ArduCAM-Mini

Multi-Camera Adapter Board User Guide

Rev 1.1, Oct 2015

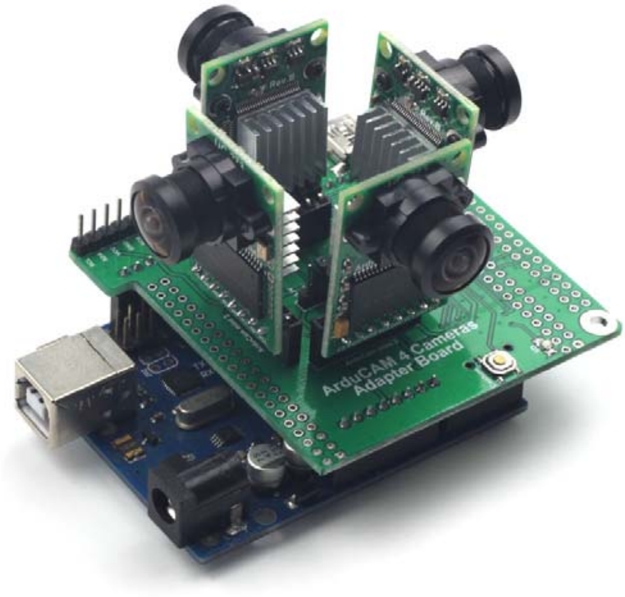


Table of Contents

1 Introduction

ArduCAM-Mini multi-camera adapter board is designed for using up to 4 ArduCAM-Mini camera boards on a single microcontroller including the Arduino, BeagleBone Black and Raspberry Pi platform. The adapter board can be well mated with these popular open source hardware platform but not limited to them, you can also connect this adapter board to any platform you are familiar with as long as they have a free I2C and SPI interface.

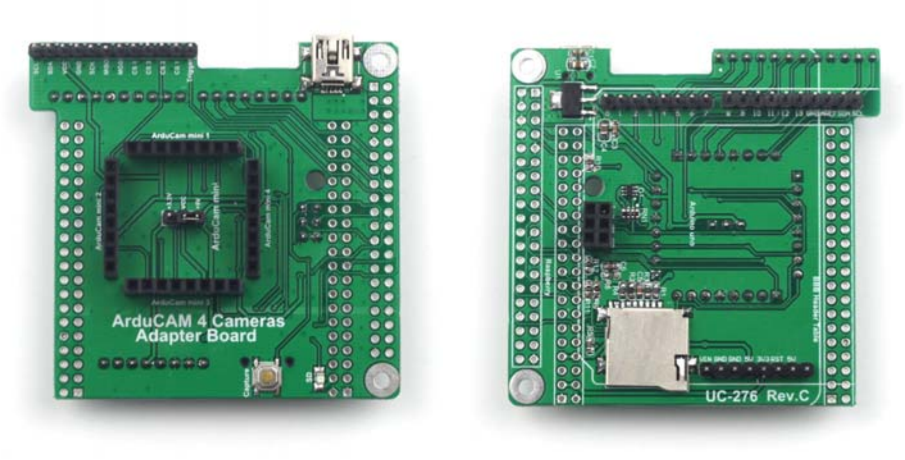


Figure 1 ArduCAM Mini Adapter Board

2 Application

- IoT cameras
- Robot cameras
- Panorama cameras
- Can be used in MCU, Raspberry Pi, ARM, DSP, FPGA platforms

3 Features

- Accommodate up to 4 ArduCAM mini boards
- Onboard SD/TF card storage (only available for Arduino)
- Additional power supply input
- Well mate with Arduino, Raspberry Pi and BeagleBone Black boards

4 Pin Definition

Table 1 lists the pins used for Arduino, Raspberry Pi and BeagleBone Black boards. Please note that the SD_CS signal is only available for Arduino, because the Raspberry Pi and BeagleBone Black have their own SD card.

Table 1 ArduCAM-Mini Adapter Pin Definition

PIN NAME	TYPE	Arduino	Raspberry Pi	BeagleBone Black
Trigger	Input	D2	GPIO8	GPIO3_21 (P9 Pin25)
CS1	Input	D4	GPIO17	GPIO1_28(P9 Pin12)
CS2	Input	D5	GPIO23	GPIO1_16(P9 Pin15)
CS3	Input	D6	GPIO22	GPIO3_19(P9 Pin27)
CS4	Input	D7	GPIO24	GPIO1_17(P9 Pin23)
MOSI	Input	MOSI	GPIO10/MOSI	SPI0_D1(P9 Pin18)
MISO	Output	MISO	GPIO9/MISO	SPI0_D(P9 Pin21)
SCLK	Input	SCLK	GPIO11/SCLK	SPI0_SCLK(P9 Pin22)
GND	Ground	GND	GND	GND
VCC	POWER	+5V	+5V	VDD_5V
SDA	Bi-directional	SDA	GPIO2/SDA	I2C2_SDA(P9 Pin19)
SCL	Input	SCL	GPIO3/SCL	I2C2_SCL(P9 Pin20)
SD_CS	Input	D9	NA	NA

5 Work with Adapter Board

The ArduCAM-Mini power supply is selected via a jumper in the center of the adapter board. It can work with both 3.3V and 5V supply, it is default set to 5V. An alternative mini USB connector is just a AUX power supply, it should be used when lack of current condition. The LED on the adapter board is the indicator for the SD card write (only applicable for Arduino). There are ready to use source code which can be found from github.org/arducam.

