CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

The Raspberry Piis a credit-card-sized single-board computer developed in the UK by the Raspberry Pi Foundation with the intention of promoting the teaching of basic computer science in schools. The Raspberry Pi is manufactured through licensed manufacturing deals with Newark element14 (Premier Farnell), RS Components and Ego man. All of these companies sell the Raspberry Pi online. The hardware is the same across all manufacturers.

The Raspberry Pi has a Broadcom BCM2835 system on a chip (SOC), which includes an ARM1176JZF-S 700MHz processor (The firmware includes a number of "Turbo" modes so that the user can attempt over clocking, up to 1 GHz, without affecting the warranty), VideoCoreIV GPU, and was originally shipped with 256 megabytes of RAM, later upgraded to 512 MB. It does not include a built-in hard disk or solid-state drive, but uses an SD card for booting and long-term storage.

The Raspberry Pi does not come with a real-time clock, so an OS must use a network time server, or ask the user for time information at boot time to get access to time and date for file time and date stamping. However, a real-time clock (such as the DS1307) with battery backup can be added via the I^{2}C interface.

1.2EXISTING SYSTEM

In the existing system, there is no automatic detection of capturing the number plate, and extraction of the textual data from the image.In past or earlier days more traditional means of license recognition access control were used. It is all the manual process of number plate extraction, to avoid this we are using this number plate extraction technique using the raspberry pi.

1.3 PROPOSED SYSTEM

The project aims at designing a system which captures the image of the number plate automatically of a vehicle and these details were verified using Raspberry Pi processor. The image was given to the optical character recognition technique algorithm to detect/ extract the textual data from the image. The OCR is in-built to the raspberry pi processor. This paper makes use of an onboard computer, which is commonly termed as Raspberry Pi processor. It acts as heart of the project.

1.4 SPECIFICATIONS OF THE PROJECT

Project description:

The paper aims at designing a system which automatically captures the image of the number plate of a vehicle and these details were verified using Raspberry Pi processor. Automation is the most frequently spelled term in the field of electronics.

This projectmakes use of an onboardcomputer, which is commonly termed as Raspberry Pi processor. The onboardcomputer can efficiently communicate with the output and input modules which are being used. The Raspberry Pi is a credit-card-sized single-board computer developed in the UK by the Raspberry Pi Foundation. The Raspberry Pi has a Broadcom BCM2835 system on a chip (SOC), which includes an ARM1176JZF-S 700 MHz processor, Video Core IV GPU, and was originally shipped with 256 megabytes of RAM, later upgraded to 512 MB. It does not include a built-in hard disk or solid-state drive, but uses an SD card for booting and long-term storage. The device which is able to perform the task is a Raspberry Pi processor. When any vehicle passes by the system, the image of the number plate of every vehicle is captured using camera. The image of the number plate details are fed as input to the Raspberry Pi processor. The Processor takes responsibility to check the details of every vehicle. To perform this task, Raspberry Pi processor is programmed using embedded ‘Linux’.

CHAPTER 2

LITERATURE SURVEY

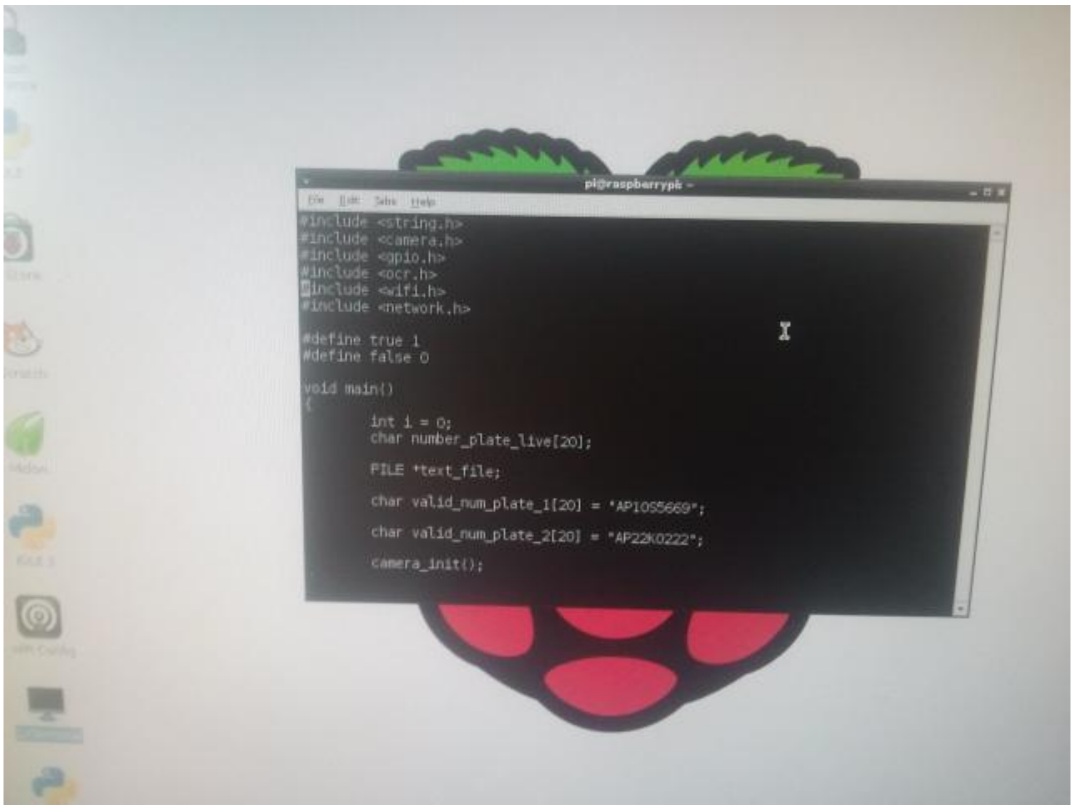
In this chapter, we will discuss about the information found by study and research that is critical and have an important value in the contribution of the whole project. It also gives some basic knowledge or theoretical base and is used as a foundation to successfully achieve the main objectives. Most of the literatures are from the related articles, journals, books and previous works of the same fields. These literatures are then compiled and use as a guidance to the work of this project.

In past or earlier days more traditional means of license recognition access control were used. Automatic number plate recognition ANPR system is a mass surveillance method that uses optical character recognition on images to read vehicle registration plates. They can also be used at an existing closed-circuit television or road-rule for enforcement cameras, or ones specifically designed for the task. They are used by various police forces and as a method of electronic toll collection on pay-per-use roads and cataloging the movements of traffic or individuals.

2.1 RELATED WORK

Automation is the most frequently spelled term in the field of electronics. The hunger for automation brought many revolutions in the existing technologies. This project makes use of an onboard computer, which is commonly termed as Raspberry Pi processor. It acts as heart of the project. This onboard computer can efficiently communicate with the output and input modules which are being used. Aforementioned identification or recognition process using raspberry pi processor will change slightly between different products and systems. These Standard systems are comprised of a USB camerafor the automated information resource of the License plate adds a new vital dimension to decision-making for Access control at toll gates and traffic junctions. License plate recognition system can be easily integrated with any physical access control device like boom barriers and sliding gates for seamless access. To perform this task, Raspberry Pi processor is programmed using embedded “Linux‟.

Linux is a Unix-like computer operating system assembled under the model of free and open source software development and distribution. The defining component of Linux is the Linux kernel, an operating system. The Linux Standard Base (LSB) is a joint project by several Linux distributions and is based on the POSIX specification, the Single UNIX Specification, and several other open standards, but extends them in certain areas.



```
#include <string.h>
#include <camera.h>
#include <gpio.h>
#include <ocr.h>
#include <wifi.h>
#include <network.h>

#define true 1
#define false 0

void main()
{
        int i = 0;
        char number_plate_live[20];

        FILE *text_file;

        char valid_num_plate_1[20] = "AP10S5669";

        char valid_num_plate_2[20] = "AP22K0222";

        camera_init();
```

Fig:2.1.1Embedded Linux programming

1. LINUXOperating System

The Linux open source operating system, orLinux OS, is a freely distributable, cross-platform operating system based on Unix that can be installed on PCs, laptops, net books, mobile and tablet devices, video game consoles, servers, supercomputers and more.

2. Qt for Embedded Linux

Qt for Embedded Linux is a C++ framework for GUI and application development for embedded devices. It runs on a variety of processors, usually with Embedded Linux. Qt for Embedded Linux provides the standard QtAPI for embedded devices with a lightweight window system.

3. OPEN CV

Open CV is an open source computer vision library originally developed by Intel. It is free for commercial and research use under a BSD (Berkeley Software Distribution) license. The library is cross-platform, and runs on Linux,Windowsand Mac OS. It focuses mainly towards real-time image processing, as such, if it finds Intel's Integrated Performance Primitives on the system, it will use these commercial optimized routines to accelerate itself.

4. USB camera

A webcam or USB camera is a video camera that feeds its image in real time to a computer or computer network. Unlike an IP camera which uses a direct connection usingEthernet or Wi-Fi, a USB camera is generally connected by a USB cable, FireWire cable, or similar cable. The common use as a video camera for the World Wide Web gave the webcam its name. Other popular uses include security surveillance, computer vision, video broadcasting, and for recording social videos. Webcams are known for their low manufacturing cost and flexibility, making them the lowest cost form of video telephony. They have also become a source of security and privacy issues, as some built-in webcams can be remotely activated via spyware.

5. Raspberry Pi processor

In the Proposed ANPR system we used the **Raspberry Pi** is a credit-card-sized single-board computer developed in the UK by the Raspberry Pi Foundation. The Raspberry Pi has a Broadcom BCM2835 system on a chip (SOC), which includes an ARM1176JZF-S 700MHz processor, Video Core IV GPU, and was originally shipped with 256 megabytes of RAM, later upgraded to 512 MB. It does not include a built-in hard disk or solid-state drive, but uses an SD card for booting and long-term storage.

2.2 SIMILAR TYPE PROJECTS

2.2.1 Vision Based Assistive System for Label Detection with Voice Output

A camera based assistive text readingframework to help blind persons read text labels andproductpackagingfrom hand-held object in theirdaily resides is proposed. To isolate the object fromcluttered backgrounds or other surroundings objectsin the camera view, we propose an efficient andeffective motion based method to define a region ofinterest (ROI) in the video by asking the user to shakethe object. In the extracted ROI, text localization andrecognition are conducted to acquire text information.To automatically localize the text regions from theobject ROI, we propose a novel text localizationalgorithm by learning gradient features of strokeorientations and distributions of edge pixels in anAdaboostmodel. Text characters in the localized textregions are then binarized and recognized by off-the-shelfoptical character recognition software. Therecognized text codes are output to blind users inspeech.

2.2.2License Plate Extractionof Images Using Raspberry Pi

The paper aims at designing a system which captures the image of the number plate automatically of a vehicle and these details were verified using Raspberry Pi processor for authentication. The system also alerts the authorities when any unauthorized image of number plate is detected using buzzer alarm system. Automation is the most frequently spelled term in the field of electronics. The hunger for automation brought many revolutions in the existing technologies. This paper makes use of an onboard computer, which is commonly termed as Raspberry Pi processor.

The device which is able to perform the task is a Raspberry Pi processor. When any vehicle passes by the system, the image of the number plate of every vehicle is captured using camera. The image of the number plate details are fed as input to the Raspberry Pi processor. The Processor takes responsibility to check the authentication details of every vehicle. Once the vehicle details are recognized then the processor operates the gate using stepper motor. The system also alerts the user through buzzer alarm whenever it detects an unauthorized image of number plate. Toper form this task, Raspberry Pi processor is programmed using embedded ‘Linux’.

CHAPTER 3

ANALYSIS AND DESIGN

3.1 INTRODUCTION TO EMBEDDED SYSTEMS

An embedded system can be defined as a computing device that does a specific focused job. Appliances such as the air-conditioner, VCD player, DVD player, printer, fax machine, mobile phone etc. are examples of embedded systems. Each of these appliances will have a processor and special hardwareto meet the specific requirement of the application along with the embedded software that is executed by the processor for meeting that specific requirement. The embedded software is also called “firm ware”. The desktop/laptop computer is a general purpose computer. You can use it for a variety of applications such as playing games, *word*processing, accounting, software development and so on. In contrast, the software in the embedded systems is always fixed listed below.

Embedded systems do a very specific task, they cannot be programmed to do different things. Embedded systems have very limited resources, particularly the memory. Generally, they do not have secondary storage devices such as the CDROM or the floppy disk. Embedded systems have to work against some deadlines. A specific job has to be completed within a specific time. In some embedded systems, called real-time systems, the deadlines are stringent. Missing a deadline may cause a catastrophe-loss of life or damage to property. Embedded systems are constrained for power. As many embedded systems operate through a battery, the power consumption has to be very low.

Some embedded systems have to operate in extreme environmental conditions such as very high temperatures and humidity.

3.2 APPLICATION AREAS

Nearly 99 percent of the processors manufactured end up in embedded systems. The embedded system market is one of the highest growth areas as these systems are used in very market segment- consumer electronics, office automation, industrial automation, biomedical engineering, wireless communication, data communication, telecommunications, transportation, military and so on.

3.2.1 Consumer appliances

At home we use a number of embedded systems which include digital camera, digital diary, DVD player, electronic toys, microwave oven, remote controls for TV and air-conditioner, VCDplayer, video game consoles, video recorders etc. Today’s high-tech car has about 20 embedded systems for transmission control, engine spark control, air-conditioning, navigation etc. Even wristwatches are nowbecoming embedded systems. The palmtops are powerful embedded systems using which we can carry out many general-purpose tasks such as playing games and word processing.

3.2.2 Office automation

Theoffice automation products using embedded systems are copying machine, fax machine, key telephone, modem, printer, scanner etc.

3.2.3 Industrial automation

Today a lot of industries use embedded systems for process control. These include pharmaceutical, cement, sugar, oil exploration, nuclear energy, electricity generation and transmission. The embedded systems for industrial use are designed to carry out specific tasks such as monitoring the temperature, pressure, humidity, voltage, current etc., and then take appropriate action based on the monitored levels to control other devices or to send information to a centralized monitoring station. In hazardous industrial environment, where human presence has to be avoided, robots are used, which are programmed to do specific jobs. The robots are now becoming very powerful and carry out many interesting and complicated tasks such as hardware assembly.

3.2.4 Medical electronics

Almost every medical equipment in the hospital is an embedded system. These equipmentsinclude diagnostic aids such as ECG, EEG, blood pressure measuring devices, X-ray scanners; equipment used in blood analysis, radiation, colonoscopy, endoscopy etc. Developments in medical electronics have paved way for more accurate diagnosis of diseases.

3.2.5 Computer networking

Computer networkingproducts such as bridges, routers, Integrated Services Digital Networks (ISDN), Asynchronous Transfer Mode (ATM), X.25 and frame relay switches are embedded systems which implement the necessary data communication protocols. For example, a router interconnects two networks. The two networks may be running different protocol stacks. The router's function is to obtain the data packets from incoming pores, analyze the packets and send them towards the destination after doing necessary protocol conversion. Most networking equipments, other than the end systems (desktop computers) we use to access the networks, are embedded systems.

3.3 INTRODUCTION TO MICROCONTROLLER

Based on the Processor side Embedded Systems is mainly divided into 3 types.

