1. How the motor controller control the motor speed

- In the motor controller, there is a hardware timer T1 that is used to generate stepping pulse for stepper motor or reference position for servomotor. The input clock's frequency of the timer, plus the preset value of this timer, determine the slewing speed of the motors.
- When T1 generates an interrupt, it might
 - Drive the motor to move 1 step (1 micro-step or 1 encoder tick) for low speed slewing.
 - Drive the motor to move up to 32 steps for high speed slewing. This method applies to
 motor controller firmware version 2.xx. For motor controller with firmware 3.xx or above,
 the motor controller always drive the motor controller 1 steps/interrupt.

2. Two motion mode

- GOTO mode: The master device tells the motor controller the desired destination, and then send a "Start" command. The motor controller will control the motor to move to that destination. The master device can check the motor status, real-time position, cancel the slewing during the GOTO.
- Speed(Tracking) mode: The master device calculate a proper preset value for T1 and send it to the motor controller, and then send a "Start" command. The motor controller will control the motor to slew at the desired speed. The master device can check the motor status, real-time position, cancel the slewing during the GOTO.
- There is a command which is used to select between the two motion mode for the next "Start" command. Generally, the motor should be at full stop status before setting the motion mode.
- Generally, the motor controller returns to "Speed Mode" when the motor stops automatically.
- A typical slewing session include:
 - o Check whether the motor is in full stop status. If not, stop it.
 - o Set the motion mode.
 - Set the parameters, for example, destination or preset value of T1.
 - Set the "Start" command.
 - For a GOTO slewing, check the motor status to confirm that the motor stops (Generally means arriving the destination.). For a Speed mode slewing, send "Stop" command to end the session.

3. Calculation on Master Device

A Skywatcher motor controller does not do complex calculation. The master device do it instead.

Calculate the angle

A Skywatcher motor controller only counts the step or the ticks of an incremental encoder on the motor shaft. But a master device can inquire the motor controller the resolution of the telescope axis (how many steps the telescope axis have for one revolution). We called it CPR (Counts per revolution). With CPR, the master device can convert an angle to steps or vise versa.

Please note that CPR might be different for the two axes of a mount.

Calculate the T1 preset value.

A Skywatcher MC can report the T1's input clock frequency TMR_Freq (Mention at the beginning of this article). A master device can use TMR_Freq and CPR to calculate the T1 preset value for desired motor speed.

Calculate the T1 preset value for high speed slewing

T1 preset value can be too small for high speed slewing, if T1's input clock frequency is low. To solve this problem, the motor use a slightly different way to control motor speed when high speed slewing is required (For example, move an axis with higher then 128x sidereal rate). When T1 generates an interrupt, the motor controller moves N micro-steps for a stepper motor, or change the reference position for N steps for a DC servo motor. That means, for the same T1 preset value, the motor will run N times faster than changing only 1 steps for each T1 interrupt event.

Currently, N is a fixed number, and a master device can inquire the motor controller for it. It might be 16, 32 or 64.

The formula for calculating T1 preset value for high speed slewing is:

```
T1_Preset = N * TMR_Freq * 360 / Speed_DegPerSec / CPR
```

When a master wants an axis to slew at high speed, it should let the motor controller know when it configures the motor to the Speed (Tracking) Mode. For GOTO mode, the motor controller will take care of it automatically.

4. Command Format:

- The command always starts with a ":" character and ends with a carriage return character 0x0D.
- If a second ":" character is received by the motor controller before the carriage return character, then the motor controller will abandon the characters received and starts receiving a new command.
- Motor controller will process the command and send response after it receives the carriage return character.
- A response from the motor controller always starts with a "=" character and ends with a carriage return character, if the response is normal.