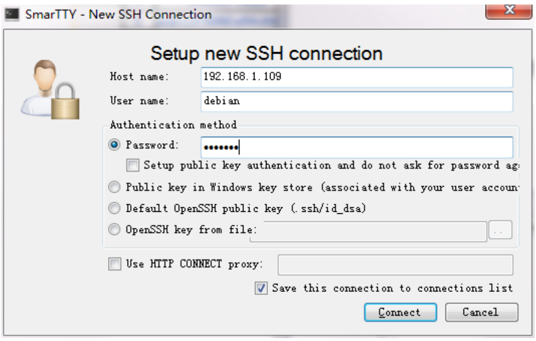
5.1 Arduino

User ArduCAM-Mini adapter board on Arduino is straight forward, just plugging the adapter board on top of the Arduino board, inserting the SD/TF card into the card slot, and plugging the ArduCAM-Mini board on to the adapter board. Then uploading example sketch ArduCAM_Mini_2MP_4Cams_TimeElapse2SD_LowPower or ArduCAM_Mini_5MP_4Cams_TimeElapse2SD_LowPower depending on which ArduCAM-Mini model you are using, it will detect how many cameras are in position and start capture on a defined intervals.

5.2 BeagleBone Black

It is more complicated to use ArduCAM-Mini on BeableBone Black board, we should do some setup before we can take a photo.

Log in the BBB board with any SSH tools you are familiar with, here I use **SmarTTY** for example. The user name is **debian**, password is **temppwd**. The IP address is assigned by your DHCP, you have to check your own IP address.



SmarTTY - New SSH Connection

Setup new SSH connection

Host name: 192.168.1.109

User name: debian

Authentication method

● Password: •••••••

□ Setup public key authentication and do not ask for password ag

○ Public key in Windows key store (associated with your user accoun

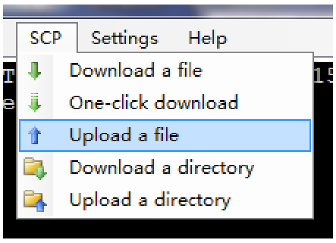
○ Default OpenSSH public key (.ssh/id_dsa)

○ OpenSSH key from file:

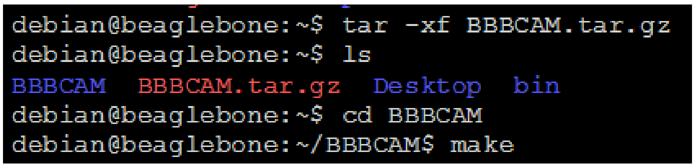
□ Use HTTP CONNECT proxy:

☑ Save this connection to connections list

Click the **SCP->Upload** a directory to upload the source code downloaded from github.

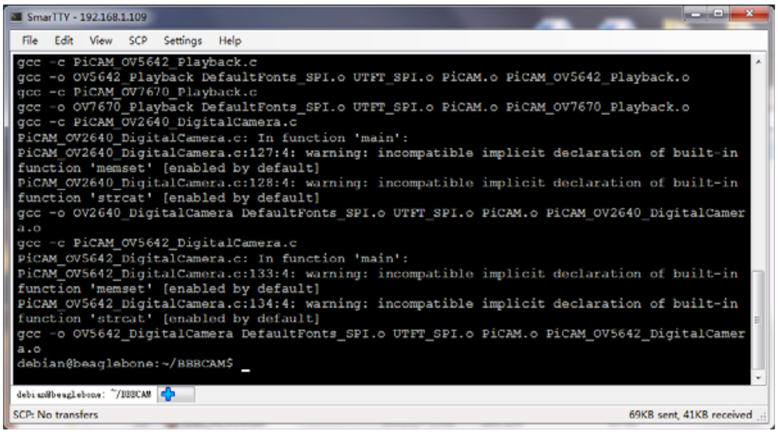


Enter the directory you just uploaded to the BBB board, here use BBBCAM for example, then use **make** command to compile the code.



```
debian@beaglebone:~$ tar -xf BBBCAM.tar.gz
debian@beaglebone:~$ ls
BBBCAM  BBBCAM.tar.gz  Desktop  bin
debian@beaglebone:~$ cd BBBCAM
debian@beaglebone:~/BBBCAM$ make
```

Make sure there is no compilation error like this.



