SynScan App Protocol

1. Introduction

This document is for app developer (desktop or mobile) who wish to control Sky-Watcher Telescope mount (read position, perform GOTO) through the SynScan app. The protocol is named "SynScan App Protocol", or formerly, "SynScanMobile Command Set".

The feature set of this protocol has about the same coverage as ASCOM ITelescopeV3. SynScan app supports being controlled over other protocols, but they are not as complete as this protocol. The "Sky-Watcher Developer Interface Overview" document describes other ways to control Sky-Watcher Telescope mount.

Since the “SynScan App” ASCOM driver uses this protocol to control mount through SynScan app, this document also serves to describe the behavior of this driver.

2. Command-Response

In this protocol, client sends a command to server, and server replies with a response. Server never initiates communication with the client. All communication from server is in response to a command from the client. Server is designed provide a response as soon as the command is processed, so that unless communication channel fails, client should receive it a short time after sending command. Therefore, the server does not support the synchronous commands specified in ASCOM ITelescopeV3, such as SlewToCoordinates. Only the async variant, such as SlewToCoordinatesAsync, is supported.

3. Connecting to SynScan app

By default, SynScan app accepts command of this protocol sent to UDP or TCP port 11881 of the device (mobile or desktop) that SynScan app is running on. Since app 2.3.3, this port can be changed in app’s “Settings > Connect Settings” page.

The client must be able to reach server in the network. This means they must either run on the same device (using the loopback IP 127.0.0.1) or run on devices in the same network (e.g. joined the same Wi-Fi network). Note that two apps running on iOS cannot communicate with each other using the loopback device.

4. Command set

4.1. Overview

In this protocol, each command-response pair is designed as a one-to-one translation of an ITelescopeV3 method or property. Therefore the detail meaning of each pair is specified in ITelescopeV3 documentation[1]. Only the format of the translation is covered in this document.

4.2. Format of translation

Sending command: ITelescopeV3 definition → protocol command string → UDP/TCP bytes.
Receiving response: UDP/TCP bytes → protocol response string → ITelescopeV3 definition.

There is no marker or envelope added when translating protocol string to or from bytes. For example, the command string TrackingGet is ASCII encoded to the 11 bytes: 0x54 0x72 ... 0x65 0x74. If UDP is used, a UDP packet of only these 11 bytes would be sent. If TCP is used, then only these 11 bytes are sent to the socket. Similarly, there's no envelope when decoding response bytes from server.

4.3. Examples

Command example 1 – Read tracking status

ITelescopeV3 property signature (C#)

```
bool Tracking { get; set; }
```

Full command: 11 bytes: 0x54, 0x72, … 0x65, 0x74

```
TrackingGet
```

(Note the 'Get' postfix)

Full response: 16 bytes

```
Ok,TrackingGet,0
```

Command example 2 – Start slew to RA/Dec

ITelescopeV3 method signature (C#)

```
void SlewToCoordinatesAsync(
       double RightAscension,
       double Declination
)
```

Full command

```
SlewToCoordinatesAsync,12.45,45.89
```

Full response

[1] http://www.ascom-standards.org/Help/Developer/html/T_ASCOM_DeviceInterface_ITelescopeV3.htm

```
Ok,SlewToCoordinatesAsync
```

(Mount would start slewing to target)

4.4. Command and response detail

Command

Part Name	Part Definition
<command>	<command_name> <arg_list>
<command_name>	A string containing upper and lower case characters
<arg_list>	"" | "," <arg> <arg_list>
<arg>	<int> | <double> | <bool>
<int>	A base 10 integer. Eg "12", "-3"
<double>	A base 10 floating point. Radix is always '.', and there's no digit group separator. Eg "1.23", "-0.1"
<bool>	"0" or "1" meaning false or true respectively

- ITelescopeV3 **method** name translate directly to <command_name>.
  - For example <command_name> corresponding to AbortSlew() is "AbortSlew"
- ITelescopeV3 **property** name is appended by the appropriate "Get" or "Set" string.
  - For example <command_name> corresponding to setting TrackingRate property is "TrackingRateSet"
- Enum (named constants) defined in ASCOM are transmitted as its integer value. For example, TelescopeAxes.axisSecondary is transmitted as 1.

Response

Part Name	Part Definition
<response>	<response_status> "," <command_name> <arg_list>
<response_status>	See response status table
<command_name>	The command that this response is replying to.

Response status	Meaning
"Ok"	Command received and performed
"Error"	An error without any further information
"Unknown"	Unknown <command_name>
"Unimplemented"	The <command_name> is known but is not implemented
"InvalidOperation"	An error corresponding to InvalidOperationException in ASCOM
"InvalidValue"	An error corresponding to InvalidValueException in ASCOM

- Client should handle timeouts, which is where client sends a command to server but does not get a response within a certain time. It could take 800 milliseconds for server to respond in a configuration where communication between client and app and between app and mount are both happening over poor Wi-Fi.

4.5. Extension via ITelescopeV3 Action() method

See ITelescopeV3 definition of Action() and SupportedActions.

“SynScan App” ASCOM driver 1.3.2 or above support Action and SupportedActions.

ITelescopeV3 method call (C#)

```
void Action("ClearSyncTo", "")
```

Full command

```
Action,ClearSyncTo
```

Full response

```
Ok,Action,ClearSyncTo
```

- ClearSyncTo
  This command takes no arguments.
  Clear all sync samples, resetting all alignment data, including CH NP. Same effect as the “Reset Alignment” button in app’s “Alignment” menu.

4.6. Extension not via ITelescopeV3

These are additional commands not in ITelescopeV3.

- Get Azimuth and Altitude property in one command

```
AzimuthAltitudeGet

Ok,AzimuthAltitudeGet,21.4288328454703,8.93772862536713
```

- Get RightAscension and Declination property in one command

```
RightAscensionDeclinationGet

Ok,RightAscensionDeclinationGet,6.76711488657704,-16.7150023897724
```

- Get server (ie SynScan app) version

```
ServerVersion

Ok,ServerVersion,1,0,0
```

- There are additional commands for satellite tracking. This is covered in another document.

4.7. ITelescopeV3 members not applicable to this protocol

The following methods and properties of ITelescopeV3 are only applicable to ASCOM so are not implemented in the app server.

```
//For connecting ASCOM driver to telescope. In the case of ASCOM “SynScan App Driver”
driver, it is for connecting to or disconnecting from SynScan app.
bool Connected { get; set; }

//Configuring the ASCOM driver
public void SetupDialog()

//Managing ASCOM driver component
public void Dispose()

//These are for obtaining information about the ASCOM driver
public string Description { get; }
public string DriverInfo { get; }
public string DriverVersion { get; }
public short InterfaceVersion { get; }
public string Name { get; }

//For sending custom command through ASCOM driver
public void CommandBlind(string command, bool raw)
public bool CommandBool(string command, bool raw)
public string CommandString(string command, bool raw)
```