1. Micro Processor : - are for general purpose eg: our personal computer.

2. Micro Controller:- are for specific applications, because of cheaper cost we will go for these.

3. DSP (Digital Signal Processor):- are for high and sensitive application purpose.

3.4 MICROCONTROLLER VERSUS MICROPROCESSOR

A systemdesigner using a general-purpose microprocessor such as the Pentium or the 68040 must add RAM, ROM, I/O ports, and timers externally to make them functional. Although the addition of external RAM, ROM, and I/O ports makes these systems bulkier and much more expensive, they have the advantage of versatility such that the designer can decide on the amount of RAM, ROM and I/O ports needed to fit the task at hand.

A Microcontroller has a CPU (a microprocessor) in addition to a fixed amount of RAM, ROM, I/O ports, and a timer all on a single chip.Therefore, the designer cannot add any external memory, I/O ports, or timer to it. The fixed amount of on-chip ROM, RAM, and number of I/O ports in Microcontrollers makes them ideal for many applicationsin which cost and space are critical.

General Micro Processor

1. cpu for computers
2. No RAM, ROM, I/O on CPU chip itself
3. Example : Intel's x86

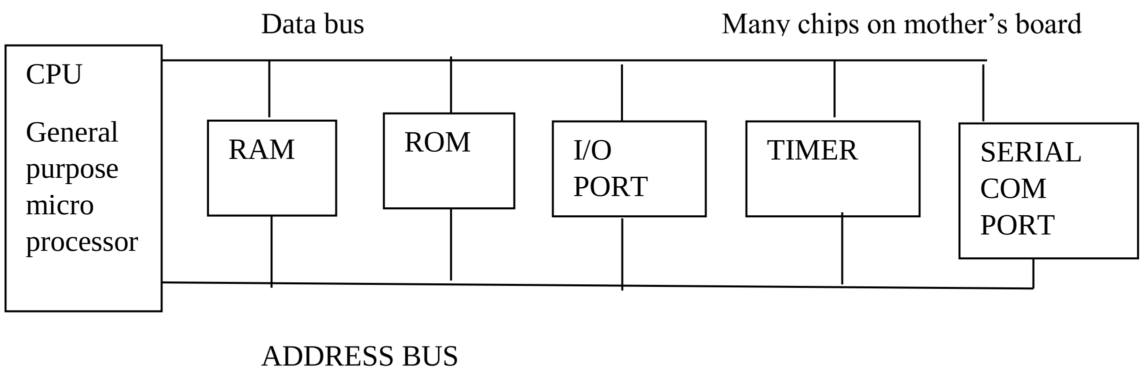


Fig:3.4.1 Micro Processor

Microcontroller

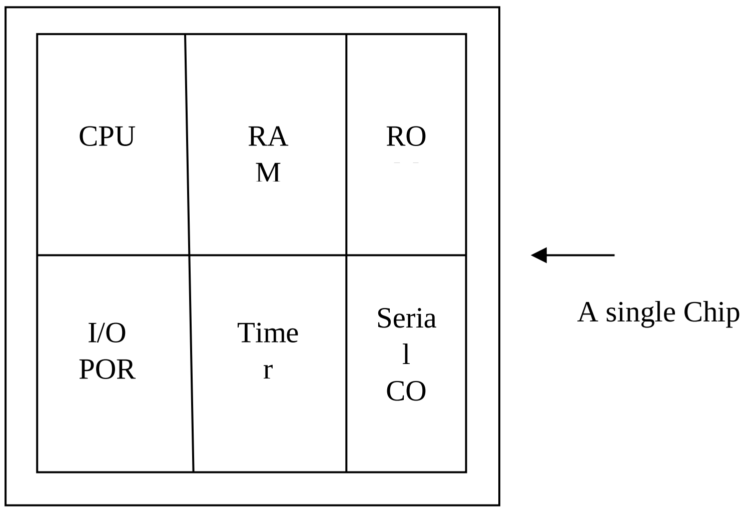


Fig:3.4.2 Microcontroller

A smaller Computer

On chip RAM, ROM, I/O PORTS……..

Example: Intel 8052 etc….

Microprocessor vs. Microcontroller	
Microprocessor	**Microcontroller**
CPU is stand alone RAM, ROM, I/O, timer are separate	CPU,RAM,ROM,I/O and timer are all on a single chip

Designer can decide on the amount of ROM, RAM andI/O ports.	Fix amount of on chip ROM,RAM, I/O Ports.
Expansive, Versatility	For applications in which cost, power and space are critical
General purpose	Single purpose

Table:3.4.1 Microprocessor Versus Microcontroller

3.5 Project Design

The implementation of the project design can be divided in two parts.

- Hardware implementation
- Firmware implementation

Hardware implementation deals in drawing the schematic on the plane paper according to the application, testing the schematic design over the breadboard using the various IC’s to find if the design meets the objective, carrying out the PCB layout of the schematic tested on breadboard, finally preparing the board and testing the designed hardware.

The project design and principle are explained in this chapter using the block diagram. The block diagram discusses about the required components of the design and working condition.

3.5.1 Block Diagram of the Project

The block diagram of the design is as shown in Fig 3.5.1. It consists of Raspberry pi processor, USB camera, SD card,Monitor,Power Supply.

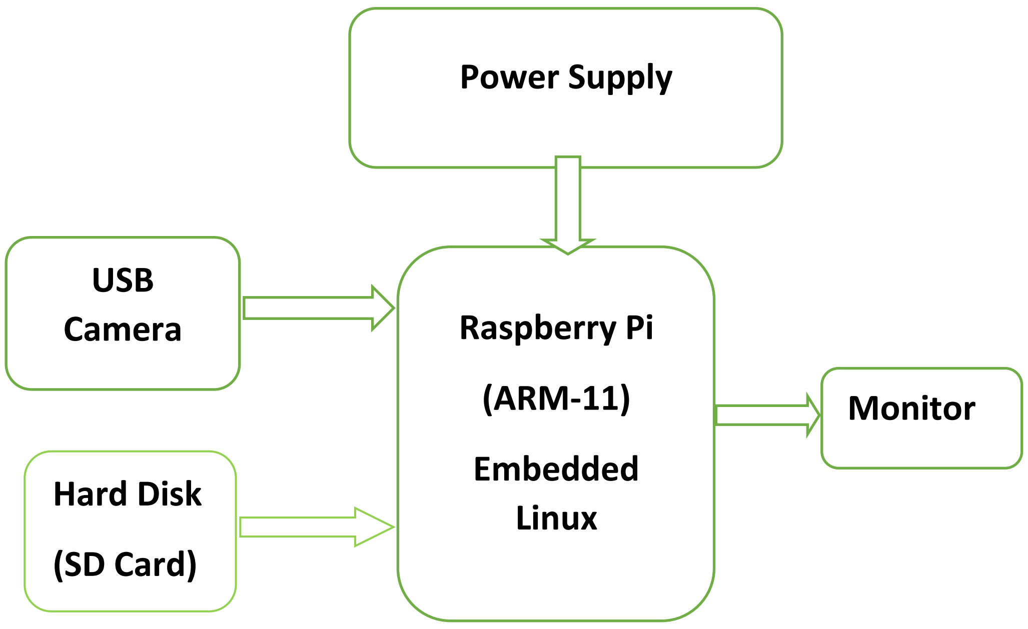


Fig:3.5.1 Block Diagram Of The Project

CHAPTER 4

IMPLEMENTATION

HARDWARE IMPLEMENTATION OF THE PROJECT

This chapter briefly explains about the Hardware Implementation of the project. It explains the features, timer programming, serial communication, interrupts ofprocessor. It also explains the various modules used in this project.

4.1 USBCamera

Fig:4.1.1 USB Camera

In our project camera is using to continuously streaming the video of located place. The specifications are shown below.

Specifications:

- Built-in mic with
- noise reduction
- Interpolated to 25 Mega Pixels
- 10 Level Zoom on live Motion Picture
- Special Visual Effects

Basic Requirements:

- 1 GHz
- 512 MB RAM or more

- 200 MB hard drive space
- Internet connection
- USB 1.1 port (2.0 recommended)

Supports:

Windows Vista, Windows 7 (32-bit or 64-bit) or Windows 8

Package Contents:

- Web Camera
- CD Driver
- Manual

4.2 Raspberry Pi

The Raspberry Pi is a low cost, **credit-card sized computer** that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing, and to learn how to program in languages like Scratch and Python. It's capable of doing everything you'd expect a desktop computer to do, from browsing the internet and playing high-definition video, to making spreadsheets, word-processing, and playing games.

What's more, the Raspberry Pi has the ability to interact with the outside world, and has been used in a wide array of digital maker projects, from music machines and parent detectors to weather stations and tweeting birdhouses with infra-red cameras.
There are currently four Raspberry Pi models. They are the Model A, the Model B, theModel B+ and the Compute Module. All models use the same CPU, the BCM2835, but other hardware features differ.

4.2.1THE MODEL B+

Released in July 2014, the Model B+ is a updated revision of the Model B. It increases the number of USB ports to 4 and the number of pins on the GPIO header to 40. In addition, it has improved power circuitry which allows higher powered USB devices to be attached and now hotplugged. The full size composite video connector has been removed and the functionality moved to the 3.5mm audio/video jack. The full size SD card slot has also been replaced with a much more robust microSD slot.

The following list details some of the improvements over the Model B.

- Current monitors on the USB ports mean the B+ now supports hot plugging.
- Current limiter on the 5V for HDMI means HDMI cable powered VGA converters will now all work
- 14 more GPIO pins
- EEPROM readout support for the new HAT expansion boards
- Higher drive capacity for analog audio out, from a separate regulator, which means a better audio DAC quality.

The power circuit changes also means a reduction in power requirements between 0.5W and 1W.

Specifications:

Chip Broadcom	: BCM2835 SOC
Core architecture	: ARM11
CPU	: 700 MHz Low Power ARM1176JZFS Applications Processor
GPU	: Dual CoreVideoCore IV® Multimedia Co-Processor Provides Open GL ES 2.0, hardware-accelerated OpenVG, and 1080p30 H.264 high-profile decodeCapable of 1Gpixel/s, 1.5Gtexel/s or 24GFLOPs with texture filtering and DMA infrastructure
Memory	: 512MB SDRAM
Operating System	: Boots from Micro SD card, running a version of the Linux operating system
Dimensions	: 85 x 56 x 17mm
Power	: Micro USB socket 5V, 2A

Connectors:

Ethernet	: 10/100BaseT Ethernet socket
Video Output	: HDMI (rev 1.3 &1.4) Composite RCA (PAL and NTSC)
Audio Output	: 3.5mm jack, HDMI
USB	: 4 x USB 2.0 Connector
GPIO Connector	: 40-pin 2.54 mm (100 mil) expansion header: 2x20 strip Providing 27 GPIO pins as well as +3.3 V, +5 V and GND supply lines
Camera Connector	: 15-pin MIPI Camera Serial Interface (CSI-2)

JTAG : Not populated

Display Connector : Display Serial Interface (DSI) 15 way flat flexiblecable connector with two data lanes ana clock lane

Memory Card Slot : Micro SDIO

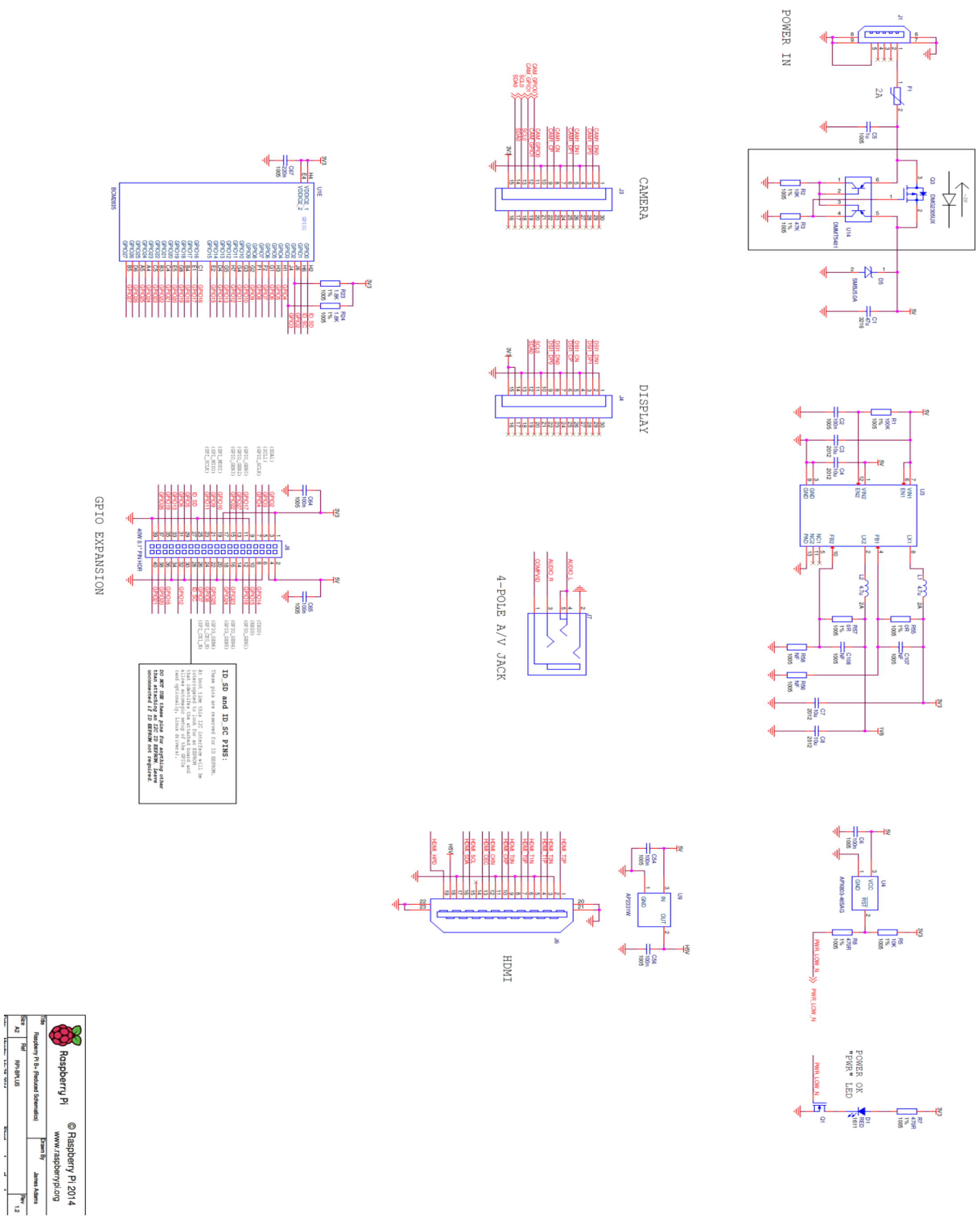


Fig:4.2.1 Schematics For Model B+

4.2.2. THECOMPUTE MODULE

The compute module is intended for industrial applications, it is a cut down device which simply include the BCM2835, 512MB of SDRAM and a 4GBeMMC flash memory, in a small form factor. This connects to a base board using a repurposed 200 pin DDR2 SODIMM connector. Note the device is NOT SODIMM compatible, it just repurposes the connector. All the BCM2835 features are exposed via the SODIMM connector, including twin camera and LCD ports, whilst the Model A or B/B+ only have one of each.

The compute module is expected to be used by companies wishing to shortcut the development process of new product, meaning only a baseboard needs to be developed, with appropriate peripherals, with the Compute Module providing the CPU, memory and storage along with tested and reliable software.

