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:
    - Check whether the motor is in full stop status. If not, stop it.
    - 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.

  Speed_CountsPerSec = Speed_DegPerSec * CPR / 360

  T1_Preset = TMR_Freq / Speed_CountsPerSec

  = TMR_Freq * 360 / Speed_DegPerSec / CPR

- 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: ":"
  - 1 byte command word, check command set table for details
  - 1 byte channel word: "1" for RA/Az axis; "2" for Dec/Alt axis.
  - 1 to 6 bytes of data, depending on command word: character "0" to "9", "A" to "F"
  - 1 byte Ending character: carriage return character.
- A normal response from the motor controller has the following parts:
  - 1 byte Leading character: "="
  - 1 to 6 bytes of data, depending on which command is processed: "0" to "9", "A" to "F"
  - 1 byte Ending character: carriage return character.
- An abnormal response from the motor controller has the following parts:
  - 1 byte Leading character: "!"
  - 2 bytes of error code: "0" to "9", "A" to "F"
  - 1 byte Ending character: carriage return character.
- Data format:
  - 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".
  - 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	:	E	*1	'0'~'F'	'0'~'F'	'0'~'F'	'0'~'F'	'0'~'F'	'0'~'F'	0x0D	A,X	Motor must be full stopped
Initialization Done	:	F	*1('3')							0x0D	A,X	
Set Motion Mode *4	:	G	*1	B0: 0=Goto, 1=Tracking B1: 0=Slow, 1=Fast(T1 0=Fast, 1=Slow(G) B2: 0=S/T, 1=Medium B3: 1x Slow Goto	B0: 0=CW 1=CCW B1: 0=Noth 1=South B2: 0=Normal Goto 1=Coarse Goto					0x0D	A,X	Motor must be full stopped
Set Goto Target Increment	:	H		'0'~'F'	'0'~'F'	'0'~'F'	'0'~'F'	'0'~'F'	'0'~'F'	0x0D	A,X	
Set Step Period Increment	:	I		'0'~'F'	'0'~'F'	'0'~'F'	'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	:	T		'0'~'F'	'0'~'F'	'0'~'F'	'0'~'F'	'0'~'F'	'0'~'F'	0x0D	A,X	
Set Brake Steps	:	U		'0'~'F'	'0'~'F'	'0'~'F'	'0'~'F'	'0'~'F'	'0'~'F'	0x0D	A,X	
Start Motion	:	J	*1							0x0D	A,X	
Stop Motion *4	:	K	*1							0x0D	A,X	
Instant Stop *4	:	L	*1							0x0D	A,X	
Set Sleep	:	B	*	'0': WakeUp '1':Sleep						0x0D	A,X	
Set Aux Switch On/Off	:	O	*1	'0': Off '1': On						0x0D	A,X	
Set AutoGuide Speed	:	P	*1	'0'='1x '1'=0.75x '2'=0.5x '3'=0.25x '4'=0.125x						0x0D	A,X	
Run Bootloader Mode	:	Q	*1	'5'	'5'	'A'	'A'			0x0D	No response	
Set Polar Scope LED brightness	:	V	*1	0'~'F'	0'~'F'					0x0D	A,X	
Inquire Counts Per Revolution	:	a	*2							0x0D	B,X	
Inquire Timer Interrupt Freq	:	b	'1'							0x0D	B,X	
Inquire Brake Steps	:	c								0x0D	B,X	
Inquire Goto Target Position	:	h	*2							0x0D	B,X	
Inquire Step Period	:	i	*2							0x0D	B,X	
Inquire Position	:	j	*2							0x0D	B,X	
Inquire Increment	:	k	'0'	'0'~'1' *3						0x0D	B,X	
Inquire Status	:	f	*2							0x0D	E,X	
Inquire High Speed Ratio	:	g	*2							0x0D	D_X	
Inquire 1X Tracking Period	:	D	'1'							0x0D	B,X	
Inquire Tele. Axis Position	:	d	*1							0x0D	B,X	
Inquire Motor Board Version	:	e	*1							0x0D	B,X *6	
Inquire PEC period	:	s	*							0x0D	B,X	
Set Debug Flag	:	z								0x0D		
Extended Setting	:	W	*1	0'~'F'	0'~'F'	0'~'F'	0'~'F'	0'~'F'	0'~'F'	0x0D	X	
Extended Inquire	:	q	*1	0'~'F'	0'~'F'	0'~'F'	0'~'F'	0'~'F'	0'~'F'	0x0D	X	
Set EEPROM Address	:	M		0'~'F'	0'~'F'	0'~'F'	0'~'F'			0x0D		
Set EEPROM Value	:	N		0'~'F'	0'~'F'					0x0D		
Inquire EEPROM Value	:	n								0x0D		
Set Register Address	:	A	*	0'~'F'	0'~'F'					0x0D		
Set Register Value	:	B		0'~'F'	0'~'F'					0x0D		
Inquire Register Value	:	r								0x0D		
Response												
A		=								0x0D		
B		=		'0'~'F'	'0'~'F'	'0'~'F'	'0'~'F'	'0'~'F'	'0'~'F'	0x0D		
C		=		'0'~'F'	'0'~'F'	'0'~'F'	'0'~'F'			0x0D		
D		=		'0'~'F'	'0'~'F'					0x0D		
E		=		B0: 1=Tracking 0=Goto B1: 1=CCW 0=CW B2: 1=Fast 0=Slow	B0: 1=Running, 0=Stopped B1: 1=Blocked, 0=Normal	B0: 0 = Not Init 1 = Init done B1: 1 = Level switch on				0x0D		
X		!	*5							0x0D		
Note												
*1: '1'=CH1; '2'=CH2 ; '3'=Both												
*2: '1'=CH1; '2'=CH2												
*3: Reset Increment,'0'=No, '1'=Yes												
*4: Channel will always be set to Tracking Mode after stopped												
*5: Error Code												
Unknown Command	0											
Command Length Error	1											
Motor not Stopped	2											
Invalid Character	3											
Not Initialized	4											
Driver Sleeping	5											
PEC Training is running	7											
No Valid PEC data	8											
*6: Motor Board Version												
EQ	x	x	x	x	0	0						
AZ	x	x	x	x	0	1						
Extended Inquired		ID		Return	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5		
Inquire Axis(Original) Indexer Position		000000			'0'~'F'	'0'~'F'	'0'~'F'	'0'~'F'	'0'~'F'	'0'~'F'		
Inquire Status Ex		000001			'0'~'F' B0: PEC Training On/Off B1: PEC Tracking ON/Off	'0'~'F' B0: Support dual encoder B1: Support PPEC B2: Support original position indexer B5: Support EQ/AZ mode	'0'~'F' B0: Has polar scope LED B1: Ra axes must start seperately B2: Support tracking-torque selection	'0'~'F'	'0'~'F'	'0'~'F'		
Extended Setting		ID		return								
Start PEC Training		000000										
Cancel PEC Training		000001										
Start PEC Tracking		000002										
Cancel PEC Tracking		000003										
Enable Dual Encoder		000004										
Disable Dual Encoder		000005										
Enable/Disable full current (torque) driving at low speed		000006: Disable 000106: Enable										
Set Stride for Slewing		xxxx07										
Reset Axis Indexer Position		000008										
Write flash buffer in RAM to flash ROM		000009										

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/