PROJECT REPORT

Entitled

"E-Writer using ARM Microcontroller"

Submitted to the Department of Electronics Engineering
In Partial Fulfillment of the Requirement for the Degree of

Bachelor of Technology
(ELECTRONICS & COMMUNICATION)

: Presented & Submitted By :

Mr. DEPANI BHUMIL RAJNIKANTBHAI
(Roll No. U16EC076)
B. TECH. IV (EC), 8th Semester

: Guided By :

Dr. A. D. DARJI
Associate Professor & Head, ECED.
Dr. J. N. Sarvaiya
Associate Professor, ECED.

(Year : 2019-20)

DEPARTMENT OF ELECTRONICS ENGINEERING
Sardar Vallabhbhai National Institute of Technology
Surat-395007, Gujarat, INDIA.

Sardar Vallabhbhai National Institute of Technology

Surat-395 007, Gujarat, INDIA.

ELECTRONICS ENGINEERING DEPARTMENT

CERTIFICATE

This is to certify that the **PROJECT PRELIMINARY REPORT** entitled **"E-Writer using ARM Microcontoller"** is presented & submitted by Candidate **Mr. DEPANI BHUMIL RAJNIKANTBHAI**, bearing **Roll No.U16EC076**, **of B.Tech. IV, 8th Semester** in the partial fulfillment of the requirement for the award of **B.Tech.** degree in **Electronics & Communication Engineering** for academic year 2019-20.

He has successfully and satisfactorily completed his **Project Exam** in all respect. We, certify that the work is comprehensive, complete and fit for evaluation.

Dr. A. D. DARJI
Associate Professor
Project Guide

Dr. J. N. Sarvaiya
Associate Professor
Project CoGuide

SEMINAR EXAMINERS:

Name of Examiner Signature with date

1. Dr. A. D. Darji _____
2. Prof. P. K. Shah _____
3. Dr. Deepak Joshi _____

Dr. A. D. Darji DEPARTMENT SEAL
Professor / Associate Professor &**(October-2019)**
Head, ECED, SVNIT.

ACKNOWLEDGEMENT

The successfulness of this project required a lot of hard work and assistance from many people and we sincerely expressing our gratitude to all of them to be with us throughout the project.

We express our sincere gratitude to our guide Dr. A. D. Darji, without whom the project would not have been successfully completed. His theoretical and technical knowledge proved very useful to me in every stage. We are also thankful to our CoGuide Dr. J. N. Sarvaiya, who guided us in the best way and motivated in each and every aspect. His guidance paved the way for the successful completion of this project. We heartily thank both of them for their exemplary counseling, constant encouragement, and careful monitoring throughout the project. We are also giving our heartily gratitude to Student Startup and Innovation Policy (SSIP), for their grant under Association for Harnessing Innovation and Entrepreneurship (ASHINE).

We would also like to thank Dr. A. D. Darji for his appreciating and encouraging nature towards students as the Head of the Department, ECED. We would also like to acknowledge all the Electronics Department faculties for boosting the students' confidence levels and enlightening them with practical real world situations with their academic guidance and organization of project preliminary.

It's a great pleasure for us to present our project on "E-Writer using ARM Microcontroller" which is useful for serving futuristic applications. We mark our gratitude for everyone who helped and encouraged us with their efforts.

Bhumil Depani

(Roll No.: U16EC076)

ABSTRACT

Are you a Student? Are you an Engineer? Are you a professor? Are you a Doctor? Are you a Lawyer? Are you an Accountant? Are you an Artist? Whatever is your field; you are using or would have used notebooks for your works. And taking an example of a college student, they use approximately 15 pages for writing and 30 pages for photocopy, so total 45 pages per day. 8,000 A4-sized papers can be made by sacrificing one green tree. So, one student cut one tree in 177 days and spends around 12000 INR for stationary.

So what is the solution? Here, we come up with "Cost Effective" solution which helps to achieve all the objectives. To resolve all these problems we have an eco-friendly solution called "FLASHBOOK"-an E-writer. A professional person or a student who carry his/her calculator, compass box, number of books in his/her heavy bags, now he/she only needs to carry approx. 450-gram device. College Students would find this more beneficial because they don't have to buy the notebooks and they need not run for photocopying every day. FLASHBOOK will also be beneficial to Professors and writers, who need to repeatedly insert pages in their write-up work, and without using physical file or diary they can easily organize their thesis and scripts.

An electrophoretic display along with stylus pen is the best suitable in between palmtop device and traditional Notebook. The main task is to design a system which provides all the feature of notebook and extra superior application to students.

To make the prototype of the same we are going to use microcontroller by STMicroelectronics. We are going to use discovery board from STM, which contains STM32F407VG- Cortex M4 microcontroller on it. STM32F407VGT6 microcontroller featuring 32-bit ARM® Cortex® -M4 with FPU core, 1-Mbyte Flash memory, 192-Kbyte RAM in an LQFP100 package. It has on-chip RC oscillator, want power supply of 1.8V to 3.6V and upto 168 MHz clock speed.

In the prototype, we are using a simple LCD from Nextion. The LCD model is Enhanced NX8048K070. This LCD uses UART (Universal Asynchronous Receiver Transmitter) protocol

to communicate with the controller. It has visual Area of 15.4 * 8.6 cm and resolution of 800 * 480 pixels. It need supply voltage of 5 V and current up to 2 A. It has user RAM of 3584 Bytes and SPI Flash of 16 MB.

We can write firmware in both C and C++. But to make code reusable and well structured, Object Oriented Programming (OOP) is a better approach. This is only compatible with C++ language, so we have used C++ for the designing of our firmware.

Now comes to the software side, STM32Cube MX is a simple GUI used to make the framework of the code, to configure the GPIO pins to as an input or as an output. It is just used to make the outer structure of the main code. Microcontroller Development Kit for ARM (MDK-ARM) by Keil™ is used to make the firmware for the STM32F407 microcontroller.

Till now we have understood all the utility and interface of STM32 discovery board. Also, have done the setup for writing the code in C or C++. We have also gained knowledge about LCD and its interfacing with microcontroller. We are now ready to write code and dumb, burn, it in the controller. Future work remains is first to interface LCD with controller and to obtain that whatever thing we write on-screen using stylus, it should be sensed by the display, send the command to the controller and sustain that written data on the display. Then after our move would be to choose a perfect electrophoretic display for our prototype and interface it with a controller and obtain the required objective.

INDEX

List of Figures

CHAPTER 1

INTRODUCTION

1.1 Problem Statement

Being a human, we are using or would have used notebooks for your works. In the current scenario, students have to manage more than one subject and professional people have to manage more than one project. For that, they have to carry heavy bags full of notebooks and stationeries. And to manage all these things is cumbersome for them.

Generally, a college student use approximately 15 pages for writing and 30 pages for photocopy, so a total of 45 pages per day [1]. 8,000 A4-sized papers can be made by sacrificing one green tree. So one student cut one tree in 177 days. Talking about engineering college, they have four years of bachelor's course, so totally 1,460 days. So by doing the calculations, each person who has done any bachelor engineering course, he or she might have cut 8.25 trees and needed 5,000 litres (if reusable water) or 98,550 litres (if non-reusable water) of water in their four years duration. This is not limited to his or her engineering span; he or she would continue this till the last day of his or her professional life. And also students spend approx 1000 INR on photocopy, notebook and other stationery. So, in the span of four years, the total reaches 12000 INR. So we come up with "**Cost Effective**" solution which helps to achieve all the objectives.

To resolve these problems we have an eco-friendly solution called "**FLASHBOOK**". An electrophoretic display along with stylus pen is the best suitable in between palmtop device and traditional Notebook. The main task is to design a system which provides all the feature of the notebook and extra superior application to the user.

1.2 Innovation/Novelty

The system will provide all the attributes of the conventional notebook along with extra features such as share, copy, erase, inbuilt geometric figures and a lot more. Moreover, the size of E-writer will remain the same as a conventional notebook.

An electrophoretic display is the best suitable device between a smartphone display and a traditional notebook. It will be as effective as writing on a conventional notebook and as efficient as using a Smartphone.

The system provides a battery life of more than a week. The reason for this is that the electrophoretic display provides the facility that system requires power only when we change the page or mode of operation in the Flashbook. Also, it comes with the scratch-resistant display.

One of the major strengths of E-writer is its reusability. Unlike conventional paper which becomes unavailable after writing on it, our device has theoretically an infinite number of pages. A professional person or a student who carry his/her calculator, compass box, number of books in his/her heavy bags, now he/she only needs to carry approx. 450-gram device.

1.3 Rationale for taking up the Project

Every day thousands of trees are being cut down and tones of water are being used for making the notebooks. In the present scenario, Global Warming is increasing due to deforestation and its after-effects are being experienced by us in the form of scanty rainfall and increasing temperature.

Apart from this situation, a student has to maintain different books for various subjects and sometimes those books do not get fully filled and the pages are wasted. Similarly, what if he/she lost his/her book? They either have to write down the whole thing back again or has to photocopy his/her classmate's notes.

These situations are not only limited to the student's life. They can arise even in the daily lives of Professionals.

1.4 Introduction to E-writer

Let's come to our E-writer. Firstly we are going to make E-writer using ARM Cortex M-series microcontroller STM32F407VG and LCD.

Our first aim is to make a system in which, the thing which we write on the screen should be displayed and to remain static on the display as long as power is ON.

Then after we upgrade our firmware and interface storage device to store the written pages on it. We would also try to replace LCD with the electrophoretic display.

1.5 Contents in the Project

Our E-writer's size would be the same as an A4-sized paper, so the person does not need to miss the feel which you got from writing on the paper. But in our prototype, we are using the small size of a display for the sole purpose of present Proof of Concept (POC).

For our prototype, we have used the Liquid Crystal Display (LCD). This display is used for interaction between E-writer and user. A user can write on the display using a stylus, like this way a user can give input into the E-writer. For output also E-writer uses the display, so that users can get information stored in the E-writer visually.

1.6 Objective of the Project

Every day thousands of trees are being cut down and tones of water are being used for making the notebooks. In the present scenario, Global Warming is increasing due to deforestation and its after-effects are being experienced by us in the form of scanty rainfall and increasing temperature.

Apart from this situation, a student has to maintain different books for various subjects and sometimes those books do not get fully filled and the pages are wasted. Similarly, what if he/she lost his/her book? They either have to write down the whole thing back again or have to photocopy his/her classmate's notes.

One of the most important objectives is to make a system without android base. Because now the android framework is free, but who knows how long it will be free?

1.7 Outline of the Project

In this project, we are first going to do a product survey. In this section, we are going to compare the existing product available in the current market and the pros and cons of these products.

