F. Report No.6 Software User’s Manual

1. Installation Guide
   1. Setting up environment
      1. Hardware requirement

* Raspberry Pi 3 with Micro SD class 10 16GB.
* LCD 3.5 inch with solution 320x480 pixel.
* Mini Wireless Keypad.
* Camera 5.0 megapixels.
* Printer.
* Power supply board.
  + 1. Software requirement
* Raspbian environment for Raspberry Pi (16-02-2015 release date or newer)
* Opencv library version 2.4.10
* Qt Development tool, Qt Creator, Qt Designer.
  1. Deployment
     1. Prepare deployment package

Project zip package which will be deployed.

* + 1. Configure Raspberry Pi 3 before deploy

Step 1: Install Raspbian environment for Raspberry Pi 3.

Step 2: We choose Raspbian because it is based on Debian, because we choose Ubuntu is programming environment on PC, and Ubuntu and Debian are closely related. It is favorable to build our project on Raspberry Pi 3. Download Raspbian from this link:

<http://downloads.raspberrypi.org/raspbian_latest>.

Step 3: After download unzip file, obtained \*.img file.

Step 4: Use Win32 Disk Imager to write a raw disk image to Micro SD device,can be downloaded here: <http://sourceforge.net/projects/win32diskimager/>

Step 5: Put Micro SD to PC, target to \*.img file and click “write”.

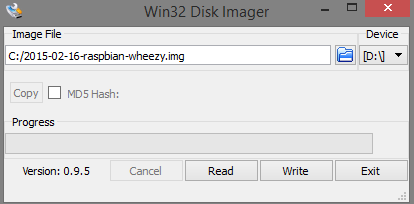


Figure 60. Write raw disk image to MicroSD

Step 6: Put Micro SD into Raspbery Pi 3, then turn on power. We must have 1 keyboard, screen connected to Raspberry VIA HDMI cable, and internet cable to install some requirement package.

Step 7: Configure raspbian on the first running

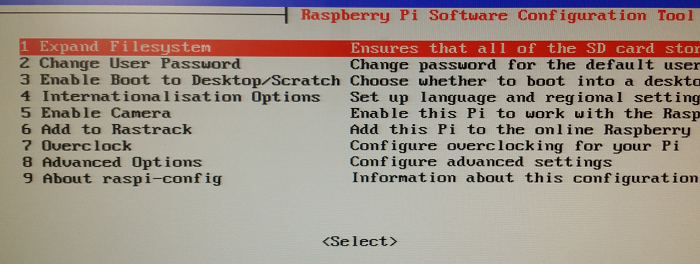


Figure 61. Configure raspbian

Step 8: Config options 1, 2, 5, then restart

* + 1. Configure IP for Raspberry

Step 1: Open command line

Step 2: At command line, enter: sudo nano /etc/network/interfaces

Step 3: And replace text on file by this:

auto lo

iface lo inet loopback //default of Kit

iface eth0 inet static //config LAN network

address 192.168.1.230 //set static IP for raspbery

// if your moderm have defferent gateway, you must set other IP //following gateway

netmask 255.255.255.0

gateway 192.168.1.1 //set gateway

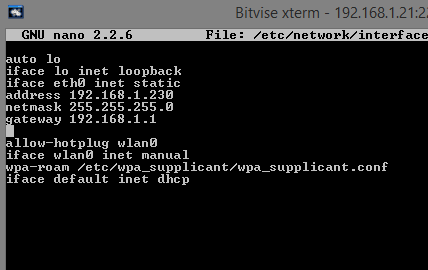


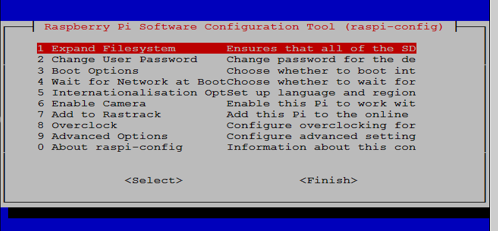
Figure 62. Configure IP network

Step 4: Press ctrl+X to save, then restart again. Now your raspberry pi have IP 192.168.1.230. You can set to other IP VIA modem IP and visible to remote Raspberry from your PC VIA Terminal. Install xrdp if you want to remote desktop. Xrdp uses the remote desktop protocol to present a GUI to the user: Sudo apt-get install xrdp

