

# Automation of the FLWO 1.2m telescope: Report No. 1

Emilio E. Falco, FLWO

20 August 2010

## 1. Phase 1: RTS2 arrives

Petr Kubanek visited FLWO between 16 and 20 Aug 2010. He began integrating RTS2 with the telescope and control software. RTS2 is written in C++ and is thus portable, although there are significant variations from site to site. The following describes the steps that Petr followed to evolve from RTS2 to an initial version of RTS2-F.

1. On the first day, Petr installed a basic version of RTS2 on the 1.2m high-level Linux control computer, flwo48. Installation required compilation, installation of libraries, and testing. These steps took several hours, but were completed during the first day, culminating with a first version of RTS2-F.
2. On the second day, Petr began building into RTS2-F the capability to write scripts to control the telescope pointing and take exposures. These are the same scripts that are used normally to observe with Keplercam on the 1.2m. He also added the capability to obtain exposures with Keplercam. He tested the code by commanding RTS2-F to slew the telescope and take an exposure at the end of the slew. Petr also started work on the GUI (graphical user interface) that displays the status of observations, RTS2-G.
3. On the third day, Petr added code to RTS2-F to convert our catalog format to the RTS2-F database format. The database describes targets and includes the elements necessary to observe them as well as the observing logs. RTS2-F interacts with the database via SQL commands embedded in the C++ code. The database includes scripts describing how each target should be observed, in particular the filters, exposure lengths and number of repetitions.
4. On the fourth day, Petr modified RTS2-F to include refined constraints on exposures: observability, priority, airmass and moon distance. These constraints are entries in the database derived from target catalogs. RTS2-F populates the observing queue each night, using a merit function based on the constraints for each target. Eventually, RTS2-F will begin executing the queue at twilight.

Petr added the constraints one at a time and tested the code as he progressed. Petr also continued developing the capabilities of RTS2-G.

5. On the fifth day, Petr completed a first version of RTS2-G that includes display of exposures in a thumbnail mode. The thumbnails also allow full-size views of individual exposures by clicking on the corresponding thumbnail. During a night of observations, the thumbnails would be available for viewing through a web browser. At this final stage of his visit, Petr was able to exercise a fully-functional first version of RTS2-F. Petr also contributed to this progress report and wrote a first draft of a manual.

## **2. Phase 1: Next steps**

We expect to complete Phase 1 by the end of September 2010.

As we only plan to cool down Keplercam just before the start of the trimester of observations, we will exercise RTS2-F during daytime until September 1st.

The next additions to RTS2-F include enabling of telescope and dome tracking, and amp selection. Petr will work on these remotely. We are planning another visit in October 2010.

During the trimester, we will use several engineering nights for tests. We will run RTS2-F using catalogs of available targets. We will monitor the operations, and simulate various conditions, such as GRB warnings and bad weather.

During the rest of Phase 1, Petr will add to RTS2-F the capability to acquire a standard set of CCD biases and dome flats, as well as other calibrations such as sky flats and photometric standards.