piCube Technical Document

1. Installation

1.1. Installation and cloning piCube Repo

piCube is a python script that creates a command line like interface for any of the three Raspberry Pi Camera models. piCube uses the picamera package to communicate with the camera and without it, the code will not execute. If you are using the Raspbian distro, picamera is probably installed by default. You can verify the version installed by simply running:

```
pi@project-pi:~ $ pip list | grep picamera
picamera                          1.13
```

Version 1.13 is the latest at the time of writing this document. If picamera is not installed on your Raspberry Pi, run:

```
pi@project-pi:~ $ sudo apt-get update
pi@project-pi:~ $ sudo apt-get install python-picamerapython3-picamera
```

The piCube git repository is where the python script, the servo configuration and this technical document reside. To get started, we must clone the repo onto the Raspberry Pi, but first we must create a git directory in Documents. Follow these steps:

```
pi@project-pi:~ $ cd Documents

pi@project-pi:~/Documents $ mkdir git
pi@project-pi:~/Documents $ ls
git
pi@project-pi:~/Documents $ cd git
pi@project-pi:~/Documents/git $ git clone git@github.com:sepseb/piCube.git
Cloning into 'piCube'...
remote: Enumerating objects: 121, done.
remote: Counting objects: 100% (121/121), done.
remote: Compressing objects: 100% (87/87), done.
remote: Total 121 (delta 69), reused 78 (delta 32), pack-reused 0
Receiving objects: 100% (121/121), 480.79 KiB | 1.44 MiB/s, done.
Resolving deltas: 100% (69/69), done.
pi@project-pi:~/Documents/git $ cd piCube/
pi@project-pi:~/Documents/git/piCube $
```

Here are a few linux commands you need to know to easily navigate around:

- `ls` - Use this to show what files or folders are in the directory you are in.

- `cd` - Use this to go to a directory and has multiple cases:
  - `cd` - takes you to home directory
  - `cd ..` - takes you one directory back
  - `cd [name]` - takes to a directory if it exists in your current directory
  - `cd [path]` takes to a directory path from anywhere
- `mkdir [name]` - create a directory in current directory

```
pi@project-pi:~/Documents/git $ cd piCube/
pi@project-pi:~/Documents/git/piCube $ cd ..
pi@project-pi:~/Documents/git $ cd
pi@project-pi:~ $ cd Documents/git/piCube/
pi@project-pi:~/Documents/git/piCube $ cd
pi@project-pi:~ $ cd Documents/git/
pi@project-pi:~/Documents/git $ cd piCube/
pi@project-pi:~/Documents/git/piCube $
```

You need to be in the piCube directory in order to execute the piCube python script.

1.2. Software Updates

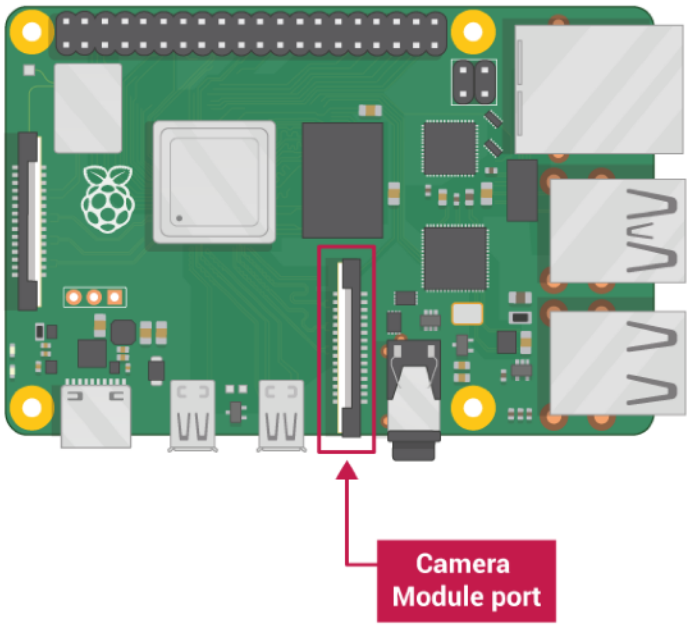
To update to the latest software, use **git pull**.

```
pi@project-pi:~/Documents/git/piCube $ git pull
hint: Pulling without specifying how to reconcile divergent branches is
hint: discouraged. You can squelch this message by running one of the
following
hint: commands sometime before your next pull:
hint:
hint:   git config pull.rebase false  # merge (the default strategy)
hint:   git config pull.rebase true   # rebase
hint:   git config pull.ff only       # fast-forward only
hint:
hint: You can replace "git config" with "git config --global" to set a
default
hint: preference for all repositories. You can also pass --rebase, --no-
rebase,
hint: or --ff-only on the command line to override the configured default
per
hint: invocation.
Updating d5572aa..9490be2
Fast-forward
 pi-cam.py | 282 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++
+-----------------------------------------------------------------
 1 file changed, 168 insertions(+), 114 deletions(-)
pi@project-pi:~/Documents/git/piCube $
```