- If there is something wrong, the motor will response a message starts with a "!" character, followed by error code and a carriage return character.
- All the character in the command and the response are ASCII characters.
- A command from the master device has the following parts:
 - 1 byte Leading character: ":"
 - o 1 byte command word, check command set table for details
 - 1 byte channel word: "1" for RA/Az axis; "2" for Dec/Alt axis.
 - o 1 to 6 bytes of data, depending on command word: character "0" to "9", "A" to "F"
 - o 1 byte Ending character: carriage return character.
- A normal response from the motor controller has the following parts:
 - 1 byte Leading character: "="
 - o 1 to 6 bytes of data, depending on which command is processed: "0" to "9", "A" to "F"
 - \circ 1 byte Ending character: carriage return character.
- An abnormal response from the motor controller has the following parts:
 - 1 byte Leading character: "!"
 - o 2 bytes of error code: "0" to "9", "A" to "F"
 - o 1 byte Ending character: carriage return character.
- Data format:
 - \circ 24 bits Data Sample: for HEX number 0x123456, in the data segment of a command or response, it is sent/received in this order: "5" "6" "3" "4" "1" "2".
 - 16 bits Data Sample: For HEX number 0x1234, in the data segment of a command or response, it is sent/received in this order: "3" "4" "1" "2".
 - \circ 8 bits Data Sample: For HEX number 0x12, in the data segment of a command or response, it is sent/received in this order: "1" "2".

