DIGITAL PHOTO FRAME

A Thesis report submitted in partial fulfilment of the requirements

for the award of the degree in

Master of Technology (Dual Degree)

In

"VLSI Design And Embedded Systems"

Prashant Kumar Jha

(Roll No. 710EC2150)

May, 2015

Department Of Electronics And Communication Engineering

National Institute Of Technology, Rourkela

Digital Photo Frame With Audio Playback And Recording

A Thesis report submitted in partial fulfilment of the requirements

for the award of the degree in

Master of Technology (Dual Degree)

in

"VLSI Design and Embedded Systems"

By

Prashant Kumar Jha

(Roll No. : 710EC2150)

May, 2015

Under the guidance of

Prof. A.K Swain

Department of Electronics and Communication Engineering

National Institute of Technology

Rourkela-769008

DEPARTMENT OF ELECTRONICS AND

COMMUNICATION ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY, ROURKELA

ODISHA, INDIA -769008

CERTIFICATE

This is to certify that the thesis entitled **"Digital Photo Frame"**, submitted by **Prashant Kumar Jha (Roll No. 710EC2150)** in partial fulfilment of the requirements for the award of **Master of Technology (Dual Degree)** in **VLSI Design And Embedded Systems** during session 2014-2015 at National Institute of Technology, Rourkela.

The candidate has fulfilled all the prescribed requirements. The Thesis is an authentic work, based on candidates' own work.

To my knowledge, this thesis is up to the standard required for the award of a Bachelor of Technology in Master of Technology (Dual Degree) in VLSI Design and Embedded Systems.

Place: Rourkela

Prof. A.K Swain

Assistant Professor

Department of Electronics and Communication Engineering

National Institute of Technology

Rourkela -769008

ACKNOWLEDGEMENTS

I wish to express my heartfelt gratitude to my supervisor **Prof. A.K Swain**, Assistant Professor, Department of Electronics and Communication Engineering, National Institute of Technology, Rourkela for his valuable support, guidance and time throughout my project. I also appreciate the freedom provided to me by provided by **Prof. A.K Swain** to explore new ideas in my project.

I am also grateful to **Prof. S. K. Sarangi**, Director, National Institute of Technology, Rourkela for providing me with outstanding facilities in the institute for my research.

I would also like to thank Prof. K.K Mahapatra, Head of Department, Department of Electronics and Communication Engineering, National Institute of Technology, Rourkela for providing facilities during this project work.

Finally, I want to thank my parents and the almighty god for their backing, without which this would not have been conceivable.

Prashant Kumar Jha

Roll No. 710EC2150

TABLE OF CONTENTS

Topics **Page Number**

ABSTRACT

With the advancement in semiconductor technology, scope for development of embedded systems has increased manifolds. New processors with improved computing capabilities and low power consumption have further accelerated the developments in embedded domain. Consumers are looking for affordable multimedia devices with high performance and durability making embedded developers to think creatively and use all resources at hand to meet the desired user specifications. This is one such attempt by designing a digital photo frame with 5 inch LCD display intended to display high quality BMP images. Small Size and low cost of development can prove to be very useful in the success of the device as a day to day consumer electronics product. Powerful computing capabilities of ARM processor when duly utilised can produce very elegant results.

LIST OF FIGURES

Figure No.	Figure Title	Page No.

LIST OF TABLES

Table No.	Table Title	Page No.

CHAPTER 1

INTRODUCTION

- Motivation
- Organisation of thesis

1 INTRODUCTION

1.1 Project Overview

The conventional table top photo frames have to be used with printed photograph and if one intend to change it he/she will have to do it manually. In that context a digital photo frame can prove to be an excellent replacement for the same where one does not have to manually change printed photographs but can do so but just pressing a button. Moreover the continuous slideshow of photographs on the screen may seem very lucrative feature to the users. Keeping in mind the specifications that a photo frame should have in todays' digitised world the current project has been thought of. The main multimedia functionality that most of the consumers want are image display. This project has paid due weightage to this functionality thus increasing the probability that it will become popular with the people using it. Because of the low cost of development and the capabilities to display high quality BMP images which itself is a lossless image format, the quality of the output is nowhere compromised. Thus if one is seeking quality display of images on their table tops at a very modest price, the device thus developed may come very handy.

With the improvements in processor technology and development of software design environments for the same, designing of an embedded system has not remained a very tedious task as it was a decade ago. Processors such ARM which have very good computing performances are becoming more popular among embedded developers and are also coming at low prices.

1.2 Organisation Of Thesis

Chapter 1: This chapter includes brief introduction and organisation of thesis

Chapter 2: This chapter deals with ARM microcontrollers giving brief idea if RISC methodology of processor design, development board used and the keil uvision MDK pro which is a design environment for ARM processors

Chapter 3: This chapter deals with the file formats that needed to worked upon describing in detail about the BMP image.

Chapter 4: This chapter deals with various communication protocols that have been used for making components communicate with processor and among each other.

Chapter 5: This chapter deals with the idea of FAT32 file system and SD card interfacing.

Chapter 6: This chapter explains conclusion and future developments that can be made using this thesis work .

Chapter 7: This chapter provides the various references used in this project work.

CHAPTER 2

ARM MICROCONTROLLERS

- Reduced Instruction Set Computer (RISC)
- ARM Processors
- STM32F4 Discovery Board

2 ARM MICROCONTROLLERS AND TOOLCHAINS

2.1 Reduced Instruction Set Computer (RISC)

RISC is a design methodology for developing CPUs based on the belief that a simple instruction set combined with a processor architecture equipped to execute these instructions provides enhanced performance with less processor cycle per instruction thus consuming less circuitry (transistors) [1] .Its most common trait is the use of Load/Store architecture in which memory is generally used only through particular instruction only instead of using it as a part of some other instruction. Due to reduced no. of cycles per instruction (CPI) it provides higher speed. This is done by optimizing every instruction on CPU and Pipelining.

2.1.1 Characteristics:

- Fewer instructions rather than a very large and complex instruction set.
- Fewer addressing modes making it more flexible and user friendly.
- Operations required to be performed takes place within CPU itself.
- Every instruction is executed in same amount of time (cycles) and hence enhances the speed of the system.
- In this design methodology processors have sufficiently large number of registers and a considerably more productive instruction pipeline.
- Instructions are of constant length and easy to decode.

2.1.2 RISC vs CISC

Following table listing the difference between two processor design methodologies.

Table 1 Difference between RISC and CISC

RISC	CISC
Stress on Software	Stress on Hardware
Fixed instruction size and very few formats.	Various instruction sizes and formats
Large number of registers	Fewer registers
Lesser addressing modes	Wide variety of addressing modes
complex compiler	Large scale use of microprogramming.
All instructions take only 1 processor cycle for execution.	Different time of executions for different instructions.
easy pipelining	Pipelining difficult.

2.1.3 Advantages of RISC over CISC

- Since a streamlined instruction set takes into account a pipelined, superscalar plan RISC processors frequently attain to 2 to 4 times the execution of CISC

processors utilizing tantamount semiconductor innovation and the same clock rates.

- Since the instruction set of a RISC processor is so basic, it uses up substantially less chip space; additional capacities, for example, memory management units or floating point arithmetic units, can likewise be set on the same chip. Littler chips permit a semiconductor producer to place more parts on a solitary silicon wafer, which can bring down the every chip cost significantly.
- Since RISC processors are more straightforward than relating CISC processors, they can be developed more rapidly, and can exploit other technological advancements faster than their CISC counterparts, prompting more noteworthy jumps in performance in upcoming eras.

