der Schnecke für eine schnelle manuelle Schwenkung weiterhin funktioniert.

Zusammenbau

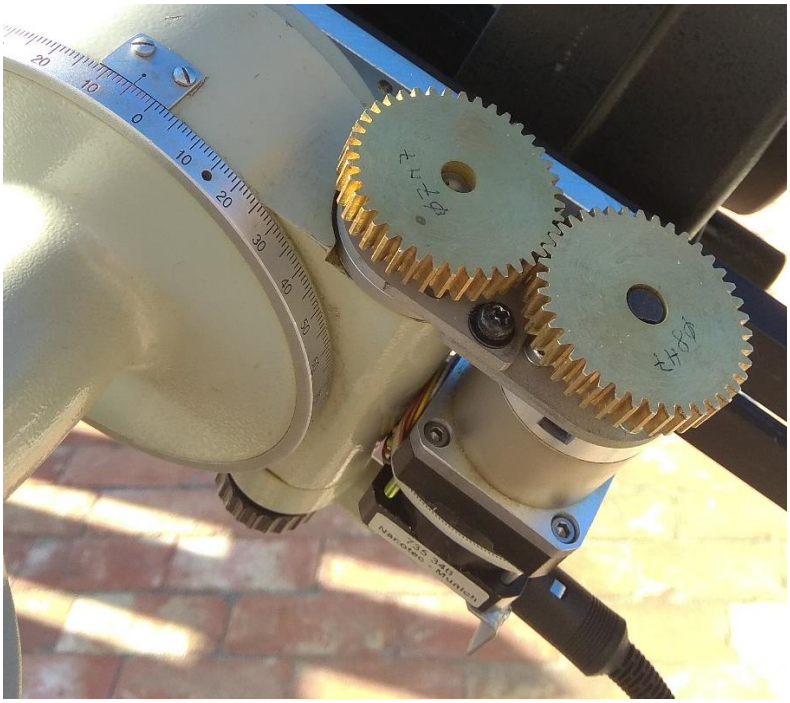




Blick ins das umgebaute Getriebegehäuse für den Rektaszensionsantrieb



Der Antrieb für die Deklinationsachse



3. Schrittmotorsteuerung

Die Steuerung sollte über folgende Möglichkeiten verfügen:

- vollautomatisches Anfahren eines Objektes
- verschiedene Nachführgeschwindigkeiten für Sterne, Sonne, Mond
- Handsteuerung mit verschiedenen Schwenkgeschwindigkeiten
- Anschlußmöglichkeit für einen Autoguider über ST4-Schnittstelle
- Steuerung über Software (Stellarium, Guide)
- Betrieb mit 12V, Stromverbrauch ca. 1 A
- als Option Anschluß eines Motorfokussieres
- als Option Referenzierung an mehreren Sternen, um Aufstellfehler zu berücksichtigen

Als Basis der Steuerung wurde sich an das OnStep-Projekt von Howard Dutton angelehnt. Unter https://onstep.groups.io/g/main/wiki/3860 findet man alle wichtigen Unterlagen dazu.

Dieses Projekt baut auf der Arduino-IDE auf und funktioniert mit verschiedenen Prozessoren und Motortreiber.

Gebaut wurden folgende Versionen:

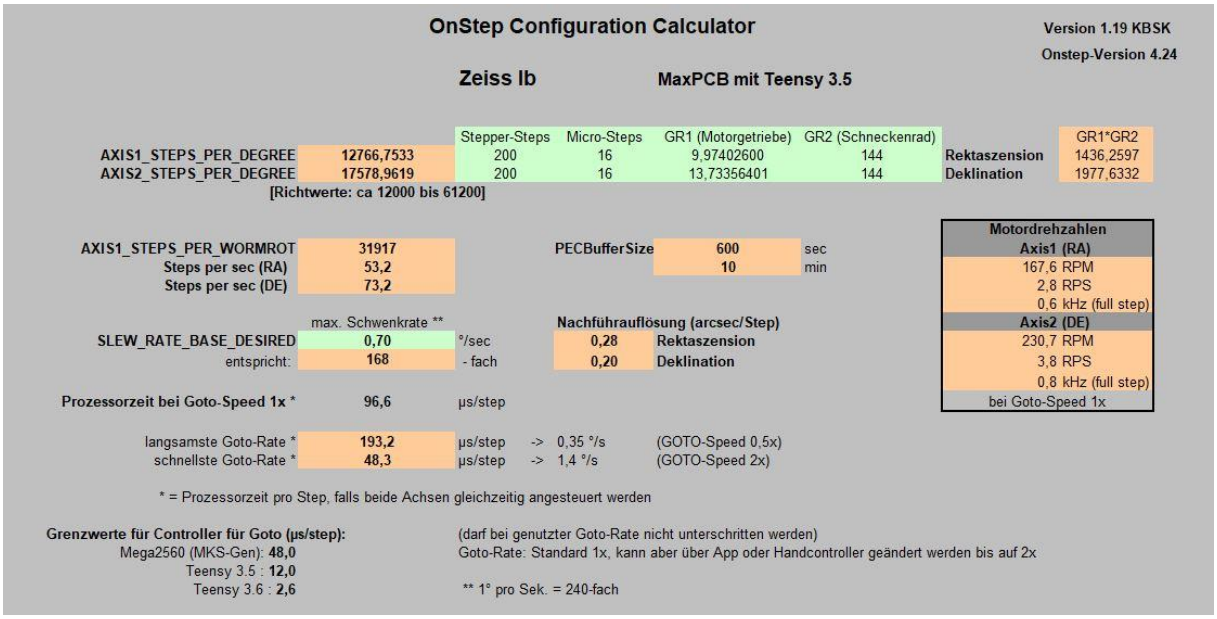
- MaxPCB mit Teensy 3.5 (zum selbst verlöten)
- MKS GenL V1 (fertiges 3D-Drucker-Board, nur Verbindungskabel im Selbstbau)

Folgende Zusatzkomponenten wurden eingesetzt:

- Bluetooth HC05
- Echtzeituhr DS3231 bzw. DS3234
- Handsteuerbox SHC mit Teensy 3.2
- Motortreiber TMC2130

Nach dem Zusammenbau der Komponenten wird die Software (OnStep 4.24) mit der Arduino-Programmierumgebung angepaßt und die Prozessoren damit programmiert.

Über Excel wurden folgende Werte ermittelt:



OnStep Configuration Calculator

Version 1.19 KBSK
Onstep-Version 4.24

Zeiss Ib — **MaxPCB mit Teensy 3.5**

		Stepper-Steps	Micro-Steps	GR1 (Motorgetriebe)	GR2 (Schneckenrad)		GR1*GR2
AXIS1_STEPS_PER_DEGREE	12766,7533	200	16	9,97402600	144	Rektaszension	1436,2597
AXIS2_STEPS_PER_DEGREE	17578,9619	200	16	13,73356401	144	Deklination	1977,6332

[Richtwerte: ca 12000 bis 61200]

AXIS1_STEPS_PER_WORMROT	31917	PECBufferSize	600	sec
Steps per sec (RA)	53,2		10	min
Steps per sec (DE)	73,2			

Motordrehzahlen	
Axis1 (RA)	
167,6	RPM
2,8	RPS
0,6	kHz (full step)
Axis2 (DE)	
230,7	RPM
3,8	RPS
0,8	kHz (full step)
bei Goto-Speed 1x	

	max. Schwenkrate **		Nachführauflösung (arcsec/Step)	
SLEW_RATE_BASE_DESIRED	0,70	°/sec	0,28	Rektaszension
entspricht:	168	- fach	0,20	Deklination

Prozessorzeit bei Goto-Speed 1x *	96,6	µs/step		
langsamste Goto-Rate *	193,2	µs/step	-> 0,35 °/s	(GOTO-Speed 0,5x)
schnellste Goto-Rate *	48,3	µs/step	-> 1,4 °/s	(GOTO-Speed 2x)

* = Prozessorzeit pro Step, falls beide Achsen gleichzeitig angesteuert werden

Grenzwerte für Controller für Goto (µs/step):
Mega2560 (MKS-Gen): **48,0**
Teensy 3.5 : **12,0**
Teensy 3.6 : **2,6**