1. BCM2

The Broadcom chip used in the Raspberry Pi Model A, B and B+. The BCM2835 is a cost-optimized, full HD, multimedia applications processor for advanced mobile and embedded applications that require the highest levels of multimedia performance. Designed and optimized for power efficiency, BCM2835 uses Broadcom's VideoCore® IV technology to enable applications in media playback, imaging, camcorder, streaming media, graphics and 3D gaming.

Features:

- Low Power ARM1176JZ-F Applications Processor
- Dual Core VideoCore IV® Multimedia Co-Processor
- 1080p30 Full HD HP H.264 Video Encode/Decode
- Advanced Image Sensor Pipeline (ISP) forupto20-megapixel cameras operating at upto220 megapixels per second
- Low power, high performance OpenGL-ES® 1.1/2.0 VideoCore GPU. 1 Gigapixelper second fill rate.

Overview

BCM2835 contains the following peripherals which may safely be accessed by the ARM:

• Timers

• Interrupt controller

• GPIO

• USB

• PWM

• UART0, UART1

2. General Purpose I/O (GPIO)

General Purpose Input/Output pins on the Raspberry Pi

OVERVIEW

This page expands on the technical features of the GPIO pins available on BCM2835 in general.

GPIO pins can be configured as either general-purpose input, general-purpose output or as one of upto6 special alternate settings, the functions of which are pin-dependent.

There are 3 GPIO banks on BCM2835.

Each of the 3 banks has its own VDD input pin. On Raspberry Pi, all GPIO banks are supplied from 3.3V.Connectionof a GPIO to a voltage higher than 3.3V will likely destroy the GPIO block within the SOC.

A selection of pins from Bank 0 is available on the P1 header on Raspberry Pi.

GPIO PADS

The GPIO connections on the BCM2835 package are sometimes referred to in the peripherals datasheet as "pads" - a semiconductor design term meaning "chip connection to outside world".

The pads are configurable CMOS push-pull output drivers/input buffers. Register-based control settings are available for

- Internal pull-up / pull-down enable/disable
- Output drive strength
- Input Schmitt-trigger filtering

POWER-ON STATES

All GPIOs revert to general-purpose inputs on power-on reset. The default pull states are also applied, which are detailed in the alternate function table in the ARM peripherals datasheet. Most GPIOs have a default pull applied.

INTERRUPTS

Each GPIO pin, when configured as a general-purpose input, can be configuredas an interrupt source to the ARM. Several interrupt generation sources are configurable:

- Level-sensitive (high/low)
- Rising/falling edge
- Asynchronous rising/falling edge

Level interrupts maintain the interrupt status until the level has been cleared by system software (e.g. by servicing the attached peripheral generating the interrupt)

The normal rising/falling edge detection has a small amount of synchronization built into the detection. At the system clock frequency, the pin is sampled with the criteria for generation of an interrupt being a stable transition within a 3-cycle window, i.e. a record of "1 0 0" or "0 1 1". Asynchronous detection bypasses this synchronization to enable the detection of very narrow events.

ALTERNATIVE FUNCTIONS

Almost all of the GPIO pins have alternative functions. Peripheral blocks internal to BCM2835 can be selected to appear on one or more of a set of GPIO pins, for example the I2C busses can be configured to at least 3 separate locations. Pad control, such as drive strength or Schmitt filtering, still applies when the pin is configured as an alternate function. There are 54 general-purpose I/O (GPIO) lines split into two banks. All GPIO pins haveatleast two alternative functions within BCM. The alternate functions are usually peripheral IOand a single peripheral may appear in each bank to allow flexibility on the choice of IOvoltage.

The block diagram for an individual GPIO pin is given below :

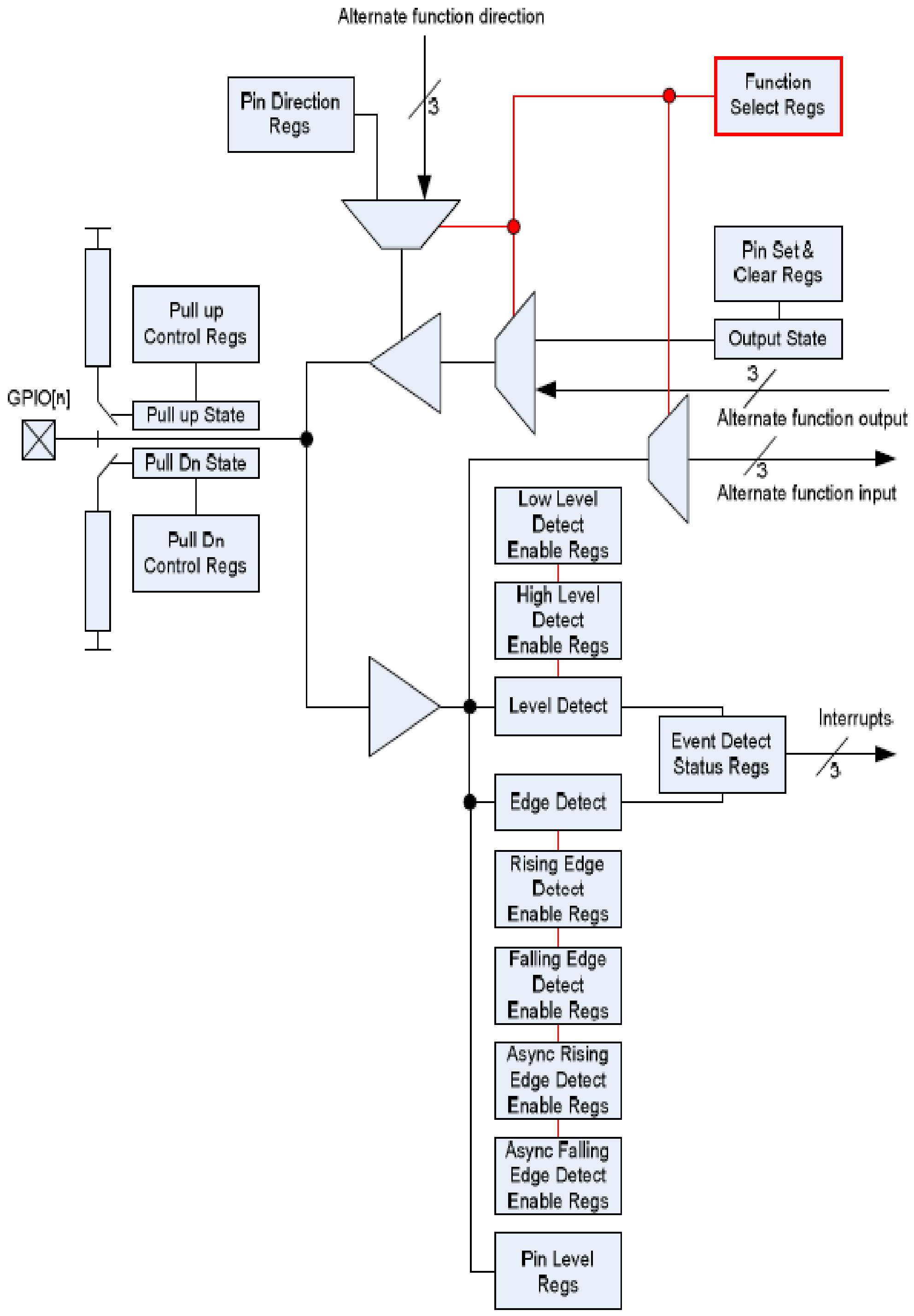

Alternate function direction
3
Pin Direction Regs
Function Select Regs
Pin Set & Clear Regs
Pull up Control Regs
Output State
Pull up State
3
GPIO[n]
Alternate function output
Pull Dn State
3
Alternate function input
Pull Dn Control Regs
Low Level Detect Enable Regs
High Level Detect Enable Regs
Level Detect
Interrupts
Event Detect Status Regs
3
Edge Detect
Rising Edge Detect Enable Regs
Falling Edge Detect Enable Regs
Async Rising Edge Detect Enable Regs
Async Falling Edge Detect Enable Regs
Pin Level Regs

Figure 6-1 GPIO Block Diagram

UART

UART performs serial-to-parallel conversion on data characters received from an external peripheral device or modem, and parallel-to-serialconversion on data characters received from the Advanced Peripheral Bus (APB).

The ARM PL011 UART has some optional functionality which can be included or left out. The following functionality is not supported :

● Infrared Data Association (IrDA)

● Serial InfraRed (SIR) protocol Encoder/Decoder (ENDEC)

● Direct Memory Access (DMA).

The UART provides:

• Separate 16x8 transmit and 16x12 receive FIFO memory.

• Programmable baud rate generator.

• Standard asynchronous communication bits (start, stop and parity). These are added prior to transmission and removed on reception.

• False start bit detection.

• Line break generation and detection.

• Support of the modem control functions CTS and RTS. However DCD, DSR, DTR, and RI are not supported.

• Programmable hardware flow control.

• Fully-programmable serial interface characteristics:

  - Data can be 5, 6, 7, or 8 bits
  - even, odd, stick, or no-parity bit generation and detection
  - 1 or 2 stop bit generation
  - baud rate generation, dc up to UARTCLK/16

Universal SPI Master (2x)

OVERVIEW

The Raspberry Pi is equipped with one SPI bus that has 2 chip selects.

The SPI master driver is disabled by default on Raspbian. To enable it, remove the blacklisting for spi-bcm2708 in/etc/modprobe.d/raspi-blacklist.conf, or use raspi-config. Reboot or load the driver manually with:

```
$ sudomodprobe spi-bcm2708
```

The SPI bus is available on the PI Header:

MOSI	P1-19		
MISO	P1-21		
SCLK	P1-23	P1-24	CE0
GND	P1-25	P1-26	CE1

SOFTWARE

WIRINGPI

WiringPiincludes a library which can make it easier to use the Raspberry Pi's on-board SPI interface. Accesses the hardware registers directly.

http://wiringpi.com/

BCM2835 LIBRARY

This is a C library for Raspberry Pi (RPi). It provides access to GPIO and other IO functions on the Broadcom BCM 2835 chip. Accesses the hardware registers directly.

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

USE SPIDEV FROM C

There's a loopback test program in the Linux documentation that can be used as a starting point.

SHELL

```
# Write binary 1, 2 and 3
echo-ne "\x01\x02\x03">/dev/spidev0.0
```

HARDWARE

The BCM2835 on the Raspberry Pi has 3 SPI Controllers. Only the SPI0 controller is available on the header.

MASTER MODES

Signal name abbreviations

SCLK - Serial Clock

CE - Chip Enable (often called Chip Select)

MOSI – MasterOut Slave In

MISO - Master In Slave Out

MOMI - Master Out Master In

MIMO – Master In Master Out

STANDARD MODE

In Standard SPI master mode the peripheral implements the standard 3 wire serialprotocol (SCLK, MOSI and MISO).

BIDIRECTIONAL MODE

In bidirectional SPI master mode the same SPI standard is implemented except that a single wire is used for data (MIMO) instead of two as in standard mode (MISO and MOSI).

LOSSI MODE (LOW SPEED SERIAL INTERFACE)

The LOSSI standard allows issuing of commands to peripherals (LCD) and to transfer data to and from them. LOSSI commands and parameters are 8 bits long, but an extra bit is used to indicate whether the byte is a command or parameter/data. This extra bit is set high for a data and low for a command. The resulting 9-bit value is serialized to the output. LOSSI is commonly used with MIPI DBI type C compatible LCD controllers.

TRANSFER MODES

- Polled
- Interrupt
- DMA

CHIP SELECT

Setup and Hold times related to the automatic assertion and de-assertion of the CS lines when operating in **DMA** mode are as follows:

- The CS line will be asserted atleast3 core clock cycles before themsb of the first byte of the transfer.
- The CS line will be de-asserted no earlier than 1 core clock cycle after the trailing edge of the final clock pulse.

LINUX DRIVER

The default Linux driver is spi-bcm2708.

The following information was valid 2014-07-05.

SPEED

The driver supports the following speeds:

cdiv	speed
2	125.0 MHz
4	62.5 MHz
8	31.2 MHz
16	15.6 MHz
32	7.8 MHz
64	3.9 MHz
128	1953 kHz
256	976 kHz
512	488 kHz
1024	244 kHz
2048	122 kHz
4096	61 kHz
8192	30.5 kHz
16384	15.2 kHz

32768	7629 Hz

When asking for say 24 MHz, the actual speed will be 15.6 MHZ.

Forum post: SPI has more speeds

SUPPORTED MODE BITS

SPI_CPOL - Clock polarity

SPI_CPHA - Clock phase

SPI_CS_HIGH – Chip Select active high

SPI_NO_CS - 1 device per bus, no Chip Select

Bidirectional mode is currently not supported.

SUPPORTED BITS PER WORD

- 8 - Normal
- 9 - This is supported using LOSSI mode.

TRANSFER MODES

Only interrupt mode is supported.

DEPRECATED WARNING

The following appears in the kernel log:

bcm2708_spi bcm2708_spi.0: master isunqueued, this is deprecated

SPI DRIVER LATENCY

This thread discusses latency problems.

DMA CAPABLE DRIVER This is a fork of spi-bcm2708 which enables DMA support for SPI client drivers that support DMA.

https://github.com/notro/spi-bcm2708 (wiki)

If you get compilation errors, try the latest version instead:

wget https://raw.github.com/torvalds/linux/master/Documentation/spi/spidev_test.c

The two universal SPI masters are secondary low throughput5 SPI interfaces. Like the UART the devicesneedsto be enabled before they can be used. Each SPI master has the following

Features:

• Single beat bit length between 1 and 32 bits.

• Single beat variable bit length between 1 and 24 bits

• Multi beat infinite bit length.

•3independent chip selects per master.

• 4 entries 32-bit wide transmit and receive FIFOs.

A major issue with an SPI interface is that there is no SPI standard in any form. Because the SPI interface has been around for a long time some pseudo-standard rules have appearedmostly when interfacing with memory devices. The universal SPI master has been developedto work even with the most 'non-standard' SPI devices.

SPI implementation details

The following diagrams shows a typical SPI access cycle. In this case we haveSPI clocks.

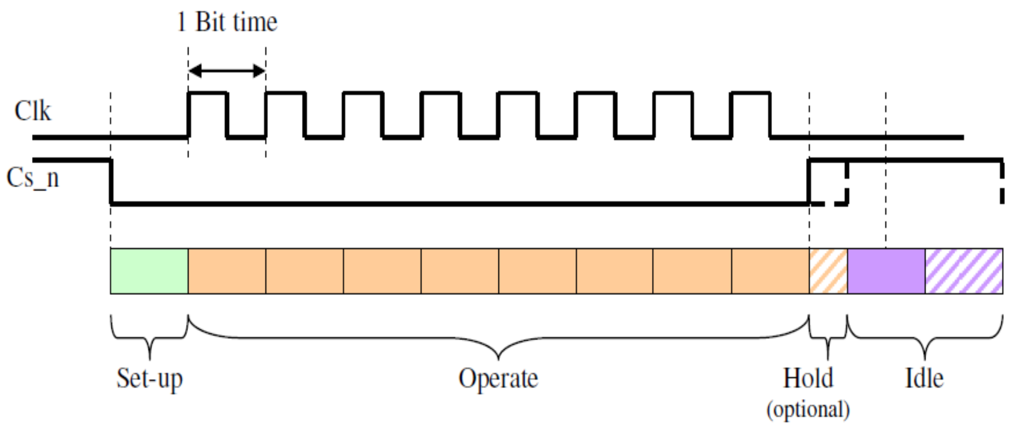


Fig: 4.2.2 Typical SPI Access Cycle

One bit time before any clock edge changes the CS_n will go low. This makes sure that theMOSI signal has a full bit-time of set-up against any changing clock edges.The operation normally ends after the last clock cycle. Note that at the end there is one halfbittime where the clock does not change but which still is part of the operation cycle.