In the next topic, we will discuss the hardware and software components of the system. We will also see the protocols used by these components. Then after in this chapter we will see the interfacing between different hardware components and the software-hardware integration.

Then after in "Result and Analysis" chapter we will argue on the result of the project and would derive some conclusion from the results obtained. And at the last in the "Conclusion" chapter we will give last comments about the project and will compare our prototype with the existing products.

CHAPTER 2

PRODUCT SURVEY

2.1 Current Scenario

Work in the reduction of paper usage and to move towards Electronic Writer is begin by Amazon by introducing Kindle back in 2007. It was just an Electronic Reader, used to download, manage and read the books. It uses an electrophoretic display for finer and eyesight-friendly reading experience.

Fig. 2.1(a) E-Reader using Electrophoretic Display, [2]

Then after people have been trying to make this display or device capable to sense the touch of a stylus (not finger) and put in written things on the screen as it is. One of that E-writer is as below.

Fig. 2.1(b) Memory less E-Writer, [3]

It has a blackish display and when we write using stylus it is written on the display in white colour. It is powered by a small lithium 3.3 V battery. One of the main advantages of this screen is that it does not consume power when it is in the steady-state, steady-state is that in which no reading or writing operation is going on. With this it has many limitations, like, it does not have memory, so it can't be used for the permanent data storage, it can only be used for rough work.

With these limitations, one more limitation is, it is colourless. This E-writer's display only has two colours, Black and White. So it is not much attractive to the users.

2.1.1 Commercially Available E-writer

The paper which we are using is first, which made out of plant-like fibres, was invented by the Chinese Cai Lun, who in 105 AD mixed textile fibers from the bark of the mulberry in water and produced sheets of paper from that. From here paper usage has been increasing rapidly. Approximately 400 million metric tons of paper is produced and consumed globally each year. Current usage demands are just over 2 pieces of paper per hour per person, for each person on Earth. Most of the papers are used for reading and writing purpose. In advancement in the electronics industry first of all electronic reader has arrived using electronics display.

There are many E-writers are available in the market, out of them we are going to discuss two E-writers. One is Ruffpad 10+ Portable Ruff Pad E-Writer, 8.5 inch LCD with 4 Magnet (Black) from Portronics, and second is paper Tablet by Remarkable.

Ruffpad 10+ has Pressure-sensitive screen that facilitates to create thick and thin lines [4]. Tablet displays notes until erased with the touch of a button; One-touch button erases notes instantly. It is the size of 8.5 inch, which is very smaller then A4 sized paper. Compare to this Paper Table from Remarkable is of size 10.1 inch, near to the size of A4 paper. Paper Tablet has a resolution of 1872x1404 pixels (226 DPI). It is partially powered by E-ink Carta technology and has an electrophoretic display (EPD) [5]. To give paper-like feel display has surfaced with Paper-like friction. It has connectivity with other devices by a Wi-Fi module. It has an internal storage capacity of 8 GB which is equivalent of 100,000 pages and RAM of 512 MB. It uses ARM A9 CPU with 1 GHz clock speed.

Now come to the limitations. Ruffpad 10+ comes without memoryless, so it can only be used for rough work, it can't store data permanently. Paper Tablet is very costly, up to 41,000 INR. And it consists of Codex, a custom Linux-based OS which increases overhead and burdens to the processor. So, our main aim is to design E-Writer which has advantages of both given models with minimum limitations.

CHAPTER 3

HARDWARE AND FIRMWARE SUBSYSTEMS

3.1 Different Hardware Required For E-writer

Now after gaining much information from the above 'Introduction' chapter, we can imagine what kind of components we do need for E-writer.

In the layman term, we can say that we need a touch screen display, stylus for writing and one thing to process and store those written things and process them or store them.

But in the designer point of view we need ROM to store our firmware, RAM for the functioning of microcontroller (for scratchpad), a cache is optional to decrease performance bottleneck, a processor for arithmetic and logic unit, stylus to take input from a user, capacitive or resistor type of touch-sensitive screen as an input device, driver circuit to drive the display, the output can be seen on display. To store the written data on the display we have two options. First, we can use external memory to store the user content and second is to use wi-fi (wireless fidelity) or Bluetooth module to store the data on the server.

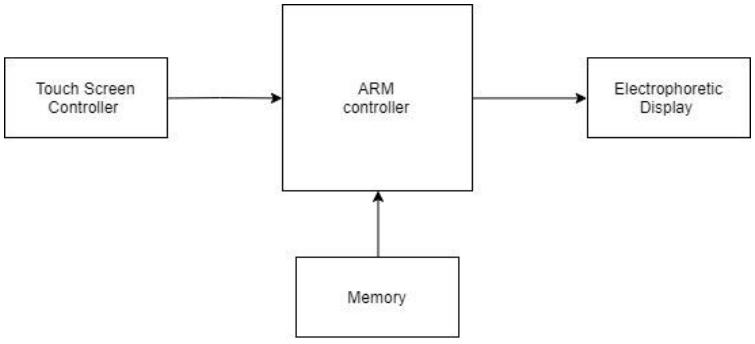


Fig. 3.1(a) Typical embedded view of E-writer

Here we are going to study selection criteria, specification required and what are the provisions there in the selected module for the memory, processor and display.

3.2 Processing element

The main part of any embedded system is the processor, so the selection of right and application-specific processing unit is very important.

3.2.1 Selection of Processing Unit

One of the most affecting parameters is the processor unit is speed. In our E-Writer, we do not need very high computation power. So controller with decent speed (in hundreds of MHz) is acceptable. Our device is a handheld device, so the power consumption is very important. So we have to use the controller with minimum power consumption. Our controller (processing unit) should also have safety and fault tolerance capability.

As applications become more complex, so does the development process. Consider the system-level debug of a large software-based system that uses an operating system or real-time kernel. Do the processor and its toolchain have a way to examine the processor state without impacting the application? Is it possible to profile and trace where the processor has been, or to trap on all events of interest? All these questions, and many more, should be answered. Before choosing processor we should also see the availability of the development and debugging IDE (Integrated Development Environment). Cost is also one of the most important economic reasons for choosing the controller or processing unit.

So after a lot of research and study, we have decided to use STM32F407VGT6 microcontroller as a processing unit. Discovery Kit STM32F407 kit contains this controller with all pinout on the board.

3.2.2 STM32F407 Discovery Kit Introduction and features

1. An ST-LINK embedded debug tool
2. Eight LEDs:
    a. LD1 (red/green) for USB communication
    b. LD2 (red) for 3.3 V power on

c. Four user LEDs, LD3 (orange), LD4 (green), LD5 (red) and LD6 (blue)

d. 2 USB OTG LEDs LD7 (green) VBUS and LD8 (red) over-current

3. Two push-buttons (user and reset). A reset push-button is connected to the reset pin (NRST) of the microcontroller.

4. STM32F407VGT6 microcontroller featuring 32-bit ARM® Cortex® -M4 with FPU core, 1-Mbyte Flash memory, 192-Kbyte RAM in an LQFP100 package

5. On-board ST-LINK/V2-A on STM32F407G
6. All of the I/O pins of the STM32F407VG are available on pin headers. Two 25 by 2 (50 pins) headers are located on either side of the board along its length, one per side. The pin names are silk-screened next to the pin headers. The pin headers protrude from the top of the board allowing easy attachment of oscilloscope or logic analyzer test probes.
7. Comprehensive free software including a variety of examples, part of STM32CubeF4 package or STSW-STM32068 to use legacy standard libraries.

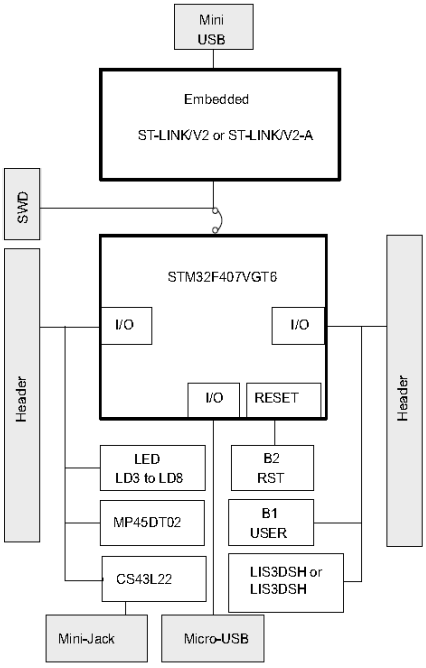


Fig. 3.2.2(a) Internal module and connection of STM32F4 Board, [6]

3.2.3 Internal Structure of the Board

3.2.3.1 Microcontroller

Soldered on the board is an STM32F407VG microcontroller packaged in a 100-pin LQFP (Low-profile Quad Flat Pack). This microcontroller contains an ARM 32-bit Cortex-M4 core with FPU (Floating Point Unit).

Some of the features of the STM32F407VG microcontroller are:

- 1M Byte Flash memory
- 192k Bytes RAM
- On-chip RC oscillator
- Powered by a single supply of 1.8V to 3.6V
- Up to 168MHz operation
- I/O pins multiplexed with many internal peripherals
- USB OTG HS/FS
- Ethernet
- Static memory controller supporting Compact Flash, SRAM, PSRAM, NOR and NAND memories
- LCD parallel interface: This microcontroller is part of the STM32F4 Hi-performance and DSP series of Cortex-M4s from ST.

3.2.3.2 Current Consumption Measurement Unit

This is a newly added feature in the microcontroller board. A jumper link on the board can be removed allowing an ammeter to be connected if current consumption of the microcontroller needs to be measured.

Audio Jack: The STM32F407VG microcontroller uses an audio DAC (CS43L22) (through I^2C protocol) to output sounds through the audio mini-jack connector.

USB OTG: The STM32F407VG microcontroller is used on this board to only drive the USB OTG full speed. The USB micro-AB connector (CN5) allows the user to connect a host or device component, such as a USB key, mouse, and so on.

Motion Sensor: The motion sensor includes a sensing element and an IC interface able to provide the measured acceleration to the external world through the I2C/SPI serial interfaces. The STM32F407VG microcontroller controls this motion sensor through the SPI interface.

3.3 Memory Organization

As we have discussed in the "Different Hardware Requirement for E-Writer" section that we need mainly two types of memory for working of our system:

3.3.1 Read Only Memory

We need ROM to store our firmware, In the STM32F407 Discovery kit, we have STM32F407VGT6 controller soldered. STM32F407VGT6 controller has 1M Byte Flash memory, which can be used to burn code.