(darf bei genutzter Goto-Rate nicht unterschritten werden)
Goto-Rate: Standard 1x, kann aber über App oder Handcontroller geändert werden bis auf 2x

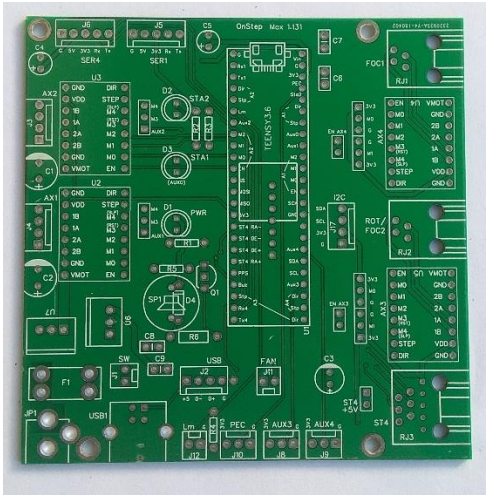
** 1° pro Sek. = 240-fach

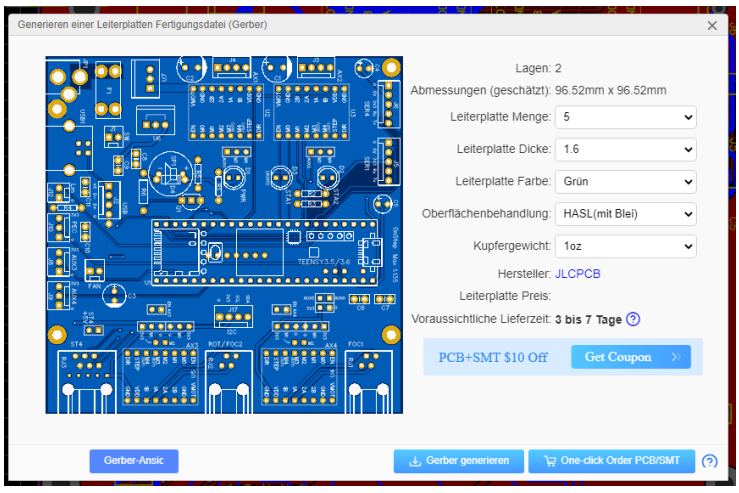
Auszug aus Software: die Motorparameter

```
// Stepper driver models (also see ~/OnStep/src/sd_drivers/Models.h for additional infrequently used models and more info.):
// A4988, DRV8825, LV8729, S109, SSS TMC2209*, TMC2130* **, and TMC5160* ***
// * = add _QUIET (stealthChop tracking) for example "TMC2130_QUIET"
// ** = SSS TMC2130 if you choose to set stepper driver current (in mA) set Vref pot. 2.5V instead of by motor current as us
// *** = SSS TMC5160 you must set stepper driver current (in mA) w/ #define AXISn_TMC_IRUN (IHOLD, etc.)

// AXIS1 RA/AZM
// see https://onstep.groups.io/g/main/wiki/6-Configuration#AXIS1
#define AXIS1_STEPS_PER_DEGREE    12766.75328 //  12800, n. Number of steps per degree:
                                              //         n = (stepper_steps * micro_steps * overall_gear_reduction)/360.0
#define AXIS1_STEPS_PER_WORMROT     31917 //  12800, n. Number of steps per worm rotation (PEC Eq mode only:)
                                          //         n = (AXIS1_STEPS_PER_DEGREE*360)/reduction_final_stage

#define AXIS1_DRIVER_MODEL            TMC2130_QUIET //    OFF, (See above.) Stepper driver model.
#define AXIS1_DRIVER_MICROSTEPS       16 //    OFF, n. Microstep mode when tracking.
#define AXIS1_DRIVER_MICROSTEPS_GOTO  8 //    OFF, n. Microstep mode used during gotos.
#define AXIS1_DRIVER_IHOLD            OFF //    OFF, n, (mA.) Current during standstill. OFF uses IRUN/2.0
#define AXIS1_DRIVER_IRUN             200 //    OFF, n, (mA.) Current during tracking, appropriate for stepper/driver/etc.
#define AXIS1_DRIVER_IGOTO            600 //    OFF, n, (mA.) Current during slews. OFF uses same as IRUN.
#define AXIS1_DRIVER_REVERSE          OFF //    OFF, ON Reverses movement direction, or reverse wiring instead to correct.
#define AXIS1_DRIVER_STATUS           TMC_SPI //    OFF, TMC_SPI, HIGH, or LOW.  Polling for driver status info/fault detect

#define AXIS1_LIMIT_MIN              -180 //   -180, n. Where n= -90..-270 (degrees.) Minimum "Hour Angle" for Eq modes.
                                          //         n. Where n=-180..-360 (degrees.) Minimum Azimuth for AltAzm mode.
#define AXIS1_LIMIT_MAX               180 //    180, n. Where n=  90.. 270 (degrees.) Maximum "Hour Angle" for Eq modes.
                                          //         n. Where n= 180.. 360 (degrees.) Maximum Azimuth for AltAzm mode.

// AXIS2 DEC/ALT
// see https://onstep.groups.io/g/main/wiki/6-Configuration#AXIS2
#define AXIS2_STEPS_PER_DEGREE    17578.9619328 //  12800, n. Number of steps per degree:
                                              //         n = (stepper_steps * micro_steps * overall_gear_reduction)/360.0

#define AXIS2_DRIVER_MODEL            TMC2130_QUIET //    OFF, (See above.) Stepper driver model.
#define AXIS2_DRIVER_MICROSTEPS       16 //    OFF, n. Microstep mode when tracking.
#define AXIS2_DRIVER_MICROSTEPS_GOTO  8 //    OFF, n. Microstep mode used during gotos.
#define AXIS2_DRIVER_IHOLD            OFF //    OFF, n, (mA.) Current during standstill. OFF uses IRUN/2.0
#define AXIS2_DRIVER_IRUN             200 //    OFF, n, (mA.) Current during tracking, appropriate for stepper/driver/etc.
#define AXIS2_DRIVER_IGOTO            600 //    OFF, n, (mA.) Current during slews. OFF uses same as IRUN.
#define AXIS2_DRIVER_POWER_DOWN       ON //    OFF, ON Powers off 10sec after movement stops or 10min after last<=1x guide.
#define AXIS2_DRIVER_REVERSE          OFF //    OFF, ON Reverses movement direction, or reverse wiring instead to correct.
#define AXIS2_DRIVER_STATUS           TMC_SPI //    OFF, TMC_SPI, HIGH, or LOW.  Polling for driver status info/fault detect
#define AXIS2_TANGENT_ARM             OFF //    OFF, ON +limit range below. Set cntr w/[Reset Home] Return cntr w/[Find Home

#define AXIS2_LIMIT_MIN               -90 //    -90, n. Where n=-90..0 (degrees.) Minimum allowed declination.
#define AXIS2_LIMIT_MAX                90 //     90, n. Where n=0..90 (degrees.) Maximum allowed declination.
```