PCM / I2S Audio

The PCM audio interface is an APB peripheral providing input and output of telephony or high quality serial audio streams. It supports many classic PCM formats including I2S.

The PCM audio interface has 4 interface signals;

PCM_CLK – bit clock.

PCM_FS – frame sync signal.

PCM_DIN - serial data input.

PCM_DOUT – serial data output.

PCM is a serial format with a single bit data_inand single bit data_out. Datais always serialized MS-bit first.

The frame sync signal (PCM_FS) is used to delimit the serial data into individual frames.

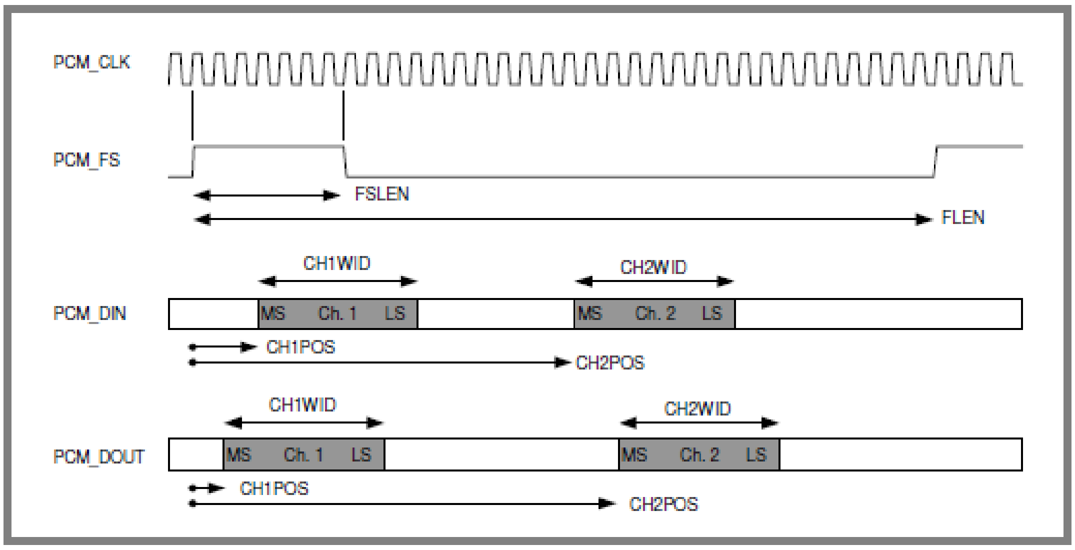


Fig: 4.2.3 PCM Audio Interface Typical Timing

The PCM_CLK can be asynchronous to the bus APB clock and can be logicallyinvertedifrequired.

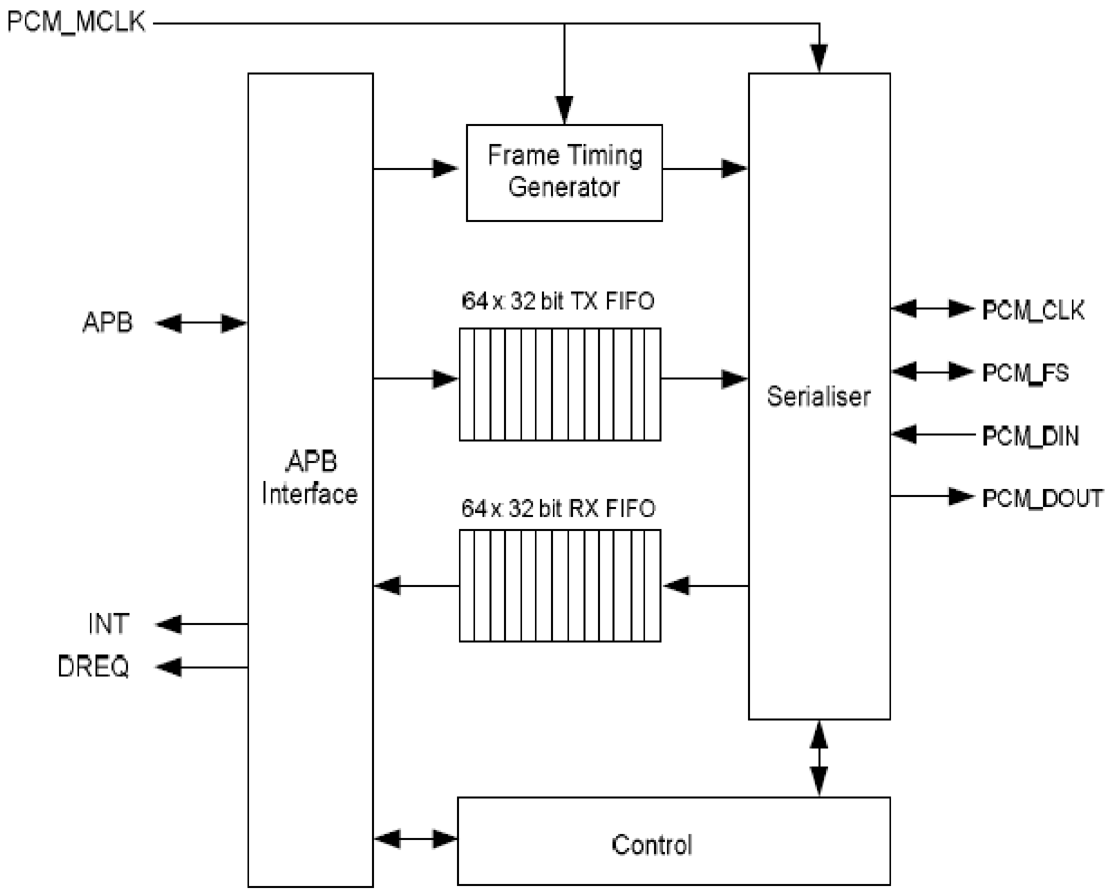


Fig:4.2.4 PCM Audio Interface Block Diagram

The PCM audio interface contains separate transmit and receive FIFOs. Note that if the frame contains two data channels, they must share the same FIFO and so the channel data will beinterleaved. The block can be driven using simple polling, an interrupt based method or directDMA control.

Operation:

In operation, the PCM format is programmed by setting the appropriate frame length, frame sync, channel position values, and signal polarity controls. The transmit FIFO should be preloaded with data and the interface can then be enabled and started, and will run continuously until stopped. If the transmit FIFO becomes empty or the receive FIFO becomes full, the RXERR or TXERR error flags will be set, but the interface will just continue. If the RX FIFO overflows, new samples are discarded and if the TX FIFO underflows, zeros are transmitted.Normally channel data is read or written into the appropriate FIFO as a single word. If the channel is less than 32 bits, the data is right justified and should be padded with zeros. If theRXSEX bit is set then the received data is sign extended up to the full 32 bits. When a frame is programmed to have two data channels, then each channel is written/read as a separateword in the FIFO, producing an interleaved data stream. When initializing the interface, thefirst word read out of the TX FIFO will be used for the first channel, and the data from thefirst channel on the first frame to be received will be the first word written into the RX FIFO.

If a FIFO error occurs in a two channel frame, then channel synchronization may be lost which may result in a left right audio channel swap. RXSYNC and TXSYNC status bits areprovided to help determine if channel slip has occurred. They indicate if the number of words in the FIFO is a multiple of a full frame (taking into account where we are in the current frame being transferred). This assumes that an integer number of frames data has beensent/read from the FIFOs.

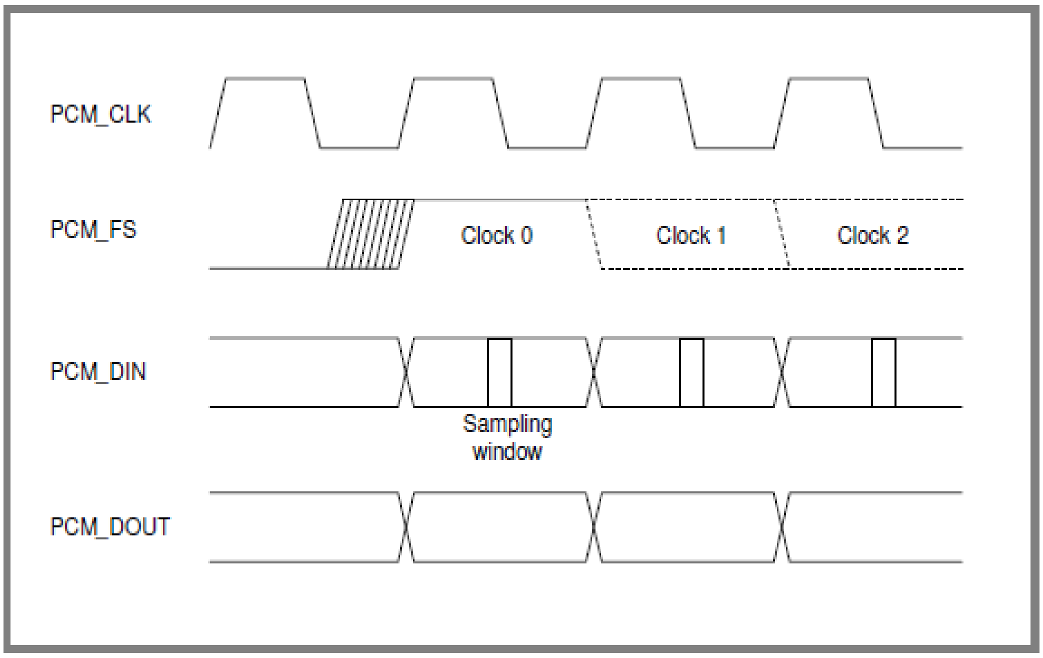


Fig:4.2.5 Timing at Start of Frame

Note that the precise timing of FS (when it is an input) is not clearly defined and it may change state before or after the positive edge of the clock. Here the first clock of the frame is defined as the clock period where the PCM_FS is sampled (negative edge) as a 1 where itwas previously sampled as a 0.

4.3 POWER SUPPLY

The device is powered by a 5V micro USB supply. Exactly how much current (mA) the Raspberry Pi requires is dependent on what you connect to it. We have found that purchasing a 1.2A (1200mA) power supply from a reputable retailer will provide you with ample power to run your Raspberry Pi.
Typically, the model B uses between 700-1000mA depending on what peripherals are connected; the model A can use as little as 500mA with no peripherals attached. The maximum power the Raspberry Pi can use is 1 Amp. If you need to connect a USB device that will take the power requirements above 1 Amp, then you must connect it to an externally-powered USB hub.
The power requirements of the Raspberry Pi increase as you make use of the various interfaces on the Raspberry Pi. The GPIO pins can draw 50mA safely, distributed across all the pins; an individual GPIO pin can only safely draw 16mA. The HDMI port uses 50mA, the

camera module requires 250mA, and keyboards and mice can take as little as 100mA or over 1000mA! Check the power rating of the devices you plan to connect to the Pi and purchase a power supply accordingly.

USB

OVERVIEW

The Raspberry Pi Model B is equipped with two USB2.0 ports. These are connected to the LAN9512 combo hub/Ethernet chip IC3, which is itself a USB device connected to the single upstream USB port on BCM2835.

On the Model A, the single USB2.0 port is directly wired to BCM2835.

The USB ports enable the attachment of peripherals such as keyboards, mice, webcams that provide the Pi with additional functionality.

There are some differences between the USB hardware on the Raspberry Pi and the USB hardware on desktop computers or laptop/tablet devices.

SUPPORTED DEVICES

In general, every device supported by Linux is possible to use with the Pi, subject to a fewcaveats.Linux has probably the most comprehensive driver database for legacy hardware of any operating system (it can lag behind for modern device support as it requires open-source drivers for Linux to recognize the device by default).

If you have a device and wish to use it with a Pi, then plug it in. Chances are that it'll "just work". If you are running in a graphical interface (such as the LXDE desktop environment inRaspbian), then it's likely that an icon or similar will pop up announcing the new device.

GENERAL LIMITATIONS

The OTG hardware on Raspberry Pi has a simpler level of support for certain devices, which may present a higher software processing overhead. The Raspberry Pi also has only one root USB port: all traffic from all connected devices is funelleddown this bus, which operates at a maximum speed of 480mbps.

The USB specification defines three device speeds - Low, Full and High. Most mice and keyboards are Low-speed, most USB sound devices are Full-speed and most video devices (webcams or video capture) are High-speed.

PORT POWER LIMITS

USB devices have defined power requirements, in units of 100mA from 100mA to 500mA. The device advertises its own power requirements to the USB host when it is first connected. In theory, the actual power consumed by the device should not exceed its stated requirement.

The USB ports on a Raspberry Pi have a design loading of 100mA each - sufficient to drive "low-power" devices such as mice and keyboards. Devices such asWiFiadapters, USB hard drives, USB pen drives all consume much more current and should be powered from an external hub with its own power supply.

DEVICES WITH KNOWN ISSUES

1. Interoperability between the Raspberry Pi and USB3.0 hubs

There is an issue with USB3.0 hubs in conjunction with the use of Full- or Low-speed devices (most mice, most keyboards) and the Raspberry Pi. A bug in most USB3.0 hub hardware means that the Raspberry Pi cannot talk to Full- or Low-speed devices connected to a USB3.0 hub.

USB2.0 high-speed devices, including USB2.0 hubs, operate correctly when connected via a USB3.0 hub.Avoid connecting Low- or Full-speed devices into a USB3.0 hub. As a workaround, plug a USB2.0 hub into the downstream port of the USB3.0 hub and connect the low-speed device, or use a USB2.0 hub between the Pi and the USB3.0 hub, then plug low-speed devices into the USB2.0 hub.

2. USB1.1 webcams

Old webcams may be Full-speed devices. Because these devices transfer a lot of data and incur additional software overhead, reliable operation is not guaranteed.

As a workaround, try to use the camera at a lower resolution.

3. Esoteric USB sound cards

Expensive "audiophile" sound cards typically use far more bandwidth than is necessary to stream audio playback. Reliable operation with 96kHz/192kHz DACs is not guaranteed.As a workaround, forcing the output stream to be CD quality (44.1kHz/48kHz 16-bit) will reduce the stream bandwidth to reliable levels.

4. Single-TT USB hubs

USB2.0 and 3.0 hubs have a mechanism for talking to Full- or Low-speed devices connected to their downstream ports called a Transaction Translator. This device buffers high-speed requests from the host (i.e. the Pi) and transmits them at Full- or Low-speed to the downstream device. Two configurations of hub are allowed by the USB specification: Single-TT (one TT for all ports) and Multi-TT (oneTTperport).

Because of the OTG hardware limitations, if too many Full- or Low-speed devices are plugged into a single-TT hub, unreliable operation of the devices may occur. It is recommended to use a Multi-TT hub to interface with multiple lower-speeddevices.

IF YOUR DEVICE HAS INTERMITTENT BEHAVIOUR

Poor quality power is the most common cause of devices not working, disconnecting or generally being unreliable.