5. Command Set

Command	Start	Header	Channel	DB1	DB2	DB3	DB4	DB5	DB6	End	Response	Note
Set Position Initialization Done	:	E F	*1('3')	'0''F'	0'-'F'	'0'~'F'	'0'~'F	'0''F'	'0'-'F'	0x0D 0x0D	A,X A,X	Motor must be full stopped
IIIIIIIIIIIIIIIIIIIII	•		1(3)							UXUD	74,74	
Set Motion Mode *4	:	G	*1	B0: 0=Goto, 1=Tracking B1: 0=Slow, 1=Fast(T) 0=Fast, 1=Slow(G) B2: 0=S/F, 1=Medium B3: 1x Slow Goto	B0: 0=CW 1=CCW B1: 0=Noth 1=South B2: 0=Normal Goto					0x0D	A,X	Motor must be full stopped
Set Goto Target Increment	:	Н	*2	'0''F'	1=Coarse Goto	0-F	0~F	'0'-'F'	'0''F'	0x0D	AX	
Set Brake Point Increment		M	*1	0'-'F	0FF	02F	0~F	0'-F'	0-F	0x0D	A,X	
Set Goto Target	:	S	*1	0'-/F	0'~'F	0'~'F'	0'~'F'	0'~'F'	0'~'F'	0x0D	A,X	Motor must be full stopped
Set Step Period (T1 preset value)	:	I	*1	'0''F'	'0''F'	'0'~'F'	'0'~'F	'0'-'F'	'0~'F'	0x0D	A,X	Do not support changing Step Period (T1 preset vaile) when motor is slewing in high speed mode.
Set Long Goto Step Period		Т	*1	0-/F	0-F	0-F	0~F	0'-'F'	0-F	0x0D	A,X	
Set Brake Steps		J	*1	0'~'F'	0-F	0'-'F'	0''F'	0'-F'	0-F	0x0D	A.X	
Start Motion Stop Motion *4	:	K	*1							0x0D 0x0D	A,X A,X	
Instant Stop *4	:	L	*1							0x0D	A,X	
Set Sleep		В	*1	'0': WakeUp '1': Sleep							A,X	
Set Aux Switch On/Off	:	0	*1	0': Off '1': On '0'=1x						0x0D	A,X	
Set AutoGuide Speed		P	*1	'1'=0.75x '2'=0.5x '3'=0.25x '4'=0.125x						0x0D	A,X	
Run Bootloader Mode	:	Q	*1	'5'		'A'	'A'			0x0D	No response	
Set Polar Scope LED brightness		V	*1	0'~'F'	0'~'F					0x0D	A,X	<u> </u>
			*2							Owork	DV	
Inquire Counts Per Revolution Inquire Timer Interrupt Freq	:	a b	*2 '1'							0x0D 0x0D	B,X B,X	
Inquire Brake Steps		c	*2							0x0D	B,X	
Inquire Goto Target Position Inquire Step Period	:	h i	*2							0x0D 0x0D	B,X B,X	
Inquire Step Period Inquire Position	:	j	*2							0x0D	B,X	
Inquire Increment	:	k	*2	10''1' *3						0x0D	B,X	
Inquire Brake Point Inquire Status	:	f	*2						1	0x0D	E,X	
Inquire High Speed Ratio		g	*2							0x0D	D, X	
Inquire 1X Tracking Period Inquire Tele. Axis Position		D d	'1' *1							0x0D 0x0D	B,X B,X	
Inquire 1 ele. Axis Position Inquire Motor Board Version		e e	*1							0x0D 0x0D	B,X *6	
Inquire PEC period	:	S	*1							0x0D	B, X	
Set Debug Flag	-:-	Z	*1					-	-	0x0D	 	
Extended Setting Extended Inquire		W	*1	0'~'F' 0'~'F'	0'-'F 0'-'F	0'-'F 0'-'F	0'~'F' 0'~'F'	0'~'F' 0'~'F'	0'~'F' 0'~'F'	0x0D 0x0D	X X	
C. FERRON I.I.				OF STR	OL 1879	or are	or my			0010		
Set EEPROM Value	:	N	111	0~F	0~F	U~F	U~T			0x0D		
Inquire EEPROM Value		n	11'							0x0D		
Set Register Address		A	*1	0~F	0~F					0x0D		
Set Register Value	:	R	91	0''F	0-F					0x0D		
Inquire Register Value	:	ľ	*1							0x0D		
Response												
	-		_									
A B		-		'0'~'F	0-F	0-F	'0'~'F'	0'~'F'	'0~F	0x0D 0x0D		
A B C		ı		'0'~'F'	'0'-'F'	'0~F	'0'~'F'	'0'-'F'	'0~F	0x0D 0x0D		
A B C D				'0'~'F' '0'~'F' B0: 1=Tracking 0=Goto	'0'-F' '0'-F'	'0''F' B0: 0 = Not Init 1 = Init done		0''F'	Ψ-F	0x0D 0x0D 0x0D		