3.3.2 Random Access Memory

RAM is used for functioning of the processing unit (for scratchpad), with 1 MB of Flash memory STM32F407VGT6 has inbuilt 192k Byte of RAM.

3.3.3 Miscellaneous

To increase performance and speed of the overall system we can use a processor with cache or off-chip cache. But cache usage in the system increases complexity very much, so cache should be incorporated in the system if and only if it required extremely or we need very fast memory access. In our project at this point, we are not going to incorporate cache to make our system simple.

As we have discussed to store the written data on the display we have two options. First, we can use external memory to store the user content and second is to use wi-fi (wireless fidelity) or Bluetooth module to store the data on a server.

3.4 Display Subsystem

In the prototype, we are using a simple LCD from Nextion. The LCD model is Enhanced NX8048K070. This LCD uses a UART (Universal Asynchronous Receiver Transmitter) protocol to communicate with the controller. It has visual Area of 15.4 * 8.6 cm and resolution of 800 * 480 pixels. It need supply voltage of 5 V and current up to 2 A. It has user RAM of 3584 Bytes and SPI Flash of 16 MB.

Communication Protocol: One serial Port

MCU: Cortex M0

MCU speed: 48 MHz

XH2.54 4P (+5V, TX, RX, GND) TTL serial interface

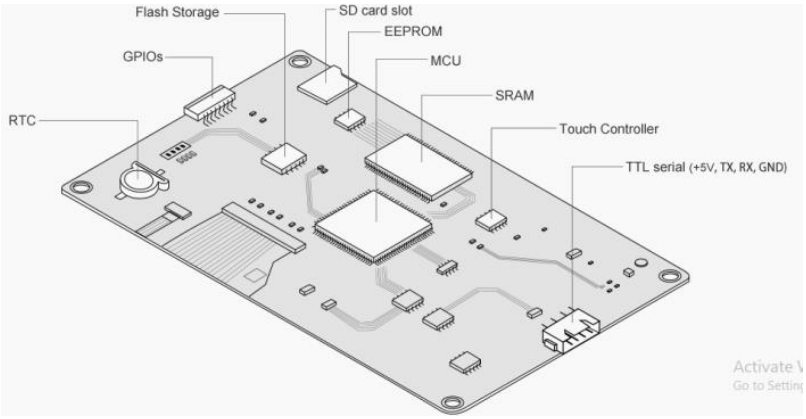


Fig. 3.4(a) Circuit to control LCD, [7]

3.4.1 Baud Rate for LCD

The most important thing in the UART communication to handle is Baud Rate.

The baud rate is the rate at which information is transferred to a communication channel. In the serial port context, "9600 baud" means that the serial port is capable of transferring a maximum of 9600 bits per second. The unit of the baud rate is bits per second.

Popular baud rates in the UART communication is 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200 bits per second.

Fig. 3.4.1(a) An electrophoretic Display

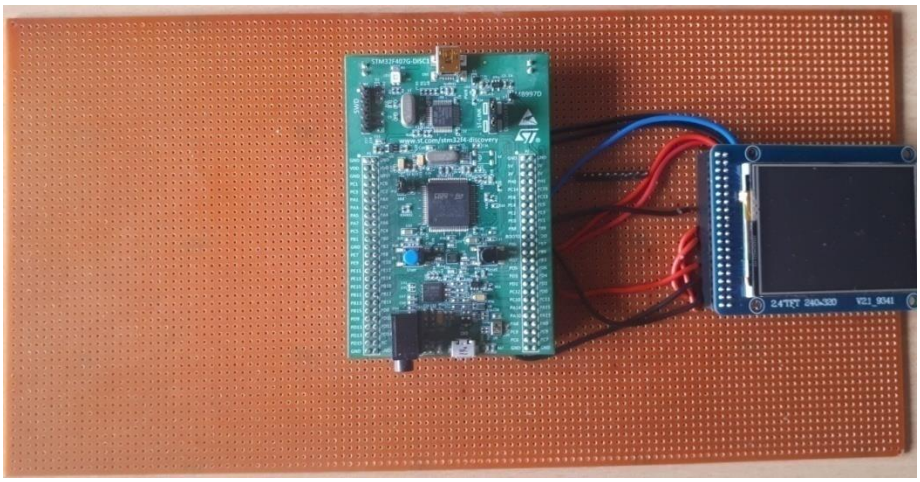


Fig. 3.4.1(b) Interfacing of STM Controller and LCD Module

3.5 Firmware for E-writer

As we have seen in the last topic that we are going to use STM32F407VGT6 microcontroller with 1M Byte flash ROM and 192k Byte RAM, Nextion Enhanced NX8048K070 display to act as an input (touch screen controller) and as an output.

All these hardware modules without firmware (software) are like infant baby which as every part of the body but don't know how to use them. To establish communication between them and to give a command to every peripheral we have to write code, firmware.

3.6 System Requirements

There are some requirements that have to fulfil before start writing code:

Windows® OS (XP, 7, 8 and 10), Linux® 64-bit or macOS™

A Windows platform is required to run one of the development toolchains for STM32 microcontrollers. Toolchains from Atollic, IAR and Keil are available for purchase. These toolchains also have restricted evaluation versions.

Alternatively, the open-source GNU ARM toolchain can be used to program STM32 microcontrollers. One example is YAGARTO for Windows.

STM32Cube MX is a simple GUI used to make the framework of the code, to configure the GPIO pins to as an input or as an output. It is just used to make the outer structure of the main code.

To write code in the C or C++ we need to install one of the Software development toolchains given below:

- IAR Embedded Workbench® for ARM (EWARM) by IAR Systems
- Microcontroller Development Kit for ARM (MDK-ARM) by Keil™
- TrueSTUDIO® by Atollic
- TASKING VX-toolset for ARM Cortex by Altium

Keil® MDK(Microcontroller Development Kit) is the most comprehensive software development solution for Arm®-based microcontrollers and includes all components that you need to create, build, and debug embedded applications. We are going to use Keil MDK-528a version. This is an IDE.

Keil.STM32F4xx_DFP.2.11.0 is a driver-specific for the Keil and STM34F4 series. It is used to connect Keil IDE and our STM32F4 Nucleo board.

ST link is used to connect the programmer chip (STM32F103CB) and main microcontroller STM32F407VG.

3.7 Software configurations and structure

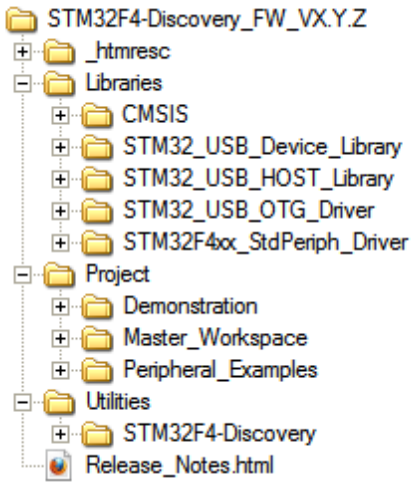


Fig. 3.7(a) Hardware Environment for STM32F407VGT6

1.1. Library: This folder contains the Hardware Abstraction Layer (HAL) for STM32F4xx Devices.

1.2. CMSIS (Cortex Microcontroller System Interface Standard): This subfolder contains the STM32F4xx and Cortex-M4F CMSIS files

2. Project folder

This folder contains the source files of the STM32F4DISCOVERY firmware applications.

3.8 Firmware Coding

Now, we are going to start writing our basic firmware code. Here we have two options,

1. The first option is to use STM32CubeMX GUI software to make the framework of the basic firmware, to configure the GPIO pins to as an input or as an output. It makes our work easy by making an outer structure of the basic firmware by itself. In this case, all the required configuration files automatically added to the project folder.

2. In the second option, we have to make basic firmware from scratch without using STM32CubeMX.

Out of the above two ways we can use any. But to get more insight into the microcontroller working and take more control over machine code generation we are going to use the second option.

Let's start with our installed Keil uVision5, and make a new project.

When we make a new project using the above procedure then in our new project we would have two files already added, they are "startup_stm32f40_41xxx.s" and "main.c".

1. startup_stm32f40_41xxx.s

It Provides the Cortex-M4F startup code and interrupts vectors (vector table) for all STM32F4xx device interrupt handler.

2. main.c

In the below flow we can see the software flow of the E-Writer.