2. Getting Started

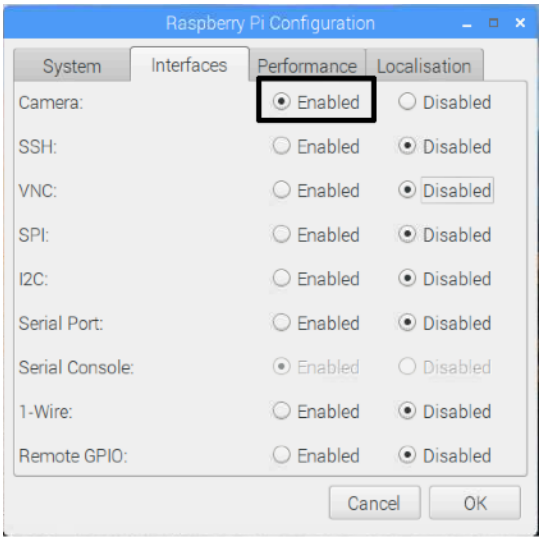
Before you can use piCube, you must make sure the camera module is properly connected to the Raspberry Pi and enabled in the Raspberry Pi Configuration. Make sure the power to the Pi is disconnected while connecting and removing the Pi Camera from the board, otherwise you risk damaging the camera.

For more detailed instructions, visit the Raspberry Pi website. The camera port for all Raspberry Pi models is located beside the ethernet port:



Note: Be very careful with the ribbon cable as they can break if pinched.

Start up the Raspberry Pi and once the operating system has booted into desktop, press the raspberry pi icon in the left top corner. Click on Preferences and then **Raspberry Pi Configuration**. Select interfaces and enable camera. Restart the Raspberry Pi.



You're now ready to startup camera interface. If you haven't already, navigate to the piCube directory and startup the application by running the following:

```
pi@project-pi:~ $ cd Documents/git/piCube/
pi@project-pi:~/Documents/git/piCube $ python3 pi-cam.py
Starting Application...
Detected Camera Module v1  1
 Camera resolution set to : 2592x1944
Development version installed  2
----------------------------------------
 piCube Interface Program v0.2.0
 Type help for a list of commands
 Type exit to leave the program
piCube $
```

Displays the type of Camera module detected - 4 possible outcomes • Camera Module v1 • Camera Module v2 • HQ Camera • Camera Model Unknown or no camera detected	**1**

Note: If you get a notification that looks like the last bullet, turn off the Raspberry Pi and verify the camera ribbon cable is properly seated in the connected and the tab has been pushed in.

Displays version status of the software • Released version v# is now available - if a new version is available • Development version installed - if current software is ahead of any released version • Latest released version installed - You have the latest version	**2**

Note: You always should have the latest released version installed. Development version can be unstable and not fully tested (Beta). If you start the piCube application and see that a new released version is available, close piCube by typing exit and performing the steps in the Software Update section.

piCube will set the camera resolution to the maximum possible setting. Because, each camera model has a different max resolution, that value changes depending on the camera model you have connected to the Raspberry Pi. Below is a table of the max resolution supported by each:

	Camera Module v1	Camera Module v2	HQ Camera
Sensor	OmniVision OV5647	Sony IMX219	Sony IMX477
Sensor resolution	2592 x 1944 pixels	3280 x 2464 pixels	4056 x 3040 pixels
Pixel Size	1.4 µm x 1.4 µm	1.12 µm x 1.12 µm	1.55 µm x 1.55 µm

3. piCube Commands

3.1. help

The **help** functions serves to show the user the list of commands currently supported by the piCube software and how to use each command.