2.1.4 Disadvantages of RISC over CISC

- Performance of a processor developed with RISC methodology is greatly determined by the quality of code which it has to execute. A poor job on programmer's part or compiler can lead to processor spending lot of time stalling. Since instruction scheduling is a tedious task, most of programmers choose high level languages such as C or C++, leaving the job of scheduling to compilers. Thus compiler has to be chosen carefully to generate quality code, making RISC processors compiler dependent.
- Instruction scheduling makes debugging a tedious task. When IS is taken into account then machine language equivalent of a line of source can appear as in between other instructions of source code.
- Code expansion can be a problem with RISC processors as a complex operation can consume many instruction which can be done by a single instruction with CISC processors.
- Very fast Memory systems are required with RISC processors which are capable of feeding instructions to processor at fast rates. Thus RISC systems

are required of having big memory caches generally within the chip. This incorporation is also called First-Level cache.

2.2 ARM Processors

This is a family of 32-bit microcontrollers developed on RISC (Reduced Instruction set computing) methodology developed by a company from Great Britain ARM Holdings. Unlike its CISC (Complex Instruction Set Computing) counterparts it consists of fewer transistors thus reducing cost, heat dissipation and power consumption, which are desirable for designing light, battery powered devices such as smart phones , tablets etc. The company designs instruction sets and architecture but not product itself.

2.2.1 History

Development of first ARM processor dates back to 1980s when Acorn RISC Machine Architecture (ARM) was developed by British Acorn Computers, its first product being coprocessor modules for a series of computers BBC Micro. It was developed using VLSI Technology as silicon partners. It included data bus(32 bits), address space(26 bits) and 27 registers (32 bits).Out of all bits of PC(program counter) register, 8 were available for other purposes, 6 for Status Flags and 2 bits were used for choosing setting mode.

2.2.2 Operating System Support

ARM architecture supports a huge number of embedded and real time operating systems like Linux, embedded C, freeRTOS etc. Both 32 and 64 bit operating system are supported on ARM architecture.

2.2.3 ARM Cortex M4

ARM Cortex M4 is a 32-bit microcontroller based on RISC (Reduced Instruction Set Computer) architecture specifically designed for efficient signal processing applications. It highlights extended Multiply- Accumulate (MAC) instructions, enhanced SMID arithmetic instructions and a Floating Point Unit (FPU) [2]. Basically ARM M4 is nothing but M3 with special Digital Signal Processing (DSP) instructions. The blend of high-proficiency signal transforming utility with low-power consumption, minimal expenses and wide usability of the Cortex-M group of processors is intended to fulfil the emerging class of adaptable solutions particularly focusing on the engine control, car, power optimization, embedded application and modern robotic markets.

Key features of ARM Cortex M4 can be summarised as follows

- ARMv7E Architecture
- Three stage Pipelining
- Instruction set contains
    - Thumb Instructions which are a subset of ARM instruction in 16 bit compressed mode.

- 32 bit hardware multiply
- 32 bit hardware divide
- DSP support

- 1 to 240 interrupts with Non Maskable Interrupts (NMI)
- Sleep modes are also available

2.2.3 STM32F4 Discovery board

STM32F4 Discovery board is a product of ST Microelectronics which has a STM32 microcontroller based on ARM Cortex M4 architecture. It includes an ST-LINK/V2 embedded debug tool interface, ST MEMS digital accelerometer, ST MEMS digital microphone, audio DAC with integrated class D speaker driver, LEDs, push buttons and a USB OTG micro AB connector.

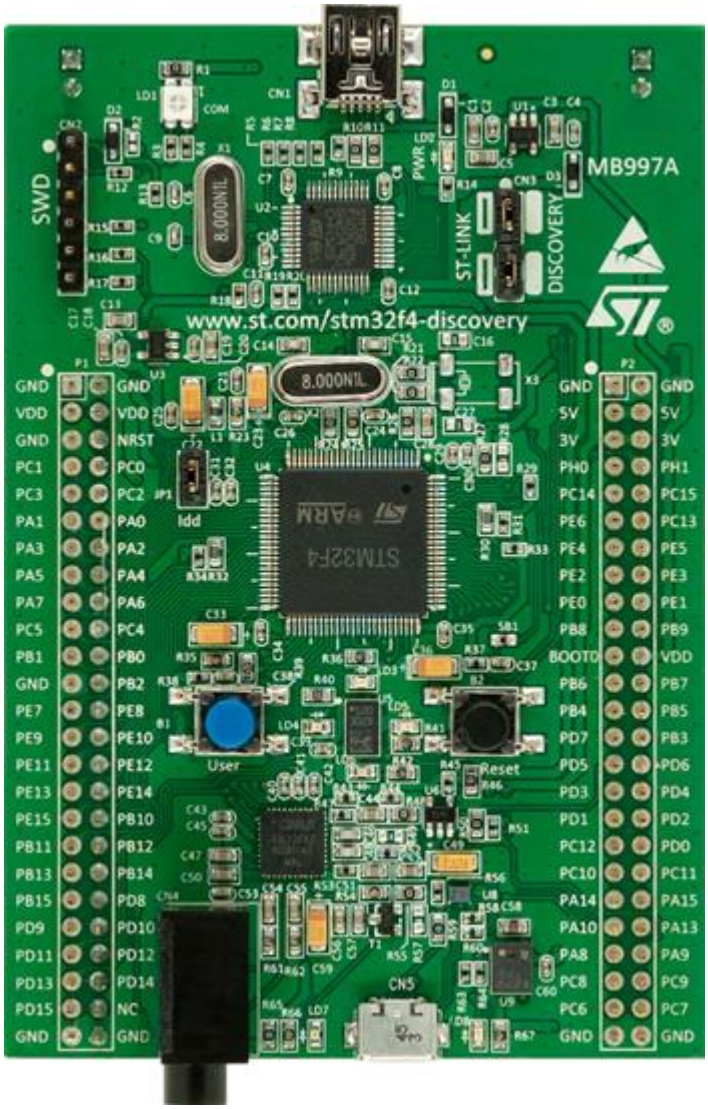

Figure 1 STM32F4 discovery board

It is a low cost easy to use development kit to quickly start development with powerful STM32 microcontrollers. Tool chain used is Keil uvision MDK pro which provides inbuilt libraries and a user friendly design environment for STM32F4. Hardware block diagram of the discovery board is shown below.

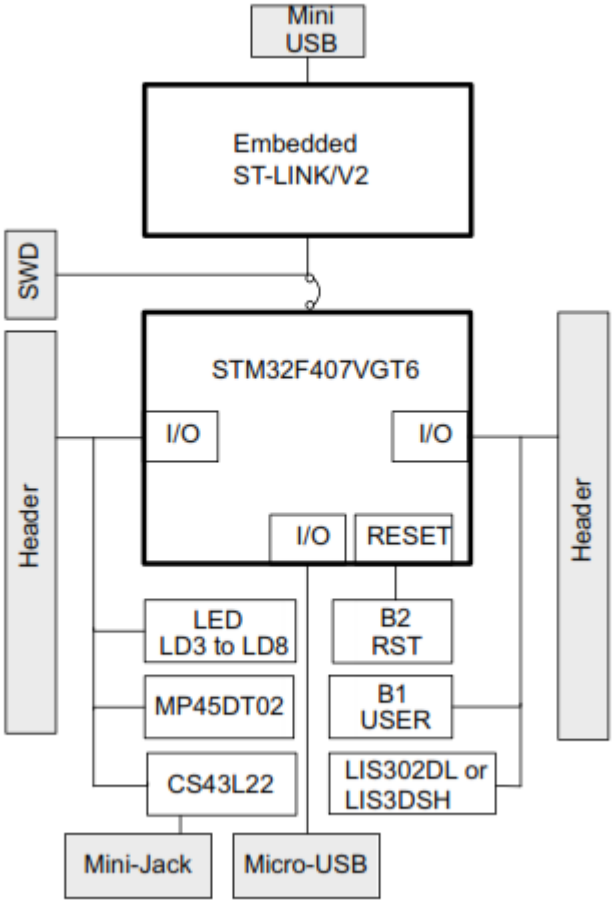

Figure 2 Hardware block diagram of STM32F4 Discovery board

It has ARM Cortex M4 32 bit MCU which is characterised by 210 DMIPS, up to 1 MB flash/192+4 KB RAM, USB OTG HS/FS, Ethernet, 3 ADCs and 15 communication interfaces.