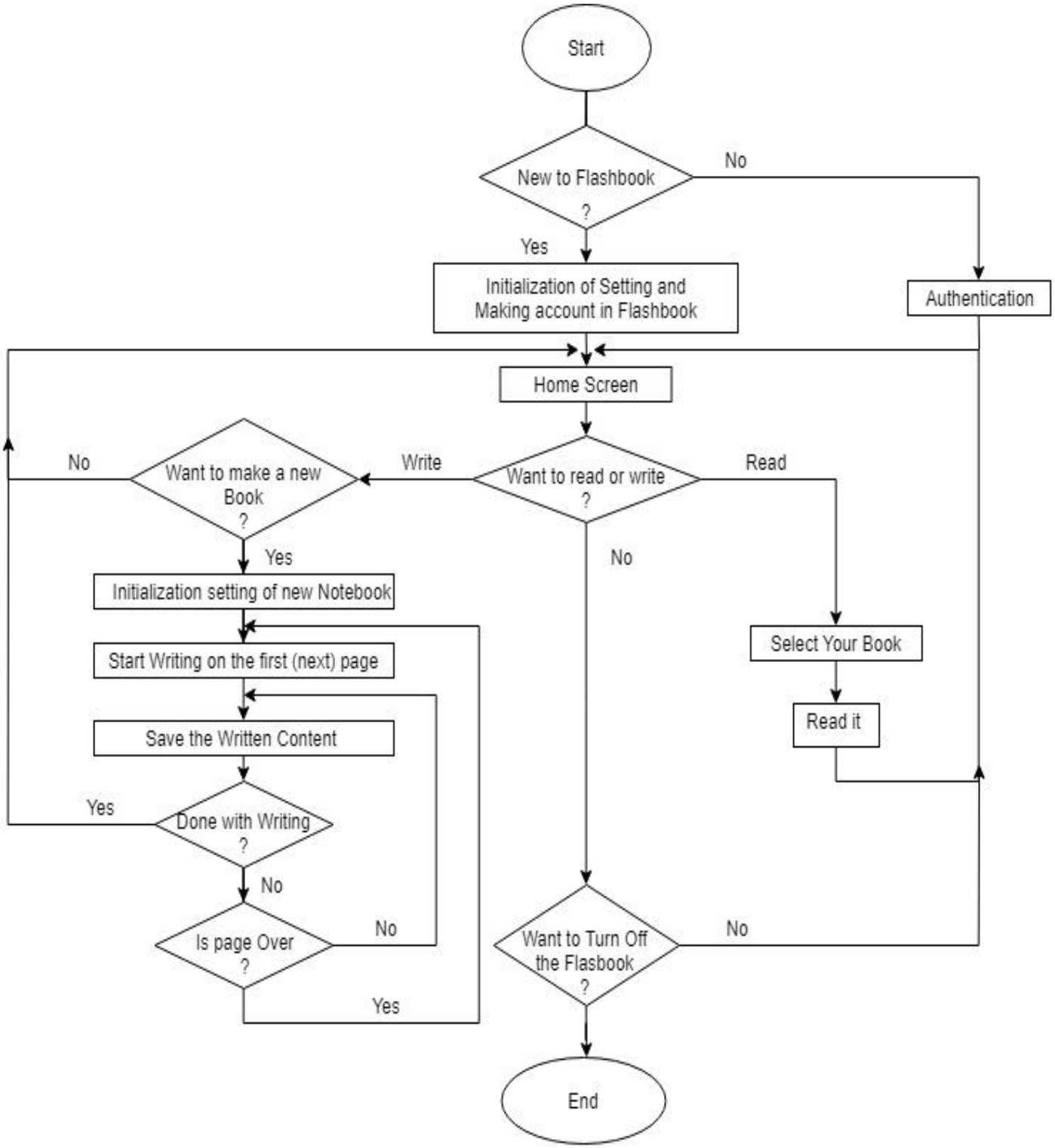


Fig. 3.8(a) Software Flow of the E-Writer

3.9 Hardware-Software Interfacing

Plugging the board into the USB port of a PC will provide power to the embedded ST-LINK and the STM32F407 microcontroller. The board can also be powered from an external 5V supply.

After plugging the board into the USB port of a PC, the pre-programmed firmware will run. This program flashes the four user-programmable LEDs on and off. If the user-programmable push button is pressed, then the LEDs are used to display the status of the accelerometer. Tilting the board in different directions will cause the corresponding LED to light up.

3.9.1 Hardware Requirements to program controller

There are two ways to connect the board to the computer:

1. The only piece of hardware required to start using the board is a USB type A to mini-B cable and a spare USB port on a PC.
2. If the USB OTG port is to be used then a USB type A to micro-B cable will be needed.

3.9.2 Software Requirements to program controller

As we have discussed in 3.1 we are using,

Windows® OS 10, STM32Cube MX to make the outer structure of the main code, ARM (MDK-ARM) by Keil™ software development toolchain,

3.10 Power supply to the system

We have two options to power STM32 discovery board:

1. Through USB bus
2. From an external 5 V supply voltage or 3 V supply voltage; D1 and D2 diodes are used to protect the board from external 5V and 3V power supply. In this case, the

STM32F4DISCOVERY board must be powered by a power supply unit or by auxiliary equipment complying with standard EN-60950-1: 2006+A11/2009 and must be Safety Extra Low Voltage (SELV) with limited power capability.

For power supply to other peripheral boards:

5V and 3V can be used as output power supplies when another application board is connected to pins P1 and P2. In this case, the 5V and 3V pins deliver a 5V or 3V power supply and power consumption must be lower than 100 mA.

3.11 Programming of STM32F4 board using application firmware

Before programming, both CN3 jumpers ON ST-LINK/V2 (or V2-A) functions enabled for onboard programming (default); but if both CN3 jumpers OFF ST-LINK/V2 (or V2-A) functions enabled for application through external CN2 connector (SWD supported)

First of all, we write our application firmware in any higher-level language like "C". Then after we compile it and make the machine level in ".hex". The STM32F4DISCOVERY firmware package contains these binary images (*.hex and *.dfu). Now to reprogram STM we have two ways:

1. In-system programming tool: Connect the STM32F4DISCOVERY board to a PC with a 'USB type A to Mini-B' cable through USB connector CN1 to power the board. Make sure that the embedded ST-LINK/V2 is configured for in-system programming (both CN3 jumpers ON). Use ".hex" binary with preferred in-system programming tool like we are using STM32 ST-LINK.

2. Bootloader (USB FS Device in DFU mode)

3. Use *.dfu binary (for example,\Project\Demonstration\Binary\STM32F4Discovery _ Demonstration_V1.0.0.dfu) with "DFUse\DFUse Demonstration" tool (available for download from www.st.com) to reprogram the demonstration firmware.

3.12 LCD Configuration

For the starting phase, we are using GUI (Graphical User Interface) making software named “Nextion Editor” from Nextio.

To dumb the GUI into the LCD ROM we have two ways:

1. To use USB to TTL converter to upload the ‘.tft’ the file generated by the “Nextion Editor” directly from the “Nextion Editor”.

2. To use FAT32 SD Card, in this method we have to first format SD card in FAT32 mode. Firstly using the inbuilt command for the format and then after copying the ‘.tft’ file in the SD card, then inserting it in the LCD, we were getting an error. Here problem is that, SD card hasn’t properly formatted in the FAT32 mode. So in alternative way we downloaded and installed “SD Card Formatter”. And using this software we have formatted SD card, and it is worked and our LCD is ready… Now we can remove SD card from the LCD slot, GUI code is saved in the ROM of LCD.

Now we have to establish UART (Universal Asynchronous Receiver Transmitter) link between LCD and STM32F407VG controller.

Start a new project and using cubeMX configure pin PA2 as USART Tx (Asynchronous mode) and PA3 as USART Rx(Asynchronous mode), and a generate project. We have to configure USART using ‘configure’ tab. The main configuration is to select USART2_Rx for DMA request.

3.13 UART Communication link

From the datasheet of the Nextion Enhanced NX8048K070, we can find that to communicate between microcontroller and display we have to use UART protocol.

For UART communication we need three wires, Rx, Tx and common GND as shown in the figure.

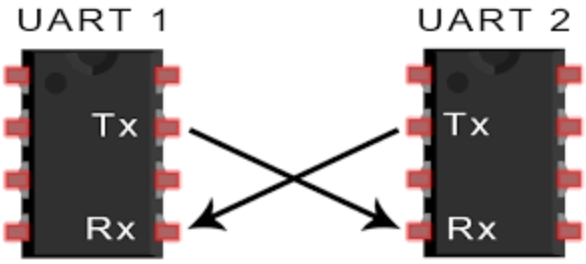


Fig. 3.13(a) UART Communication hardware interface, [8]

3.13.1 Baud Rate Configuration for UART

3.13.1.1 System clock Configuration of STM32F407VGT6

For proper functioning of any microcontroller we need to give precise clock pulses to the ALU and control unit. As we can see in the flow chart we have two options for clock input. First one is HSI (High-Speed Internal) and second is HSE (High-Speed External).

Depending on the values of the selection bit, the clock will be fed. And depending on the value of the PLL_M, PLL_N, PLL_P system clock can be set. The maximum value of the system clock can be 168 MHz.

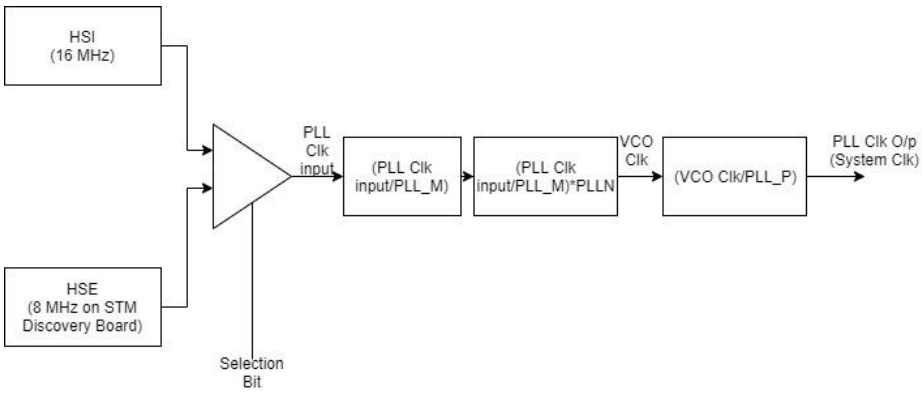


Fig. 3.13.1.1(a) Generation of system clock for STM32F407VGT6, [9]

3.13.1.2 Baud Rate for STM32F407VG controller

From the datasheet of the LCD, we can derive that typical baud rate used by the LCD is 9600 bps. So we have to configure our microcontroller in a way that it also communicates at the 9600 bps. Using flowchart given we can set the baud rate at 9600 bps.

3.13.1.3 Baud Rate for NX8048K070

From the datasheet of the Nextion Enhanced NX8048K070 and given in the datasheet, we can find that to display can be operated at 2400 bps and can go up to 115200 bps. But typically we use 9600 bps because it is the standard UART protocol baud rate.

3.14 Offline and Cloud Storage of Pages

After the device has started on, a user can interact with E-writer using a display of E-writer. A user can write using a stylus on the display.

Now the situation arises, that we have to store the data, which is written by a user, into the memory. To point this objective, we have two ways, one is to store the content in the offline mode and another way is to store the content in the online mode. Let's briefly discuss both the modes.

3.14.1 Storage of Data in ROM

In this mode of data storage, we are going to use memory to store the data. In this method also we have two ways to store the data. In the first way, we have internal ROM, internal to the micro-controller. The second way in this technique, we can store the data in external ROM, external to the controller.

As we have discussed in the "Memory Organization" topic in "Hardware-Firmware Subsystems" chapter, we have 1 M Byte of flash ROM in the STM32F407 controller. We are using this ROM to store our firmware code. We are burning (fusing) firmware code in this ROM memory. We can use this memory as a storage element to store the user's data.

The second way in this technique is to use external EEPROM. Here, EEPROM stands for Electrically Erasable Programmable Read-Only Memory. For this, we have to interface the microcontroller with the external EEPROM.

There are many different EEPROM available for embedded system. These EEPROM comes in two configurations. These configurations are serial and parallel. But to simplify the circuit diagram and to make it less complex we are going with serial mode EEPROM. And in the serial EEPROM, the best suited for our requirement is AT24Cxxx series of EEPROM. To fulfil the sole purpose of demonstrating the concept we are using the EEPROM of 256 Kilo Bit of series AT24Cxxx. Its IC number is AT24C256 [10] from Microchip.

In most on-chip or off-chip EEPROM write cycles are limited up to some finite number. In our case, both EEPROMs have 1,00,000 (1 million) write cycles. So every time we should avoid unnecessary write on the EEPROM and by doing this we can improve the life of EEPROM.

