USING BLACS

Contents

Version 1.0, SPJ 5 June 2013

Introduction

BLACS provides an interface for controlling hardware in the lab. It has two main modes of operation, the manual control of hardware and the coordination of hardware timed experiment execution in buffered mode.

Initial setup

Before using BLACS, you must have a lab connection table. This is the same as the connection table for an experiment, as described in the "Using the labscript API" document, but describes all devices in use by your control computer, not just the ones used in a particular experiment. This connection table should be generated by writing a minimalistic experiment file, containing only the connection table section, the `start()` instruction and calling `stop(1)`, where 1 is just any time greater than the shortest time interval between two instructions supported by your pseudoclock(s). This file can be compiled using runmanager (see "Using runmanager" document) and then re-named and placed in the location specified by your labconfig file. The lab connection table H5 file should be in a location accessible by both BLACS and BIAS (i.e. a shared drive if they are on separate computers).

Upon starting BLACS for the first time, you should enter the file paths for any global variables that may be required to compile the lab connection table by clicking "Select Globals" in the "Connection Table" drop down menu. Click "Add global file" and navigate to the location where the h5 file created in `runmanager` was saved. Calibration scripts, used to provide alternative "real world" units to the output of devices may also be loaded from this dialog box. (see Figure 1)

In the "General" tab of the preferences menu you can enter the path to your favourite text editor to allow BLACS to launch it for editing the connection table.

Future updates to the lab connection table can be made by selecting "Edit" in the connection table drop down menu in BLACS. BLACS will detect when the connection table python file has been

changed, and prompt the user to re-compile it (see Figure 4), alternatively you can click “Recompile and Restart” in the drop down menu to recompile the connection table. Upon recompilation, BLACS must be restarted for the changes to take effect.