A demonstration software has been already loaded in the flash memory of the board. It makes use of MEMS motion sensor for blinking four LEDs according to motion direction and speed. By connecting the discovery to a computer with ‘type A to micro B’cable , it can act as a standard mouse.

Following figure describes the overall MCU structure of STM32F4.

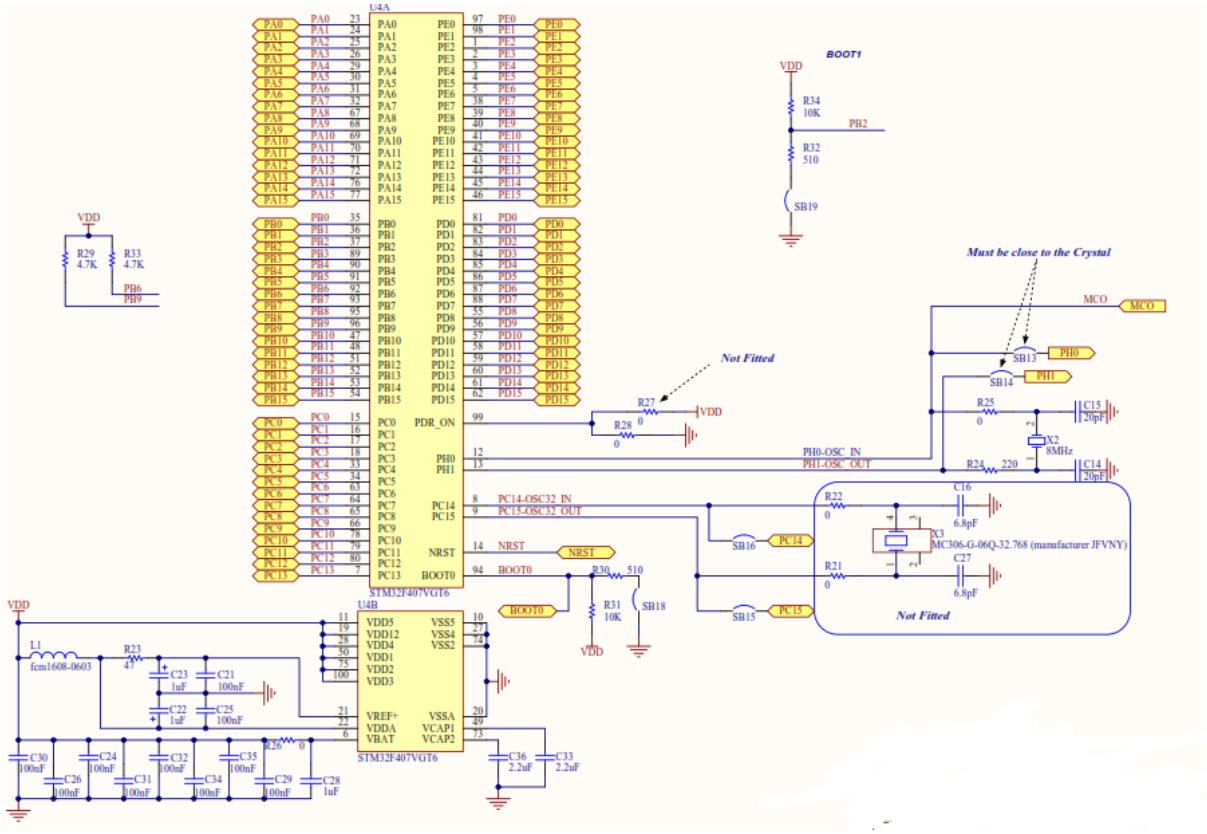

Figure 3 STM32F4 Discovery MCU

2.3 Keil uvision MDK pro Tool chain

Keil MDK pro is a complete software development environment for a variety of ARM based microcontrollers including Cortex M and Cortex R series. It includes uvision IDE/Debugger, ARM C/C++ compiler and all the additional middleware components required for ARM based embedded development [3]. It also includes Keil RTX which is a Real Time Operating System (RTOS). It also features a complete GUI library for designing Graphical User Interface (GUI) for embedded systems.

Run time environment window shows all software components that are compatible with the selected device. These pre designed software components helps to design embedded systems faster. It creates complete design environment for the chosen device. Various components required in the project can be manually chosen from the list provided for the target board. It also provides us with the middleware software requirements of each of these components. It also indicates the communication protocol required to be configured for each component for their smooth communication with the target microcontroller.