A B C C D C		-	*5	'0''F' '0''F' B0: 1=Tracking	0'-'F'	'0''F' B0: 0 = Not Init		0°-'F'	'0-'F	0x0D 0x0D		
E X		111	*5	'0''F' '0''F' B0: 1=Tracking 0=Goto B1: 1=CCW 0=CW B2: 1=Fast	'0'F' '0'F' B0: 1=Running, 0=Stopped B1: 1=Blocked,	'0''F' B0: 0 = Not Init 1 = Init done		0'-F'	Ψ-F	0x0D 0x0D 0x0D 0x0D		
E X	th	111	*5	'0''F' '0''F' B0: 1=Tracking 0=Goto B1: 1=CCW 0=CW B2: 1=Fast	'0'F' '0'F' B0: 1=Running, 0=Stopped B1: 1=Blocked,	'0''F' B0: 0 = Not Init 1 = Init done		'0'~'F'	'0-'F	0x0D 0x0D 0x0D 0x0D		
E X Note *1: '1'=CH1; '2'=CH2; '3'=Bo *2: 'T=CH1; '2'=CH2		111	*5	'0''F' '0''F' B0: 1=Tracking 0=Goto B1: 1=CCW 0=CW B2: 1=Fast	'0'F' '0'F' B0: 1=Running, 0=Stopped B1: 1=Blocked,	'0''F' B0: 0 = Not Init 1 = Init done		0F	у-'F	0x0D 0x0D 0x0D 0x0D		
E Note *1: '1-CH1; '2-CH2; '3-Bo *2: '1-CH1; '2-CH2, '3-Bo *2: '1-CH1; '2-CH2, '3-Bo	'=Yes	-	*5	0''F '0''F B0: !=Tracking 0=Gsss B1: !=CCW 0=CW B2: !=Fast 0=Slow	'0'F' '0'F' B0: 1=Running, 0=Stopped B1: 1=Blocked,	'0''F' B0: 0 = Not Init 1 = Init done		(0,-,k,	у-'F	0x0D 0x0D 0x0D 0x0D		
E X Note *1: '1'=CH1; '2'=CH2; '3'=Bo *2: 'T=CH1; '2'=CH2	'=Yes	-	*5	0''F '0''F B0: !=Tracking 0=Gsss B1: !=CCW 0=CW B2: !=Fast 0=Slow	'0'F' '0'F' B0: 1=Running, 0=Stopped B1: 1=Blocked,	'0''F' B0: 0 = Not Init 1 = Init done		(0,-,t,-		0x0D 0x0D 0x0D 0x0D		
E X Note *1:1"=CH1; 7"=CH2; '3"=Bo *2: 1"=CH1; 7"=CH2 *3". Reset Increment, 0"=No. 1" *4". Channel will always be set Unknown Command Unknown Command	'=Yes t to Tr	-	*5	0''F '0''F B0: !=Tracking 0=Gsss B1: !=CCW 0=CW B2: !=Fast 0=Slow	'0'F' '0'F' B0: 1=Running, 0=Stopped B1: 1=Blocked,	'0''F' B0: 0 = Not Init 1 = Init done		0-F	'0-F	0x0D 0x0D 0x0D 0x0D		
X Note *1:1"-CHI: 2"-CH2: '3"-B *2: "1"-CHI: 2"-CH2 *3: Reset Increment,"0"-No. '1" *4: Channel will always be se Unknown Command Command Length Error Moleron Stopped	'=Yes t to Tr	-	*5	0''F '0''F B0: !=Tracking 0=Gsss B1: !=CCW 0=CW B2: !=Fast 0=Slow	'0'F' '0'F' B0: 1=Running, 0=Stopped B1: 1=Blocked,	'0''F' B0: 0 = Not Init 1 = Init done		0-F	·g-·F	0x0D 0x0D 0x0D 0x0D		
E Note 11: 13-CH1; 7-CH2; 3'-Bo 21: 13-CH1; 7-CH2; 3'-Bo 22: 11-CH1; 7-CH2 45: Reset Increment, 10'-No, 1' 45: Channel will always be se 55: Error Code Unknown Command Command Length Error Motor not Stopped Invalid Character	'=Yes t to Tr	-	*5	0''F '0''F B0: !=Tracking 0=Gsss B1: !=CCW 0=CW B2: !=Fast 0=Slow	'0'F' '0'F' B0: 1=Running, 0=Stopped B1: 1=Blocked,	'0''F' B0: 0 = Not Init 1 = Init done		0E.	'0-F	0x0D 0x0D 0x0D 0x0D		
X Note *1: "-C-H; ?'-C-H2 ; '3'-H9 *2: "1-C-H; ?'-C-C-H2 *3: Reset Increment, W'-No., '1 *4: Channel will always be see "5: Error Code Ulskowen Command Length Error Motor not Stopped Invalid Character Not Initialized	'=Yes t to Tr	-	*5	0''F '0''F B0: !=Tracking 0=Gsss B1: !=CCW 0=CW B2: !=Fast 0=Slow	'0'F' '0'F' B0: 1=Running, 0=Stopped B1: 1=Blocked,	'0''F' B0: 0 = Not Init 1 = Init done		0	ψ-F	0x0D 0x0D 0x0D 0x0D		