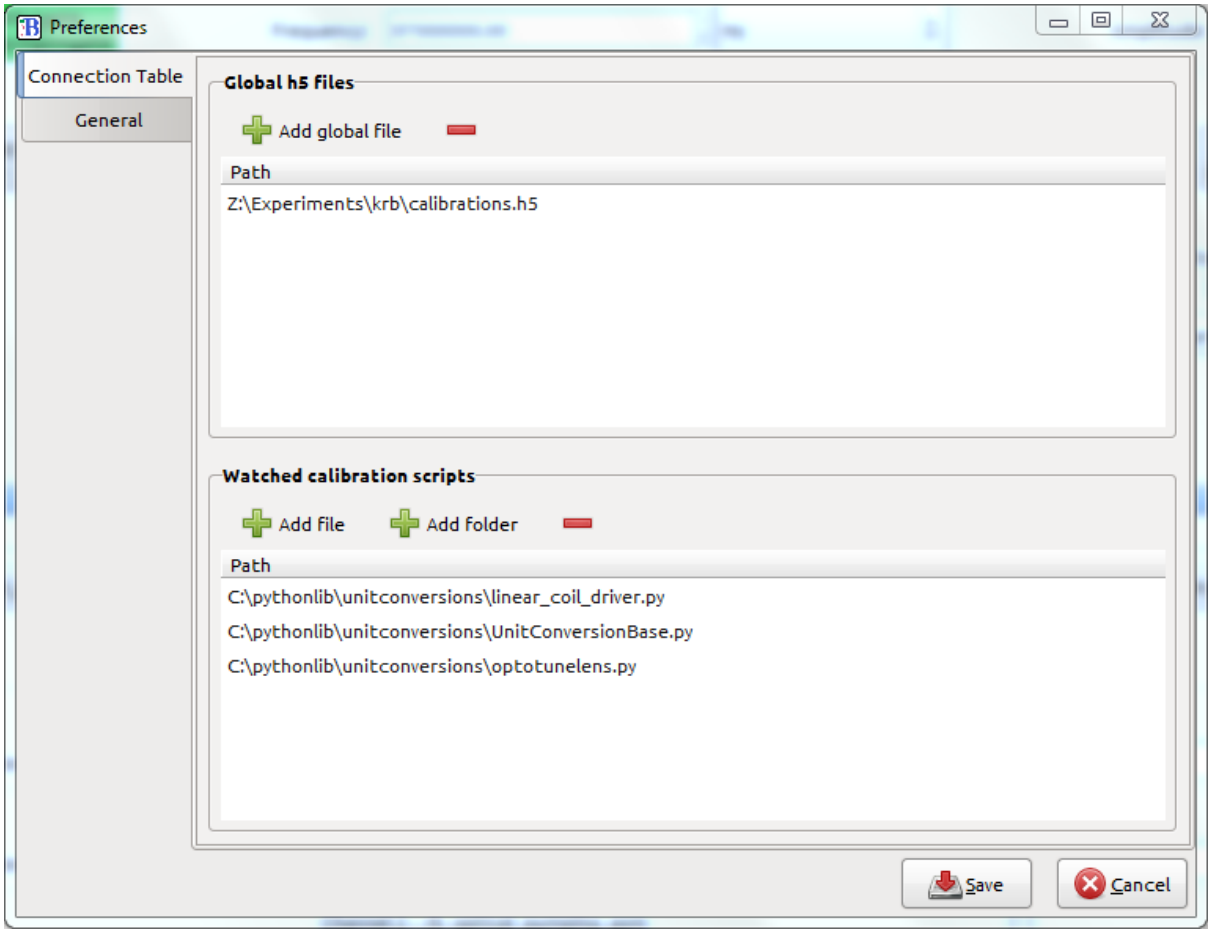

Figure 1 The BLACS preferences menu. Globals files required for compiling the connection table or for unit calibrations should be added here. Your preferred text editor can also be selected in the “General” tab to the left.

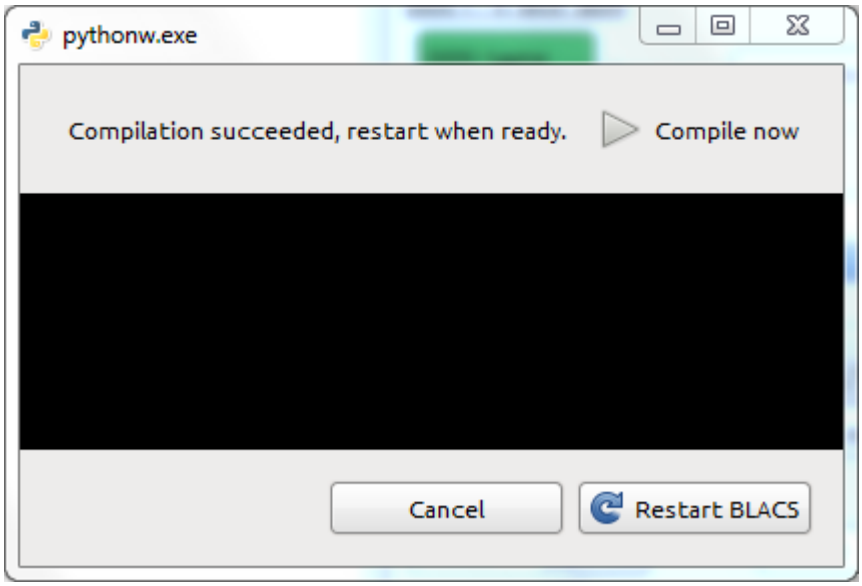

Figure 2 After re-compiling the connection table, you will be asked to restart BLACS. If there are any errors or compilation output, they will be shown in the black dialog box.

Manual Mode

The BLACS interface contains a tab for each device in the lab connection table. This tab contains widgets for controlling the output of that device (see Figure 3 for examples).

Widgets that take numerical values (e.g. analog outputs, frequencies etc) generally have an associated “units” selector, which allows values to be entered in units provided by your unit conversion scripts. The step size that these widgets will change by when arrow keys or the stepper buttons are pressed can be customised by right clicking on them and selecting “Change step size.”

Widgets may be locked by right clicking on them and selecting “Lock Widget”. This will cause the widget to go grey and prevent further changes to the state of that output until it is unlocked (by right clicking again and choosing “Unlock Widget”). This allows some degree of protection against unintended changes to important outputs, such as trigger lines or acousto-optic modulators used in laser locks.

The tabs for devices may be re-ordered and re-distributed between the four panes by dragging their labels around.