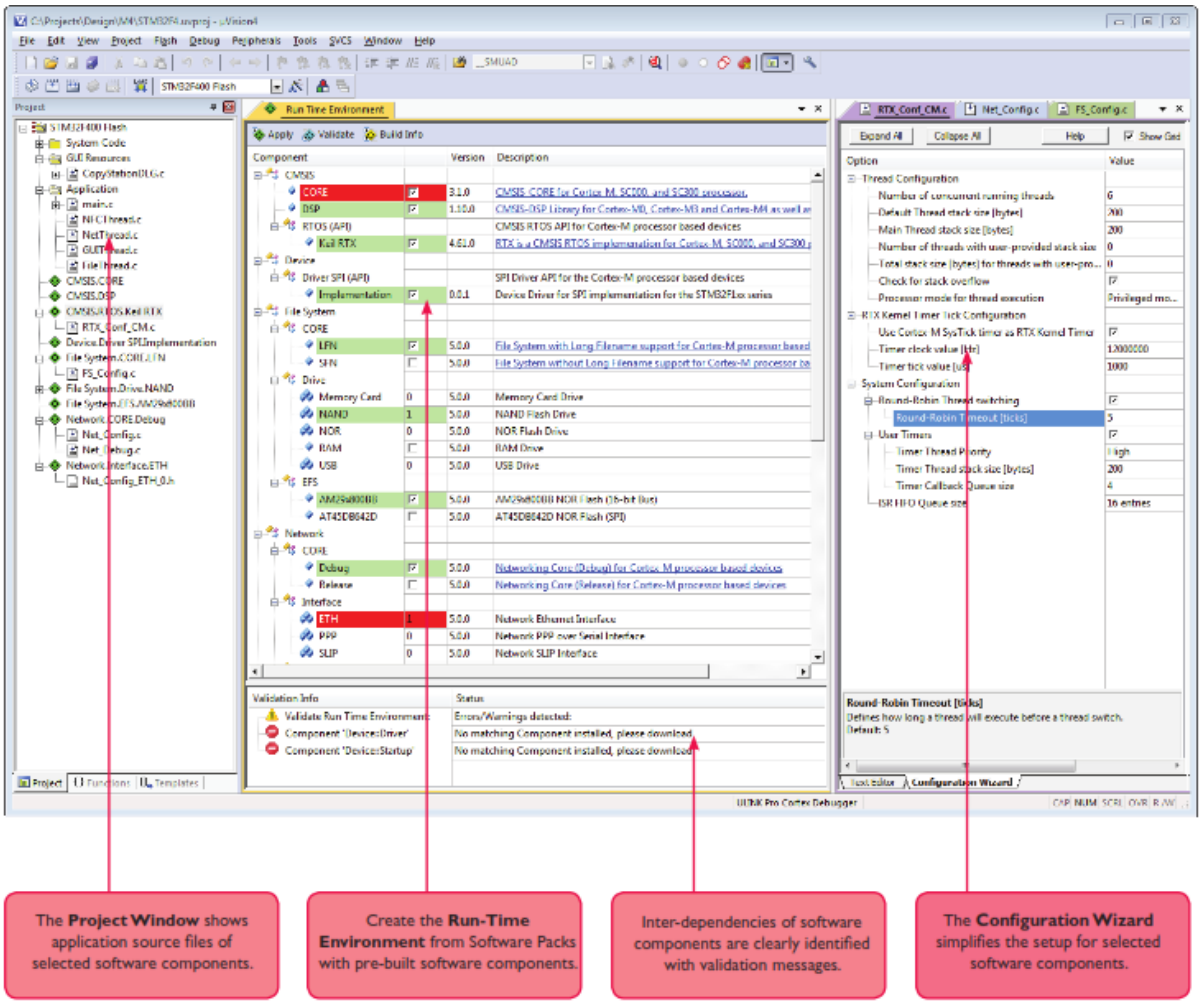

Figure 4 Run time environment of Keil MDK pro

It has a powerful source code editor which enhances productivity and is a user friendly environment for compiling source codes. While compiling the source code, editor provides

with excellent debugging mechanism enlisting each error during compilation and indicating their location in source code or header files and libraries. With such qualities writing a source code in this design environment become very simplified and easy to use.

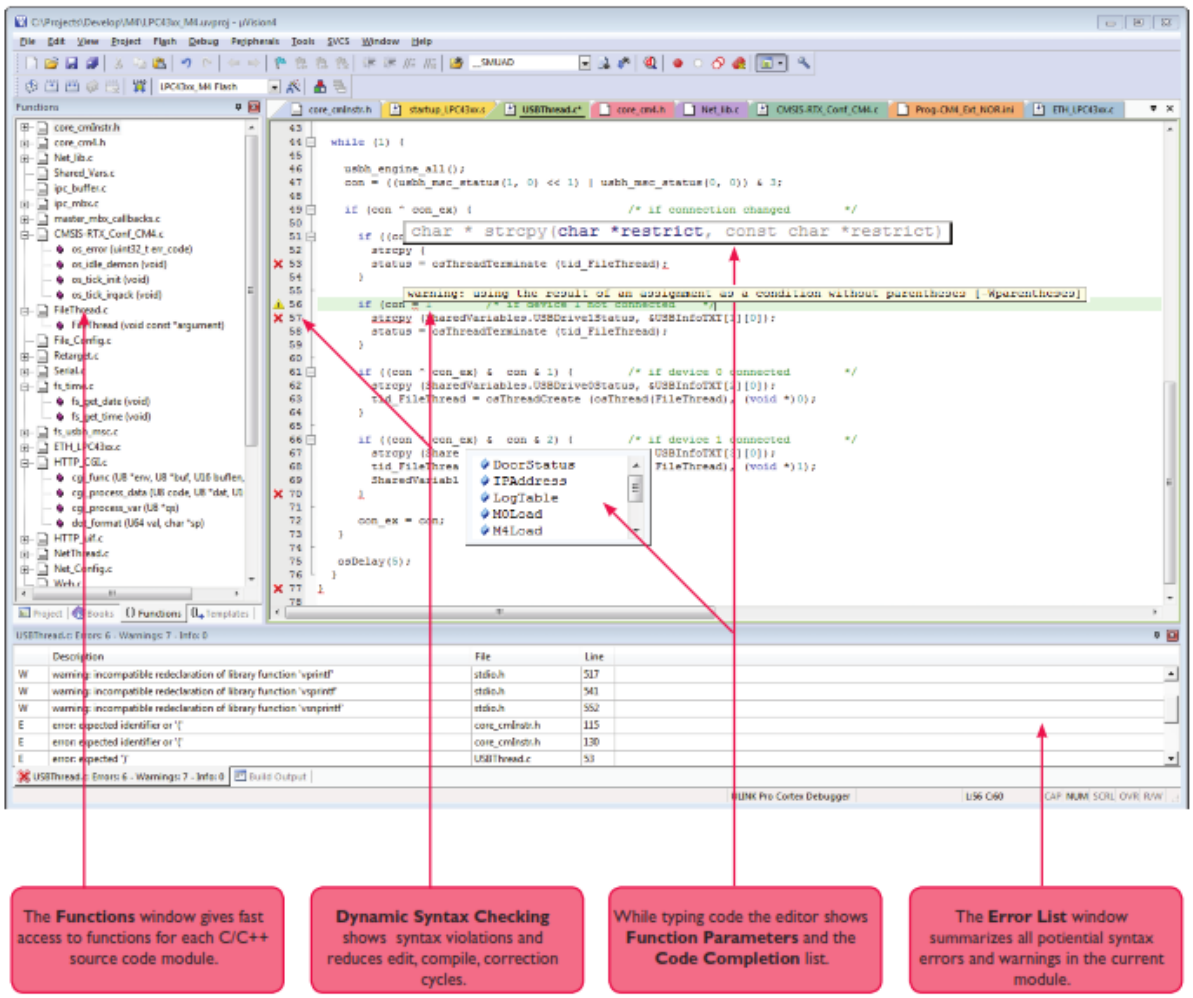

Figure 5 Source code editor

CHAPTER 3

BMP FILE FORMAT

- BMP image
- BMP File Structure
- Usage of BMP

3 BMP FILE FORMAT

3.1 BMP image

One of the most frequently used application of a computer is to display, store and make changes to graphical images. An image in real world that our eyes see are analogue where colours blend smoothly. But computers being digital can only process digital data. Thus if an image is to be processed by a computer it should be first converted into binary data by digitizing the analogue image. A digital camera for example directly creates a digital image with help of image sensor, which is then transferred to PC or some other display device with help of USB , Bluetooth or any other inter device transfer protocol.

Most straightforward method of representing an image in digital (binary) form is by creating a bitmap. In this method image is stored as array of binary numbers known as pixels forming a rectangular grid throughout the display area. Each pixel consists of 24 bits where each byte (8 bits) represents colour Red, Green and Blue respectively [4]. This representation has binary data on every pixel thus its size is very large. A bitmap image is of very high quality as all the data is present in the pixel wise description and no loss of data is there as no compression scheme is employed. Basically BMP is an uncompressed image file which makes its size very large in magnitude thus generally not used in web application but its simplicity makes it a very useful format where quality of image is of utmost importance than the size of the file. The effortlessness of the BMP file format, and its far reaching recognition in Windows and , and also the way that this configuration is well documented and free of licenses, makes it an exceptionally normal arrangement that image processing applications from numerous working frameworks can read and compose.

