Command Sets Serviced by Sky-Watcher SynScan App

1. About command sets

SynScan app allows client application to command it to perform actions such as read telescope mount position and perform go-to. SynScan app accepts commands by two different command sets: SynScan communication protocol and SynScanMobile command set. SynScan Communication Protocol is the command set implemented by SynScan hand controller. SynScanMobile is designed specifically to match the ITelescopeV3 interface specified in ASCOM.

Figure 1 shows the various software, hardware, and protocol that may be involved in controlling a Sky-Watcher telescope mount.

1.1. Common notes for both command sets

Command and Response

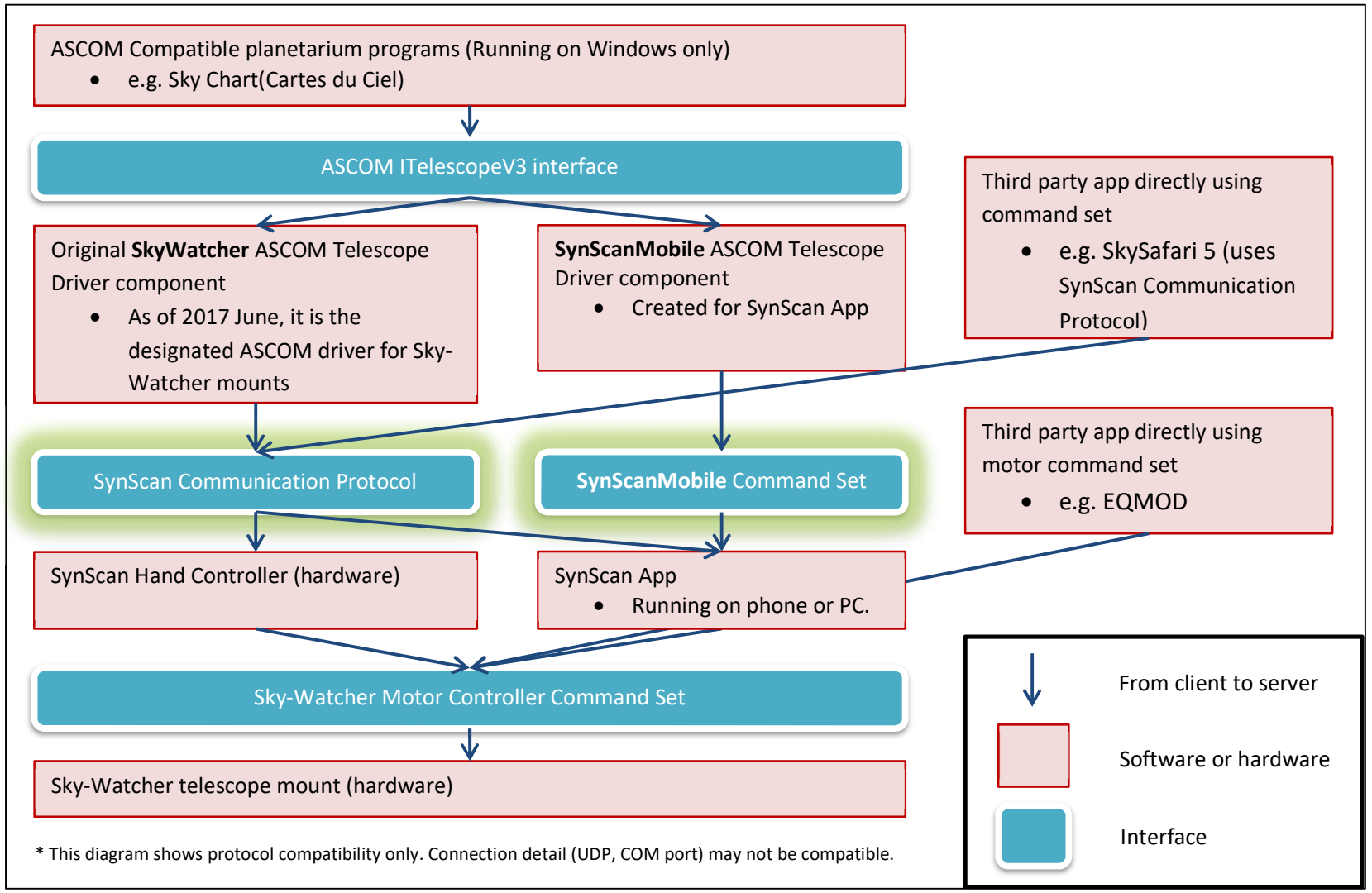
With either command sets, the client application (eg. SynScanMobile ASCOM driver) would send a command to SynScan app (server application), and SynScan app would reply with a response. SynScan app never initiates communication with the client, instead, all communication from SynScan app is in response to a command from the client. The difference between the two command sets is in the format of the command-response pairs and the connection details.