There are many techniques available for this. But one of the best ways to reduce the number of writes on the particular bit is, instead of using write data (written page in our case) on the EEPROM we should use update data function. In this, whenever we have to write data into the EEPROM we should first read every location of memory and compare it with the data to write, if both the location have the same state of memory then there is no need to write at that location. And leave it as it is. And if both the location does not have same state then write the appropriate data into the EEPROM location.

AT24C256 has serial I^2C (Inter Integrated Circuit) interface and has three I^2C address lines. So, one microcontroller can control maximum 8 this type of memory ICs. Here, our controller works as a master and memory IC works as a slave.

As given in the below circuit GPIO 0, GPIO 1 and GPIO 2 pins are used to set the address of a particular memory IC. We can hardwire these lines to the GND or VCC according to our convenience. But, here we are using three GPIO pins of the controller. The SCK and SDA are two lines used for I^2C communication protocol. The SCK is “Serial Clock” and used to share common clocks between memory IC and Microcontroller to synchronize

serial communication. The SDK is “Serial Data” and used to transfer data between the controller and memory IC. These both lines are open-drain, so have to connect pull-high resistors.

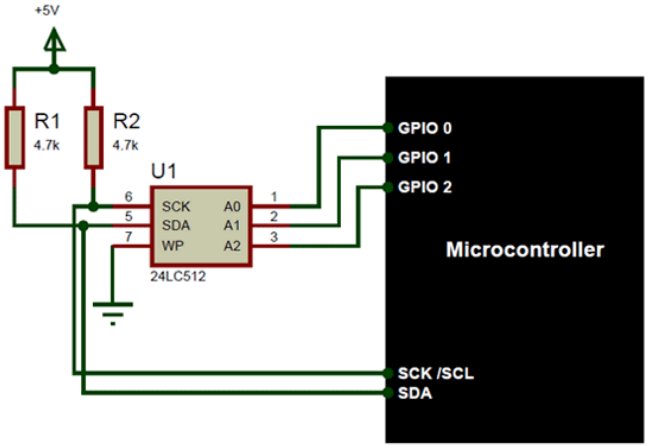


Fig. 3.14.1(a) I^2C communication between microcontroller and EEPROM, [11]

3.14.2 Storage of data in the cloud

As we discussed before we have to ways to store data, first we have seen and the second one is cloud storage. For storage connectivity, we need to use NodeMCU development board.

This NodeMCU board contain ESP8266 microcontroller with wi-fi capability. We can connect Arduino Mega[12] with ESP8266 with I2C serial communication protocol and use Mega with master and ESP8266 as a slave. ESP8266 can be used to connect with the cloud. As a master, Arduino Mega would give commands to ESP8266 which data to store on cloud and which data to retrieve from the cloud.

CHAPTER 4

RESULTS AND ANALYSIS

Graphical User Interface made using Nextion Editor:

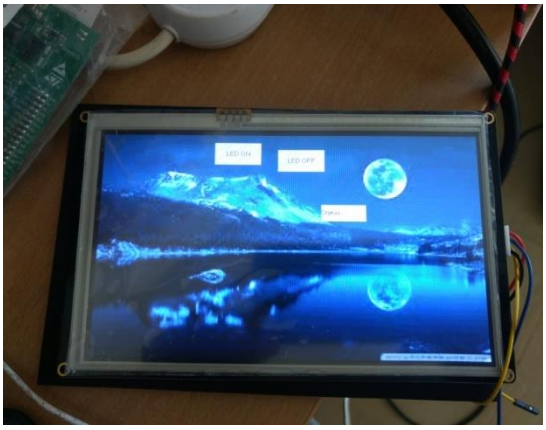


Fig. 4(a) GUI generated by the Nextion Editor

Interfacing between STM microcontroller and LCD:

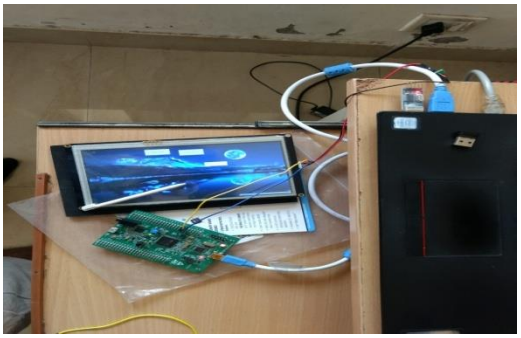


Fig. 4(b) UART communication between microcontroller and LCD

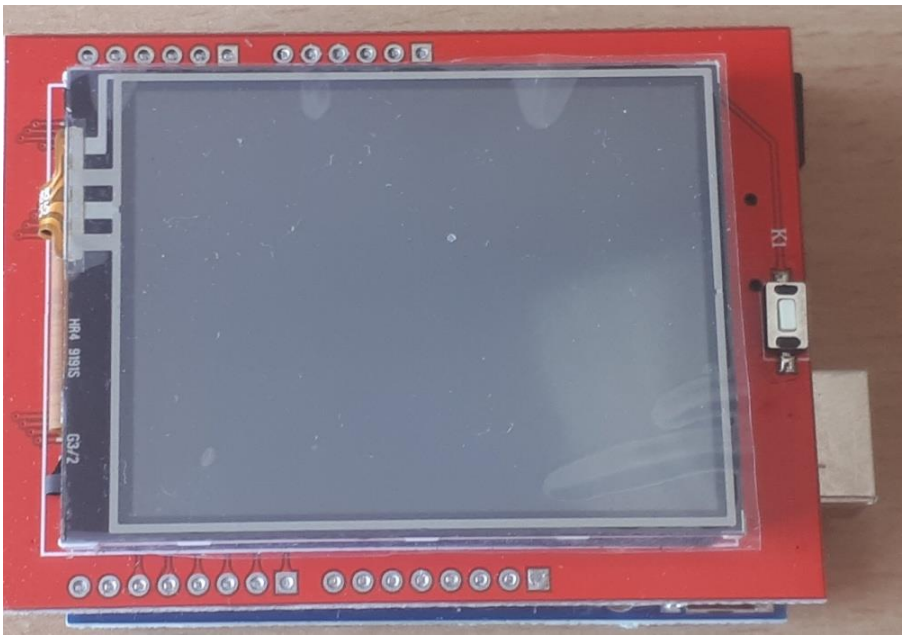


Fig. 4(c) Display Module with Arduino UNO

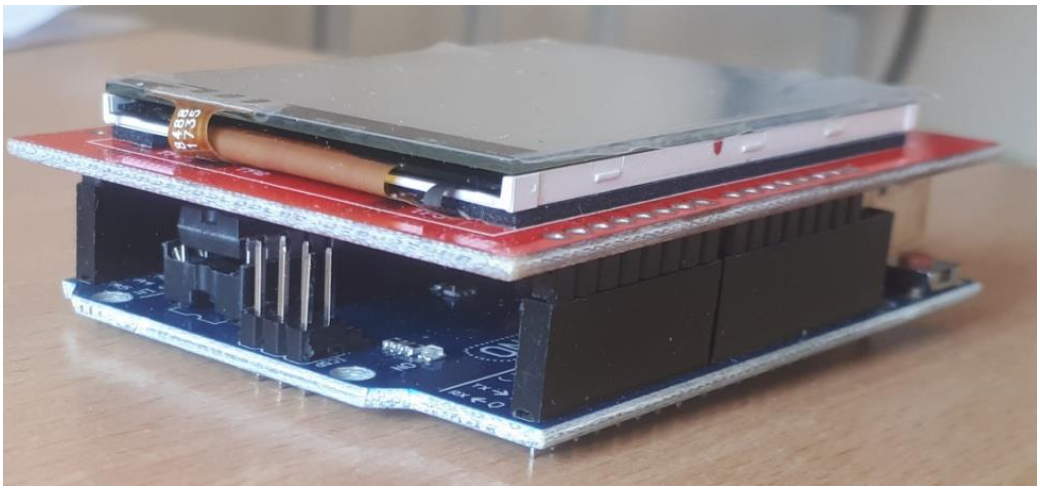


Fig. 4(d) Side view of display module and Arduino UNO

Fig.4(e) Login Page

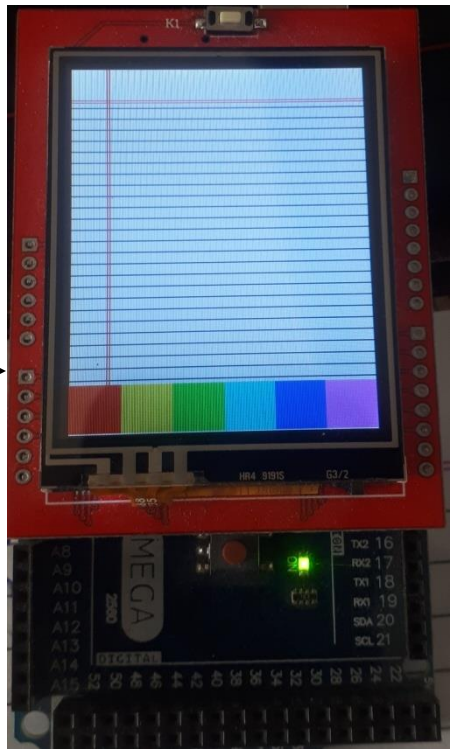


Fig. 4(f) New Page to Write

Fig. 4(g) Written things on the E-writer

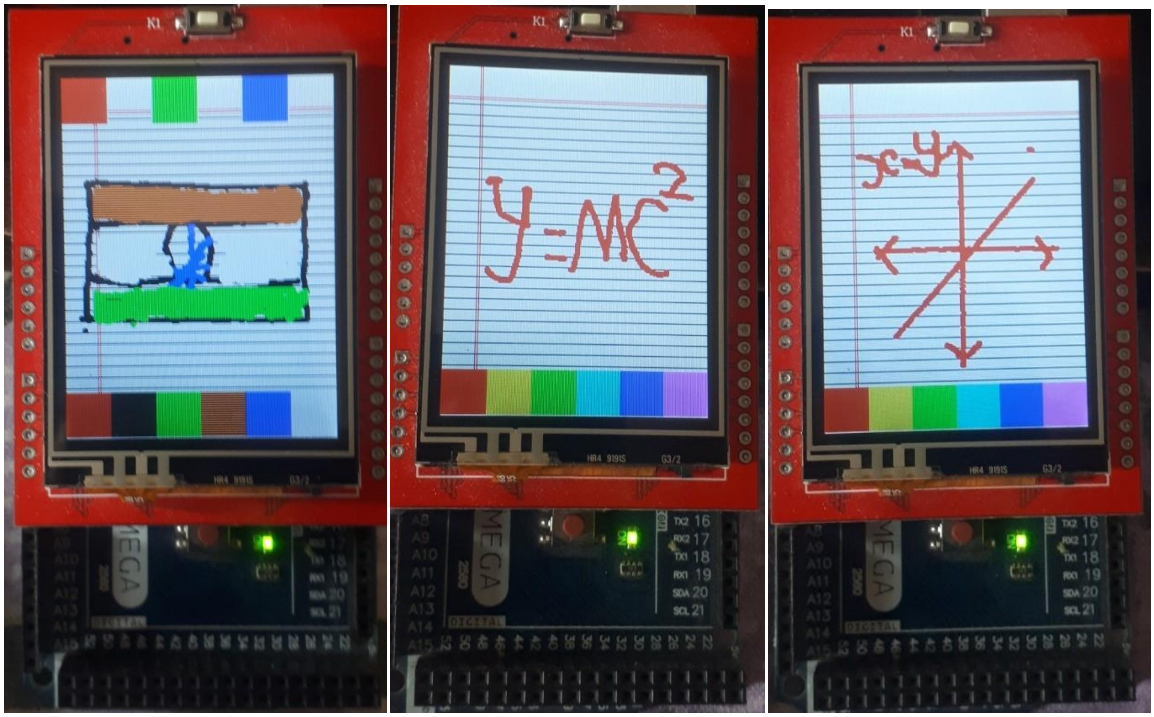


Fig. 4(h) Figure, equation and graph on the E-writer

We have also calculated latency of screen response time. And it comes out like 500 milliseconds. We have used the traditional method to calculate latency. First we put a scale parallel to display's length. And drag a line using a scale at 2 cm/s. The longest side of the display is 5 cm long, so needed 2.5 second to draw a 5 cm line. When our stylus reached at length of 5 cm from origin, display updation is just reached to approximately 4 cm. So, difference is 1 cm which will be used to calculate latency. So, after using equation,

Time=Distance/Velocity

Here, velocity=2 cm/s, distance=1 cm

Using equation we would get latency of 500 milliseconds.

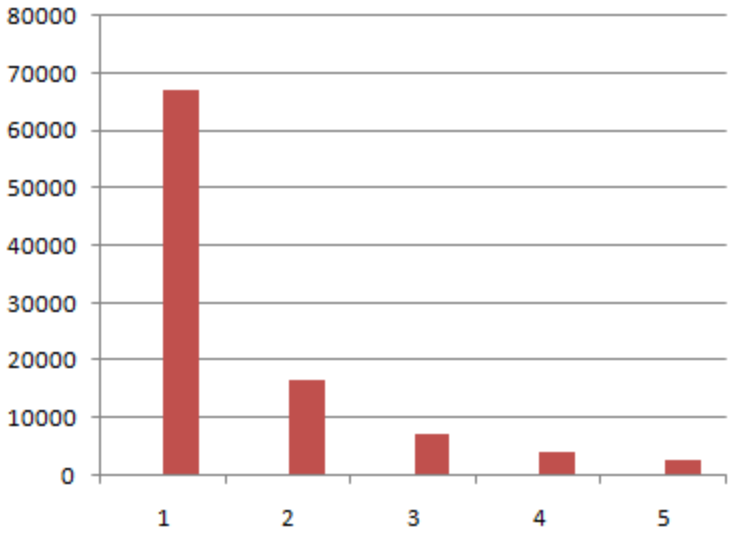


Fig. 4(i) Graph between text size and memory requirement

Above graph is laid down between Memory requirements in byte verses text size used to write. So, from graph we can analyze that as text size is increases memory requirement is decreasing exponentially.

CHAPTER 5

CONCLUSION

In this project, we have made E-writer using microcontroller. The E-writers which are available in the market are very much expensive. Besides having expensive, they do not have all the features that one good E-writer should have. Like, we have seen two products in the topic "Product Survey". One of the electronics Readers was Amazon Kindle by company Amazon. The company commits that a kindle can long last around a week before it get off.

But, with these merits, it is only one E-reader. So, one can't use it as a writer. In comparison to this, our E-writer has writing functionality. We also the have functionality to write using a stylus.

We have also seen one E-writer from Portronics Company. This is not truly E-writer. But, actually it can be used as a ruffpad. One can write on it using stylus and the written things would be static on the display. We are not able to store any data in the device, only can be used for ruff work. A strong point in this E-writer is that it does not need a power to display the contents on the display. It requires power only to erase the content from the display. So, this product is very much power efficient.

With all these privileges, it is just ruffpad. In contrast to this, by adding the memory unit our E-writer can be able to store the data. Our E-writer can be used to replace physical notebook.

Let's compare them in terms of costing. Amazon kindle costs around 10k and only give reading functionality. But in our case, in out prototype, if we use small sized 4.2 inch display, the prototype would cost us around 4k and if we use large size of display, then at the max it would cost around 8k.

Our E-writer not only has plus points. One of the main pitfalls in our design is that our design is not that much power efficient. From out practical experiment we have calculated that with 2000 mAh battery (power bank) our small display prototype can last for around 6 hours. But, if we use this design with large display it only lasts for around two hours 30 minutes. We need to work on this issue to reduce power consumption, to make device more portable.

Improvement in our prototype could be add very easy going User-Interface. And as we discussed, we can add new functionality in cloud storage to facilitate for flexibility to the user.

At the last we conclude that, in order to reduce the paper usage and paper wastage, we have tried to make one low cost E-writer to give virtual paper like feel and to replace traditional writing method on the notebook.

REFERENCES

[1] Dan Gavrilescu, Adrian Catalin Puitel, Gheorghe Dutuc, Grigore Craciun, "Environmental impact of pulp and paper mills", Environmental engineering and management journal, Jan, 2012, [online] Available: https://www.researchgate.net/publication/281761323_Environmental_impact_of_pulp_and_paper_mills

[2] Good Reader, https://goodereader.com/blog/electronic-readers/amazon-kindle-paperwhite-4-review (accessed Sept. 12th, 2019)

[3] price hunt, https://www.price-hunt.com/tablets/rewy-15-r-8-5-inch-e-writer-lcd-writing-pad-paperless-memo-digital-tablet-notepad-stylus-drawing-for-erase-button-and-pen-to-write.php (accessed Sept. 15th, 2019)

[4] Portronics, https://www.portronics.com/products/ruffpad-10-plus (accessed Sept. 12th, 2019)

[5] Remarkable, https://remarkable.com/ (accessed Aug. 25th, 2019)

[6] ST Microelectronics, "Discovery kit with STM32F407VG MCU", Datasheet UM1472, Sep. 2011 [Revised May. 2017].

[7] Nextion Technology, "NX8048K070", Datasheet, May. 2011[Revised Oct. 2018].

[8] Circuit Basics, http://www.circuitbasics.com/basics-uart-communication/ (accessed Nov. 2nd, 2019)

[9] ST Microelectronics, "STM32F405/415, STM32F407/417, STM32F427/437 and STM32F429/439 advanced Arm®-based 32-bit MCUs", Datasheet RM0090, Sep. 2011[Revised Feb. 2019].

[10] Microchip, https://www.microchip.com/wwwproducts/en/AT24C256C (accessed Mar. 23th, 2020)

[11] Components101,https://components101.com/ics/24lc512-eeprom-pinout-equivalent-datasheet (accessed Mar. 30th, 2020)

[12] Arduino, https://www.arduino.cc/(accessed Feb. 24th, 2020)

ACRONYMS

MCU	Micro Controller Unit
EPD	Electorphoretic Display
LCD	Liquid Crystal Display
IDE	Integrated Development Environment
LQFP	Low-profile Quad Flat Pack
FPU	Floating Point Unit
UART	Universal Asynchronous Receiver Transmitter
HSI	High Speed Internal
HSE	High Speed External
EEPROM	Electrically Erasable Programmable Read Only Memory
I^2C	Inter Integrated Circuit

APPENDIX A

1 Features

The STM32F4DISCOVERY offers the following features:

- STM32F407VGT6 microcontroller featuring 32-bit ARM Cortex® - M4 with FPU core, 1-Mbyte Flash memory, 192-Kbyte RAM in an LQFP100 package
- On-board ST-LINK/V2 on STM32F4DISCOVERY or ST-LINK/V2-A on STM32F407G-DISC1
- ARM® mbed Enabled™ (http://mbed.org) with ST-LINK/V2-A only
- USB ST-LINK with re-enumeration capability and three different interfaces:

– Virtual COM port (with ST-LINK/V2-A only)

  – Mass storage (with ST-LINK/V2-A only)
  – Debug port

- Board power supply:
  – Through USB bus
- LIS302DL or LIS3DSH ST MEMS 3-axis accelerometer
- MP45DT02 ST MEMS audio sensor omni-directional digital microphone
- CS43L22 audio DAC with integrated class D speaker driver
- Eight LEDs:
  – LD1 (red/green) for USB communication
  – LD2 (red) for 3.3 V power on
  – Four user LEDs, LD3 (orange), LD4 (green), LD5 (red) and LD6 (blue)
  – 2 USB OTG LEDs LD7 (green) VBUS and LD8 (red) over-current
- Two push buttons (user and reset)
- USB OTG FS with micro-AB connector
- Extension header for all LQFP100 I/Os for quick connection to prototyping board and easy probing
- Comprehensive free software including a variety of examples, part of the STM32CubeF4 package or STSW-STM32068 for legacy standard library usage

Quick start

The STM32F4DISCOVERY is a low-cost and easy-to-use development kit to quickly evaluate and start a development with an STM32F407VG high-performance microcontroller.

Before installing and using the product, accept the Evaluation Product License Agreement from the www.st.com/stm32f4-discovery webpage.

For more information on the STM32F4DISCOVERY and for demonstration software, visit the www.st.com/stm32f4-discovery webpage.

Getting started

Follow the sequence below to configure the STM32F4DISCOVERY board and launch the DISCOVER application:

1. Check jumper position on the board, JP1 on, CN3 on (DISCOVERY selected).
2. Connect the STM32F4DISCOVERY board to a PC with a USB cable 'type A to mini-B' through USB connector CN1 to power the board. Red LED LD2 (PWR) then lightsup.
3. Four LEDs between B1 and B2 buttons are blinking.
4. Press user button B1 to enable the ST MEMS sensor, move the board and observe the four LEDs blinking according to the motion direction and speed. (If a second USB cable 'type A to micro-B' is connected between PC and CN5 connector, then the board is recognized as standard mouse and its motion will also control the PC cursor).
5. To study or modify the DISCOVER project related to this demonstration, visit the www.st.com/stm32f4-discovery webpage and follow the tutorial.
6. Discover the STM32F407VG features, download and execute programs proposedin the list of projects.
7. Develop the application using available examples.