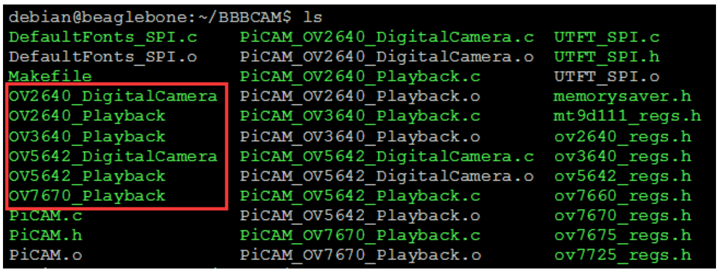
SmarTTY - 192.168.1.109

```
gcc -c PiCAM_OV5642_Playback.c
gcc -o OV5642_Playback DefaultFonts_SPI.o UTFT_SPI.o PiCAM.o PiCAM_OV5642_Playback.o
gcc -c PiCAM_OV7670_Playback.c
gcc -o OV7670_Playback DefaultFonts_SPI.o UTFT_SPI.o PiCAM.o PiCAM_OV7670_Playback.o
gcc -c PiCAM_OV2640_DigitalCamera.c
PiCAM_OV2640_DigitalCamera.c: In function 'main':
PiCAM_OV2640_DigitalCamera.c:127:4: warning: incompatible implicit declaration of built-in
function 'memset' [enabled by default]
PiCAM_OV2640_DigitalCamera.c:128:4: warning: incompatible implicit declaration of built-in
function 'strcat' [enabled by default]
gcc -o OV2640_DigitalCamera DefaultFonts_SPI.o UTFT_SPI.o PiCAM.o PiCAM_OV2640_DigitalCamer
a.o
gcc -c PiCAM_OV5642_DigitalCamera.c
PiCAM_OV5642_DigitalCamera.c: In function 'main':
PiCAM_OV5642_DigitalCamera.c:133:4: warning: incompatible implicit declaration of built-in
function 'memset' [enabled by default]
PiCAM_OV5642_DigitalCamera.c:134:4: warning: incompatible implicit declaration of built-in
function 'strcat' [enabled by default]
gcc -o OV5642_DigitalCamera DefaultFonts_SPI.o UTFT_SPI.o PiCAM.o PiCAM_OV5642_DigitalCamer
a.o
debian@beaglebone:~/BBBCAM$ _
```

SCP: No transfers

69KB sent, 41KB received

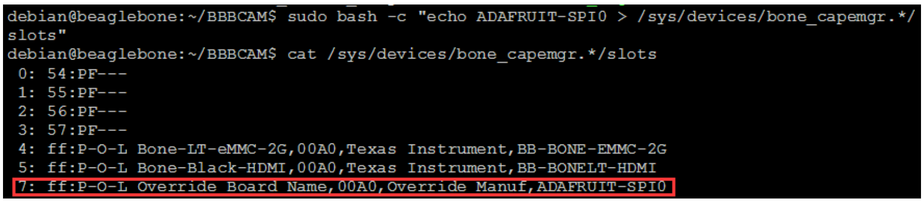
It will generate related examples like this.



```
debian@beaglebone:~/BBBCAM$ ls
DefaultFonts_SPI.c        PiCAM_OV2640_DigitalCamera.c  UTFT_SPI.c
DefaultFonts_SPI.o        PiCAM_OV2640_DigitalCamera.o  UTFT_SPI.h
Makefile                  PiCAM_OV2640_Playback.c       UTFT_SPI.o
OV2640_DigitalCamera      PiCAM_OV2640_Playback.o       memorysaver.h
OV2640_Playback           PiCAM_OV3640_Playback.c       mt9d111_regs.h
OV3640_Playback           PiCAM_OV3640_Playback.o       ov2640_regs.h
OV5642_DigitalCamera      PiCAM_OV5642_DigitalCamera.c  ov3640_regs.h
OV5642_Playback           PiCAM_OV5642_DigitalCamera.o  ov5642_regs.h
OV7670_Playback           PiCAM_OV5642_Playback.c       ov7660_regs.h
PiCAM.c                   PiCAM_OV5642_Playback.o       ov7670_regs.h
PiCAM.h                   PiCAM_OV7670_Playback.c       ov7675_regs.h
PiCAM.o                   PiCAM_OV7670_Playback.o       ov7725_regs.h
```

We use SPI0 hardware pins on BBB board, which is not enabled by default, so we have to enable the SPI0 from the device tree by running the command like this.

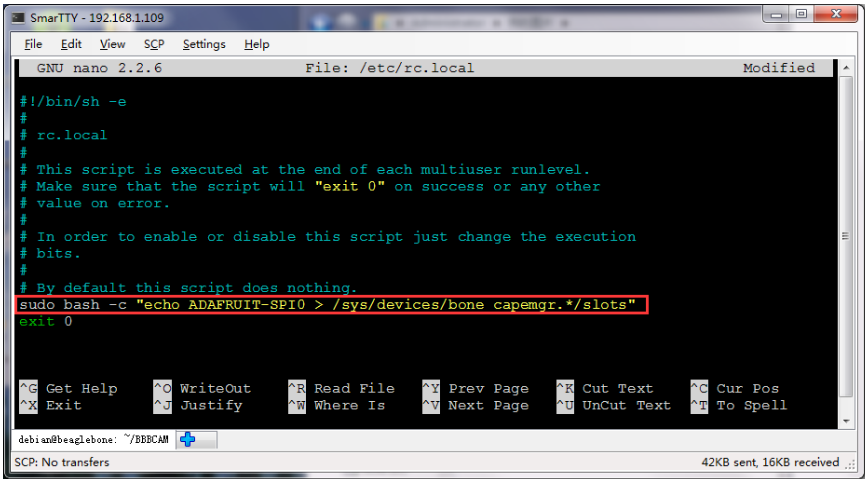
sudo bash -c "echo ADAFRUIT-SPI0 > /sys/devices/bone_capemgr.*/slots"



```
debian@beaglebone:~/BBBCAM$ sudo bash -c "echo ADAFRUIT-SPI0 > /sys/devices/bone_capemgr.*/
slots"
debian@beaglebone:~/BBBCAM$ cat /sys/devices/bone_capemgr.*/slots
 0: 54:PF---
 1: 55:PF---
 2: 56:PF---
 3: 57:PF---
 4: ff:P-O-L Bone-LT-eMMC-2G,00A0,Texas Instrument,BB-BONE-EMMC-2G
 5: ff:P-O-L Bone-Black-HDMI,00A0,Texas Instrument,BB-BONELT-HDMI
 7: ff:P-O-L Override Board Name,00A0,Override Manuf,ADAFRUIT-SPI0
```