Network environment

SynScan app supports both command sets over the internet protocol. Specifically, it acts as TCP server for SynScan communication protocol and UDP server for SynScanMobile command set.

The client and SynScan app must be able to reach each other in the network. This means they must either run on the same device or run on devices in the same network (e.g. joined the same Wi-Fi network). Note that due to iOS's limitation, two apps running on iOS would have problem communicating with each other.

Figure 1 Interface Between Software/Hardware



2. SynScan communication protocol

2.1. Overview

This section only covers detail specific to SynScan app's implementation of SynScan communication protocol. The definition of the command-response pairs is in the "SynScan Serial Communication Protocol V3.3" document.

2.2. Connection detail with SynScan app

Command and response are exchanged over TCP. Once SynScan app has established connection between it and the telescope mount, it would act as a TCP server on port 11882 waiting for client app to connect and send command.

In the following example, the command and response are denoted by C language strings. For example, `\x06#` indicates 2 bytes: 0x06 0x23. The client would make a TCP connection to server's port 11882 and send the 1 byte command `m` over the socket. Server would then reply with the 2 byte response `\x06#` over that socket.

2.3. Examples

Command example 1 – Get telescope mount model

Full command: 1 byte

```
m
```

Full response: 2 byte

```
\x06#
```

Command example 2 – Slew Azm in negative direction

Full command: 8 byte

```
P\x02\x10\x25\x05\0\0\0
```

Full response: 1 byte

```
#
```

3. SynScanMobile command set

3.1. Overview

Each SynScanMobile 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.

3.2. Connection detail with SynScan app

Command and response are exchanged over UDP. Once SynScan app has established connection between it and the telescope mount, it would act as a UDP server on port 11881 waiting for client app to send command.

Command and response denoted in the next section are ASCII encoded strings without additional marker or envelope. For example, the command `TrackingGet` indicates 11 bytes: 0x54 0x72 ... 0x65 0x74. Client would send `TrackingGet` as the entire payload of a UDP packet to server's port 11881. Server would then respond with a UDP packet containing `Ok,TrackingGet,0`.

3.3. Examples

Command example 1 – Read tracking status

ITelescopeV3 property signature (C#)

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

Full command: 11 byte

```
TrackingGet
```

(Note the 'Get' postfix)

Full response: 16 byte

```
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
```

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

Full response

```
Ok,SlewToCoordinatesAsync
```

(Mount would start slewing to target)

3.4. 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. 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_name> in 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. In poor communication environment, it could be up to 800 millisecond for SynScan app to respond.

3.5. SynScanMobile 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 SynScan app version

```
ServerVersion

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

3.6. ITelescopeV3 members not applicable to SynScanMobile

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

```
//For connecting ASCOM driver to telescope. In the case of ASCOM SynScanMobile 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 ArrayList SupportedActions { get; }
public string Action(string actionName, string actionParameters)
public void CommandBlind(string command, bool raw)
public bool CommandBool(string command, bool raw)
public string CommandString(string command, bool raw)
```