System requirements

- Windows® OS (XP, 7, 8 and 10), Linux® 64-bit or macOS™
- USB type A to Mini-B cable.

Development toolchains supported

- Keil® MDK-ARM(a)
- IAR™ EWARM(a)
- GCC-based IDEs including free SW4STM32 from AC6
- ARM® mbed Enabled™ online

Embedded ST-LINK/V2 (or V2-A)

ST-LINK/V2 on STM32F4DISCOVERY or ST-LINK/V2-A on STM32F407G-DISC1 is an
embedded tool for programming and debugging.

The embedded ST-LINK/V2 (or V2-A) supports only SWD for STM32 devices. For information about debugging and programming features refer to *ST-LINK/V2 in-circuit debugger/programmer for STM8 and STM32*, UM1075 User manual, which describes in details all the ST-LINK/V2 features.

The changes on ST-LINK/V2-A versus ST-LINK/V2

version are listed below. New features supported on ST-

LINK/V2-A:

- Virtual COM port interface on USB (see *Section 6.1.3: ST-LINK/V2-A VCP configuration*)
- Mass storage interface on USB

Features not supported on ST-LINK/V2-A:

- SWIM interface
- Minimum supported application voltage limited to 3 V
- USB power management request for more than 100 mA power on USB

Known limitation:

- Activating the readout protection on ST-LINK/V2-A target, prevents the target application from running afterwards. The target readout protection must be kept disabled on ST-LINK/V2-A boards.

Drivers

Before connecting the STM32F4DISCOVERY board to a Windows® PC (XP, 7, 8 and 10) through the USB, a driver for the ST-LINK/V2 (or V2-A) must be installed. It is available at the *www.st.com* website. In case the STM32 Discovery is connected to the PC before the driver is installed, some Discovery interfaces may be declared as "Unknown" in the PC device manager. To recover from this situation, after installing the dedicated driver, the association of "Unknown" USB devices found on the STM32F4DISCOVERY board to this dedicated driver, must be updated in the device manager manually.

ST-LINK/V2 (or V2-A) firmware upgrade

The ST-LINK/V2 (or V2-A) embeds a firmware upgrade mechanism for in-situ upgrade through the USB port. As the firmware may evolve during the life time of the ST-LINK/V2 (or V2-A) product (for example new functionalities, bug fixes, support for new microcontroller families), it is recommended to visit the *www.st.com* website before starting to use the Discovery board and periodically, to stay up-to-date with the latest firmware version.

ST-LINK/V2-A VCP configuration

The ST-LINK/V2-A supports a virtual COM port (VCP) on U2 pin 12 (ST-LINK_TX) and U2 pin 13 (ST-LINK_RX) but these pins are not connected to the USART of the STM32F407 microcontroller for mbed support.

Two solutions are possible to connect an STM32F407 USART to the VCP on the PC:

- Using an USART to USB dongle from the market connected for instance to STM32F407 USART2 available on connector P1 pin 14 (PA2: USART2_TX) and P1 pin 13 (PA3: USART2_RX).

Using flying wires to connect ST-LINK/V2-A virtual COM port (ST-LINK VCP on U2 pin 12 and 13) to STM32F407 USART2 (PA2 and PA3: P1 pin 14 and 13)

Power supply and power selection

The power supply is provided either by the host PC through the USB cable, or by an external 5V power supply.

The D1 and D2 diodes protect the 5V and 3V pins from external power supplies:

- 5V and 3V can be used as output power supplies when another application board is connected to pins P1 and P2.
  In this case, the 5V and 3V pins deliver a 5V or 3V power supply and power consumption must be lower than 100 mA.
- 5V can also be used as input power supplies e.g. when the USB connector is not connected to the PC.
  In this case, the STM32F4DISCOVERY board must be powered by a power supply unit or by auxiliary equipment complying with standard EN-60950-1: 2006+A11/2009, and must be Safety Extra Low Voltage (SELV) with limited power capability.

LEDs

- LD1 COM: LD1 default status is red. LD1 turns to green to indicate that communications are in progress between the PC and the ST-LINK/V2.
- LD2 PWR: red LED indicates that the board is powered.
- User LD3: orange LED is a user LED connected to the I/O PD13 of the STM32F407VGT6.
- User LD4: green LED is a user LED connected to the I/O PD12 of the STM32F407VGT6.
- User LD5: red LED is a user LED connected to the I/O PD14 of the STM32F407VGT6.
- User LD6: blue LED is a user LED connected to the I/O PD15 of the STM32F407VGT6.
- USB LD7: green LED indicates when VBUS is present on CN5 and is connected to PA9 of the STM32F407VGT6.
- USB LD8: red LED indicates an overcurrent from VBUS of CN5 and is connected to the I/O PD5 of the STM32F407VGT6.

Push buttons

- B1 USER: User and Wake-Up buttons are connected to the I/O PA0 of the STM32F407VG.
- B2 RESET: Push button connected to NRST is used to RESET the STM32F407VG.

On-board audio capability

The STM32F407VG microcontroller uses an audio DAC (CS43L22) to output sounds through the audio mini-jack connector.

The STM32F407VG microcontroller controls the audio DAC through the I^2C interface and processes digital signals through an I^2S connection or an analog input signal.

- The sound can come independently from different inputs:
  - ST-MEMS microphone (MP45DT02): digital using PDM protocol or analog when using the low pass filter
  - USB connector: from external mass storage such as a USB key, USB HDD, and so on
  - Internal memory of the STM32F407VG microcontroller
- The sound can be output in different ways through the audio DAC:
  - Using I^2S protocol
  - Using DAC to analog input AIN1x of the CS43L22
  - Using the microphone output directly via a low-pass filter to analog input AIN4x of the CS43L22

USB OTG supported

The STM32F407VG microcontroller is used on this board to only drive the USB OTG full speed. The USB micro-AB connector (CN5) allows the user to connect a host or device component, such as a USB key, mouse, and so on.

Two LEDs are dedicated to this module:

- LD7 (green LED) indicates when VBUS is active
- LD8 (red LED) indicates an overcurrent from connected device

Motion sensor (ST-MEMS LIS302DL or LIS3DSH)

Two different versions of motion sensors (U5 in schematic) are available on the board depending on the PCB version. The LIS302DL is present on board MB997B (PCB revision
A) and the LIS3DSH is present on board MB997C (PCB rev C).

The LIS302DL and LIS3DSH are both ultra-compact low-power three-axis linear accelerometers.

The motion sensor includes a sensing element and an IC interface able to provide the measured acceleration to the external world through the I^2C/SPI serial interfaces.

The LIS302DL has dynamically user selectable full scales of +-2g/+-8g and it is capable of measuring acceleration with an output rate of 100Hz to 400Hz.

The LIS3DSH has ±2g/±4g/±6g/±8g/±16g dynamically selectable full-scale and it is capable of measuring acceleration with an output data rate of 3.125 Hz to 1.6 kHz.

APPENDIX B

1. General Description

This display module is a transmissive type color active matrix TFT(Thin Film Transistor) liquid crystal display (LCD) that uses amorphous silicon TFT as a switching device. This module is composed of a TFT LCD module, a driver circuit, and a back-light unit.

The resolution of a 2.4" contains 240（RGB)X320 dots and can display up to 262k colors.

2. Module Parameter

Features	Details	Unit
Display Size(Diagonal)	**2．4"**	**-**
LCD type	**α-Si TFT**	**-**
Support interface mode	**MCU SPI RGB**	**-**
Display Mode	**TN/ Normally white**	**-**
Resolution	**240 RGB x 320**	**-**
View Direction	**12 O' clock**	**Best image**
Grayscale Inversion Direction	**6 O' clock**	**-**
Module Outline	**42.72(H) ×60.26(V)×2.2 (T)**	**mm**
TP Outline	**42.32(H) x59.26(V)x1.0(T)**	**mm**
TP Viewing Area	**38.72(H) x57.92(V)**	**mm**
TP Active Area	**37.72(H) x53.16(V)**	**mm**
Active Area	**36.72 (H)×48.96(V)**	**mm**
Viewing Area	**N/A**	**mm**
Pixel Size	**0.153(H) x0.153 (V)**	**mm**
Pixel Arrangement	**Stripe**	**-**
Display Colors	**262K**	**-**
Interface	**System parallel interface**	**-**
Driver IC	**ILI9341**	**-**
Operating Temperature	**-20～70**	**ºC**
Storage Temperature	**-30～80**	**ºC**
LCM brightness	**210**	cd / m^2
Weight	**TBD**	**g**

3. Absolute Maximum Ratings

V_{SS}=0V, Ta=25ºC

Item		Symbol	Min.	Max.	Unit
Supply Voltage	Power supply	VCC	-0.3	+4.6	V
	Analog	VCI	-	-	V
	IO	IOVCC	-	-	V
Input Voltage		Vi	-0.3	IOVCC+0.3	V
Storage temperature		T_{stg}	-30	+80	ºC
Operating temperature		T_{op}	-20	+70	ºC

Storage humidity	H_{stg}	10	Note 1	%RH
Operating humidity	H_{op}	10	Note 1	%RH

Note 1: 90%RH max, If Ta is below 50ºC; 60%RH max, If Ta is over 60ºC.