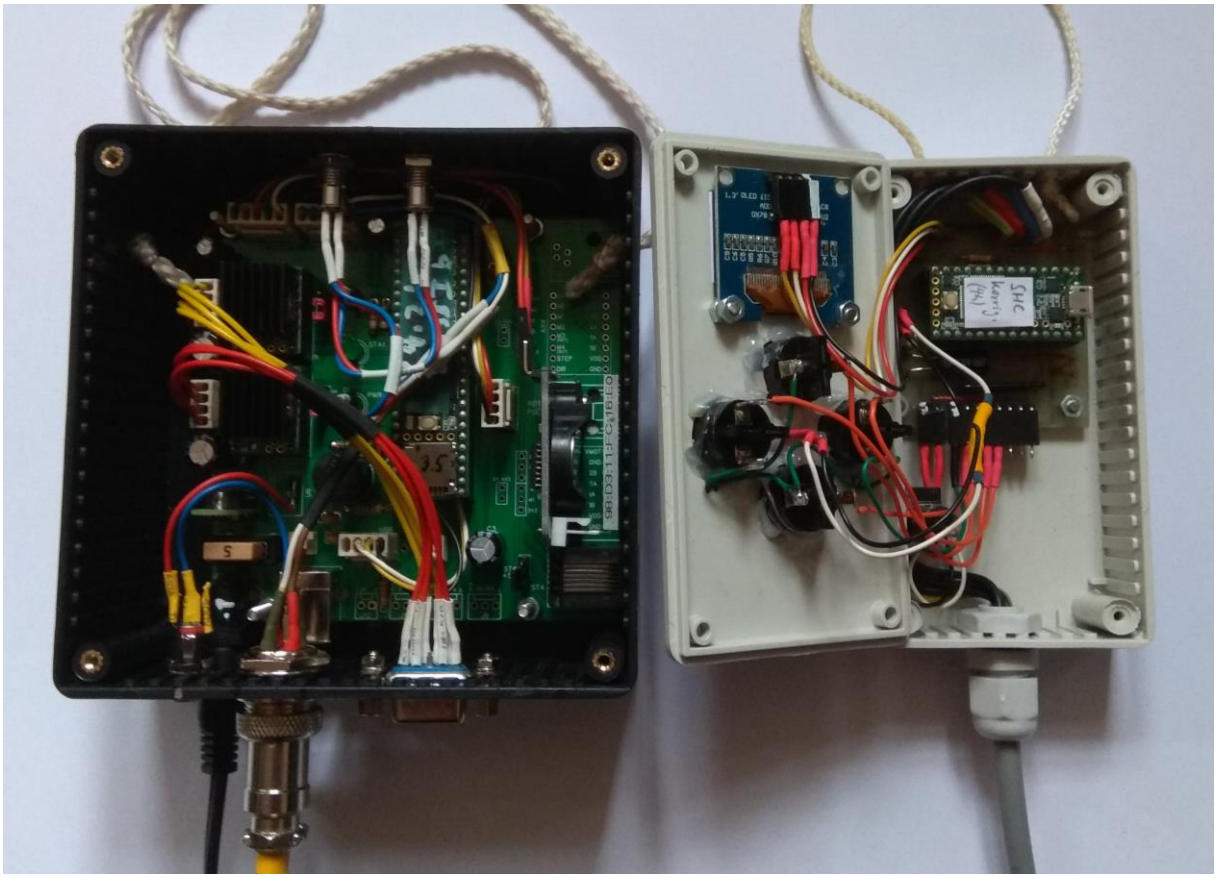
unbestückte Leiterplatte für MaxPCB-Version