E Note 11:11-CHI; 22-CHI; 23-CHI 23:11-CHI; 22-CHI 23: Reset Increment, W-No.; 14 44: Channel will always be se 55: Error Code Uknkown Command Command Length Error Motor not Stopped Invalid Character Not initialized Driver Sleeping	"=Yes t to Tr 0 1 2 3 4 5	-	*5	0''F '0''F B0: !=Tracking 0=Gsss B1: !=CCW 0=CW B2: !=Fast 0=Slow	'0'F' '0'F' B0: 1=Running, 0=Stopped B1: 1=Blocked,	'0''F' B0: 0 = Not Init 1 = Init done		V-F	W-F	0x0D 0x0D 0x0D 0x0D		
E Note Y. Note 11: 11-CH1; 22-CH2; 33-Bo 22: 11-CH1; 22-CH2 33: Reset Increment, U-No.; 1 34: Channel will always be se 55: Eare Langth Fror Moster not Stopped Invalid Character Not Initialized Driver Sleeping PEC Training is running	"=Yes t to Tr 0 1 2 3 4 5	-	*5	0''F '0''F B0: !=Tracking 0=Gsss B1: !=CCW 0=CW B2: !=Fast 0=Slow	'0'F' '0'F' B0: 1=Running, 0=Stopped B1: 1=Blocked,	'0''F' B0: 0 = Not Init 1 = Init done		V-F	ү-г	0x0D 0x0D 0x0D 0x0D		
E Note Y. Note 11: 1"-CH1; 2"-CH2; 3"-lis 22: 1"-CH1; 2"-CH2 33: Reset Increment, V-No.; 1 34: Channel will always be se 55: Error Code Unknown Command Command Length Frore Montalitation Driver Sleeping PET Training is running FET Training is running No Valid PET data *6: Motor Board Version	"=Yes t to Tr 0 1 2 3 4 5	- !	*5	V-P V-P B0 1-Tracking 0-Good B1:1-CCW 0-CW D2:1-Pins 0-Polion	90-F 90-F Bit 1-Raming 0-Supped 0-Supped 0-Normal	90-F Bot 0 = Not hit 1 = hind-one H1 1 = Level witch on		()E.	уF	0x0D 0x0D 0x0D 0x0D		
E X X X Yell (1) (2) (1) (2) (1) (2) (1) (2) (1) (2) (1) (2) (1) (2) (1) (2) (1) (2) (1) (2) (1) (2) (2) (2) (2) (2) (2) (2) (2) (2) (2	"=Yes t to Tr 0 1 2 3 4 5	- ! !	*5	0''F '0''F B0: !=Tracking 0=Gsss B1: !=CCW 0=CW B2: !=Fast 0=Slow	90-FP 90-FP B0: 1-Raming B1: 1-Blocked, 0-Normal	90-PF BO 0 - Not hat BO 0 - Not hat BI 1 = Level owitch on			у-F	0x0D 0x0D 0x0D 0x0D		
E Note Y. Note 11: 1"-CH1; 2"-CH2; 3"-lis 22: 1"-CH1; 2"-CH2 33: Reset Increment, V-No.; 1 34: Channel will always be se 55: Error Code Unknown Command Command Length Frore Montalitation Driver Sleeping PET Training is running FET Training is running No Valid PET data *6: Motor Board Version	"=Yes t to Tr 0 1 2 3 4 5	- !	*5	V-P V-P B0 1-Tracking 0-Good B1:1-CCW 0-CW D2:1-Pins 0-Polion	90-F 90-F Bit 1-Raming 0-Supped 0-Supped 0-Normal	90-PF BO 0 - Not hat BO 0 - Not hat BI 1 = Level owitch on		ΟΕ.	Ф-F	0x0D 0x0D 0x0D 0x0D		
E X X X Yell (1) (2) (1) (2) (1) (2) (1) (2) (1) (2) (1) (2) (1) (2) (1) (2) (1) (2) (1) (2) (1) (2) (2) (2) (2) (2) (2) (2) (2) (2) (2	"=Yes t to Tr 0 1 2 3 4 5	- ! !	*5	V-P V-P B0 1-Tracking 0-Good B1:1-CCW 0-CW D2:1-Pins 0-Polion	90-FP 90-FP B0: 1-Raming B1: 1-Blocked, 0-Normal	90-PF BO 0 - Not hat BO 0 - Not hat BI 1 = Level owitch on		/0E-	Q-F	0x0D 0x0D 0x0D 0x0D		
E Note 11: 11"-CH1; 2"-CH2; 3"-H8 21: 11"-CH1; 2"-CH2; 3"-H8 22: 11"-CH1; 2"-CH2 23: Reset Increment, 0"-No; 3" 24: Channel will always be see "S: Error Code Unknown Command Command Length Error Motor rot Stopped Invalid Character Not Initialized Driver Sleeping No Valid FEC data No Valid FEC data 42: AZ Extended Inquired	"=Yes t to Tr 0 1 2 3 4 5	- ! !	e de after st	V-P V-P B0 1-Tracking 0-Good B1:1-CCW 0-CW D2:1-Pins 0-Polion	90-F 90-F Bot 1-Raming Bit 1-Raming Bit 1-Blocked, 0-Normal	90-PF BO 0 - Not hat BO 0 - Not hat BI 1 = Level owitch on	₩.₽	Bhte3		0x0D 0x0D 0x0D 0x0D		