- If you are using an external powered hub, try swapping the power adapter supplied with the hub for another compatible power supply with the same voltage rating and polarity.
- Check to see if the problem resolves itself if you remove other devices from the hub's downstream ports.
- Temporarily plug the device directly into the Pi and see if the behaviour improves.

RASPBIAN

Raspbian is the recommended operating system for normal use on a Raspberry Pi. Raspbianis a free operating system based on Debian, optimized for the Raspberry Pi hardware. Raspbian comes with over 35,000 packages; precompiled software bundled in a nice format for easy installation on your Raspberry Pi.Raspbian is a community project under active development, with an emphasis on improving the stability and performance of as manyDebian packages as possible.

INSTALLING OPERATING SYSTEM IMAGES

How to install a Raspberry Pi Operating System image on an SD card. You will need another computer with an SD card reader to install the image.We recommend most users download NOOBS which is designed to be very easy to use. However more advanced users looking to install a particular image should usethis guide.

DOWNLOAD THE IMAGE

Official images for recommended Operating Systems are available to download from the Raspberry Pi website: raspberrypi.org/downloads. Alternative distributions are available from third party vendors.

WRITING AN IMAGE TO THE SD CARD

With the image file of the distribution of your choice, you need to use an image writing tool to install it on your SD card.

APT

The easiest way to manage installing, upgrading, and removing software is using APT (Advanced Packaging Tool) which comes fromDebian. If a piece of software is packaged inDebianand works on the Raspberry Pi's ARM architecture, it should also be available inRaspbian.

To install or remove packages you need root user permissions, so your user needs to be in sudoers or you must be logged in as root.

SOFTWARE SOURCES

APT keeps a list of software sources on your Pi in a file at /etc/apt/sources.list. Before installing software, you should update your package list with apt-get update:

```
sudoapt-get update
```

INSTALLING A PACKAGE WITH APT

```
sudoapt-get install tree
```

Typing this command should inform the user how much disk space the package will take up and asks for confirmation of the package installation. Entering Y (or justhitting Enter, as yes is the default action) will allow the installation to occur. This can be bypassed by adding the -y flag to the command:

```
Sudo apt-get install tree -y
```

Installing this package makes tree available for the user.

UNINSTALLING A PACKAGE WITH APT

REMOVE

You can uninstall a package with apt-get remove:

```
sudoapt-get remove tree
```

The user is prompted to confirm the removal. Again, the -y flag will auto-confirm.

PURGE

You can also choose to completely remove the package and its associated configuration files with apt-get purge:

```
sudo apt-get purge tree
```

UPGRADING EXISTING SOFTWARE

If software updates are available, you can get the updates with sudoapt-get update and install the updates with sudoapt-get upgrade, which will upgrade all of your packages. To upgrade a specific package, without upgrading all the other out-of-date packages at the same time, you can use sudoapt-get install somepackage (which may be useful if you're low on disk space or you have limited download bandwidth).

UPDATING AND UPGRADING RASPBIAN

There are two steps to upgrading. First, run sudo apt-get update in order to synchronize the database of available software packages and the versions available. Next, run sudo apt-get upgrade which will cause any packages with newer versions available to be updated. Generally speaking, doing this regularly will keep your installation up to date; in other words, it will be equivalent to the latest released image. However, there are occasionally times where a change is made in the Foundation's Raspbianimage that require your intervention to reproduce. A recent example is the addition of the xserver-xorg-video-fbturbo X.Org driver to the standard image; this requires users running older images to manually install the packageto benefit from it. Cases like this are documented on the relevant image update announcement on the Raspberry Pi blog.

UPDATING THE KERNEL AND FIRMWARE

The kernel and firmware are installed as aDebian package, and so will also get updates when using the procedure above. These packages are updated infrequently (after extensive testing); if you want to try more recent experimental software, it's also easy to update to the latest available version using rpi-update. This is pre-installed on the current Raspbian image, so you can just use sudorpi-update to try the latest firmware; this will sometimes be suggested when troubleshooting. If you receive errors about invalid certificates, then run sudo apt-get update&&sudo apt-get installrpi-update to upgrade to the latestrpi-update version.

RUNNING OUT OF SPACE

When running sudo apt-get upgrade, it will show how much data will be downloaded and how much space it will take up on the SD card. It's worth checking with df -h that you have enough disk space free, as unfortunately apt will not do this for you. Also be aware that downloaded package files (.deb files) are kept in /var/cache/apt/archives. You can remove these in order to free up space with sudoapt-get clean.

raspi-config

raspi-config is the Raspberry Pi configuration tool written and maintained by Alex Bradbury. It targets Raspbian.

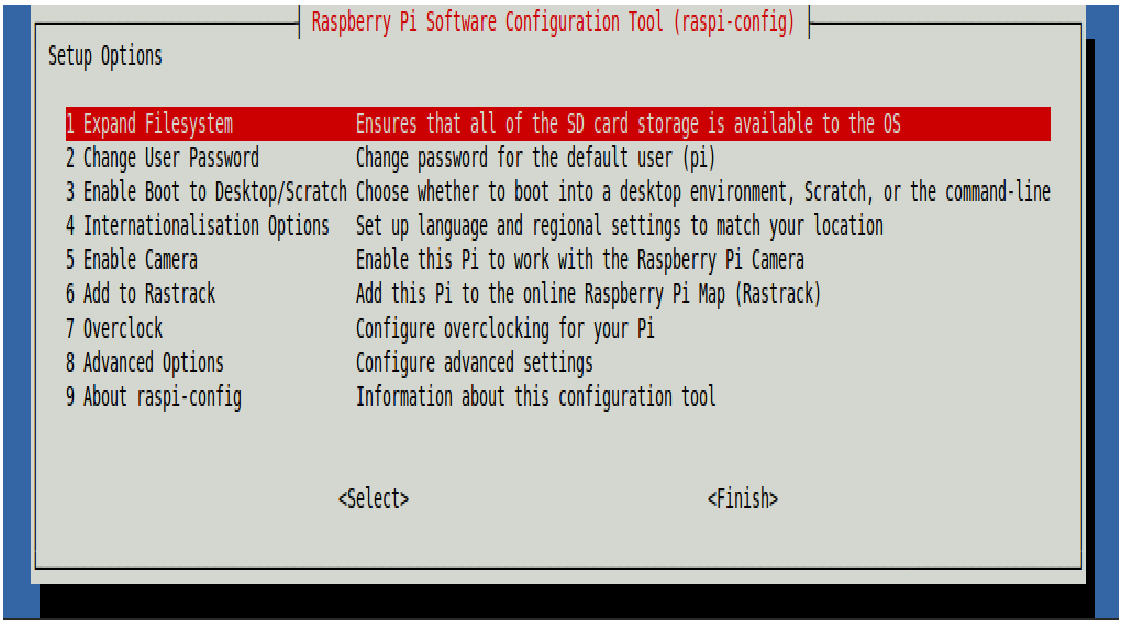
USAGE

You willbe shown raspi-config on first booting intoRaspbian. To open the configuration tool after this, simply run the following from the command line:

sudoraspi-config

The sudo is required because you will be changing files that you do not own as the pi user.

You should see a blue screen with options in a grey box in thecentre, like so:



```
                        Raspberry Pi Software Configuration Tool (raspi-config)
Setup Options

   1 Expand Filesystem            Ensures that all of the SD card storage is available to the OS
   2 Change User Password         Change password for the default user (pi)
   3 Enable Boot to Desktop/Scratch Choose whether to boot into a desktop environment, Scratch, or the command-line
   4 Internationalisation Options Set up language and regional settings to match your location
   5 Enable Camera                Enable this Pi to work with the Raspberry Pi Camera
   6 Add to Rastrack              Add this Pi to the online Raspberry Pi Map (Rastrack)
   7 Overclock                    Configure overclocking for your Pi
   8 Advanced Options             Configure advanced settings
   9 About raspi-config           Information about this configuration tool

                    <Select>                          <Finish>
```

Fig:4.3.6 Raspberry Pi Software Configuration Tool

It has the following options available:

Raspberry Pi Software Configuration Tool (raspi-config)

Setup Options

1. ExpandFilesystem- Ensures that all of the SD card storage is available to the OS
2. Change User Password –Change password for the default user (pi)
3. Enable Boot to Desktop/Scratch- Choose whether to boot into a desktop environment, Scratch, or the command line
4. InternationalisationOptions – Set up language and regional settings to match your location
5. Enable Camera-Enable this Pi to work with the Raspberry Pi Camera
6. Add toRastrack– Add this Pi to the online Raspberry Pi Map (Rastrack)
7. Overclock- Configure overclocking for your Pi
8. Advanced Options- Configure advanced settings
9. About `raspi-config`- Information about this configuration tool

<Select><Finish>

MOVING AROUND THE MENU

Use the up and down arrow keys to move the highlighted selection between the options available. Pressing the right arrow key will jump out of the options menu and take you to the <Select> and <Finish> buttons. Pressing left will take you back to the options. Alternatively, use the Tabkey to switch between these.
Note that in long lists of option values (like the list oftimezonecities), you can also type a letter to skip to that section of the list. For example, enteringL will skip you to Lisbon, just two options away from London, to save you scrolling all the way through the alphabet.

WHAT raspi-config DOES

Generally speaking, raspi-config aims to provide the functionality to make the most common configuration changes. This may result in automated edits to /boot/config.txt and various standard Linux configuration files. Some options require a reboot to take effect. If you changed any of those, raspi-configwill ask if you wish to reboot now when you select the <Finish>button.

SUDO

You will not normally log into the computer as root, but can instead use the sudo command to provide access as thesuperuser. If you log into your Raspberry Pi as the pi user then you are logging in as a normal user. You can run commands as the root user by using the sudo command before the program you want to run.
For example if you want to install additional software on Raspbianthen you normally use the apt-get tool. To be able to update the list of available software then you need to prefix the apt-get command with sudo. Sudo apt-get update

WHO CAN USE SUDO?

It would defeat the point of the security if anyone could just putsudoin front of their commands, so only approved users can usesudo to gain administrator privileges. The pi user is included in thesudoersfile. To allow other users to act as a superuser then you could add the user to the sudo group or add them using visudo.

INSTALLING SOFTWARE

There are different ways of installing software on your Raspberry Pi, depending on where the source of the software lives.

The most common is through command line tool APT (Advanced Packaging Tool), and there is the Pi Store – the Raspberry Pi's own user contributed app store. Also some software can be installed using other package managers.

CONTENTS

PI STORE

The Pi Store is the Raspberry Pi app store – you can browse and download community developed software and games, and even submit your own applications for others to download for free or to buy.

You can browse the Pi Store online at store.raspberrypi.com

INSTALLING PYTHON PACKAGES

APT

Some Python packages can be found in theRaspbian archives and can be installed using APT, for example:

```
sudo apt-get update
sudo apt-get install python-picamera
```

This is a preferable method of installing software, as it means that the modules you install can be kept up to date easily with the usual sudoapt-get update and sudoapt-get upgrade commands.

Python packages inRaspbianwhich are compatible with Python 2.x will always have a python- prefix. So, the picamera package for Python 2.x is named python-picamera (as shown in the example above). Python 3 packages always have a python3- prefix. So, to install rpi.gpio for**Python 3 you would use:**

sudoapt-get install python3-rpi.gpio

Uninstalling packages installed via APT can be accomplished as follows:

sudo apt-get remove python3-rpi.gpio

PIP

Not all Python packages are available in theRaspbian archives, and those that are can sometimes be out of date. If you can't find a suitable version in theRaspbianarchives you can install packages from the Python Package Index (also known asPyPI). To do so, use the pip tool.

KERNEL

The Raspberry Pi kernel is stored inGitHuband can be viewed atgithub.com/raspberrypi/linux; it follows behind the main linux kernel.

The main Linux kernel is continuously updating; we take long-term releases of the kernel, which are mentioned on the front page, and integrate the changes into the Raspberry Pi kernel. We then create a 'next' branch which contains an unstable port of the kernel; after extensive testing and discussion we push this to the main branch.

- Updating your kernel
- Building a new kernel
- Configuring the kernel
- Applying patches to the kernel
- Getting the kernel headers

GETTING YOUR CODE INTO THE KERNEL

There are many reasons you may want to put something into the kernel:

- You've written some Raspberry Pi-specific code that you want everyone to benefit from
- You've written a generic Linux kernel driver for a device and want everyone to use it
- You've fixed a generic kernel bug

- You've fixed a Raspberry Pi-specific kernel bug

Initially you should fork the Linux repository and clone that on your build system; this can be either on the Raspberry Pi or on a Linux machine you're cross-compiling on. You can then make your changes, test them, and commit them into your fork.

4.1 Next, depending upon whether the code is Raspberry Pi-specific or not:

For Pi-specific changes or bug fixes, submit a pull request to the kernel.

For general Linux kernel changes (i.e. a new driver) these need to be submitted upstream first. Once they've been submitted upstream and accepted, submit the pull request and we'll receive it.

4.2 Building Qt5 on Raspberry Pi

This is a how toguide for buildingQt5 for the Raspberry Pi, and building and deployingQt 5 apps using Qt Creator. This guide will be using Raspbian "Wheezy", a Debian baseddistrodesigned for the Raspberry Pi. This guide also assumes the use of Linux or UNIX on the workstation side.

Note: Throughout this guide the symbol "$" denotes a command prompt on the host workstation, and "pi$" denotes a command prompt on the target Raspberry Pi.

4.3 Install a Toolchain

To build on the Raspberry Pi we need a cross-compiletoolchain. Thetoolchainwill contain compilers, linkers and other tools that run on the host workstation but create executablesfor the target Raspberry Pi.

For embedded development, one normally uses a vendor-suppliedtoolchain, but in the case of the Raspberry Pi, there is no official vendor suppliedtoolchain. There are several generic ARMtoolchains that will suffice, however, we have chosen to use the same one the *bakeqtpi* script uses. This is a Linaro based toolchainfor the ARMv6 platform with hard floating-point support. Alternatively, we could have built our owntoolchain.

We will create a working directory to use named "raspberry". Our first step is to get and install a cross compilingtoolchain.

```
$ mkdir ~/raspberry
$ cd ~/raspberry
$ wget http://swap.tsmt.eu/gcc-4.7-linaro-rpi-gnueabihf.tbz
$ tar xfj gcc-4.7-linaro-rpi-gnueabihf.tbz
```

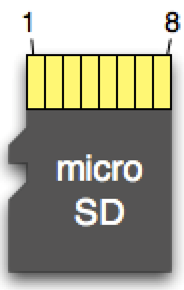
Since this toolchain is built for 32-bit systems, you will need a set of 32-bit libraries installed if you are on a 64-bit system. On Ubuntu systems, this can be accomplished by installing the ia32-libs package. Unfortunately, this is a deprecated transitional package, with no replacement.