(wurde bei einem Leiterplattenhersteller bestellt, Gerber-Daten als Grundlage)

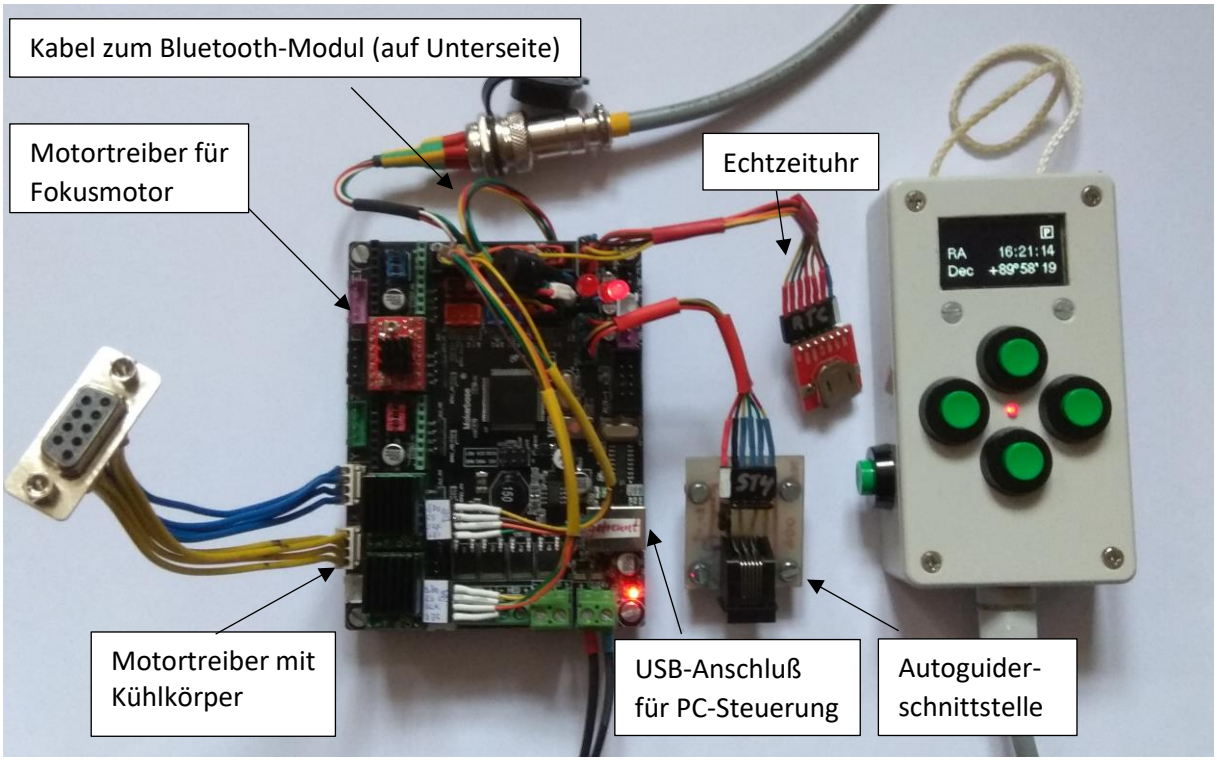




fertige MaxPCB-Einheit und universelle Handsteuerbox



MKS GenL V1-Version



4. Erprobung

Ein erster Test fand auf der Sternwarte Quedlinburg mit einem C8-Teleskop statt. Kurz vorher wurde eine Adapterplatte mit einer Zeiss-Prismenleiste angefertigt, um das C8 auf die Ib montieren zu können.

Es wurde die Montierung grob eingenordet und von der Fernrohrgrundstellung ausgehend (Fernrohr zeigt zum Himmelspol) über die Handsteuerbox der Befehl zum Schwenk auf die Sonne gegeben. Als das akustische Fertigsignal kam, wurde die Richtung