To get list of of all commands, type help:

```
piCube $ help

Documented commands (type help <topic>):
========================================
auto  exit  filter  help  manual  new_session  set_resolution

piCube $
```

To get specific help on a command, type help followed by the name of the command from the list

```
piCube $ help auto
Syntax : auto x
Captures one picture with automatic settings
If x is not defined, filename is current data_time.jpg
If x is give a string, filename will be the string.jpg
piCube $
```

help <topic> is there to provide information on how to use a specific command and the syntax.

3.2. new_session

After you start the piCube application and before you can take any pictures, the software needs to know the desired location to store the pictures. The **new_session** command creates a folder with the user provided name in the /home/pi/Documents/ directory.

If a folder already exists with the same name as the one provided by the user, the software will ask if you like to use existing folder as your current session. This will not overwrite the current folder and all previous files will remain on the Raspberry Pi Drive.

If you fail to provide a name, the command will throw a syntax error.

```
piCube $ new_session
Syntax Error: No session name provided
piCube $ new_session test
Created directory : /home/pi/Documents/test/
piCube $ new_session test
Session already exists - Continue with old session? (y/n)?y
Using /home/pi/Documents/test/ as current session
piCube $
```

3.3. auto

The **auto** command captures one jpg image in the current session with either timestamp used as filename or one provided by the user given how the command is called. This feature is useful for quick image capture and the camera will decide for the optimal values to use for iso and shutter speed.

If you just call the auto command, it will use the following filename format : **auto_m-d-y_H-M-S.jpg** where m = month, d = day, y = year, H = hour, M = minutes and S = seconds as filename.

If you call the command followed by a string (can include digits and characters), the filename will be set to the **string.jpg**.

```
piCube $ auto
Image captured to : /home/pi/Documents/test/auto_07-17-2022_12-10-33.jpg
piCube $ auto test
Image captured to : /home/pi/Documents/test/test.jpg
piCube $ auto test2
Image captured to : /home/pi/Documents/test/test2.jpg
piCube $
```

3.4. manual

The **manual** command captures one jpg image in the current session and has three settings that alter the image. The three arguments are:

- `iso` : When set, the sensitivity of the sensor to light is adjusted. Lower value implies less sensitivity than higher value. Valid values are between 0 (auto) and 1600.
- `shutter` : When set, the property adjusts the shutter speed of the camera, The value input is in units of µs and the default value is 0 (auto). The exposure_speed attribute shows the actual shutter speed being used when this shutter_speed is set to 0 (auto).

- `exp` : When set, the property adjusts the exposure compensation. Valid value is between -25 and 25 indicating the exposure level of the camera. Larger values result in brighter image. Each increment represents 1/6th of a stop. Setting the attribute to 6 increases exposure by 1 stop. The default value is 0 and it not affected if user doesn't provide a value (no auto for this attribute)

```
piCube $ manual
Warning: No user input for iso - using auto
Warning: No user input for shutter speed - auto
Warning: No user input for exposure compensation - using default value

ISO : auto
Exposure time : 1/153 (6517) microseconds
Image captured to : /home/pi/Documents/test/iso-0_shutter-0_55-01.jpg

piCube $ manual --iso 800
Warning: No user input for shutter speed - auto
Warning: No user input for exposure compensation - using default value

ISO : 800
Exposure time : 1/153 (6517) microseconds
Image captured to : /home/pi/Documents/test/iso-800_shutter-0_55-11.jpg

piCube $ manual --shutter 5000
Warning: No user input for iso - using auto
Warning: No user input for exposure compensation - using default value

ISO : auto
Shutter speed : 1/200 (4989) microseconds
Image captured to : /home/pi/Documents/test/iso-0_shutter-4989_55-25.jpg

piCube $ manual --iso 800 --shutter 10000
Warning: No user input for exposure compensation - using default value

ISO : 800
Shutter speed : 1/100 (9994) microseconds
Image captured to : /home/pi/Documents/test/iso-800_shutter-9994_55-41.jpg

piCube $ manual --iso 800 --shutter 1000 --exp 2

ISO : 800
Shutter speed : 1/1009 (991) microseconds
exposure compensation : 2
Image captured to : /home/pi/Documents/test/iso-800_shutter-991_56-01.jpg

piCube $ manual --iso 1900
Invalid iso value: 1900 (valid range 0..800)  - try a lower number
piCube $
```

Manual mode does not require the user to provide all three inputs. If the user chooses to just modify iso, they can just input a value for iso and piCube will set the other attributes to auto or default. The user will be notified of the attributes that have been set to auto. The order the user provides the inputs does not matter.