Currently BLACS has support for the following devices:

- SpinCore PulseBlaster DDS-II-300-AWG (seen in Figure 3, top left tab)
- Novatech DDS9m (seen in Figure 3, bottom right tab)
- National Instruments PCIe6363 (seen in Figure 3, top right tab)
- National Instruments PCI6733 (seen in Figure 3, bottom left tab)
- PineBlaster
- PhaseMatrix Quicksyn FSW-0010
- BLACS also has a tab used to communicate with the imaging system, BIAS. This tab simply contains an input box to tell BLACS the hostname of the computer running BIAS, and the port number to connect to (assigned in the connection table).

Error states

If something goes wrong with a device, its tab will go into an error state (e.g. see the Novatech in the bottom pane of Figure 4). An error box will appear at the top of the tab, and the icon next to its title will change to an orange exclamation mark. The python traceback will be displayed, explaining what error has occurred. For minor errors the dialog can be closed, and the device will continue functioning, though there is the option to re-start the tab, which will re-initialise the device. For fatal errors, the tab must be re-started to recover. Note that restarting a tab should not affect other devices. Tabs may also be restarted at any time by right clicking on their label and selecting “Restart” or choosing “Restart” from the “Device” drop down menu when the desired device has focus.

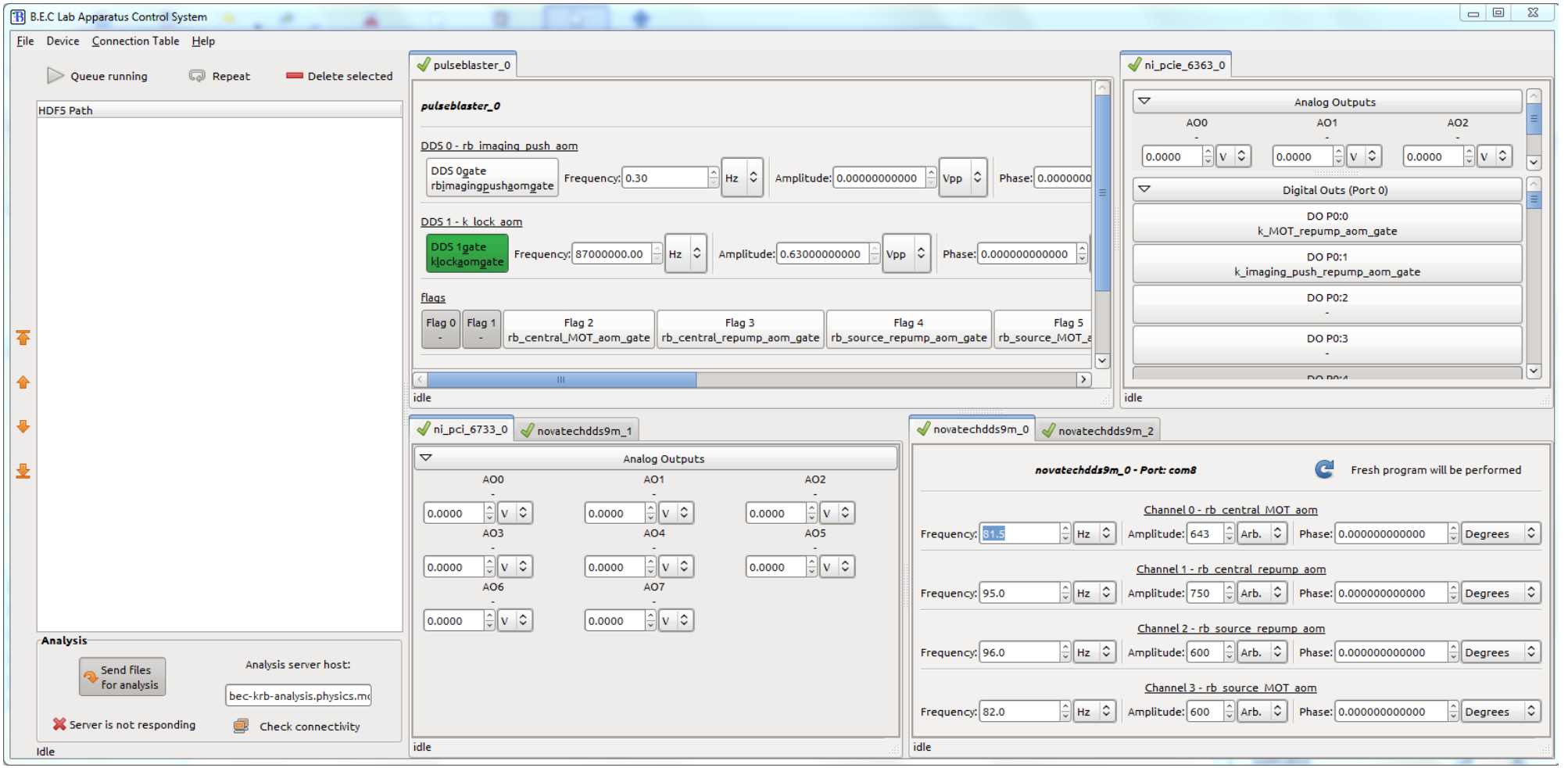

Figure 3 The BLACS interface. To the right are four resizable tab panes, containing the manual controls for each device described in the connection table. Note that channels that have been explicitly defined in the connection table are labelled with their names, while those that are not (e.g. if a channel is not used in any experiments) have a dash where the name would be. To the left is the experiment queue, and controls for sending file to analysis.