3.2 BMP File Format

BMP file comprises of fixed (header) as well variable sized structures which appears in a predestined fashion. [5]

a) **File header:** This part is utilized for identification of file. It is at the start of the file. Any application is supposed to read this part of the file first to determine whether the file is BMP or not and also to ensure that the file is undamaged. The very first 2 bytes of the header file are characters "B" and "M'" in ASCII which suggests the application that file indeed is BMP. The next 4 bytes gives the size of the BMP file (in bytes). Next 4 bytes are reserved and its value is dependent of the application creating the image file. And the last 4 bytes gives the starting address of the bitmap data. Thus we see that file header of a bitmap file is of 14 bytes.

b) **Bitmap information header:** This part contains elaborate information about the image and application is supposed to get this information from information header itself. The first 4 bytes tells about the size of the header which is 40 bytes. Next 4 bytes gives information about width in pixels. Next 4 bytes tells about height in pixels. The successive 2 bytes gives the information about how many colour planes are there. Next 2 bytes tells us that how many bits are there per pixel that is colour depth. Then upcoming 4 bytes gives information about compression method used. Next 4 bytes gives the size of real bitmap data. Next 2 nibbles gives horizontal and vertical resolution. Last 2 nibbles tells about number of colours in colour palette and important colours respectively.

c) **Colour table:** After information header, next in BMP file comes the colour table or colour palette. It is basically a collection of bytes posting the hues used by the image. Every pixel in a recorded colour image is portrayed by various bits (1, 4, or 8)

which is an index of a solitary colour depicted by this table. The motivation behind the colour palette in ordered shading bitmaps is to advise the application about the real shading that each of these record qualities corresponds to.

d) **Pixel Storage:** In bitmap all the bits are packed in rows. Each rows' size has to be rounded to multiple of 4 bytes which is done by padding.

e) Pixel Array: It is a 2 dimensional array of bits which describes the entire image pixel by pixel. This is the part of the BMP file which contains the actual bitmap image data. Usually the pixels are stored from left to right in up to down fashion.

By studying the basic structure of a BMP image one can understand the importance of headers and the constituency of a pixel one can figure out the logic to be used in fetching an image from SD card and then figuring out how to display raw bitmap on screens with different resolution. For example if one wishes to display a bitmap on a screen with 16 bit data handling capability, one has to discard some bits from each colour in each pixel.

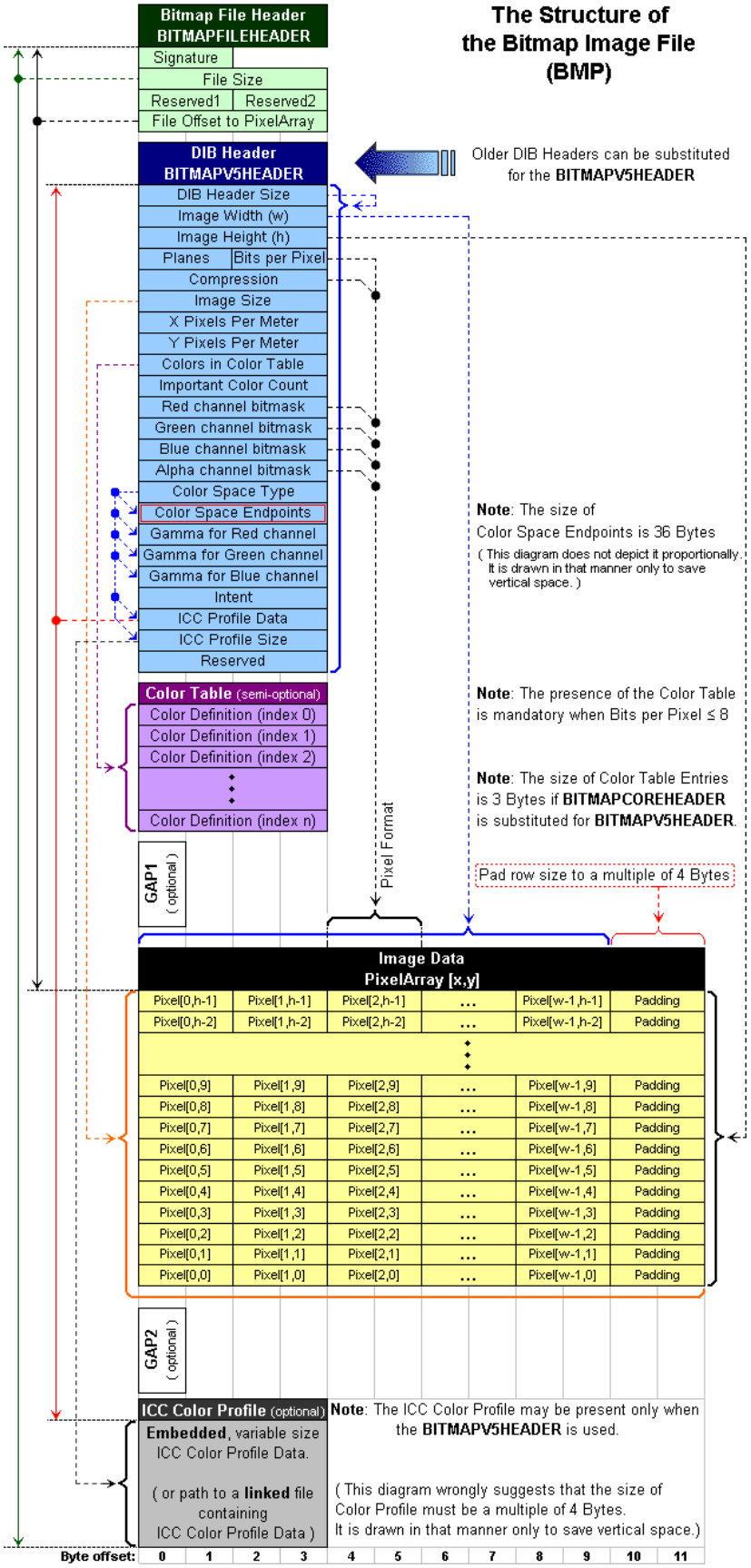

Figure 6 Structure of BMP image [5]

3.3 Usage of BMP image

The effortlessness of the BMP file format, and its far reaching recognition in Windows and , and also the way that this configuration is well documented and free of licenses, makes it an exceptionally normal arrangement that image processing applications from numerous working frameworks can read and compose. Many of the old graphical user interfaces were primarily designed with bitmaps in their graphical subsystems for example in Microsoft Windows' Graphics Device Interface (GDI) subsystems where particularly bitmap file format has been used with .bmp as file extension.

With most of the BMP files are comparatively big in size owing to lack of employment of any compression scheme, many times their size can be reduced by employing lossless compression mechanisms such as ZIP as they consist of redundant data.

CHAPTER 4

COMMUNICATION PROTOCOLS

- Inter Integrated Circuit (I2C) Protocol
- Serial Peripheral Interface (SPI)

4. COMMUNICATION PROTOCOLS

4.1 Inter Integrated Circuit (I2C) Protocol

Inter Integrated circuits or more popularly known as I2C is a serial communication Protocol used for connecting peripheral ICs to microprocessors where speed is not a parameter of consideration. This mode of connecting peripherals was first designed by Philips in 1982 with the objective of connecting a CPU with peripheral chips on a TV set.

I2C being a multi master serial bus can be used to connect many devices. It uses two lines. These two lines are called Serial Data (SDA) and Serial Clock (SCL) [8]. Several number of slaves and master devices can be connected to these two lines and easily communicate among themselves. The following figure illustrates how two devices can be connected to I2C bus.

