Software for the Piccolo Spectroradiometer

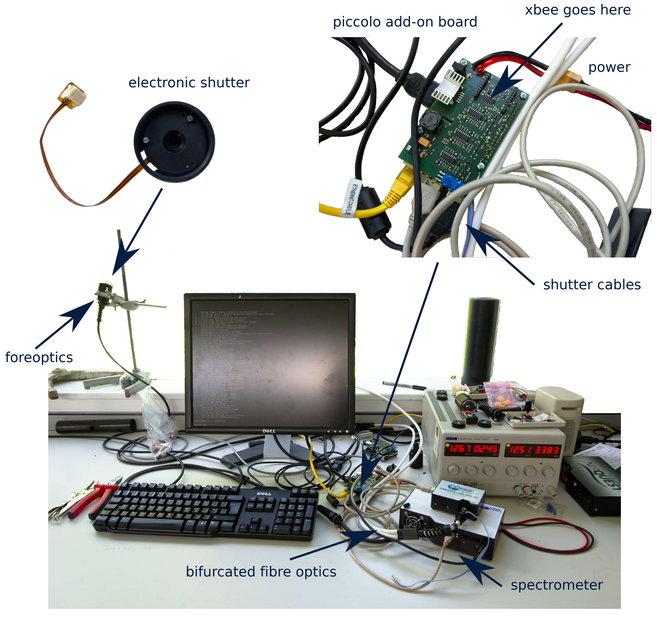
Magnus Hagdorn[1], Alasdair MacArthur[1], Iain Robinson[2]

[1]School of GeoSciences, University of Edinburgh, UK
[2]now at Rutherford Appleton Laboratory, Science and Technology Facilities Council

THE UNIVERSITY of EDINBURGH

Introduction

The Piccolo instrument is a lightweight dual-field-of-view spectrometer system for the near simultaneous measurement of both reflectance and sun induced fluorescence (SIF). This system utilises a double bifurcated fibre optic incorporating a novel switching mechanism such that it enables reflectance and SIF to be measured simultaneously from the same Earth surface area and reference irradiance to be measured at frequencies greater than 1Hz. This system is lightweight, self-contained, and wirelessly controlled to enable it to be deployed on rotary-wing UAVs, or at fixed locations for high temporal frequency logging (time series) measurement approaches. It also contains the latest very high resolution Ocean Optics QE Pro optical bench which enables both telluric O2 bands to be measured simultaneously and with the same integration time. This system can therefore be used for multi temporal and spatial scale Earth observation measurements to assess surface state, photosynthetic rate and Earth/atmosphere gas exchange flux.

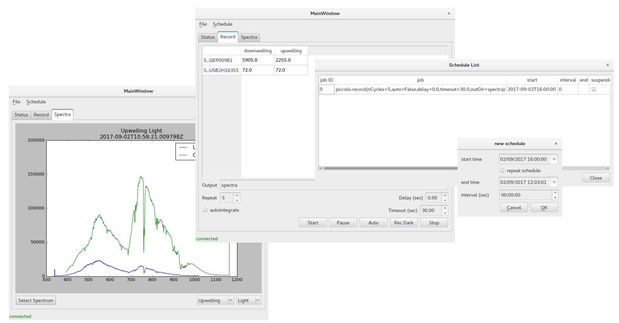


Hardware

- raspberry pi
- custom board for power supply and control for three shutters
- xbee serial over radio board
- multiple spectrometers attached via USB
- double bifurcated fibre optics to record using two diferent spectrometers simultanuously and to switch between two directions
- optionally battery powered

Software Design Goals

- control multiple instruments
- modular: easy to add more components/instruments
- allow multiple clients
- clients connect via serial over radio (xbee) and TCP/IP
- clients must be able to disconnect/reconnect
- need to be able to schedule measurement sequences



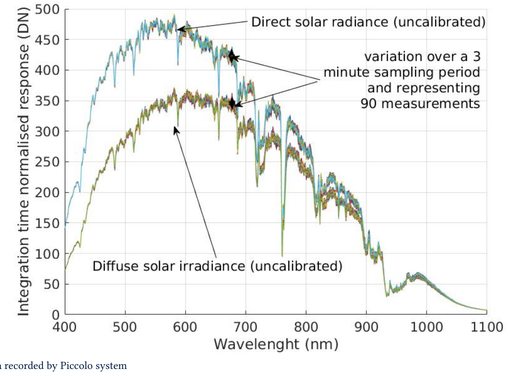
Piccolo Player GUI: showing the spectra tab with a spectrogram; the record tab used to setup the measurements and the scheduler windows

Client Software

- uses JSON RPC for TCP/IP communication
- written in python
- uses custom protocol for serial over radio
- uses proxy class to access server via RPC
- GUI uses QT and is created using designer
- matplotlib for plotting

Proxy class methods are automatic

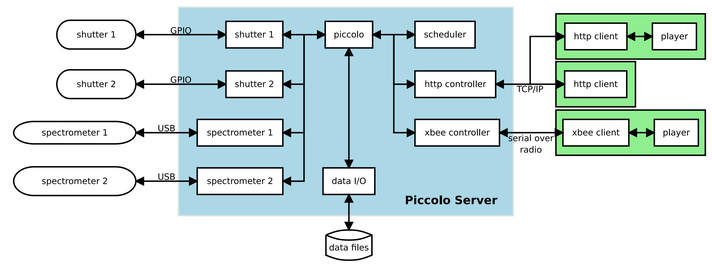
```
def __getattr__(self,attr):
   def func(**keywords):
        return self._client.call(attr,keywords=keywords)
          return func
```



Processed spectra recorded by Piccolo system

Server Software

- written in Python
- multi-threaded application with one thread for each component
- CherryPy for JSON RPC communication
- each component consists of a proxy class and a real class implementing the functionality
- proxy class and real class communicate via pipes and a status lock
- calls are asynchronous



Components of the Piccolo system: Spectrometers are attached via USB; the shutters are controlled via GPIO. Each component consists of a proxy/worker object pair. The workers run in a separate thread. Clients comunicate via RPC.

Proxy/Worker Class Pair

The proxy object

- checks status of worker
- adds task to task queue
- gets results from result queue

The worker object sits in an endless loop waiting for new tasks

```
while True:
    new_task = self.taskQ.get()
    self.busy.acquire()
    result = do_task(new_task)
    self.resultQ.put(result)
    self.busy.release()
```

Proxy Class	RealClass
+taskQ +doneQ +busy	+taskQ +doneQ +busy
+record(time) +getSpectogram()	+run() -_record(time)

Future Plans

- add mini-language to describe recording schedule
- add more sensors, eg GPS for time and position
- add triggers, eg start recording when position (x,y) has been reached
- web interface

Other Projects?

- The software is mostly GPL licensed appart from the low level USB drivers for the spectrometers
- The architecture can be used for other client/server logging problems
- The source code is available on bitbucket:

https://bitbucket.org/account/user/teampiccolo/projects/PP

contact: magnus.hagdorn@ed.ac.uk