kontrolliert und die ganze Montierung so um den Aufnahmezapfen gedreht, daß die Sonne im Gesichtsfeld erschien.

Dann wurde per Goto auf die Venus geschwenkt (die Venus befand sich ca. 45° westlich der Sonne und ist so hell, daß man sie auch am Tage beobachten kann). Falls das funktioniert, sind die richtigen Werte programmiert worden. Nach kurzer Zeit war es soweit: der Sonnenfilter wurde abgenommen und die Venus stand als kleine Sichel im Gesichtsfeld. Die Steuerung läuft perfekt.

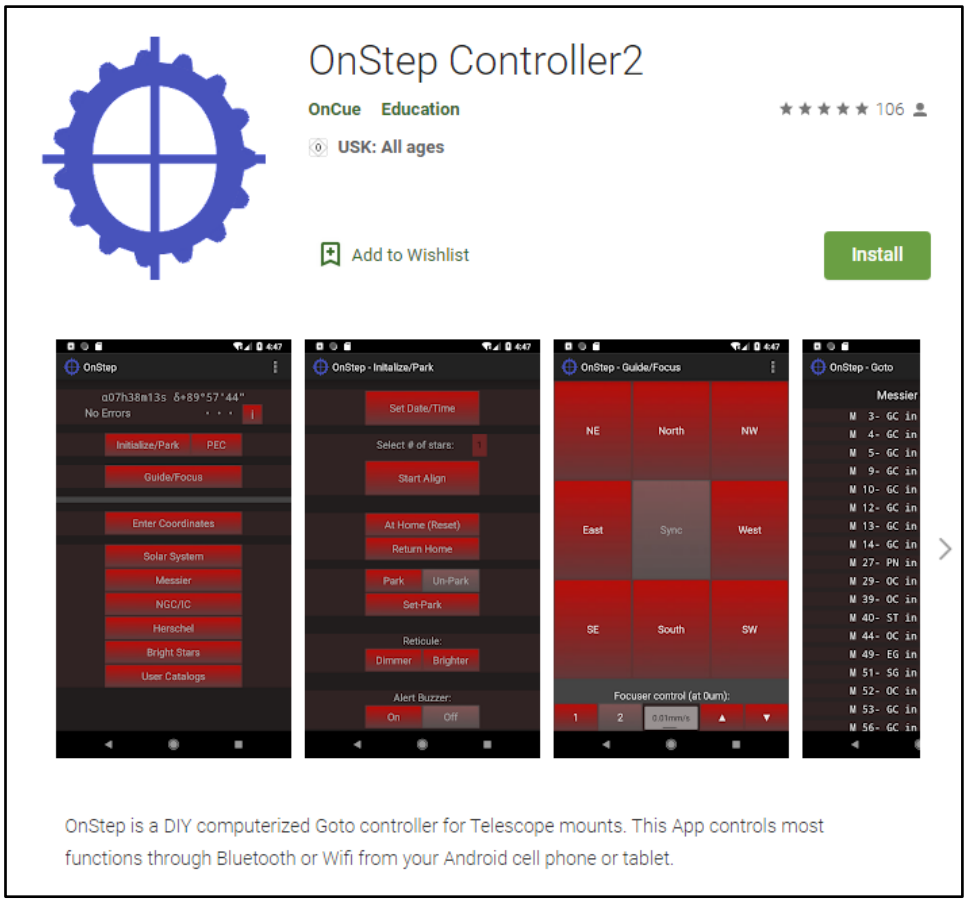
Als Zusatzfeatures kann die ganze Bedienung auch über ein Smartphone erfolgen, falls man keine Handsteuerbox zur Verfügung hat.