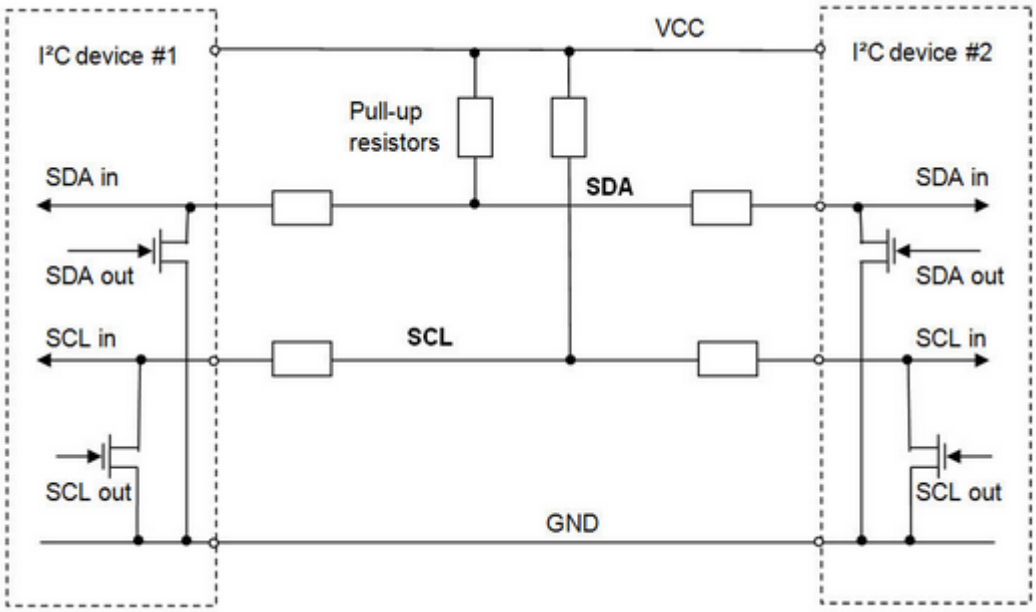

Figure 7 Schematic connection of two devices to I2C bus [9]

I2C protocol is characterised by:

- Slave address of 7-bits where each device connected to the bus is assigned a unique address.
- 8 bit data bytes.
- Control bits for identification of start and end of data transfer, determining the direction of data transfer, and acknowledgement system.

According to this protocol the IC initiating data transfer acts as bus master while all other ICs connected to bus act as bus slaves. Firstly the master IC issues a "START" condition which acts as an "ATTENTON" for all other devices connected to the bus thus making them ready for receiving data on the bus [10]. After this the master device puts the address of the slave device it wants to communicate with, on the bus along with specifying whether it is a Read or Write operation. Slave devices compares this address with their own unique address. If it matches then the corresponding slave device gives an acknowledgement response. Other devices whose address does not matches with the address issued by master wait in idle state until they get stop signal on bus. When master sense this acknowledgement signal it starts sending or receiving data.

A typical I2C transfer can be explained by the condition of the bus while reading / writing two bytes of data.

Table 2 Condition of I2C bus while read/write operation

START	Slave Address	Read/Write	ACK	DATA	ACK	DATA	ACK	STOP

1 bit	7 bits	1 bit	1 bit	8 bits	1 bit	8 bits	1 bit	1 bit

The condition of the bus while writing 2 bytes of data to the slave device is shown below where shaded part are the data which is put on the bus by the master device.

Table 3 Bus condition while writing 2 bytes to slave

START	Slave Address	0	0	DATA	0	DATA	0	STOP
1 bit	7 bits	1 bit	1 bit	8 bits	1 bit	8 bits	1 bit	1 bit

The condition of the bus while reading 2 bytes of data from slave is also shown below where shaded part are the data which is put on the bus by the master device.

Table 4 Bus condition while reading 2 bytes from slave

START	Slave Address	1	0	DATA	0	DATA	1	STOP
1 bit	7 bits	1 bit	1 bit	8 bits	1 bit	8 bits	1 bit	1 bit

The conditions of "START" and "STOP" closely depends on the physical structure of bus.

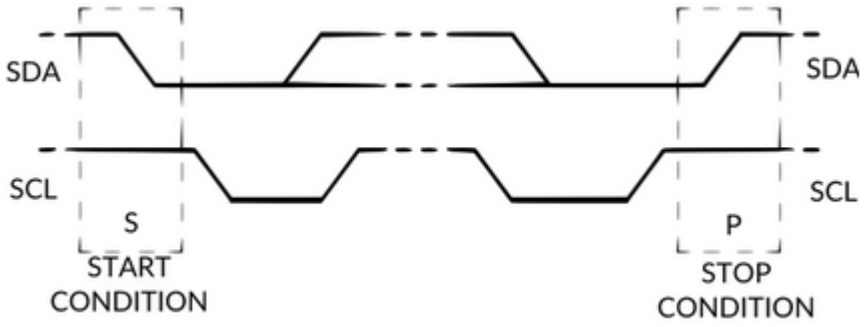

Figure 8 Start and stop condition in I2C bus

According to the specifications of I2C protocol, data on SDA line can alter only when the clock line, SCL, is at low level. Also the data is stable on SDA only when SCL is high.

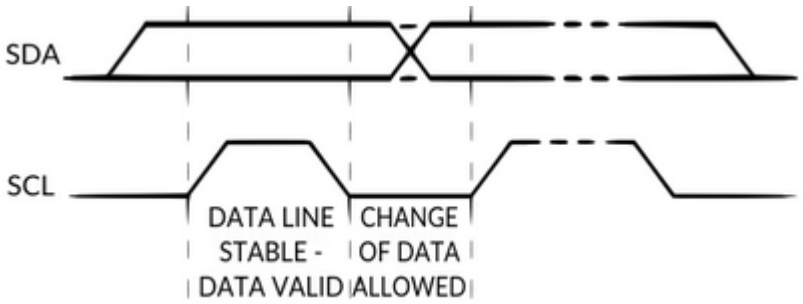

Figure 9 Data stability condition

4.2 Serial Peripheral Interface (SPI)

Serial Peripheral Interface (SPI) is a very simple protocol developed first by Motorola. It is a protocol which any digital electronics engineer would think of for quickly designing a way of communication for two devices.

This protocol has four lines:

- A clock line denoted as SCLK which is issued by the master device to all the slave devices and all the other signals used in this mechanism are synchronised with the clock signal
- Another signal is there for selecting slaves known as Slave Select (SS) used to allow the master device to select slave device it wants to communicate with.
- Third signal is for sending data from master to slave known as Master out Slave in (MOSI).
- Fourth signal is for sending data from slave to master known as Master in Slave out (MISO).

The following figure shows how master device is connected to one slave using SPI communication protocol. [10]

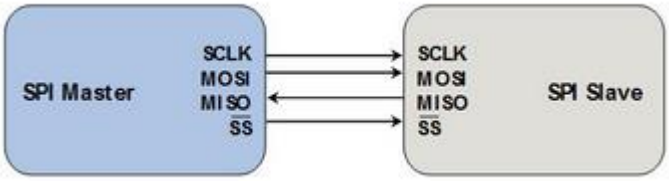

Figure 10 single master single slave SPI connection

If there are more than one slave devices then following topology is adopted for connection to master device.

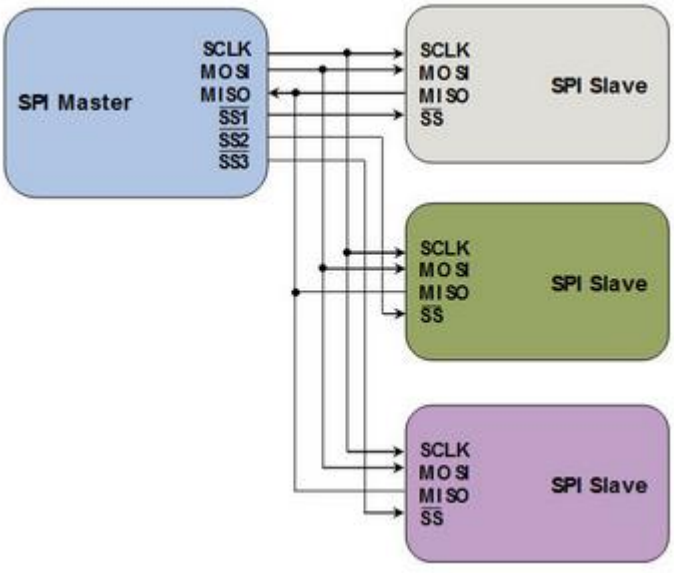

Figure 11 Single master multiple slaves SPI connection

In this methodology there will be only one device which acts as a master device whose responsibility is to initiate the communication with the slave devices. When master wants to

communicate with some slave device, it has to select the slave device by making the corresponding slave select line (SS) low and activating the clock signal at a frequency which is same for master and slave. Master then puts data onto MOSI line while sampling MISO. The following figure demonstrates a sample SPI communication