* + 1. Install OpenCV library

**Step #1: Expand filesystem**

* The first thing you should do is expand your filesystem to include all available space on your micro-SD card: $ sudo raspi-config



* Select the first option, “1. Expand File System”, hit Enter on your keyboard, arrow down to the “<Finish>” button, and then reboot your Pi: $ sudo reboot
* Delete the Wolfram engine to free up some space on your Pi: $ sudo apt-get purge wolfram-engine

**Step #2: Install dependencies**

* The first step is to update and upgrade any existing packages: $ sudo apt-get update

$ sudo apt-get upgrade

* We then need to install some developer tools, including CMake, which helps us configure the OpenCV build process: $ sudo apt-get install build-essential cmake pkg-config
* Next, we need to install some image I/O packages that allow us to load various image file formats from disk $ sudo apt-get install libjpeg-dev libtiff5-dev libjasper-dev libpng12-dev
* We also need video I/O packages. These libraries allow us to read various video file formats from disk as well as work directly with video streams: $ sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev

$ sudo apt-get install libxvidcore-dev libx264-dev

* The OpenCV library comes with a sub-module named highgui which is used to display images to our screen and build basic GUIs. In order to compile the highgui module, we need to install the GTK development library: $ sudo apt-get install libgtk2.0-dev
* Many operations inside of OpenCV (namely matrix operations) can be optimized further by installing a few extra dependencies: $ sudo apt-get install libatlas-base-dev gfortran
* Let’s install both the Python 2.7 and Python 3 header files so we can compile OpenCV with Python bindings: $ sudo apt-get install python2.7-dev python3-dev

### **Step #3: Download the OpenCV source code**

* Now that we have our dependencies installed, let’s grab the 2.4.13 archive of OpenCV from the official OpenCV repository.

