LSE-2015 Project

Camera Control

Cholbi Alenda, Pablo (*Dev, Doc, Test*)
p.cholbi@alumnos.upm.es

20th of May, 2015

Date	Version	Issue	Author
2015-05-20	1.0	Initial release.	*PCA*

Contents

1 Description

The **camctrl** application provides the functionality controlling the pan/tilt mechanism of the camera. It receives raw commands over UART, process the data and sends processed commands to the servomotors.

2 Dependencies

camctrl depends on **servoctrl**, which is also provided within this repository and the documentation of which can be found in `../servoctrl`.

3 Application

The application receives input through an UART-USB converter. this devices is assumed to be mapped to `/dev/ttyUSB0`.

camctrl expects received data to be in the following format: **8N1 19200bps**.

The application interprets incoming data as shown in the table below:

3.0.1 `Input`

Bit	Name	Description
0	`down`	State of "down" button
1	`up`	State of "up" button
2	`right`	State of "right" button
3	`left`	State of "left" button
7-4	`x`	RESERVED

Table 1: `Input`

Please not that this application considers the state bits in the byte to be active-low; meaning that the least significant nibble of the byte is b1111 when all of the buttons are at rest.

If the "up" and "down" or the "left" and "right" buttons are asserted simultaneously, the cameras's video stream is paused and a capture of the frozen frame is saved to `/home/pi/capture.jpg`. Once paused, the normal execution of the application can be resumed by asserting the pause condition once again.

All other messages are processed and commands are sent to the **servoctrl** over socket **2301** in the format specified by the documentation of the aforementioned application.

4 Setup

A set up script is provided at `../../scripts/setup/camctrl_setup.sh` to ease the building and installation.

If the setup script finished successfully; the binary should be at `/usr/local/bin/camctrl.py` and a UNIX System V init script should be at `/etc/init.d/camctrl`.

Aside form the setup script; the source code is also provided.

5 Service

To start **camctrl** as a daemon; execute `/etc/init.d/camctrl` as root. This daemon can take as argument **start**, **stop**, **restart** or **force-reload**.

6 Testing

A test script is provided at `../../scripts/test/camctrl_test.sh` to test the application. The test script execute a series of test cases and then asks for user input to determine if the test executed correctly.

If the test was successful, the scripts returns 0, if the test failed a value different from 0 is returned.

To aid with the testing of this application, a helper script has been developed (`camctrl-dummy.py`) to simulate UART messages with the same UART-USB converter. For this to work correctly it is necessary to connect the TX and RX pins of the converter during the test.

The test sequence currently implemented is the following:

1. At the start, the user is asked if the video stream is being displayed correctly.
2. A simulated "pause" condition is sent.
3. The user is asked if the video stream has paused.
4. Simulated messages move the pan/tilt mechanism in a square path.
5. The user is asked if the pan/tilt mechanism followed expected path.

7 Issues

The following issues have been observed in this module:

1. At the time of writing, a recent update of the Raspberry Pi bootloader and the third party kernel and packages by Adafruit needed to control the LCD-TFT touchcreen have caused the screen to not function. This is expected to be resolved by the upstream developer.
2. There is a bug in the sevice script `/etc/init.d/camctrl` or the source code that prevents the application from being daemonized. as a workaround, the following command can be executed the start **camctrl** detached:
   ```
   screen -d -m -S camctrl python /usr/local/bin/camctrl.py
   ```