Python camera Module - camera.py

This library takes care of taking camera photographs, choosing the "best" for transmission, and converting the JPG files to SSDV format for transmission.

It works from a schedule list which dictate how often photos should be taken, to what resolution, and where they should be stored. Images are taken and converted in a thread so as not to delay the user functions.

Normally there should be one schedule per radio channel, plus an extra one for full-sized images that are not sent over radio.

Sample Usage

```
from camera import *

MyCam = SSDVCamera()

MyCam.add_schedule('RTTY', 'PYSKY', 'images/RTTY', Period=30, Width=320, Height=240)
MyCam.add_schedule('LoRa0', 'PYSKY2', 'images/LoRa0', Period=30, Width=640, Height=480)
MyCam.add_schedule('SD', '', 'images/FULL, Period=60, Width=3280, Height=2464)

MyCam.take_photos()

Packet = self.camera.get_next_ssdv_packet("RTTY")
```

Reference

Object Creation

```
SSDVCamera()
```

Functions

```
add_schedule(Channel, Callsign, TargetFolder, Period, Width, Height, VFlip=False, HFlip=False)
```

where

- Channel is a unique name for this entry, and is used to retrieve/convert photographs later
- Callsign is used for radio channels, and should be the same as used by telemetry on that channel (it is embedded into SSDV packets)
- TargetFolder is where the JPG files should be saved. It will be created if necessary. Each channel should have its own target folder.
- Period is the time in seconds between photographs. This should be much less than the time taken to transmit an image, so that there are several images to choose from when transmitting. Depending on the combination of schedules, and how long each photograph takes, it may not always (or ever) be possible to maintain the specified periods for all channels.
- Width and Height are self-evident. Take care not to create photographs that take a long time to send.
- VFlip and HFlip can be used to correct images if the camera is not physically oriented correctly.

  clear_schedule()

Clears the schedule.

```
take_photos()
```

Begins execution of the schedule.

```
get_next_ssdv_packet()
```

Retrieves the next SSDV packet for a particular channel. If there is none available (i.e. no photograph has been taken and converted yet for this channel) then **None** is returned. Returned packets contain a complete (256-byte) SSDV packet.

Dependencies

- Needs the Pi Camera to be enabled
- Needs SSDV to be installed

Behind The Scenes

The library includes "best image selection", by looking for the largest available JPG file. This works better than you might expect, largely eliminating poor images from the radio transmissions.

The process begins with the schedules as described above. So, for example, RTTY images might be stored in the images/RTTY folder. At some point (see below), the image files are checked and the largest one is selected for conversion to SSDV format. The conversion is done by calling the SSDV program, resulting in an SSDV file ready for transmission. All the JPG files in that folder (i.e. the selected one and the rejected ones) are then moved to a dated folder images/RTTY/, leaving the original folder ready for subsequent images.

This conversion process is triggered when there are fewer than 10 image packets remaining for transmission. So, on startup, this will happen immediately after the first image is taken, and then again shortly before that first image has been completely sent.