If you don't want to run the command each time, you can add it to **rc.local** before the line **exit 0**, then save the file.

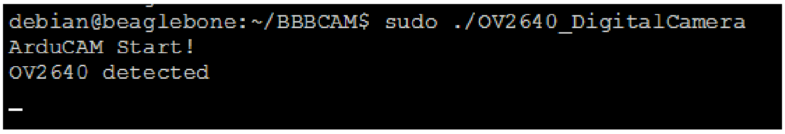
sudo nano /etc/rc.local



```
  GNU nano 2.2.6                 File: /etc/rc.local                          Modified

#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.
sudo bash -c "echo ADAFRUIT-SPI0 > /sys/devices/bone_capemgr.*/slots"
exit 0



^G Get Help    ^O WriteOut    ^R Read File   ^Y Prev Page   ^K Cut Text    ^C Cur Pos
^X Exit        ^J Justify     ^W Where Is    ^V Next Page   ^U UnCut Text  ^T To Spell
```

Now you can run the demos.

sudo ./OV2640_DigitalCamera



```
debian@beaglebone:~/BBBCAM$ sudo ./OV2640_DigitalCamera
ArduCAM Start!
OV2640 detected

_
```

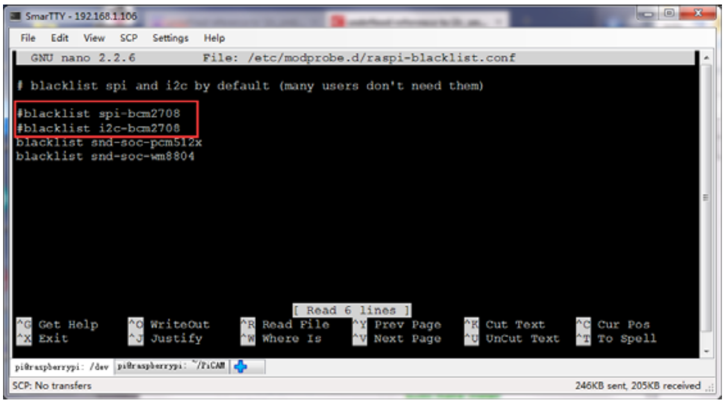
5.3 Raspberry Pi

Same tools as BBB board, we use SmarTTY to log in the Raspberry Pi with user name : **pi**, password: **raspberry .** The IP address is assigned by your DHCP, you have to check your own IP address.

Edit the **raspi-blacklist.conf** file to enable the SPI and I2C hardware on the Rapberry Pi with command

sudo nano /etc/modprobe.d/raspi-blacklist.conf

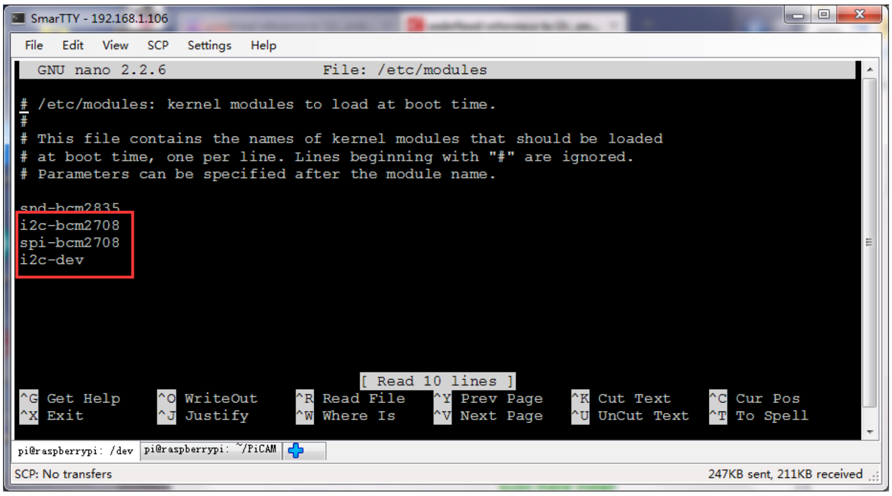
Add # before the blacklist spi-bcm2708 and blacklist i2c-bcm2708, then save.



```
  GNU nano 2.2.6           File: /etc/modprobe.d/raspi-blacklist.conf

# blacklist spi and i2c by default (many users don't need them)

#blacklist spi-bcm2708
#blacklist i2c-bcm2708
blacklist snd-soc-pcm512x
blacklist snd-soc-wm8804



                                     [ Read 6 lines ]
^G Get Help    ^O WriteOut    ^R Read File   ^Y Prev Page   ^K Cut Text    ^C Cur Pos
^X Exit        ^J Justify     ^W Where Is    ^V Next Page   ^U UnCut Text  ^T To Spell
```

