# Documentation of Extended Raspicam

Titusz Ozsváth

07.01.2016

**Abstract**

Now, there is no available application for Raspberry Pi or Raspberry Pi 2 which can handle the camera and which can reach more than 30 frame per secundum. So I extended the Raspicam application.

## 1 Preparation

### 1.1 Notes

In this tutorial, I will work in Bash.

### 1.2 Download from internet

1. Navigate to the library where you want to work.
2. Make a new directory:

   ```
   sudo mkdir ExtendedRaspicam
   ```

3. Download the original project from github:

   ```
   sudo git clone https://github.com/cedricve/raspicam
   ```

4. Add permissions:...

   ```
   sudo chmod 777 –R ExtendedRaspicam/
   ```

5. Step into the directory and make a build directory:

   ```
   cd ExtendedRaspicam/raspicam && mkdir build
   ```

6. Create a Makefile with CMake:

   ```
   sudo cmake ..
   ```

7. Make and install:

   ```
   sudo make && sudo make install
   ```

8. Try it out:

```
raspicam_test
```

# 2 Modifications

## 2.1 Notes

I made these modifications in Netbeans.

## 2.2 The actions

1. Navigate to ExtendedRaspicam/raspicam/src/private/private_impl.h. Here you have to declare two functions:

   - `void setFps(unsigned int fps);`
   - `unsigned int getFps() const;`

2. In private_impl.cpp the implementations come:

   - ```
     void setFps(unsigned int fps) {
             State.framerate = fps;
     }
     ```
   - ```
     unsigned int getFps() const {
             return State.framerate;
     }
     ```
   - overwrite setDefaultStateParams() function:

     ```
     void Private_Impl::setDefaultStateParams() {...
             State.framerate          = 120;
             State.width                    = 320;
             State.height                  = 240;
             ...
     }
     ```

3. In raspicam_cv.cpp:

   - overwrite get() function:

```
double RaspiCam_Cv::get ( int propId ) {
    switch ( propId ) {
                ...
                case CV_CAP_PROP_FPS:
                        return _impl->getFps();
                        break;
                ...
        }
}
```

- overwrite set() function:

```
bool RaspiCam_Cv::set ( int propId, double value )
        switch ( propId ) {
                ...
                case CV_CAP_PROP_FPS:
                        _impl->setFps(value);
                        break;
        }
}
```

4. In raspicam_cv_test.cpp:

```
...
int nCount=500;
...
Camera.set(CV_CAP_PROP_FRAME_WIDTH,320);
Camera.set(CV_CAP_PROP_FRAME_HEIGHT,240);
Camera.set(CV_CAP_PROP_FPS,120);
...
//if ( i%30==0 && i!=0 )
//     cv::imwrite ("image"+std::to_string(i)+".jpg",image );
...
```

# 3 Test

- navigate to:

```
cd .../ExtendedRaspicam/raspicam/build
```

- rebuild all in Bash:

```
make && make sudo install
```

- raspicam_cv_test

### 3.1 Result

```
Usage (-help for help)
Connecting to camera Connected to camera =00000000e0aad12f
Capturing  capturing ...495/500
Images saved in imagexx.jpg
5.68073 seconds for 500
frames : FPS = 88.0168
```