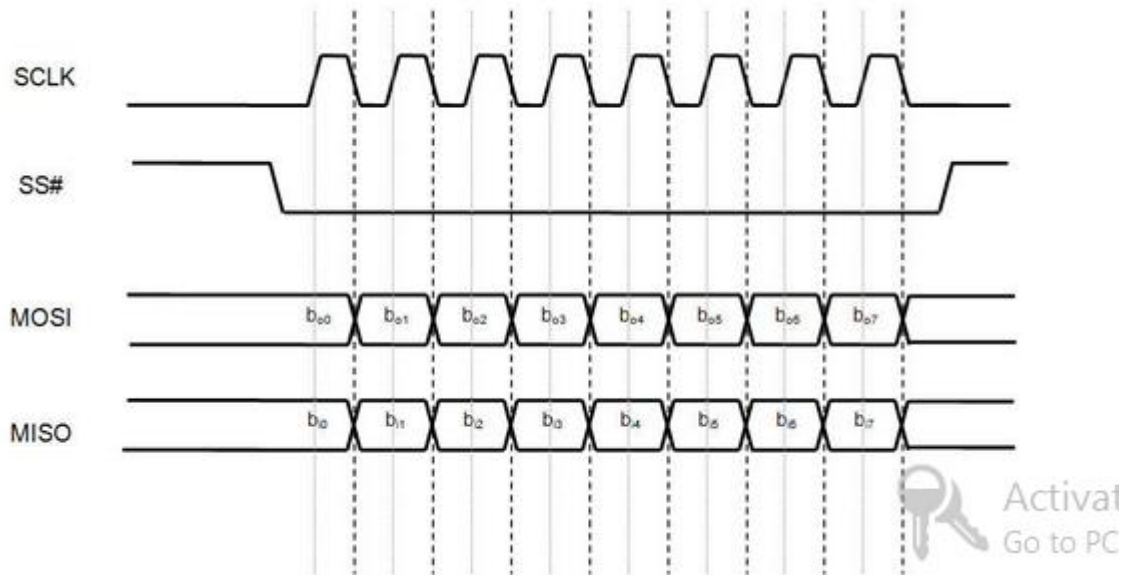

Figure 12 Condition of SPI signals during communication

There are four modes possible with SPI which normally define the clock (SCLK) edge over which MOSI line is supposed to toggle, clock (SCLK) edge over which master is supposed to sample MISO line, clock line's steady line. Each mode is characterised by two parameters namely Clock Polarity (CPOL) and Clock Phase (CPHA). The following figure describe each of these modes

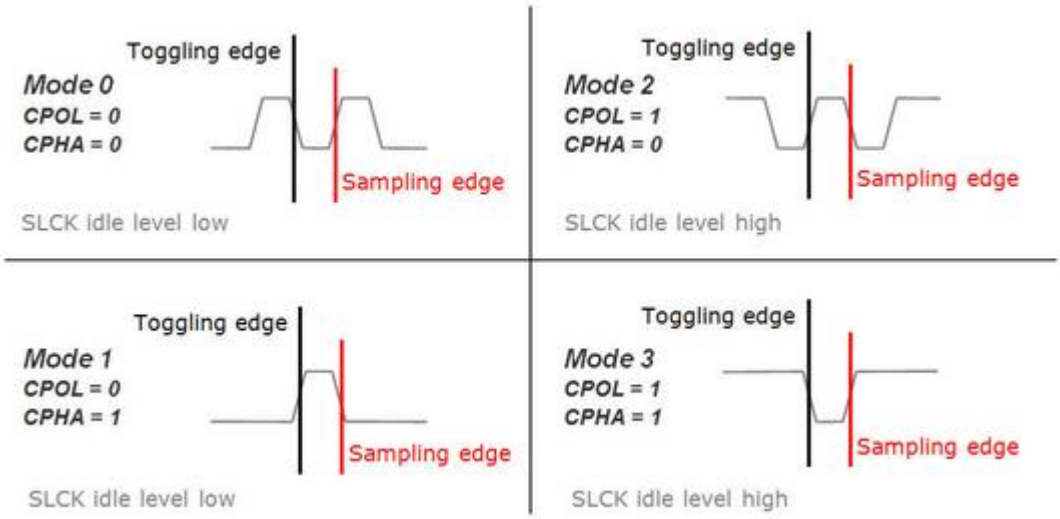

Figure 13 Four modes of SPI

For a communication to be feasible master/slave pair is supposed to have same set of specifications characterised by clock frequency, CPOL and CPHA. If there are more than one slaves then for each communication with different slave master is supposed to reconfigure itself.

CHAPTER 5

FAT 32 FILE SYSTEM AND SD CARD

- FAT 32 File System
- SD Card Interfacing

5 FAT 32 FILE SYSTEM AND SD CARD INTERFACING

5.1 FAT 32 File System

File Allocation Table (FAT) is a file system developed by Microsoft for MS-DOS and was the primary file system for consumer versions of Microsoft Windows up to and including Windows Me. FAT file system is relatively easy and is virtually supported by all major operating systems. This makes it an ideal file system for organization of data in hard disks, floppy disks, and memory cards.

FAT comes in three variations namely FAT 12, FAT16 and FAT 32. The numbers here corresponds to the number of bits that are being used in defining a cluster. Here in this project the SD card is using FAT 32 file system for storing the data [11].

The very first sector of the drive is known as Master Boot Record (MBR) with its first 446 bytes being used for booting the system. Then follows a partition table of length 64 bytes with last 2 bytes always being 0x55 and 0xAA which are usually used for checking the integrity of MBR.

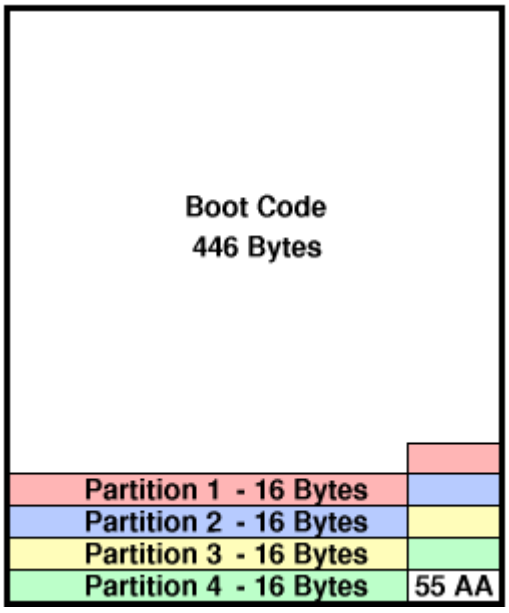

Figure 14 Master Boot Record [11]

Every partition is of 16 bytes length but most of the bytes can be ignored. Fifth byte is known as “Type Code” which gives information about the type of file system that is supposed to be contained in each partition. Ninth through twelfth bytes is “LBA begin” which is the address where that partition begins on the disk.