Open file /etc/modules to add three lines, like follows,

sudo nano /etc/modules

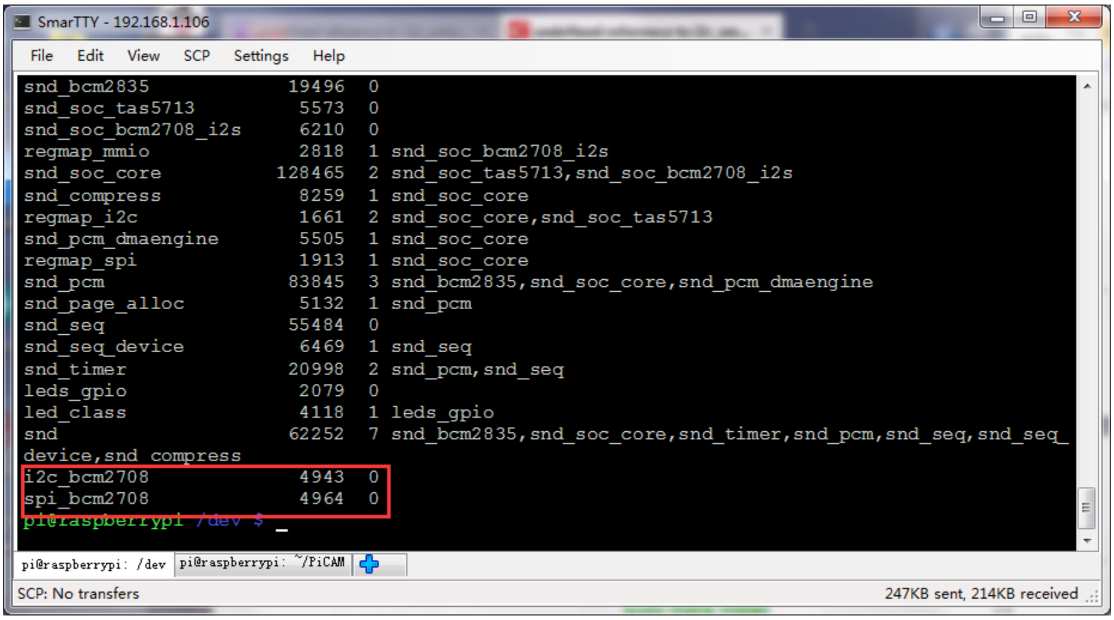


```
  GNU nano 2.2.6                File: /etc/modules

# /etc/modules: kernel modules to load at boot time.
#
# This file contains the names of kernel modules that should be loaded
# at boot time, one per line. Lines beginning with "#" are ignored.
# Parameters can be specified after the module name.

snd-bcm2835
i2c-bcm2708
spi-bcm2708
i2c-dev



                              [ Read 10 lines ]
^G Get Help   ^O WriteOut   ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit       ^J Justify    ^W Where Is   ^V Next Page  ^U UnCut Text ^T To Spell
```

then reboot the Pi.

sudo reboot

After reboot use **lsmod** command to check if the driver is properly loaded.



```
snd_bcm2835             19496  0
snd_soc_tas5713          5573  0
snd_soc_bcm2708_i2s      6210  0
regmap_mmio              2818  1 snd_soc_bcm2708_i2s
snd_soc_core           128465  2 snd_soc_tas5713,snd_soc_bcm2708_i2s
snd_compress             8259  1 snd_soc_core
regmap_i2c               1661  2 snd_soc_core,snd_soc_tas5713
snd_pcm_dmaengine        5505  1 snd_soc_core
regmap_spi               1913  1 snd_soc_core
snd_pcm                 83845  3 snd_bcm2835,snd_soc_core,snd_pcm_dmaengine
snd_page_alloc           5132  1 snd_pcm
snd_seq                 55484  0
snd_seq_device           6469  1 snd_seq
snd_timer               20998  2 snd_pcm,snd_seq
leds_gpio                2079  0
led_class                4118  1 leds_gpio
snd                     62252  7 snd_bcm2835,snd_soc_core,snd_timer,snd_pcm,snd_seq,snd_seq_
device,snd_compress
i2c_bcm2708              4943  0
spi_bcm2708              4964  0
pi@raspberrypi /dev $ _
```

Before install the **wringPi** and ArduCAM, please install the **git** first.

sudo apt-get install git-core

If there is error message, please try to update the Raspbain, using command below:

sudo apt-get update

sudo apt-get upgrade

Fetching the **wiringPi** by **git** command

git clone git://git.drogon.net/wiringPi

If need up to date **wiringPi**, run the following commands:

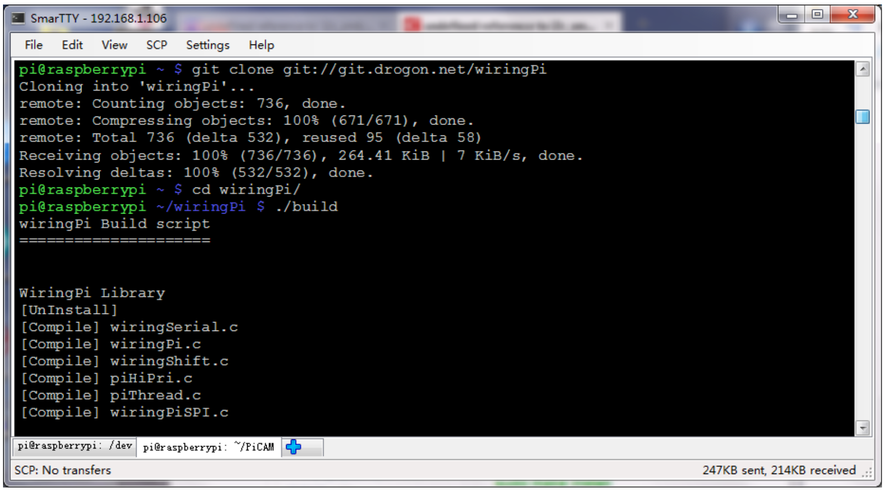
cd wiringPi

git pull origin

Next enter the **wiringPi** directory to compile and install the package as follows:

cd wiringPi

./build

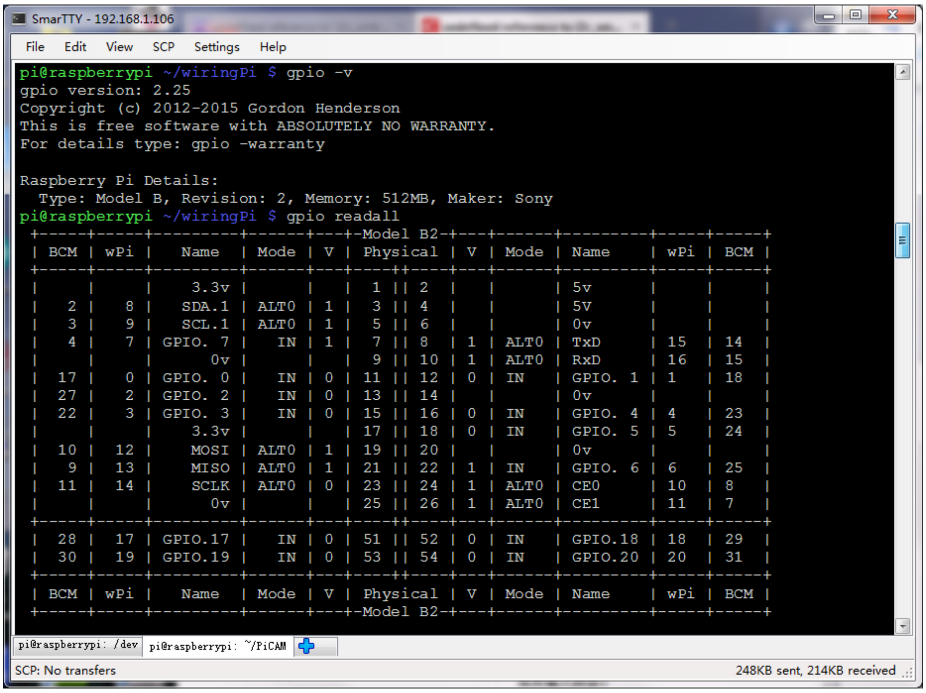


```
pi@raspberrypi ~ $ git clone git://git.drogon.net/wiringPi
Cloning into 'wiringPi'...
remote: Counting objects: 736, done.
remote: Compressing objects: 100% (671/671), done.
remote: Total 736 (delta 532), reused 95 (delta 58)
Receiving objects: 100% (736/736), 264.41 KiB | 7 KiB/s, done.
Resolving deltas: 100% (532/532), done.
pi@raspberrypi ~ $ cd wiringPi/
pi@raspberrypi ~/wiringPi $ ./build
wiringPi Build script
=====================


WiringPi Library
[UnInstall]
[Compile] wiringSerial.c
[Compile] wiringPi.c
[Compile] wiringShift.c
[Compile] piHiPri.c
[Compile] piThread.c
[Compile] wiringPiSPI.c
```

Using the following command to check if the wiringPi is properly installed.

