# SpOT ATC Observatory User Guide

#### Shelbe Timothy, Aug 2018

## Overview

The SpOT ATC Observatory is located LMATC building 252 in Palo Alto, CA. All software is maintained in the 2D15 Lab. The automated data collection is comprised of the following Python object classes and scripts:

* Dome.py
* WeatherWatchdog.py
* [pyindi image sequencer and/or data pipeline]

During the observational time range (Astronomical Dusk - Astronomical Dawn), images are taken through ~~Ekos Scheduler via~~ an pyINDI interface.

## Devices

Ports/connections to devices

Davis Weather station USB device /dev/tty.SLAB\_USBtoUART

Optec IFW filter wheel USB device /dev/tty.usbserial-OP330GTQ

GPS usb USB device /dev/tty.usbserial

Boltwood Cloud Sensor II USB device /dev/tty.usbserial-AD0273ZI

Robofocus focus motor Serial-Ethernet 192.168.1.20:4661 (planned)

Astrohaven dome Serial-Ethernet 192.168.1.20:4660 (I think)

Apogee camera Serial-Ethernet 192.168.1.70

Paramount Serial-Ethernet? (planned)

## Software

The following section provide a brief description of the source and scripts used by the SpOT ATC Observatory.

### Dome

Dome is a python object Class that initializes a serial port connection and provides methods to communicate with the Astrohaven Dome

open(): opens Side B of Dome

close(): closes Side B of Dome

disconnect(): closes serial port connection

refresh(): refreshes serial port connection

get\_dome\_state(byte=True/False): returns state of the Dome as bytes or string.

**Bytes State String State**

b'0' 'Dome Closed'

b'1' 'B Side Ajar'

b'Y' 'B Side Closed'

b'y' 'B Side Open'

Note: Opening Side A of the Dome trips the circuit breaker. DO NOT USE.

### WeatherWatchdog

In order for the observatory to collect data without supervision, the health and safety of the equipment is the first priority. The WeatherWatchdog runs continuously, monitoring and logging real time weather conditions and sends open/close commands to the dome.

Dome Open conditions:

* Start Time ≤ Current Time ≥ Stop Time
  + Start Time: Astronomical Dusk (downloaded from Astral v1.6)
  + Stop Time: Astronomical Dawn (downloaded from Astral v1.6)
* Weather Status: Green
  + Humidity < 85%
  + Outside Temperature - Dew Point > 2°F (Similar to Humidity, Outside Temperature must remain 2°F above the Dewpoint)
  + Wind Speed (<10)

Dome Close conditions:

* Current Time outside observational time range (see above)
* Weather Status: Red – any triggered red limits will return the following log messages:
  + Humidity Red limit (> 85%)
  + Dew Point Red limit (<2°F)
  + Wind Speed Red limit (> 10 mph)

Note: Weather Status is logged 24/7 or as long as WeatherWatchdog is active. Dome commands are not sent outside the observational time range.

## Start Up

### Power ON Camera

In a web browser, open the iBoot url: 192.168.1.70.

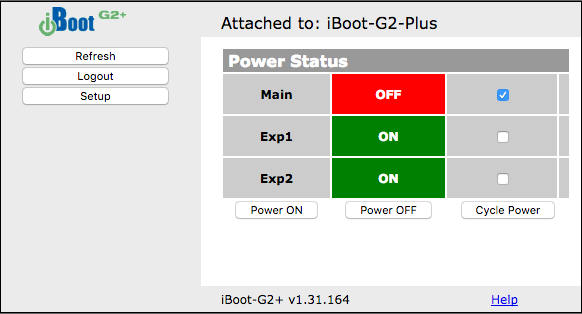


Figure 1 - Apogee CCD iBoot interface

Select “Main” box and and “Power ON”

### Start WeatherWatchdog (if not already running)

Open a new terminal window and use the following command:

>> WeatherWatchdog.py

### Start INDI

Open a new terminal window and use the following command:

>> indiserver -v indi\_apogee\_ccd indi\_gpsd indi\_simulator\_telescope indi\_optec\_wheel indi\_robo\_focus indi\_astrometry

(still required for pyINDI, testINDIserver.py)

### ~~Open KStars~~

~~Select Ekos Tab~~

~~~~

~~Figure 2 - KStars toolbar, Ekos highlighted in red~~

~~In Setup tab, select “SpOT Remote” profile and “Start INDI”~~

~~In Scheduler Tab, load image sequence/scheduler job~~

~~(ex. “sequences/SPOT\_main\_image job.esl”)~~

~~Select “Start Scheduler”~~

## ShutDown/Manual Commands

WeatherWatchdog.py is meant to run 24/7 with dome communication only during the observation time range. If the operator needs to interact with the dome manually, Dome\_startup.py and Dome\_shutdown.py can be called from a separate terminal window without interrupting the watchdog. Note that Dome\_startup.py still checks the weather status and will not open with a red status.

Alternatively, the dome can also be accessed through a python console:

>>> from spot.Dome import Dome

>>> dome = Dome()

>>> dome.open()

See Dome section above for full list of functions.