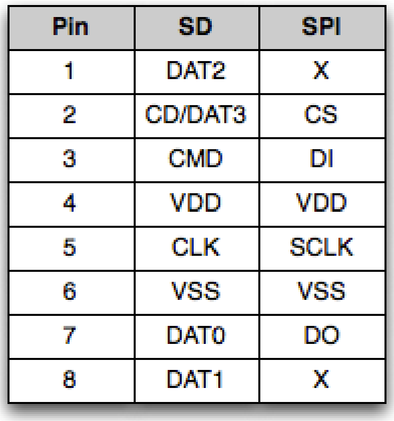
```
$ sudo apt-get install ia32-libs
```

4.4 SD-CARD

The Secure Digital Card is a flash-based memory card that is specifically designed to meet the security, capacity, performance and environmental requirements inherent in newly emerging audio and video consumer electronic devices. The SD Card includes a copyright protection mechanism that complies with the security of the SDMI standard, and is faster and capable of higher Memory capacity. The SD Card security system uses mutual authentication and a "new cipher algorithm" to protect from illegal usage of the card content. A non-secured access to the user‘s own content is also available. The physical form factor, pin assignment and data transfer protocol are forward compatible with the SD Card, with some additions.

TheSD Card communication is based on an advanced nine-pin interface (Clock, Command, 4xDataand 3xPower lines) designed to operate in a low voltage range. The communication protocol is defined as part of this specification. The SD Card host interface supports regular Multi Media Card operation as well. In other words, Multi Media Card forward compatibility was kept. Actually the main difference between SD Card and Multi Media Card is the initialization process. The SD Card specifications were originally defined by MEI (Matsushita Electric Company), Toshiba Corporation and SanDisk Corporation. Currently, the specifications are controlled by the Secure Digital Association (SDA). The SanDisk SD Card was designed to be compatible with the SD Card Physical Specification.





Pin	SD	SPI
1	DAT2	X
2	CD/DAT3	CS
3	CMD	DI
4	VDD	VDD
5	CLK	SCLK
6	VSS	VSS
7	DAT0	DO
8	DAT1	X

Table:4.4.1   SD Card Pin Configuration

Advantages Of Memory Cards

1. Memory cards have non-volatile memory, which keeps data stable on the card. Data on them are not threatened by loss of power source, and need not to be periodically refreshed.
2. They are solid state media hence free from mechanical difficulties or damages.
3. The new generation memory cards are smaller, lighter and compact with higher storage capacity.
4. They require less amount of power.
5. They are highly portable. They can be easily used in number of small, lightweight and low-power devices.

Disadvantages Of Memory Cards

As true for all the devices, memory card too have some disadvantages. Let's check them out.

1.They can break easily.

2.They can belost, misplaced or smashed.

3.These cards may be affected by electronic corruption and make entire card unreadable.

SOFTWARE TOOLS USED

4.5 OpenCV (*Open Source Computer Vision*) is a library of programming functions mainly aimed at real-time computer vision, originally developed by Intel research center inNizhny Novgorod (Russia), later supported byWillow Garage and now maintained byItseez.[1] The library iscross-platform and free for use under the open-sourceBSD license.

Contents

History

Officially launched in 1999, the OpenCV project was initially an Intel Research initiative to advanceCPU-intensive applications, part of a series of projects including real-timeray tracing and 3D display walls. The main contributors to the project included a number of optimization experts in Intel Russia, as well as Intel's Performance Library Team. In the early days of OpenCV, the goals of the project were described[*citation needed*] as:

- Advance vision research by providing not only open but also optimized code for basic vision infrastructure. No more reinventing the wheel.
- Disseminate vision knowledge by providing a common infrastructure that developers could build on, so that code would be more readily readable and transferable.
- Advance vision-based commercial applications by making portable, performance-optimized code available for free—with a license that did not require to be open or freethemselves.

The first alpha version of OpenCV was released to the public at theIEEE Conference on Computer Vision and Pattern Recognition in 2000, and five betas were released between 2001 and 2005. The first 1.0 version was released in 2006. In mid-2008, OpenCV obtained corporate support from Willow Garage, and is now again under active development. A version 1.1 "pre-release" was released in October 2008.

Applications

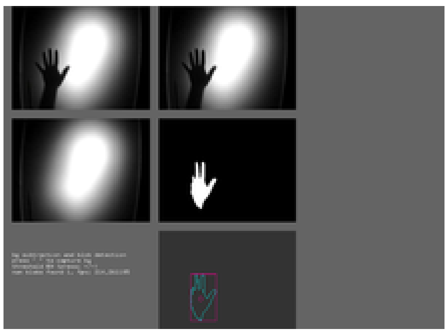


Fig:4.5.1 openFrameworksrunning the OpenCV

OpenCV'sapplication areas include:

- 2D and 3D feature toolkits

- Egomotionestimation
- Facial recognition system
- Gesture recognition
- Human–computer interaction (HCI)

To support some of the above areas, OpenCV includes a statistical machine learninglibrary that contains:

- Boosting (meta-algorithm)
- Decision tree learning
- Gradient boosting trees
- Expectation-maximization algorithm
- k-nearest neighbor algorithm
- Naive Bayes classifier
- Artificial neural networks
- Random forest
- Support vector machine (SVM)

Programming language

OpenCV is written inC++ and its primary interface is in C++, but it still retains a less comprehensive though extensive olderC interface. There are now full interfaces inPython, Javaand MATLAB/OCTAVE (as of version 2.5). The API for these interfaces can be found in the online documentation.[5] Wrappers in other languages such asC#, Perl,[6]Ch,[7] and Rubyhave been developed to encourage adoption by a wider audience.

Hardware Acceleration

If the library finds Intel'sIntegrated Performance Primitiveson the system, it will use these proprietary optimized routines to accelerate itself.

ACUDA-based GPU interface has been inprogress since September 2010.**OS support**

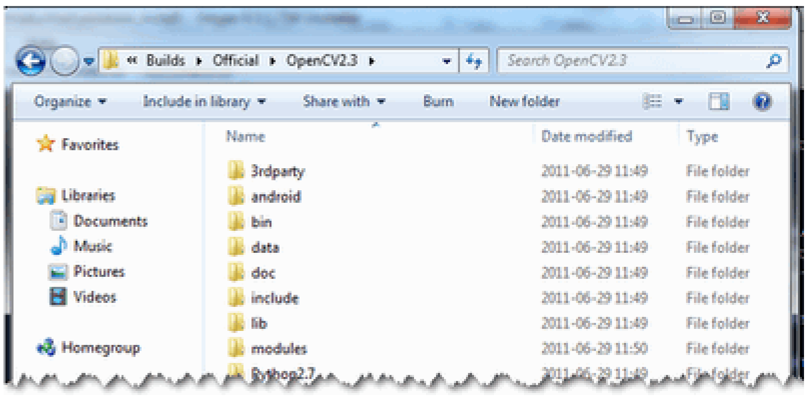
OpenCV runs on a variety of platforms. Desktop: Windows, Linux, OS X, FreeBSD, OpenBSD; Mobile: Android, iOS, Maemo,[11]BlackBerry 10.[12] The user can get official releases fromSourceForge or take the latest sources from GitHub.[13] OpenCV usesCMake.

OPEN CV Installation in Windows

The description here was tested on Windows 7 SP1. Nevertheless, it should also work on any other relatively modern version of Windows OS. If you encounter errors after following the steps described below.

Installation by Using the Pre-built Libraries

1. Launch a web browser of choice and go to our page on Source forge.
2. Choose a build you want to use and download it.
3. Make sure you have admin rights. Unpack the self-extracting archive.
4. You can check the installation at the chosen path as you can see below.



Name	Date modified	Type
3rdparty	2011-06-29 11:49	File folder
android	2011-06-29 11:49	File folder
bin	2011-06-29 11:49	File folder
data	2011-06-29 11:49	File folder
doc	2011-06-29 11:49	File folder
include	2011-06-29 11:49	File folder
lib	2011-06-29 11:49	File folder
modules	2011-06-29 11:50	File folder
Python2.7	2011-06-29 11:49	File folder

5. To finalize the installation go to the *Set the OpenCV enviroment variable and add it to the systems path*section.

Installation by Making Your Own Libraries from the Source Files

If you are building your own libraries you can take the source files from ourGitrepository.

Building theOpenCV library from scratch requires a couple of tools installed beforehand:

- An IDE of choice (preferably), or just a CC++ compiler that will actually make the binary files. Here we will use theMicrosoft Visual Studio. However, you can use any other IDE that has a valid CC++ compiler.

- CMake, which is a neat tool to make the project files (for your chosen IDE) from the OpenCV source files. It will also allow an easy configuration of the OpenCV build files, in order to make binary files that fits exactly to your needs.

- Git to acquire the OpenCV source files. A good tool for this is TortoiseGit. Alternatively, you can just download an archived version of the source files from our page onSourceforge

OpenCV may come in multiple flavors. There is a “core” section that will work on its own. Nevertheless, there is a couple of tools, libraries made by 3rd parties that offer services of which the OpenCV may take advantage. These will improve its capabilities in many ways. In order to use any of them, you need to download and install them on your system.

- ThePython libraries are required to build the*Python interface* of OpenCV. For now use the version 2.7.*x*. This is also a must if you want to build the *OpenCV documentation*.
- Numpy is a scientific computing package for Python. Required for the *Python interface*.
- Intel © Threading Building Blocks (*TBB*) is used inside OpenCV for parallel code snippets. Using this will make surethat the OpenCV library will take advantage of all the cores you have in your systems CPU.
- Intel © Integrated Performance Primitives (*IPP*) may be used to improve the performance of color conversion, Haar training and DFT functions of the OpenCV library. Watch out, since this isn't a free service.
- OpenCV offers a somewhat fancier and more useful graphical user interface, than the default one by using theQt framework. For a quick overview of what this has to offer look into the documentations*highgui* module, under the*Qt New Functions*section. Version 4.6 or later of the framework is required.
- Eigen is a C++ template library for linear algebra.
- The latestCUDA Toolkitwill allow you to use the power lying inside your GPU. This will drastically improve performance for some algorithms (e.gthe HOG descriptor). Getting more and more of our algorithms to work on the GPUs is a constant effort of the OpenCV team.
- OpenEXR source files are required for the library to work with this high dynamic range (HDR) image file format.
- TheOpenNI Framework contains a set of open source APIs that provide support for natural interaction with devices via methods such as voice command recognition, hand gestures and body motion tracking.

- Miktex is the bestTEXimplementation on the Windows OS. It is required to build the *OpenCV documentation*.

- Sphinxis a python documentation generator and is the tool that will actually create the*OpenCVdocumentation*. This on its own requires a couple of tools installed.

Building the library

1. Make sure you have a working IDE with a valid compiler. In case of the Microsoft Visual Studio just install it and make sure it starts up.

2. InstallCMake. Simply follow the wizard, no need to add it to the path. The default install options are OK.

3. Download and install an up-to-date version ofmsysgit from itsofficial site. There is also the portable version, which you need only to unpack to get access to the console version ofGit. Supposing that for some of us it could be quite enough.

4. Install TortoiseGit. Choose the 32 or 64 bit version according to the type of OS you work in. While installing, locate yourmsysgit (if it doesn't do that automatically). Follow the wizard – the default options are OK for the most part.

5. Choose a directory in your file system, where you will download the OpenCV libraries to. I recommend creating a new one that has short path and no specialcharacters in it, for example D:/OpenCV. For this tutorial I'll suggest you do so. If you use your own path and know, what you're doing – it's OK.

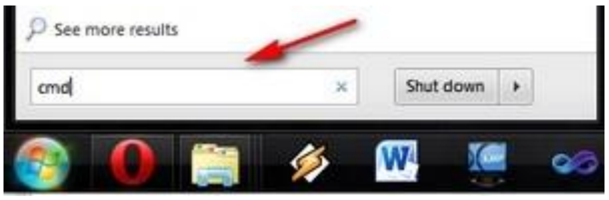
    1. Clone the repository to the selected directory. After clicking*Clone* button, a window will appear where you can select from what repository you want to download source files (https://github.com/Itseez/opencv.git) and to what directory (D:/OpenCV).

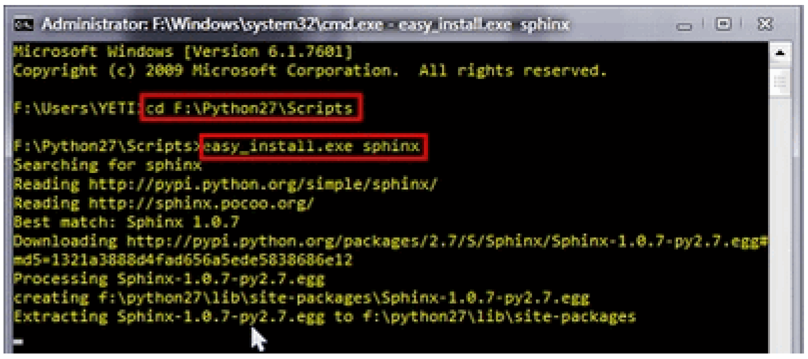
    2. Push the OK button and be patient as the repository is quite a heavy download. It will take some time depending on your Internet connection.

6. In this section I will cover installing the 3rd party libraries.

    1. Download thePython libraries and install it with the default options. You will need a couple other python extensions. Luckily installing all these may be automated by a nice tool called Setuptools. Download and install again.

    2. Installing Sphinx is easy once you have installed*Setuptools*. This contains a little application that will automatically connect to the python databases and download the latest version of many python scripts. Start up a command window (enter *cmd* into the windows start menu and press enter) and use the *CD*command to navigate to your Python folders Script sub-folder. Here just pass to the *easy_install.exe* as argument the name of the program you want to install. Add the *sphinx*argument.





```
Administrator: F:\Windows\system32\cmd.exe - easy_install.exe  sphinx
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

F:\Users\YETI>cd F:\Python27\Scripts

F:\Python27\Scripts>easy_install.exe sphinx
Searching for sphinx
Reading http://pypi.python.org/simple/sphinx/
Reading http://sphinx.pocoo.org/
Best match: Sphinx 1.0.7
Downloading http://pypi.python.org/packages/2.7/S/Sphinx/Sphinx-1.0.7-py2.7.egg#
md5=1321a3888d4fad656a5ede5838686e12
Processing Sphinx-1.0.7-py2.7.egg
creating f:\python27\lib\site-packages\Sphinx-1.0.7-py2.7.egg
Extracting Sphinx-1.0.7-py2.7.egg to f:\python27\lib\site-packages
_
```

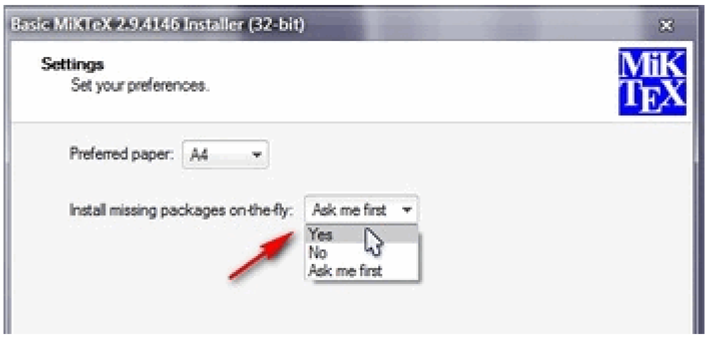
Note

The*CD* navigation command works only inside a drive. For example if you are somewhere in the*C:*drive you cannot use it this to go to another drive (like for example *D:*). To do so you first need to change drives letters. For this simply enter the command*D:*. Then you can use the *CD* to navigate to specific folder inside the drive. Bonus tip: you can clear the screen by using the*CLS* command.

This will also install its prerequisitesJinja2and Pygments.

3. The easiest way to installNumpy is to just download its binaries from the sourceforga page. Make sure your download and install exactly the binary for your python version (so for version 2.7).