Dazu wurde die „OnStep Controller 2“-App von der Selbstbau-Community zur Verfügung gestellt. Diese App läuft nur unter Android.

Über das in der Steuerung eingebaute HC05-Modul läßt sich sehr leicht eine direkte Verbindung herstellen. Praktischerweise notiert man sich die MAC-Adresse des Moduls, um es schnell im Bluetooth-Gerätemanager zu finden. Einmal verbunden, merkt sich das die Software für die Zukunft.

Besonders nützlich ist die App bei der Erstinbetriebnahme. Sehr leicht läßt sich die Uhr stellen, die Beobachtungskoordinaten festlegen und die Schwenkgrenzen setzten. Außerdem sind noch mehr Objekte abgespeichert als in der Handsteuerbox.

Wer noch mehr Beobachtungsobjekte braucht (z.B. Kleinplaneten oder Kometen), kann per Bluetooth oder per USB-Kabel einen PC mit einem Planetariumsprogramm koppeln. Es muß nur das LX200-Protokoll unterstützen. Erfolgreich getestet wurde Guide und Stellarium.



Bildschirmfoto vom Google Play Store

5. Kostenaufstellung

Für die Steuerungselektronik incl. Handsteuerbox und Gehäuse muß man mit etwa 100 Euro Bauteilekosten rechnen.

Die beiden Schrittmotore, Zahnräder und das Getriebe für die Deklinationsachse schlagen etwa mit 60 Euro zu Buche.

Die Kosten der anzufertigenden Bauteile (Drehen, Fräsen) können nur geschätzt werden, da sie als Spende zur Verfügung gestellt wurden. Bis auf das kleine Drehteil für das Motorritzel und das Ausdrehen der Zahnräder könnte man diese aber auch selbst im Heimwerkerkeller herstellen.

Ein stabilisiertes Steckernetzteil für 12V DC/2A war vorhanden.

6. Danksagung

Für dieses Umbauprojekt gilt der besondere Dank unserem Vereinsmitglied Stefan Huskamp, der dieses Projekt vor einiger Zeit entdeckte. Er baut selbst eine solche Steuerung und hat dafür bei einem Leiterplattenhersteller 10 Stück MaxPCB bestellt und mir einige davon überlassen.

Die Anfertigung der mechanischen Teile übernahm mein Freund Wolfgang Kirschner. Auch ihm gilt mein besonderer Dank.

Wer Interesse am Umbau seiner eigenen Montierung bekommen hat, kann sich zwecks Unterstützung an mich wenden.

Es besteht jederzeit die Möglichkeit, die Montierung wieder in den Originalzustand mit dem 220V-Synchronmotor zu versetzten.

Stefan Kunz
Sternwarte Quedlinburg

07. März 2022