E Note Note 11: 19-CH1; 2-CH2; 39-B0 21: 19-CH1; 2-CH2; 39-B0 22: 19-CH1; 2-CH2 23: Reset incoment, U-No.7, 12 24: Channel vid always be se 25: E. Channel vid always be se 25: E. Urknown Command Command Length Error Motor not Stopped Invalid Character Not Initialized Driver Sleeping PEC Training is running No Valid PEC data No Valid PEC data AZ Extended Inquired AZ Extended Inquired AIS Extended Inquired Notes (19-10-10-10-10-10-10-10-10-10-10-10-10-10-	"=Yes t to Tr 0 1 2 3 4 5	! !	e de after st	W-F	00-F 00-F 100 1-Raming 100 1-Raming 101 101 101 101 101 101 101 101 101 101 101 101 101 101 101 101 101 101 101 101 101 101 101 101 101 101 101 101 101 101 101 101 101 101 101 101 101 10	90-PF BO 0 = Not Init BO 0 = Not Init BI: I = Level switch on II: I = Level switch on II: I = Level switch on II: I = Level switch on	₩.₽	Byte3		0x0D 0x0D 0x0D 0x0D 0x0D		
E Note 11: 11"-CH1; 2"-CH2; 3"-H8 21: 11"-CH1; 2"-CH2; 3"-H8 22: 11"-CH1; 2"-CH2 23: Reset Increment, 0"-No; 3" 24: Channel will always be see "S: Error Code Unknown Command Command Length Error Motor rot Stopped Invalid Character Not Initialized Driver Sleeping No Valid FEC data No Valid FEC data 42: AZ Extended Inquired	"=Yes t to Tr 0 1 2 3 4 5	! ! X X X	e de after st	W-F	10 - F 1	10 - F 1	₩-₽	Byte3	Byte4	0x0D 0x0D 0x0D 0x0D 0x0D		
E X X X X X X X X X X X X X	"=Yes t to Tr 0 1 2 3 4 5	! !	e de after st	W-F	10 - F 1	10-P	W-F	Byre3 '0' 'F'	Byte4 * 0*** p*	0x0D 0x0D 0x0D 0x0D 0x0D		
E X X X X X X X X X X X X X	"=Yes t to Tr 0 1 2 3 4 5	! !	e de after st	W-F	10 - F 1	10 - F 1	₩-₽	Byre3 '0' 'F'	Byte4 * 0*** p*	0x0D 0x0D 0x0D 0x0D 0x0D		
E X X X X X X X X X X X X X	"=Yes t to Tr 0 1 2 3 4 5	! !	e de after st	W-F	W-F W-F	10 - F 1	₩-₽	Byre3 '0' 'F'	Byte4 * 0*** p*	0x0D 0x0D 0x0D 0x0D 0x0D		
E Note Note 11: 19—CH1: 29—CH2: 39—Bo 11: 19—CH1: 29—CH2: 29—Bo 22: 19—CH1: 29—CH2: 29—Bo 23: 19—CH1: 29—CH2: 29—Bo 24: Channel will under No. 19—CH2: 29—CH2:	"=Yes t to Tr 0 1 2 3 4 5	-	e de after st	W-FP W-FP 00: I-Tracking 0-Code 10-Code 10-Cod	W-F W-F	10 - F 1	₩-₽	Byre3 '0' 'F'	Byte4 * 0*** p*	0x0D 0x0D 0x0D 0x0D 0x0D		
E X X X X X X X X X X X X X	"=Yes t to Tr 0 1 2 3 4 5	-	e de after st	W-F	W-F W-F	10 - F 1	₩-₽	Byre3 '0' 'F'	Byte4 * 0*** p*	0x0D 0x0D 0x0D 0x0D 0x0D		
E X Nate Nate Nate Tay "CCH1, 2"-CH2, 13"-the Tay "CCH1, 2"-CH2 "Seest Incernent,"Wab, 71 "48. Channel will always be se "58. Error Code Unknown Command Command Length Error Motor not Stopped Invald Character Not Intialized Driver Sleeping PEC Training is running PEC Training is running PEC Training is running AZ Extended Inquired Inquire Again (Original) Inquire Status EX Extended Setting Start PEC Training Start PEC Training Start PEC Training	"=Yes t to Tr 0 1 2 3 4 5	-	e de after st	W-FP W-FP 00: I-Tracking 0-Code 10-Code 10-Cod	W-F W-F	10 - F 1	₩-₽	Byre3 '0' 'F'	Byte4 * 0*** p*	0x0D 0x0D 0x0D 0x0D 0x0D		