4. Download theMiktexand install it. Again just follow the wizard. At the fourth step make sure you select for the*"Install missing packages on-the-fly"*the *Yes*option, as you can see on the image below. Again this will take quite some time so be patient.

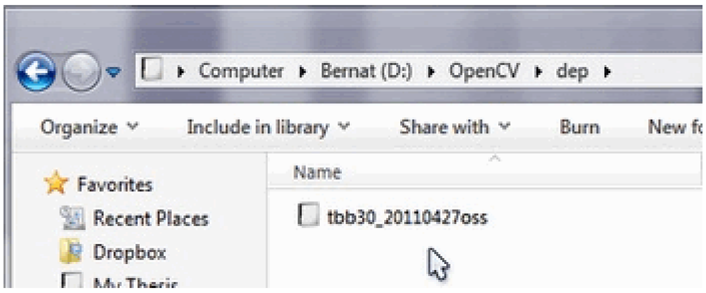

Basic MiKTeX 2.9.4146 Installer (32-bit)

Settings
Set your preferences.

Preferred paper: A4

Install missing packages on-the-fly: Ask me first

5. For the Intel © Threading Building Blocks (*TBB*) download the source files and extract it inside a directory on your system. For example let there be D:/OpenCV/dep. For installing the Intel © Integrated Performance Primitives (*IPP*) the story is the same. For extracting the archives I recommend using the7-Zip application.


Computer ▸ Bernat (D:) ▸ OpenCV ▸ dep ▸

tbb30_20110427oss

6. In case of the Eigen library it is again a case of download and extract to the D:/OpenCV/dep directory.

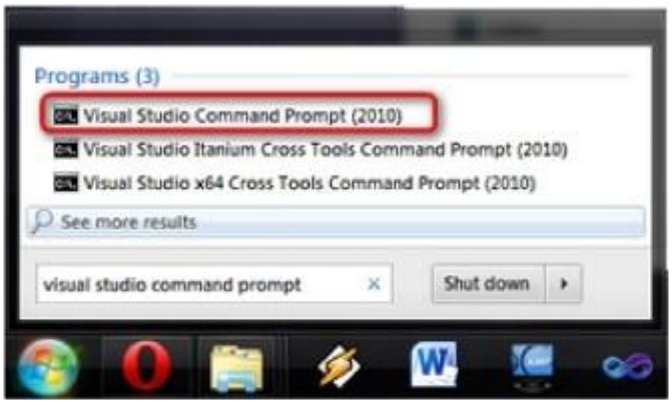
7. Same as above withOpenEXR.

8. For the OpenNI Frameworkyou need to install both the development build and thePrimeSensorModule.

9. For the CUDA you need again two modules: the latestCUDA Toolkit and the*CUDA Tools SDK*. Download and install both of them with a *complete*option by using the 32 or 64 bit setups according to your OS.

10. In case of theQtframework you need to build yourself the binary files (unless you use the Microsoft Visual Studio 2008 with 32 bit compiler). To do this go to the Qt Downloads page. Download the source files (not the installers!!!):

    Extract it into a nice and short named directory like D:/OpenCV/dep/qt/ . Then you need to build it. Start up a *VisualStudioCommandPrompt* (*2010*) by using the start menu search (or navigate through the start menu*All Programs* ‣ *Microsoft Visual Studio 2010* ‣ *Visual Studio Tools* ‣ *Visual Studio Command Prompt (2010)*).


Programs (3)
Visual Studio Command Prompt (2010)
Visual Studio Itanium Cross Tools Command Prompt (2010)
Visual Studio x64 Cross Tools Command Prompt (2010)
See more results
visual studio command prompt
Shut down

Now navigate to the extracted folder and enter inside it by using this console window. You should have a folder containing files like*Install*, *Make* and so on. Use the*dir*command to list files inside your current directory. Once arrived at this directory enter the following command:

```
configure.exe -release -no-webkit -no-phonon -no-phonon-backend -no-script -no-scripttools

 -no-qt3support -no-multimedia -no-ltcg
```

Completing this will take around 10-20 minutes. Then enter the next command that will take a lot longer (can easily take even more than a full hour):

```
nmake
```

After this set the Qt environment variables using the following command on Windows 7:

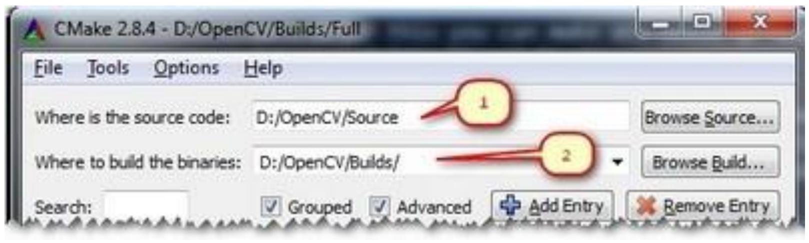
```
setx -m QTDIR D:/OpenCV/dep/qt/qt-everywhere-opensource-src-4.7.3
```

Also, add the built binary files path to the system path by using thePath Editor. In our case this is D:/OpenCV/dep/qt/qt-everywhere-opensource-src-4.7.3/bin.

Note

If you plan on doingQtapplication development you can also install at this point the*Qt Visual Studio Add-in*. After this you can make and buildQt applications without using the*Qt Creator*. Everything is nicely integrated into Visual Studio.

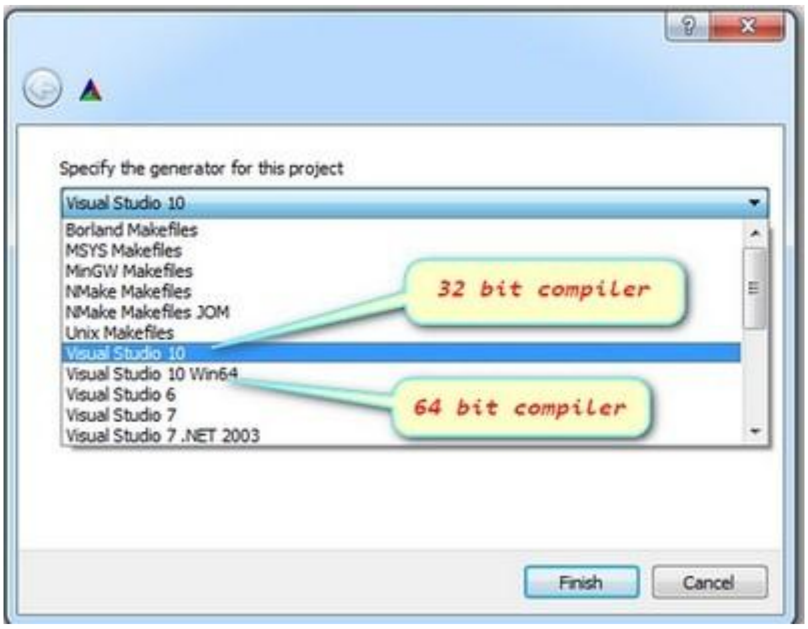
7. Now start the*CMake (cmake-gui)*. You may again enter it in the start menu search or get it from the *All Programs ‣CMake 2.8 ‣CMake (cmake-gui)*. First, select the directory for the source files of the OpenCV library (1). Then, specify a directory where you will build the binary files for OpenCV (2).


CMake 2.8.4 - D:/OpenCV/Builds/Full

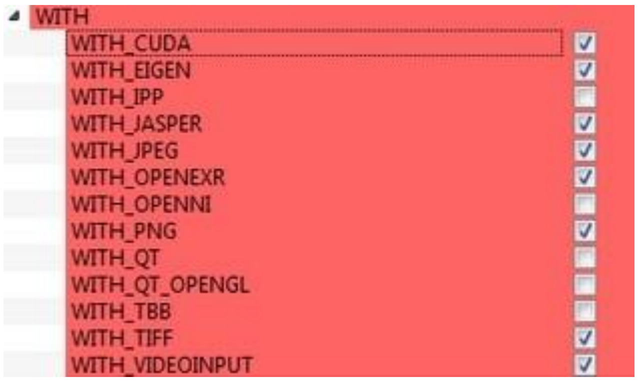
Where is the source code: D:/OpenCV/Source

Where to build the binaries: D:/OpenCV/Builds/

Press the Configure button to specify the compiler (and*IDE*) you want to use. Note that in case you can choose between different compilers for making either 64 bit or 32 bit libraries. Select the one you use in your application development.

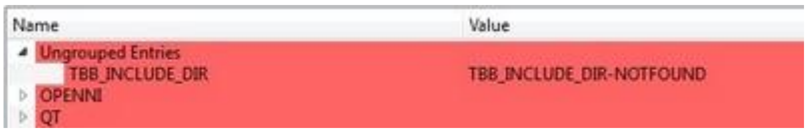


CMake will start out and based on your system variables will try to automatically locate as many packages as possible. You can modify the packages to use for the build in the*WITH* ‣ *WITH_X*menu points (where*X*is the package abbreviation). Here are a list of current packages you can turn on or off:

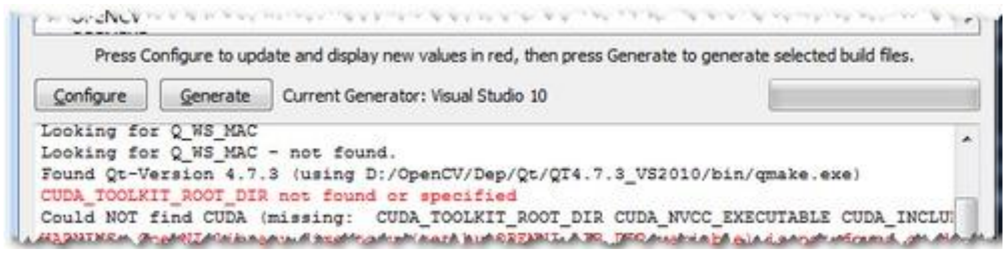


WITH	
WITH_CUDA	✓
WITH_EIGEN	✓
WITH_IPP	
WITH_JASPER	✓
WITH_JPEG	✓
WITH_OPENEXR	✓
WITH_OPENNI	
WITH_PNG	✓
WITH_QT	
WITH_QT_OPENGL	
WITH_TBB	
WITH_TIFF	✓
WITH_VIDEOINPUT	✓

Select all the packages you want to use and press again the *Configure*button. For an easier overview of the build options make sure the*Grouped*option under the binary directory selection is turned on. For some of the packages CMake may not find all of the required files or directories. In case of these CMake will throw an error in its output window (located at the bottom of the GUI) and set its field values, to not found constants. For example:



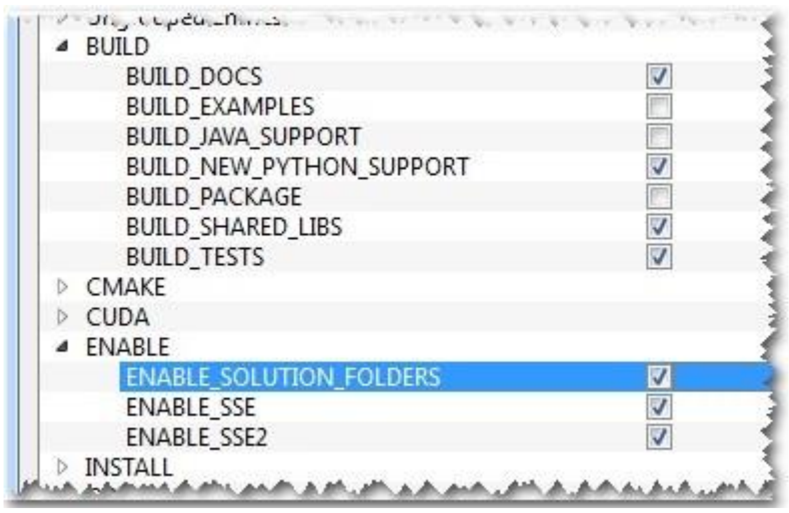
Name	Value
Ungrouped Entries	
TBB_INCLUDE_DIR	TBB_INCLUDE_DIR-NOTFOUND
OPENNI	
QT	



Press Configure to update and display new values in red, then press Generate to generate selected build files.

```
Looking for Q_WS_MAC
Looking for Q_WS_MAC - not found.
Found Qt-Version 4.7.3 (using D:/OpenCV/Dep/Qt/QT4.7.3_VS2010/bin/qmake.exe)
CUDA_TOOLKIT_ROOT_DIR not found or specified
Could NOT find CUDA (missing:  CUDA_TOOLKIT_ROOT_DIR CUDA_NVCC_EXECUTABLE CUDA_INCLU
```

For these you need to manually set the queried directories or files path. After this press again the *Configure*button to see if the value entered by you was accepted or not. Do this until all entries are good and you cannot see errors in the field/value or the output part of the GUI. Now I want to emphasize an option that you will definitely love: *ENABLE* ‣ *ENABLE_SOLUTION_FOLDERS*. OpenCV will create many-many projects and turning this option will make sure that they are categorized inside directories in the*Solution Explorer*. It is a must have feature.



BUILD	
BUILD_DOCS	✓
BUILD_EXAMPLES	
BUILD_JAVA_SUPPORT	
BUILD_NEW_PYTHON_SUPPORT	✓
BUILD_PACKAGE	
BUILD_SHARED_LIBS	✓
BUILD_TESTS	✓
CMAKE	
CUDA	
ENABLE	
ENABLE_SOLUTION_FOLDERS	✓
ENABLE_SSE	✓
ENABLE_SSE2	✓
INSTALL	

Furthermore, you need to select what part of OpenCV you want to build.

- *BUILD_DOCS* -> It creates two projects for building the documentation of OpenCV (there will be a separate project for building the HTML and the PDF files). Note that these aren't built together with the solution. You need to make an explicit build project command on these to do so.

- *BUILD_EXAMPLES* -> OpenCV comes with many example applications from which you may learn most of the libraries capabilities. This will also come handy to easily try out if OpenCV is fully functional on your computer.

- *BUILD_PACKAGE* -> Prior to version 2.3 with this you could build a project that will build an OpenCV installer. With this you can easily install your OpenCV flavor on other systems. For the latest source files of OpenCV it generates a new project that simply creates zip archive with OpenCV sources.

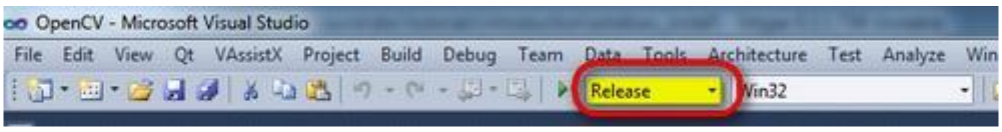
- *BUILD_SHARED_LIBS* -> With this you can control to build DLL files (when turned on) or static library files (*.lib) otherwise.

- *BUILD_TESTS* -> Each module of OpenCV has a test project assigned to it. Building these test projects is also a good way to try out, that the modules work just as expected on your system too.

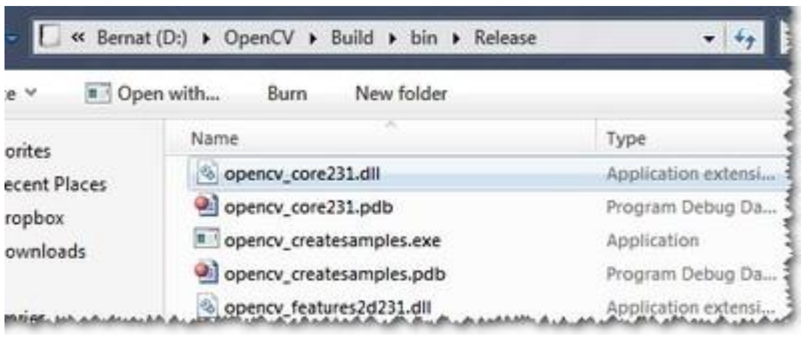
- *BUILD_PERF_TESTS* -> There are also performance tests for many OpenCV functions. If you're concerned about performance, build them and run.