Images captured with manual command have a filename with the following format: **iso-i_shutter-j_M-S.jpg** (where i = iso value, j = shutter speed, M = minutes, S = seconds).

3.5. set_resolution

The **set_resolution** command allows the user to modify the cameras resolution at which images are taken with. Initially the piCube software sets the camera to the max supported resolution possible but, if the user wants to lower the resolution, they can use this command.

```
piCube $ set_resolution 80 80
Camera Resolution set to : 80 x 80
piCube $
```

4. Advanced Features

4.1. Sensor Mode

The user can now pass an argument to python when starting piCube to force the sensor_mode variable to a desired value. To switch the sensor mode, use the --mode argument after pi-cam.py followed by the number of the mode. piCube will indicate if it has set the sensor mode. To change sensor modes, you must exit the piCube software and start it again with desired mode. Currently the software does not support sensor mode change while application is running.

```
pi@project-pi:~/Documents/git/piCube $ python3 pi-cam.py --mode 3
Starting Application...
 Camera sensor mode set to : 3
 Detected Camera Module v1
 Camera resolution set to : 2592x1944
 Development version installed
----------------------------------------
 piCube Interface Program v0.2.0
 Type help for a list of commands
 Type exit to leave the program
piCube $
```

4.2. Development Mode

Usually it is recommended that you use the latest released version of piCube. However, you can switch to development version if there's a beta feature that you'd like to use. Use the following commands to switch from released to development:

```
pi@project-pi:~/Documents/git/piCube $ git pull
remote: Enumerating objects: 15, done.
remote: Counting objects: 100% (15/15), done.
remote: Compressing objects: 100% (9/9), done.
remote: Total 13 (delta 7), reused 9 (delta 4), pack-reused 0
Unpacking objects: 100% (13/13), 7.19 KiB | 409.00 KiB/s, done.
From github.com:sepseb/piCube
   d5572aa..9490be2  main        -> origin/main
 * [new branch]      development -> origin/development
 * [new tag]         v0.2.0      -> v0.2.0
Updating d5572aa..9490be2
error: Your local changes to the following files would be overwritten by
merge:
    pi-cam.py
Please commit your changes or stash them before you merge.
Aborting
pi@project-pi:~/Documents/git/piCube $ git checkout development
Branch 'development' set up to track remote branch 'development' from
'origin'.
Switched to a new branch 'development'
pi@project-pi:~/Documents/git/piCube $
pi@project-pi:~/Documents/git/piCube $ python3 pi-cam.py
Starting Application...
 Detected Camera Module v1
 Camera resolution set to : 2592x1944
 Development version installed
----------------------------------------
 piCube Interface Program v0.3.0
 Type help for a list of commands
 Type exit to leave the program
piCube $
```

To switch back to the latest released version run the following commands:

```
pi@project-pi:~/Documents/git/piCube $ git checkout main
Switched to branch 'main'
Your branch is behind 'origin/main' by 4 commits, and can be fast-
forwarded.
  (use "git pull" to update your local branch)
pi@project-pi:~/Documents/git/piCube $ git pull
Already up to date.
pi@project-pi:~/Documents/git/piCube $
```

5. Release Notes

5.1. v0.2.0

- Notification of New Release Version availability
- Added full manual control command
- Added auto command
- Added sensor mode to startup command
- capture cmd is deprecated; use auto instead
- cap cmd is deprecated; use manual instead
- cap_raw cmd due to incompatibility with HQ camera
- Improvements to code stability and bug fixes

5.2. v0.1.0

- Camera Model is now shown when piCube is first started
- Camera Resolution is set to Max
- Release Version is now indicated
- Added capability to adjust resolution
- Added raw data capture (Beta)