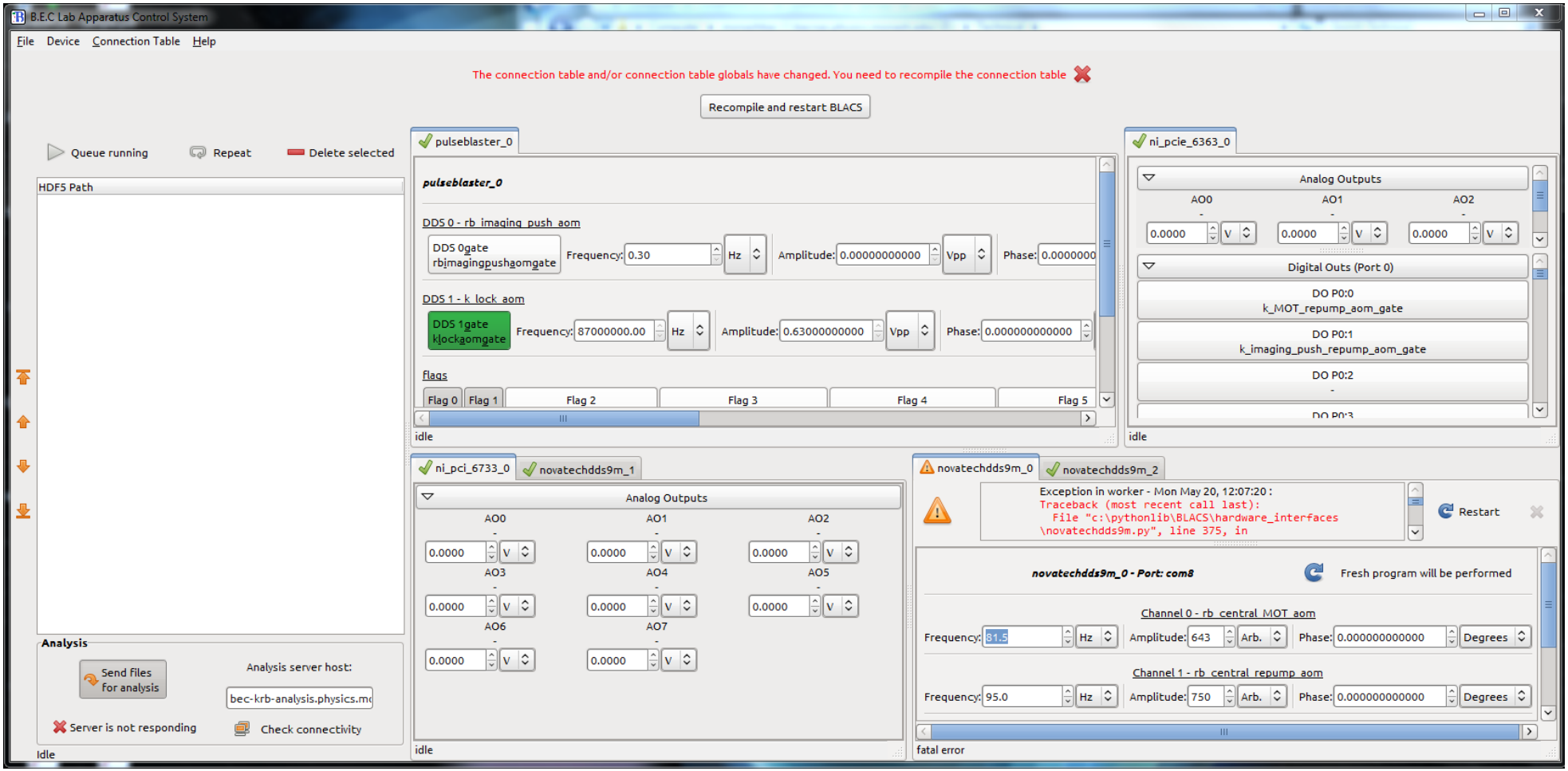

Figure 4 Error states. The Novatech in the bottom right hand corner has gone into a fatal error and must be restarted (note the greyed out X button would be able to be used to ignore less serious errors). The traceback contains information to help debug what has gone wrong. Note also, the notification at the top of the window indicating that the connection table has been modified, requiring a restart of BLACS.

Buffered Mode

When experiment shot HDF files are sent to BLACS from `runmanager`, they are placed in a queue, visible on the left hand side of the BLACS interface. BLACS will run experiments from the queue in order, unless the pause button is clicked.

As experiments are added to the queue, the experiment connection table is compared with the lab connection table. If it is not a subset then the experiment will be rejected.

When an experiment is about to run, BLACS will ask each device required for the experiment (i.e. any devices in the experiment's connection table) to enter buffered mode, and program them accordingly. For some devices with lots of data this may take tens of seconds. Once all devices are loaded, a start signal will be sent to your master pseudoclock (as specified in your experiment script). The manual controls for devices used in the experiment will not work during the buffered sequence. Note that any channels on devices involved with the experiment that are not used in the experiment will be set to zero. Devices that are not involved in the experiment at all will still be available for manual control during the buffered sequence.

If a device used in your experiment encounters an error before or during an experiment, the queue will be paused and the experiment will be placed back at the top of the queue. You will need to resolve the error before the expersiment will run.