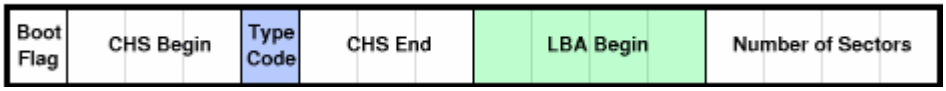

Figure 15 16 byte Partition entry

Generally only “type code” is looked upon for 0x0B or 0x0C and then “LBA begin” is read to know the location of FAT 32 File System on the disk.

While reading FAT 32 File System, first sector has to be read known as Volume ID which is supposed to be read by using “LBA begin” which is to be found from partition table. This sector gives detailed information about the layout of the FAT 32 File system. But many of these information are not of interest here. Only four variables are desired and another three for ensuring their authenticity.

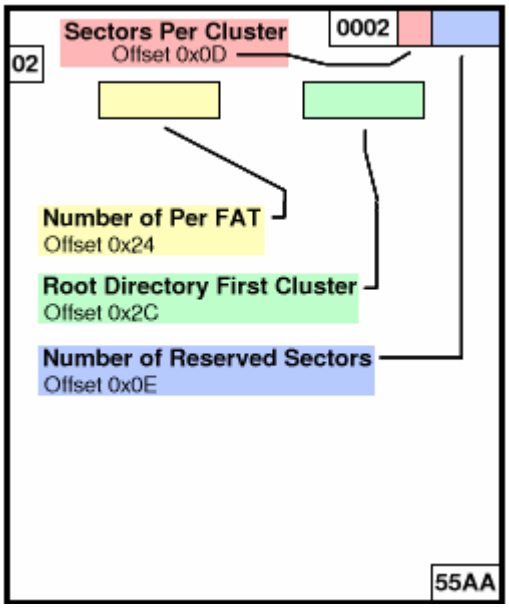

Figure 16 Critical fields of FAT 32 Volume ID

Following table dig into these seven variables and their significance while using FAT 32 File System.

Table 5 Main Variables of Volume ID

FIELD	MICROSOFT'S NAME	OFFSET	SIZE	VALUE
Bytes Per Sector	BPB_BytsPerSec	0x0B	16 bits	Always 512 Bytes
Sectors Per Cluster	BPB_SecPerClus	0x0D	8 bits	1,2,4,8,16,32,64,128
Number Of Reserved Sectors	BPB_RsvdSecCnt	0x0E	16 bits	Usually 0x20
Number Of FATs	BPB_NumFATs	0x10	8 bits	Always 2
Sectors Per FAT	BPB_FATSz32	0x24	32 bits	Depends on size of the disk
Root Directory First Cluster	BPB_RootClus	0x2C	32 bits	Usually 0x00000002
Signature	None	0x1FE	16 bits	Always 0xAA55

Arrangement of FAT 32 File System is very simple. Volume ID is always the first sector followed by some empty space known as Reserved Sectors. After this reserved sector there exists two set of File Allocation Tables (FATs). The remaining file system is data organised in clusters with a very small unused space after last cluster. Majority of space is being used by

clusters used to store files. Cluster numbering is that first cluster is named cluster 2, where there is no cluster 0 or cluster 1.

5.2 SD Card Interfacing

Secure Digital (SD) cards are the solid state memory devices popularly used with small portable devices such as digital camera, mobile phones, tablets etc. Their small size, low power requirement and non-volatile nature has enhanced their utilities in day to day consumer electronics.

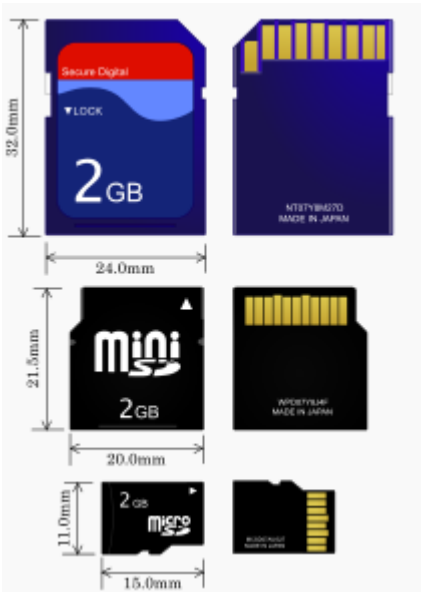

Figure 17 Different types of SD cards

Now as far as interfacing of SD card with STM32F4 discovery board is concerned it is a convenient task using Chan's [12] FATFS library and designing tools provided by Keil uvision

MDK pro. The communication of SD card with STM32 microcontroller is configured to take place using SPI communication protocol.

The Pin out configuration of SD card is shown below

Table 6 Pin connection of SD card with STM32 microcontroller using SPI

Pin number	Name	STM32F4	Description
1	CS	PB5	Chip select for SPI
2	MOSI	PA7	Data input for SPI
3	VSS1	GND	GND
4	VDD	3.3V	3.3V
5	SCK	PA5	Clock signal for SPI
6	VSS2	GND	GND
7	MISO	PA6	Data output for SPI

Card Detect and Card Protect are a part of SD card connector but not of SD card itself. These are listed below

Table 7 Card Detect and Card Protect

Name	Pin	Description
WP	PB7	Low when write enabled and act as write protect pin.
CD	PB6	Low when card is detected

Communication modes whether be STDIO or SPI can be chosen in define.h file [13].

CHAPTER 6

CONCLUSION AND FUTURE PROSPECTS

6 CONCLUSION AND FUTURE WORK

6.1 Conclusion

A table top digital photo frame has thus been designed using a 32 bit microcontroller board STM32F4 discovery which a microcontroller based on ARM cortex M4 architecture. The bitmap data has been initially stored in micro SD card using chan's FAT 32 file system and microcontroller fetches image from SD card using SPI communication protocol. Microcontroller then decodes the header file and process the bitmap data according to instructions given to it in source code. The final stage involves the display of image data onto the Graphic LCD (GLCD). Touch navigation has also been implemented for the device to be able to swiftly navigate through images. Touch controller uses I2C communication protocol for communication with microcontroller.

6.2 Future Work

This work can act as a reference for the embedded engineers to look upon while working with ARM cortex M4 and SD card interfacing. Project can be further extended for integrating more functions like a fully functional Graphical User Interface (GUI) can be developed for making the device more user friendly. With the use of a Real Time Operating System (RTOS) one can even develop a tablet using this configuration as a primer.

CHAPTER 7

REFERENCES

7 REFERENCES

[1] Sivarama P Dandamudi, *Guide to RISC Processors for programmers and engineers,* Spinger International Series in Engineering and Computer Science,2005

[2] Joseph Yiu , *Definitive guide to ARM cortex M3 and ARM cortex M4 Processors,* Newnes Publishers,2013

[3] www.keil.com/product/brochures/uv4.pdf

[4] www.graphic-design-employment.com/just-what-is-a-bitmap.html

[5] en.wikipedia.org/wiki/BMP_file_format

[6] topherlee.com/software/pcm-tut-waveformat.htm

[7] www.itk.ilstu.edu/faculty/javila/datatypes/soundwave.htm

[8] Miroslav Popovic, *Communication Protocol Engineering,* CRC Press,2009

[9] www.byteparadigm.com/application/introduction-to-i2c-and-spi-protocols

[10] Richard Lai, Ajin jirachiefpattana, *Communication Protocol Specification and Verification,* Spinger US, 1998

[11] www.pjrc.com/tech/8051/ide/fat32.html

[12] elm-chan.org/fsw/ff/00index_e.html

[13] www.stm32f4-discovery.com/2014/library-21-read-sd-card-fatfs-stm32f4xx-devices.html

[14] www.st.com/stwebu/static/active/en/resource/technical/document/active/technical.pdf

[15] http://www.inversereality.org/files/dmaprogramming.pdf

[16] http://educypedia.karadimov.info/electronics/I2C.htm