gpio -v

gpio readall



```
pi@raspberrypi ~/wiringPi $ gpio -v
gpio version: 2.25
Copyright (c) 2012-2015 Gordon Henderson
This is free software with ABSOLUTELY NO WARRANTY.
For details type: gpio -warranty

Raspberry Pi Details:
  Type: Model B, Revision: 2, Memory: 512MB, Maker: Sony
pi@raspberrypi ~/wiringPi $ gpio readall
 +-----+-----+---------+------+---+-Model B2-+---+------+---------+-----+-----+
 | BCM | wPi |   Name  | Mode | V | Physical | V | Mode | Name    | wPi | BCM |
 +-----+-----+---------+------+---+----++----+---+------+---------+-----+-----+
 |     |     |    3.3v |      |   |  1 || 2  |   |      | 5v      |     |     |
 |   2 |   8 |   SDA.1 | ALT0 | 1 |  3 || 4  |   |      | 5V      |     |     |
 |   3 |   9 |   SCL.1 | ALT0 | 1 |  5 || 6  |   |      | 0v      |     |     |
 |   4 |   7 | GPIO. 7 |   IN | 1 |  7 || 8  | 1 | ALT0 | TxD     | 15  | 14  |
 |     |     |      0v |      |   |  9 || 10 | 1 | ALT0 | RxD     | 16  | 15  |
 |  17 |   0 | GPIO. 0 |   IN | 0 | 11 || 12 | 0 | IN   | GPIO. 1 | 1   | 18  |
 |  27 |   2 | GPIO. 2 |   IN | 0 | 13 || 14 |   |      | 0v      |     |     |
 |  22 |   3 | GPIO. 3 |   IN | 0 | 15 || 16 | 0 | IN   | GPIO. 4 | 4   | 23  |
 |     |     |    3.3v |      |   | 17 || 18 | 0 | IN   | GPIO. 5 | 5   | 24  |
 |  10 |  12 |    MOSI | ALT0 | 1 | 19 || 20 |   |      | 0v      |     |     |
 |   9 |  13 |    MISO | ALT0 | 1 | 21 || 22 | 1 | IN   | GPIO. 6 | 6   | 25  |
 |  11 |  14 |    SCLK | ALT0 | 0 | 23 || 24 | 1 | ALT0 | CE0     | 10  | 8   |
 |     |     |      0v |      |   | 25 || 26 | 1 | ALT0 | CE1     | 11  | 7   |
 +-----+-----+---------+------+---+----++----+---+------+---------+-----+-----+
 |  28 |  17 | GPIO.17 |   IN | 0 | 51 || 52 | 0 | IN   | GPIO.18 | 18  | 29  |
 |  30 |  19 | GPIO.19 |   IN | 0 | 53 || 54 | 0 | IN   | GPIO.20 | 20  | 31  |
 +-----+-----+---------+------+---+----++----+---+------+---------+-----+-----+
 | BCM | wPi |   Name  | Mode | V | Physical | V | Mode | Name    | wPi | BCM |
 +-----+-----+---------+------+---+-Model B2-+---+------+---------+-----+-----+
```

Download the ArduCAM library with github.

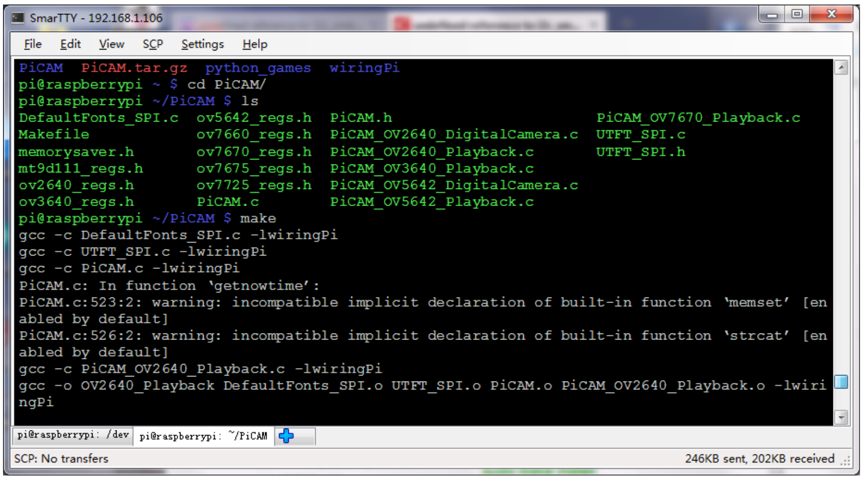
git clone https://github.com/ArduCAM/RaspberryPi.git