E Note Y Note 11: 1"=CH1; 2"=CH2; 3"=60 71: 1"=CH1; 2"=CH2 72: 1"=CH1; 2"=CH2 73: Reset lnerment, 0"=No.; 1 74: Channel will always be se 75: Error Cook Command Command Command Command Command Command Command Command For Moter Board Version PFC Training in running No Valid PFC data No Valid PFC data For Moter Board Version AZ Extended Inquired Inquired Inquire Asta (Siviginat) Indexer Position Inquire Status EX Extended Setting Start PFC Training Start PFC Tracking Concert PFC Tracking Cancel PFC Tracking Cancel PFC Tracking Cancel PFC Tracking	"=Yes t to Tr 0 1 2 3 4 5	" ! " ! " ! " ! " ! " ! " ! " ! " ! " !	e de after st	W-FP W-FP 00: I-Tracking 0-Code 10-Code 10-Cod	W-F W-F	10 - F 1	₩-₽	Byre3 '0' 'F'	Byte4 * 0*** p*	0x0D 0x0D 0x0D 0x0D 0x0D		
E X X X X X X X X X X X X X	"=Yes t to Tr 0 1 2 3 4 5	-	e de after st	W-FP W-FP 00: I-Tracking 0-Code 10-Code 10-Cod	W-F W-F	10 - F 1	₩-₽	Byre3 '0' 'F'	Byte4 * 0*** p*	0x0D 0x0D 0x0D 0x0D 0x0D		
E X X X X X X X X X X X X X	"=Yes t to Tr 0 1 2 3 4 5	-	e de after st	W-FP W-FP 00: I-Tracking 0-Code 10-Code 10-Cod	W-F W-F	10 - F 1	₩-₽	Byre3 '0' 'F'	Byte4 * 0*** p*	0x0D 0x0D 0x0D 0x0D 0x0D		
E X X X X X X X X X X X X X	"=Yes t to Tr 0 1 2 3 4 5	-	e de after st	W-FP W-FP 00: I-Tracking 0-Code 10-Code 10-Cod	W-F W-F	10 - F 1	₩-₽	Byre3 '0' 'F'	Byte4 * 0*** p*	0x0D 0x0D 0x0D 0x0D 0x0D		
E X X X X X X X X X X X X X	"=Yes t to Tr 0 1 2 3 4 5	-	e de after st	W-FP W-FP 00: I-Tracking 0-Code 10-Code 10-Cod	W-F W-F	10 - F 1	₩-₽	Byre3 '0' 'F'	Byte4 * 0*** p*	0x0D 0x0D 0x0D 0x0D 0x0D		
E Note Note 11: 19-CH1: 29-CH2: 139-Bo 11: 19-CH1: 29-CH2: 139-Bo 12: 19-CH1: 29-CH2: 139-Bo 13: 19-CH1: 29-CH2: 139-Bo 14: 19-CH1: 29-CH2: 139-Bo 15: 19-CH1: 29-CH2: 139-Bo 16: 19-CH1: 29-CH2: 139-Bo 17: 19-CH1: 29-CH1: 29-CH2: 139-Bo 18: 19-CH1: 29-CH1:	"=Yes t to Tr 0 1 2 3 4 5	= 1 1 1	e de after st	W-FP W-FP 00: I-Tracking 0-Code 10-Code 10-Cod	W-F W-F	10 - F 1	₩-₽	Byre3 '0' 'F'	Byte4 * 0*** p*	0x0D 0x0D 0x0D 0x0D 0x0D		
E Note Note 11:19-CH1: 72-CH2: 73-Bo 21:19-CH1: 72-CH2: 73-Bo 22: 19-CH1: 72-CH2: 73-Bo 23: Reset Inccument, V3-No.; 1 24: Reset Inccument, V3-No.; 1 25: Enror Code 15: Enror Code 15: Enror Code 16: Enror Code 17: Enror Code 17: Enror Code 18: Enror Code	"=Yes t to Tr 0 1 2 3 4 5	" " " " " " " " " "	e de after st	W-FP W-FP 00: I-Tracking 0-Code 10-Code 10-Cod	W-F W-F	10 - F 1	₩-₽	Byre3 '0' 'F'	Byte4 * 0*** p*	0x0D 0x0D 0x0D 0x0D 0x0D		
E Note Note 11:19-CH1: 72-CH2: 73-Bo 21:19-CH1: 72-CH2: 73-Bo 22: 19-CH1: 72-CH2: 73-Bo 23: Reset Inccument, V3-No.; 1 24: Reset Inccument, V3-No.; 1 25: Enror Code 15: Enror Code 15: Enror Code 16: Enror Code 17: Enror Code 17: Enror Code 18: Enror Code	"=Yes t to Tr 0 1 2 3 4 5	" " " " " " " " " "	e de after st	W-FP W-FP 00: I-Tracking 0-Code 10-Code 10-Cod	10 - F 1	10 - F 1	₩-₽	Byre3 '0' 'F'	Byte4 * 0*** p*	0x0D 0x0D 0x0D 0x0D 0x0D		