The experiment will run until your pseudoclock notifies BLACS that it is complete. For continuity, all outputs used in the experiment will remain in the state that they were in at the end of the experiment (i.e. not necessarily the state that they were in before the experiment started) and be available for manual control again.

If the experiment queue is paused during an experiment then the current experiment will run normally, but subsequent experiments will not run until it is un-paused. Experiments in the queue can be re-ordered using the arrow buttons, or removed from the queue.

If the repeat button is pressed, experiments will be automatically re-added to the bottom of the queue upon completion, creating a new HDF file with a repeat number added to the file name. Experiments may be manually added to the queue by selecting "Open" from the "File" drop down menu. If the experiment to be added has already been run, a new HDF file will be created with a repeat number added to the filename.

Smart programming

We have employed a "smart programming" feature on devices that often require large amounts of data and are slow to program. When enabled for a device, BLACS will compare the data for each experiment to the previous experiment, and only send the subset of data that has changed to that device. A "fresh program" can be requested by un-ticking the smart programming checkbox on the device tab before it begins transitioning to buffered mode. This will result in the entire dataset for the next experiment being loaded, regardless of similarity to the previous experiment. This may be required after a device has been restarted, or if you suspect corrupt memory in your device.

Sending files for analysis

Experiment shot files can be automatically sent to `lyse` for analysis when they finish running. This is done by typing in the host name (or IP address) of the computer running lyse (or `localhost` if `lyse` is running on the same computer as BLACS) and ensuring the "Send files for analysis" button is enabled.

If connectivity to `lyse` is lost, experiments to be analysed will be queued and sent when connectivity is regained.