Compile the ArduCAM library

cd RaspberryPi/ArduCAM4Pi

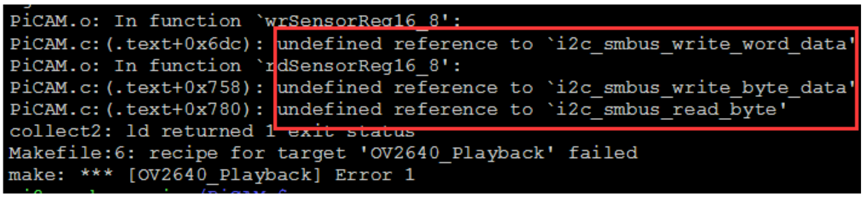
Compile the code using **make**

make



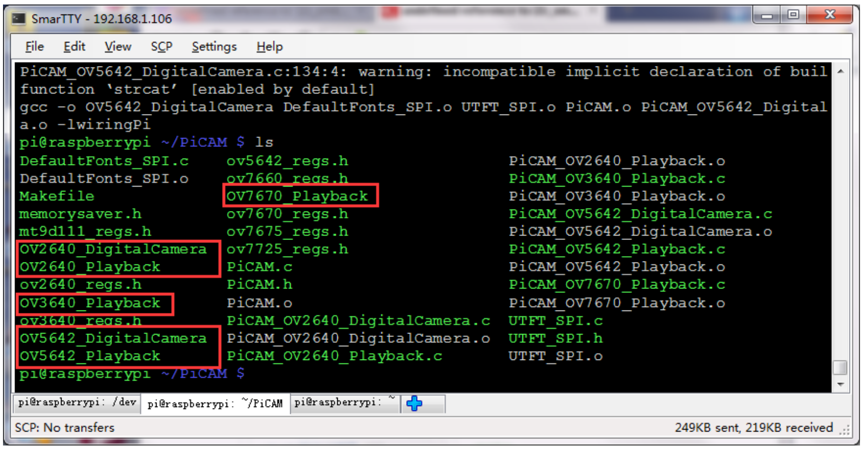
```
PiCAM  PiCAM.tar.gz  python_games  wiringPi
pi@raspberrypi ~ $ cd PiCAM/
pi@raspberrypi ~/PiCAM $ ls
DefaultFonts_SPI.c  ov5642_regs.h  PiCAM.h                        PiCAM_OV7670_Playback.c
Makefile            ov7660_regs.h  PiCAM_OV2640_DigitalCamera.c  UTFT_SPI.c
memorysaver.h       ov7670_regs.h  PiCAM_OV2640_Playback.c       UTFT_SPI.h
mt9d111_regs.h      ov7675_regs.h  PiCAM_OV3640_Playback.c
ov2640_regs.h       ov7725_regs.h  PiCAM_OV5642_DigitalCamera.c
ov3640_regs.h       PiCAM.c        PiCAM_OV5642_Playback.c
pi@raspberrypi ~/PiCAM $ make
gcc -c DefaultFonts_SPI.c -lwiringPi
gcc -c UTFT_SPI.c -lwiringPi
gcc -c PiCAM.c -lwiringPi
PiCAM.c: In function 'getnowtime':
PiCAM.c:523:2: warning: incompatible implicit declaration of built-in function 'memset' [en
abled by default]
PiCAM.c:526:2: warning: incompatible implicit declaration of built-in function 'strcat' [en
abled by default]
gcc -c PiCAM_OV2640_Playback.c -lwiringPi
gcc -o OV2640_Playback DefaultFonts_SPI.o UTFT_SPI.o PiCAM.o PiCAM_OV2640_Playback.o -lwiri
ngPi
```

If report “undefined reference to `i2c_smbus_write_word_data' error message， you have to reinstall the i2c-tools，libi2c-dev ，python-smbus packages, then recompile. It will generate all the demo program marked as red square box.



```
PiCAM.o: In function `wrSensorReg16_8':
PiCAM.c:(.text+0x6dc): undefined reference to `i2c_smbus_write_word_data'
PiCAM.o: In function `rdSensorReg16_8':
PiCAM.c:(.text+0x758): undefined reference to `i2c_smbus_write_byte_data'
PiCAM.c:(.text+0x780): undefined reference to `i2c_smbus_read_byte'
collect2: ld returned 1 exit status
Makefile:6: recipe for target 'OV2640_Playback' failed
make: *** [OV2640_Playback] Error 1
```

sudo apt-get install i2c-toolslibi2c-devpython-smbus



```
PiCAM_OV5642_DigitalCamera.c:134:4: warning: incompatible implicit declaration of buil
function 'strcat' [enabled by default]
gcc -o OV5642_DigitalCamera DefaultFonts_SPI.o UTFT_SPI.o PiCAM.o PiCAM_OV5642_Digital
a.o -lwiringPi
pi@raspberrypi ~/PiCAM $ ls
DefaultFonts_SPI.c    ov5642_regs.h                 PiCAM_OV2640_Playback.o
DefaultFonts_SPI.o    ov7660_regs.h                 PiCAM_OV3640_Playback.c
Makefile              OV7670_Playback               PiCAM_OV3640_Playback.o
memorysaver.h         ov7670_regs.h                 PiCAM_OV5642_DigitalCamera.c
mt9d111_regs.h        ov7675_regs.h                 PiCAM_OV5642_DigitalCamera.o
OV2640_DigitalCamera  ov7725_regs.h                 PiCAM_OV5642_Playback.c
OV2640_Playback       PiCAM.c                       PiCAM_OV5642_Playback.o
ov2640_regs.h         PiCAM.h                       PiCAM_OV7670_Playback.c
OV3640_Playback       PiCAM.o                       PiCAM_OV7670_Playback.o
ov3640_regs.h         PiCAM_OV2640_DigitalCamera.c  UTFT_SPI.c
OV5642_DigitalCamera  PiCAM_OV2640_DigitalCamera.o  UTFT_SPI.h
OV5642_Playback       PiCAM_OV2640_Playback.c       UTFT_SPI.o
pi@raspberrypi ~/PiCAM $
```

Run the demo.

sudo ./OV2640_DigitalCamera