Special Notes for data related to position: All the position data is offset by 0x800000. For example, axis position (in Counts) 0x000012 should be converted to 0x800012 when preparing the command; while the true position of data 0x801234 reported by the motor controller is 0x001234.

6. Hardware

- UART: 9600bps, 1 start bit, 1 stop bit, no parity check.
- Signal level: 5V or 3.3V.
- On most of the EQ mount, the TX and RX lines are separated. The motor controller will send its response immediately after it received and process the command.
- On most the Alt/Az mount, TX and RX lines are connected together, and there is another line(Drop) to indicate that the TX/RX bus is busy. The Drop line is controlled by the master only, which means the master device should pull the Drop line to low level when it starts to send a command and keep pulling it low until it receives the full response from the motor controller, or, a time-out occurs. The motor controller will send its response immediately after it received and process the command, thus the master device should release the TX/RX bus as soon as possible after the last bit of the command is shift out of the hardware register.
- The motor controller pull its TX line to high level with a 5.1K to 10K resistor, other than that, it
 does not strongly pull the TX line to high level and other devices can pull the TX line to low level
 without problem.

6. Wi-Fi Connection

The same protocol runs on the SynScan Wi-Fi dongle or mount with built-in Wi-Fi module.

- The Wi-Fi dongle/module runs a UDP server and listen to UDP port 11880 to accept commands from host.
- The command must be sent in a single UDP package; the response is also included in a single package.
- When the Wi-Fi dongle/module works in access point mount, its IP address is 192.168.4.1. If it runs in station mode, the router that it links to allocates its IP address.

6. Useful Resources

- Sample Code: https://code.google.com/archive/p/skywatcher/
- Documents: http://www.skywatcher.com/download/manual/application-development/