- *BUILD_opencv_python* -> Self-explanatory. Create the binaries to use OpenCV from the Python language.

Press again the*Configure*button and ensure no errors are reported. If this is the case you can tell CMake to create the project files by pushing the *Generate*button. Go to the build directory and open the created**OpenCV** solution. Dependingon just how much of the above options you have selected the solution may contain quite a lot of projects so be tolerant on the IDE at the startup. Now you need to build both the*Release*and the *Debug* binaries. Use the drop-down menu on your IDE to change to another of these after building for oneof them.

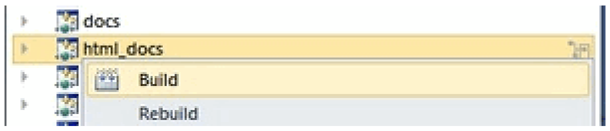

OpenCV - Microsoft Visual Studio
File Edit View Qt VAssistX Project Build Debug Team Data Tools Architecture Test Analyze Win
Release
Win32

In the end you can observe the built binary files inside the bin directory:

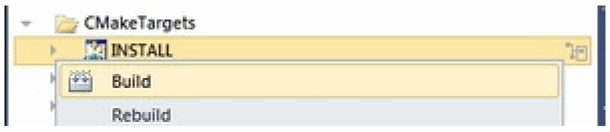


Name	Type
opencv_core231.dll	Application extensi...
opencv_core231.pdb	Program Debug Da...
opencv_createsamples.exe	Application
opencv_createsamples.pdb	Program Debug Da...
opencv_features2d231.dll	Application extensi...

For the documentation you need to explicitly issue the build commands on the *doc* project for the PDF files and on the*doc_html*for the HTML ones. Each of these will call*Sphinx*to do all the hard work. You can find the generated documentation inside the Build/Doc/_html for the HTML pages and within the Build/Doc the PDF manuals.

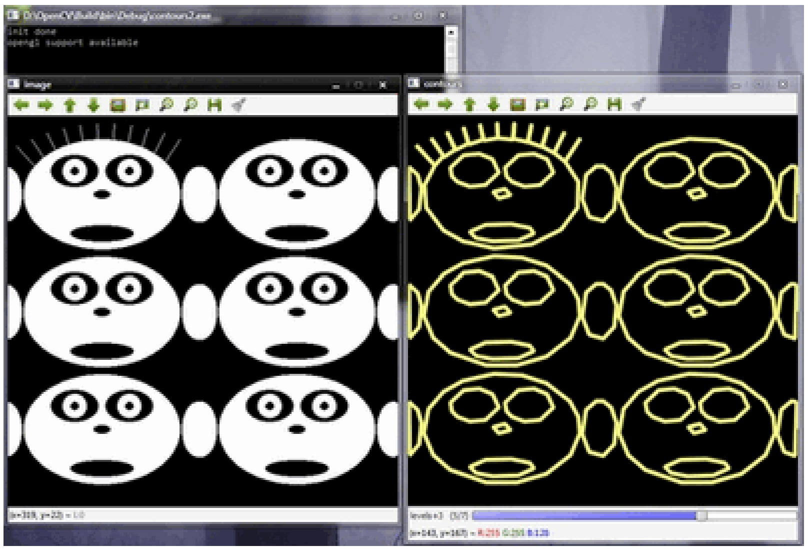


To collect the header and the binary files, that you will use during your own projects, into a separate directory (similarly to how the pre-built binaries ship) you need to explicitly build the*Install* project.



This will create an*Install*directory inside the*Build*one collecting all the built binaries into a single place. Use this only after you built both the *Release* and*Debug*versions.

To test your build just go into the Build/bin/Debug or Build/bin/Release directory and start a couple of applications like the*contours.exe*. If they run, you are done. Otherwise, something definitely went awfully wrong. In this case you should contact us at ourQ&A forum. If everything is okay the*contours.exe* output should resemble the following image (if built withQtsupport):



```
init done
opengl support available
```

Note

If you use the GPU module (CUDA libraries) make sure you also upgrade to the latest drivers of your GPU. Error messages containing invalid entries in (or cannot find) the nvcuda.dll are caused mostly by old video card drivers. For testing the GPU (if built) run the*performance_gpu.exe* sample application.

Qtfor Embedded Linux

Qtfor Embedded Linux is a C++ framework for GUI and application development for embedded devices. It runs on a variety of processors, usually with EmbeddedLinux. Qtfor Embedded Linux provides the standardQt API for embedded devices with a lightweight window system.

QtCreator

QtCreator is a cross-platform C++, JavaScript and QML integrated development environment which is part of the SDK for the Qt GUI Application development framework.[4] It includes a visual debugger and an integrated GUI layout and forms designer.

The editor's features include syntax highlighting and autocompletion, but not tabs. Qt Creator uses the C++ compiler from the GNU Compiler Collection on Linux and FreeBSD. On Windows it can use MinGW or MSVC with the default install and can also use cdb when compiled from source code. Clang is also supported.

IDE Overview

Qt Creator is an integrated development environment (IDE) that provides you with tools to design and develop applications with theQtapplication framework. Qt is designed for developing applications and user interfaces once and deploying them across several desktop and mobile operating systems. Qt Creator provides you with tools for accomplishing your tasks throughout the whole application development life-cycle, from creating a project to deploying the application on the target platforms.

4.6 OCR (Optical Character Recognition)

FreeOCR is a free Optical Character Recognition Software for Windows and supportsscanning from most Twain scanners and can also open most scanned PDF's and multi page Tiff images as well as popular image file formats. FreeOCR outputs plain text and can export directly to Microsoft Word format.

Free OCR uses the latest Tesseract (v3.01) OCR engine. It includes a Windows installer and It is very simple to use and supports opening multi-page tiff documents, Adobe PDF and fax documents as well as most image types including compressed Tiff's which the Tesseractengine on its own cannot read .It now can scan using Twain and WIA scanning drivers.

Scanning Software

As well as OCR FreeOCR can scan and save images as JPG's and we are currently working on "Scan to PDF" capability with the option to save as searchable PDF

OCR Engine

The included TesseractOCR PDF engine is an open source product released by Google. It was developed at Hewlett Packard Laboratories between 1985 and 1995. In 1995 it was one of the top 3 performers at the OCR accuracy contest organized by University of Nevada in Las Vegas.

License

FreeOCR is a freeware OCR& scanning software and you can do what you like with it including commercial use. The includedTesseract OCR engine is distributed under the Apache V2.0 license.

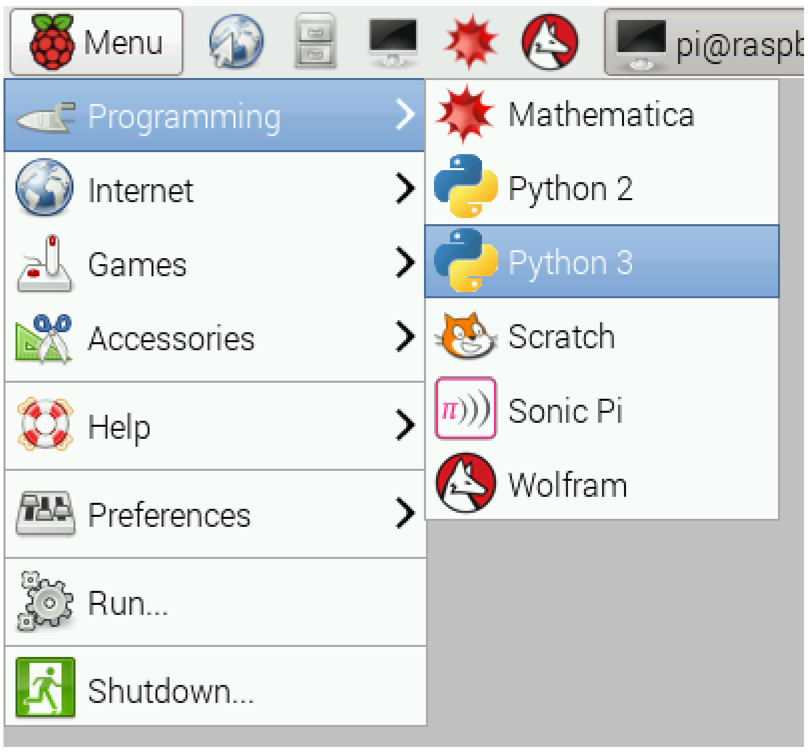
4.7 Python

Python is a wonderful and powerful programming language that's easy to use (easy to read**and**write) and with Raspberry Pi lets you connect your project to the real world.

Python syntax is very clean, with an emphasis on readability and uses standard English keywords. Start by opening IDLE from the desktop.

IDLE

The easiest introduction to Python is through IDLE, a Python development environment. Open IDLE from the Desktop or applications menu:



IDLE gives you a REPL (Read-Evaluate-Print-Loop) which is a prompt you can enter Python commands in to. As it's a REPL you even get the output of commands printed to the screen without using print.

You can use variables if you need to but you can even use it like a calculator. For example:

```
>>> 1 + 2
3
>>> name = "Sarah"
>>> "Hello " + name
'Hello Sarah'
```

IDLE also has syntax highlighting built in and some support for autocompletion. You can look back on the history of the commands you've entered in the REPL with Alt + P (previous) and Alt + N (next).

Basic Python usage

Hello world in Python:

print("Hello world")

Simple as that!

Comments

Comments are ignored in the program but there for you to leave notes, and are denoted by the hash # symbol. Multi-line comments use triple quotes like so:

```
"""

This is a very simple Python program that prints "Hello".

That's all it does.

"""



print("Hello")
```

Lists

Python also has lists (called arrays in some languages) which are collections of data of any type:

```
numbers = [1, 2, 3]
```

Lists are denoted by the use of square brackets [] and each item is separated by a comma.

4.8 SOURCE CODE:

Program Code:

```
import CV2

from PIL import Image
from pytesser import *

Image  FILE = ‘My Image.jpg’

#loop forever

While True:

#Save image from webcam

Print “save Image”

img = CV2.VideoCapture(0).read()[1]

CV2.imwrite(IMAGE_FILE,img)

#load image

Print “load Image”

Img = Image .open(IMAGE_FILE)

#detect words in image

Print “Detcting words”

Words = image_to_string(img).strip()

time.sleep(1)

print “words Detected pls show Below”

print words

time.sleep(2)
```

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 CONCLUSION

The project wasdesigned a system which captures the image of the number plate of a vehicle and these details of number plate were displayed on monitor using Raspberry Pi processor.

5.2 FUTURE WORK

This paper can be extended using high efficiency GPS receiver and a GPRS module. The GPRS module gives the intimation of the vehicle racking directly on to the predefined web link for tracking the vehicle on Google maps. The project can be extended using USB camera for vehicle monitoring from longer distances. The project can be extended using memory card using which the traveled path can be stored which helps in storing the tracked path along with speed and time.

CHAPTER 6

APPENDICES

- ANPR-Automatic Number Plate Recognition
- API-Application Programming Interface
- ATM-Asynchronous Transfer Mode
- CPU-Central Processing Unit
- CSI-Camera Serial Interface
- DSI-Display Serial Interface
- EPROM-Erasable Programmable Read Only Memory
- GPIO-General Purpose Input Output
- GUI-Graphical User Interface
- HDMI-High Definition Multiple Interface
- IP-Internet Protocol
- JTAG-Joint Test Action Group
- LAN-Local Area Network
- MIPI-Mobile Industry Processor Interface
- OCR-Optical Character Recognition
- RAM-Random Access Memory
- ROM-Read Only Memory
- SD-Storage Disk
- USB-Universal Serial Bus
- VGA-Video Graphics Array

CHAPTER 7

REFERENCES/BIBLIOGRAPHY

[1] B. Hongliangand L. Changping, “A hybrid license plate extraction method based on edge statistics and morphology,” in Proc. ICPR, pp. 831-834, 2004.

[2] D. Zheng, Y. Zhao, and J. Wang, “An efficient method of license plate location,” Pattern Recognition. Lett., vol. 26, no.15, pp. 2431-2438, Nov. 2005.

[3] H.J. Lee, S.Y. Chen, and S.Z. Wang, “Extraction and recognition of license plates of motorcycles and vehicles on highways,” in Proc. ICPR, pp. 356-359, 2004.

[4] A. Broumandnia and M. Fathy, “Application of pattern recognition for Farsi license plate recognition,” presented at the ICGST Int. Conf. Graphics, Vision and Image Processing (GVIP), Dec. 2005. [Online]. Available: http://www.icgst.com/gvip/v2/P1150439001.pdf

[5] T. D. Duan, T. L. Hong Du, T. V. Phuocand N. V. Hoang, “Building an automatic vehicle license plate recognition system,” in Proc. Int. Conf. Comput. Sci. RIVF, pp. 59-63, 2005.

[6] C.T. Hsieh, Y.S Juan, and K.M. Hung, “Multiple license plate detection for complex background,” in Proc. Int. Conf. AINA, vol. 2, pp. 389-392, 2005.

[7] D.S. Kim and S.I. Chien, “Automatic car license plate extraction using modified generalized symmetry transform and image warping,” in Proc. ISIE, pp. 2022-2027, 2001.

[8] S.H. Park, K.I. Kim, and H.J. Kim, “Locating car license plate using neural networks,” Electron. Lett., vol. 35, no.17, pp. 1475- 1477, 2005.

[9] A. Ebrahimiand A. Raie, “License Plate Character Recognition Using Multiclass SVM,” J Am Sci, vol. 8, no. 1s, pp. 38-43, 2012.

[10] National Policing Improvement Agency, “Practice Advice on the Management and Use of Automatic Number Plate Recognition ,” 2009.

[11] L. Dignan, “ARM Holdings 2015 Plan : Grab PC, server share,” Feb 3rd 2011, http://www.zdnet.com/blog/btl/arm-holdings-2015-plan-grab-pcserver- share/44386

[12] G. Bradskiand A. Kaehler, “LearningOpenCV,” ISBN: 978-0- 59651613-0.

[13] M. Marengoni and D. Stringhini, “High Level Computer Vision using OpenCV,” 24th SIBGRAPI Conference on Graphics, Patterns and Images Tutorials 2011, pp. 11-24.

[14] Y. Liu, L. Cui, J. Shu and G. Xin, “License Plate Location Method Based on Binary Image Jump and Mathematical Morphology,” International Journal of Digital Content Technology and its Applications 2011, pp. 259–265.

[15] H. Lee, D, Kim, D. Kim and S.Y. Bang, “Real Time Automatic VehicleManagementSystem using Vehicle Tracking and Car Plate Number Identification,”

[16] J.H. Hsieh, S. H. Yu and Y. S. Chen, “Morphology Based license plate detection from complex scenes,” 16th International Conference on Pattern Recognition (ICPR) 2012, pp. 79-179.

[17] M. T. Qadriand M. Asif, “Automatic Number Plate Recognition System for Vehicle Identification using Optical Character Recognition,” International Conference on Education Technology and Computer 2009, pp. 335-338.