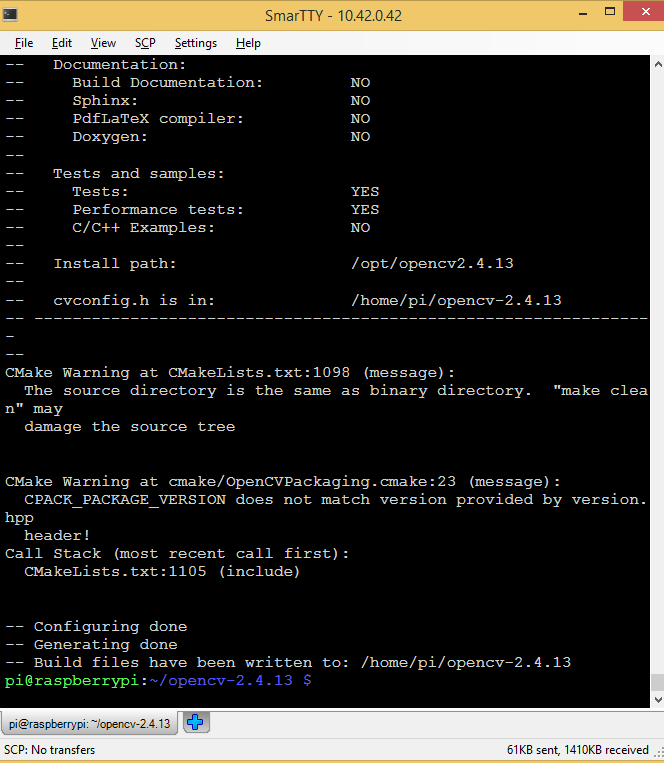
$ cd ~

$ wget -O opencv.zip https://github.com/Itseez/opencv/archive/2.4.13.zip

$ unzip opencv.zip

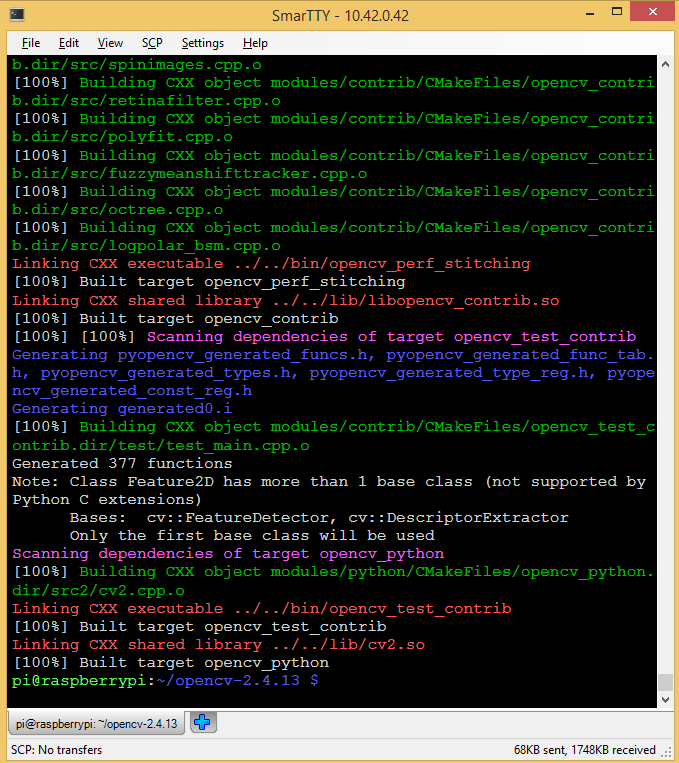
**Step #4 Setup our build using Cmake with prefix: /opt/opencv2.4.13**

$ cd opencv-2.4.13  
$ cmake -DCMAKE\_INSTALL\_PREFIX:PATH=/opt/opencv2.4.13



* When this done, go to compilation step:   
   $ sudo make -j5

$ sudo make install



* + 1. Install QT

Step 1: Open command line.

Step 2: Install cross-platform C++ application framework can use to build Qt project: sudo apt\_get install qt5-default

Step 3: Install Qt programming: sudo apt-get install qtcreator

* + 1. Install Zbar

Step 1: Open command line.

Step 2: In command line, enter:

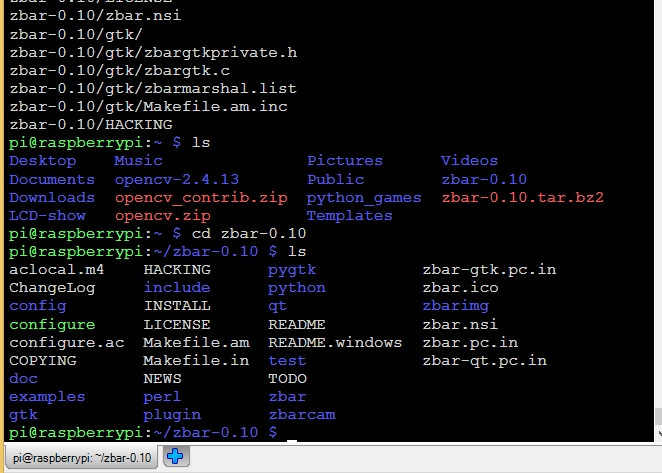
Wget <https://sourceforge.net/projects/zbar/files/zbar/0.10/zbar-0.10.tar.bz2>

tar *xvjf zbar-0.10.tar.bz2*

./configure --without-qt --without-gtk --without-python --without-imagemagick --prefix=/opt/zbar.arm --with-x=no

sudo make –j5

sudo make install



* + 1. Deploy application on Raspberry Pi 3

Step 1: Copy .zip file of application to raspberry VIA scp command, bitvise ssh, or any program at directory “/home/pi”

Step 2: At raspberry, unzip file zip, open command line, enter:

unzip DemoCapstone.zip

cd DemoCapstone

Step 3: Build qt application.

Create makefile by qt development tool: qmake DemoCapstone.pro;

Build application: make

Step 4: Finally, reboot raspberry and that auto run application (with path to execution file (DemoCapstone file on DemoCapstone directory) you entered on .xinitrc file

* + 1. Connection map between components

Figure 67. Connection map between all components

User Guide

G. Appendix

1. Rasbian operating system:

http://www.raspbian.org/RaspbianAbout

2. OpenCV library to detect object: <http://docs.opencv.org/modules/objdetect/doc/cascade_classification.html>

3. Raspberry Pi Camera module document:

<http://www.ics.com/blog/raspberry-pi-camera-module#.VDqccvldX6U>

4. GPIO library for Raspberry:

<http://www.airspayce.com/mikem/bcm2835/>

5. Setup rasbian OS:

http://chiaseprojects.blogspot.com/2014/06/huong-dan-cai-at-he-ieu-hanh-raspbian.html

6. Development in/out with raspberry:

http://codientu.org/threads/10519/

7. UART raspberry guideline:

<http://www.raspberry-projects.com/pi/programming-in-c/uart-serial-port/using-the-uart>