4. DC Characteristics

Item		Symbol	Min.	Typ.	Max.	Unit
Supply Voltage	Power supply	VCC	2.5	2.8	3.3	V
	Analog	VCI	2.5	2.8	3.3	V
	IO	IOVCC	1.65	2.8	3.3	V
Logic Low input voltage		V_{IL}	0.0	-	0.2*IOVCC	V
Logic High input voltage		V_{IH}	0.7*IOVCC	-	IOVCC	V
Logic Low output voltage		V_{OL}	-	-	0.2*IOVCC	V
Logic High output voltage		V_{OH}	0.8*IOVCC	-	IOVCC	V
Current Consumption	Normal display	Ivdd	-	-	-	mA
	Standby mode	Ivdd-	-	-	-	uA
Frame Frequency		f_{FR}	-	TBD	-	Hz

5. Backlight Characteristics

Backlight Characteristics

Item	Symbol	Condition	Min	Typ	Max	Unit
Forward Voltage	V_f	Ta=25 ºC,I_F=(15*4)mA	3.1	3.2	3.3	V/LED
Forward Current	I_f	Ta=25 ºC, V_F=3.2V	-	60	-	mA
Luminance	L_V	-		3500		cd / m^2
Uniformity	Avg	-	-	-	-	%
CIE	X	-		-	-	-
	Y	-	-	-	-	-
Power dissipation	P_d	-	-	-	-	mW
Backlight Driving Voltage	V_{AK}	-	-	-	-	V
Drive method	Constant current					
LED Configuration	4 White LEDs in Parallel					

6. Optical Characteristics

Optical Characteristics

Ta=25ºC, VDD=2.8V, TN LC+ Polarizer

	Item	Symbol	Condition	Specification			Unit
				Min.	Typ.	Max.	
B	Luminance on TP surface(I_f =60mA)	Lv	Normally viewing angle	-	-TBD	-	cd/m²

Contrast ratio(See 6.3)		CR	$\theta_X = \theta_Y = 0°$	400	500	-	-
Response time (See 6.2)		$T_{R+}T_F$		-	16	32	ms
Chromaticity Transmissive (See 6.5)	Red	X_R	-	0.606	0.626	0.646	-
		Y_R		0.314	0.334	0.354	-
	Green	X_G		0.299	0.319	0.339	-
		Y_G		0.537	0.557	0.577	-
	Blue	X_B		0.122	0.142	0.162	-
		Y_B		0.102	0.122	0.142	-
	White	X_W		0.298	0.318	0.338	-
		Y_W		0.317	0.337	0.357	-
Viewing Angle (See 6.4)	Horizontal	θ_{X+}	Center CR≥10	-	45	-	Deg.
		θ_{X-}		-	45	-	
	Vertical	θ_{Y+}		-	45	-	
		θ_{Y-}		-	20	-	
NTSC Ratio(Gamut)		-	-	-	TBD	-	%

1. Interface Pins Definition

Module interface

No.	Symbol	I/O	DESCRIPTION
1	VCI	P	Power Supply
2	IOVCC	P	Power supply for IO port
3	IM0	I	Select the MCU interface mode MPU Parallel interface bus and serial interface select If use RGB Interface must select serial interface. (Note1).
4	IM1		
5	IM2		
6	IM3		
7	RESET	I	LCM reset signal
8	VSYNC	I	Frame synchronizing signal for RGB interface operation.
9	HSYNV	I	Line synchronizing signal for RGB interface operation.
10	DOTCLK	I	Dot clock signal for RGB interface operation.
11	ENABLE	I	Data enable signal for RGB interface operation.
12^29	DB17^DB0	I/P	Data bus
30	SDO	I/P	LCD Read for the MPU interface
31	SDI	I	Write control pin for the MPU interface
32	RD	I	LCD Read for the MPU interface
33	WR	I	Write control pin for the MPU interface
34	RS/SCL	I	(Please according to, the IC specification choice model).

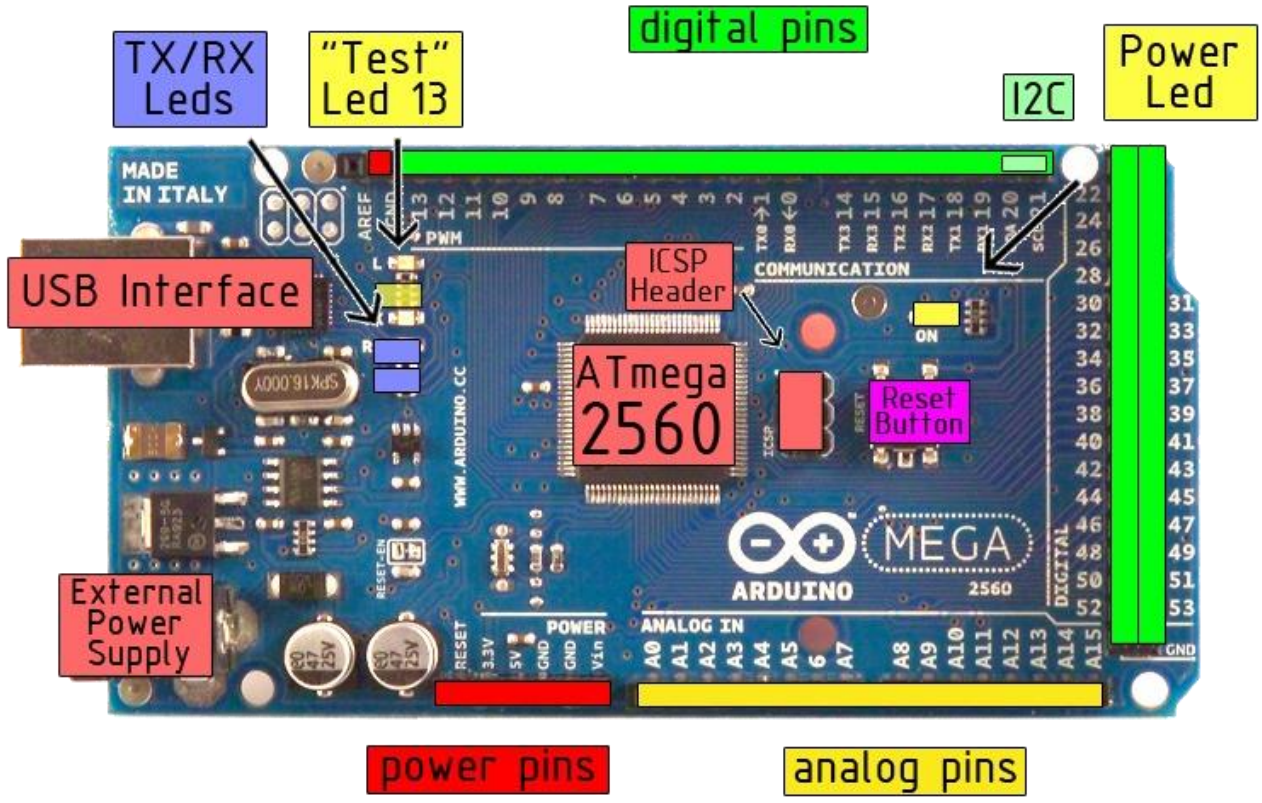
35	CS	I	The data is applied on the rising edge of the SCL signal.If not used, fix this pin at VDDI or VSS.
36	GND	P	Ground
37	LEDA	I	LED Anode
38	LEDK1	I	LED Cathode
39	LEDK2		
40	LEDK3		
41	LEDK4		
42	XR	I	TOUCH P ANLE PIN
43	YD		
44	XL		
45	YU		

Note1:

IM3	IM2	IM1	IM0	MCU-Interface Mode	Pins in use	
					Register/Content	GRAM
0	0	0	0	8080 MCU 8-bit bus interface I	D[7:0]	D[7:0],WRX,RDX,CSX,D/CX
0	0	0	1	8080 MCU 16-bit bus interface I	D[7:0]	D[15:0],WRX,RDX,CSX,D/CX
0	0	1	0	8080 MCU 9-bit bus interface I	D[7:0]	D[8:0],WRX,RDX,CSX,D/CX
0	0	1	1	8080 MCU 18-bit bus interface I	D[7:0]	D[17:0],WRX,RDX,CSX,D/CX
0	1	0	1	3-wire 9-bit data serial interface I	SCL,SDA,CSX	
0	1	1	0	4-wire 8-bit data serial interface I	SCL,SDA,D/CX,CSX	
1	0	0	0	8080 MCU 16-bit bus interface II	D[8:1]	D[17:10],D[8:1],WRX,RDX,CSX,D/CX
1	0	0	1	8080 MCU 8-bit bus interface II	D[17:10]	D[17:10],WRX,RDX,CSX,D/CX
1	0	1	0	8080 MCU 18-bit bus interface II	D[8:1]	D[17:0],WRX,RDX,CSX,D/CX
1	0	1	1	8080 MCU 9-bit bus interface II	D[17:10]	D[17:9],WRX,RDX,CSX,D/CX
1	1	0	1	3-wire 9-bit data serial interface II	SCL,SDI,SDO,CSX	
1	1	1	0	4-wire 8-bit data serial interface II	SCL,SDI,D/CX,SDO,CSX	

APPENDIX C

Microcontroller	ATmega2560
Operating Voltage	5V Input Voltage (recommended) 7-12V Input Voltage (limits)
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz



Power

The Arduino Mega2560 can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the EEPROM library).

Input and Output

Each of the 54 digital pins on the Mega can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip .
- **External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2).** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.
- **PWM: 0 to 13.** Provide 8-bit PWM output with the analogWrite() function.
- **SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS).** These pins support SPI communication, which, although

provided by the underlying hardware, is not currently included in the Arduino language. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Duemilanove and Diecimila.

- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- **I^2C: 20 (SDA) and 21 (SCL).** Support I^2C (TWI) communication using the Wire library (documentation on the Wiring website). Note that these pins are not in the same location as the I^2C pins on the Duemilanove.

The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and analogReference() function.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with analogReference().
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

Communication

- The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega8U2 on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

- A SoftwareSerial library allows for serial communication on any of the Mega's digital pins.

The ATmega2560 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation on the Wiring website for details.

Programming

-
- The Arduino Mega2560 can be programmed with the Arduino software (download). For details, see the reference and tutorials.
-
- The Atmega2560 on the Arduino Mega comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).
-
- You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details.

APPENDIX D

Features

- Low-voltage and Standard-voltage Operation
  - V_{CC} = 1.7V to 5.5V
- Internally Organized as 32,768 x 8
- 2-wire Serial Interface
- Schmitt Trigger, Filtered Inputs for Noise Suppression
- Bidirectional Data Transfer Protocol
- 400kHz (1.7V) and 1MHz (2.5V, 2.7V, 5.0V) Compatibility
- Write Protect Pin for Hardware Protection
- 64-byte Page Write Mode
  - Partial Page Writes Allowed
- Self-timed Write Cycle (5ms Max)
- High Reliability
  - Endurance: 1,000,000 Write Cycles
  - Data Retention: 40 Years
- Lead-free/Halogen-free Devices Available
- Green Package Options (Pb/Halide-free/RoHS Compliant)
  - 8-lead JEDEC SOIC, 8-lead TSSOP, 8-pad UDFN, and 8-ball VFBGA Packages
- Die Sale Options: Wafer Form, Waffle Pack, and Bumped Wafers

Description

The Atmel® AT24C256C provides 262,144-bits of Serial Electrically Erasable and Programmable Read-Only Memory (EEPROM) organized as 32,768 words of 8 bits each. The device's cascading feature allows up to eight devices to share a common 2-wire bus. The device is optimized for use in many industrial and commercial applications where low-power and low-voltage operation are essential. The devices are available in space-saving 8-lead JEDEC SOIC, 8-lead TSSOP, 8-pad UDFN, and 8-ball VFBGA packages. In addition, this device operates from 1.7V to 5.5V.

SUGGESTIONS