A

PROJECT REPORT

ON

AN OPEN SOURCE SMARTWATCH FOR INTERNET CONNECTED DEVICES AND HOME AUTOMATION

Submitted in partial fulfillment of the requirements for the award of the degree

Of

BACHELOR OF TECHNOLOGY
In
COMPUTER SCIENCE AND ENGINEERING
By

KARAN NAYAN(1030910103)
KAIVAN WADIA(1030910101)

Under the guidance of

DR. E. POOVAMMAL

(Professor and HOD, Department of Computer Science & Engineering)

FACULTY OF ENGINEERING & TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SRM UNIVERSITY, SRM Nagar, Kattankulathur - 603203

Kancheepuram District, Tamil Nadu

May 2013

BONAFIDE CERTIFICATE

Certified that this project report titled **‘An Open Source Smartwatch for Internet Connected Devices and Home Automation’** is the bonafide work **KARAN NAYAN(1030910103**) and **KAIVAN WADIA(1030910101**) who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

Signature of the Guide	Signature of the HOD
Dr. E. Poovammal	**Dr. E. Poovammal**
Professor & HOD	Professor & HOD
Department of Computer Science	Department of Computer Science
And Engineering,	And Engineering,
Kattankulathur Campus,	Kattankulathur Campus,
SRM University -603203.	SRM University -603203.

Date:

Internal Examiner	External Examiner

ABSTRACT

The rapid pace of development in the semiconductor and the technological field has now made it possible to couple together a sufficiently powerful processor, its peripherals, and a low power display together in a small device such as a wristwatch. This opens up a field of possibilities for the applications of the concept.

The advantage of using a wristwatch form factor is that it is very small, hence convenient to handle and can easily interact with a smartphone to display the various notifications and messages without taking the smartphone out of the pockets unnecessarily hence providing an instant on feature .

The Smartwatch can also be used as a device to display the various sensor information collected from the smartphone or collected from the internet using the smartphone or any other device. The Smartwatch with its access to the smartphone can also be used to display various notification from the phone such as calls and messaging as well as information from the internet, such as Social Networking apps (Facebook, Twitter etc.),To do lists and many other application. Despite these advantages there are many development issues associated with the Smartwatch concept such as power usage, battery size, minimization of entire circuit and many others.

Apart from these, the Smartwatch can also be used for home automation by using it as a remote control for various household applications and electrical circuitry by using the wireless interface for communicating with the electrical devices.

The objective of the project is to make a basic working prototype of the Smartwatch, solve the associated problems and if possible make a working end product in its minimized wrist-watch form factor and release it as an open-source hardware and software platform so that developers can use it for further development.

ACKNOWLEDGMENT

We take this opportunity to thank our **Director (E&T), Dr.C.MuthamizhChelvan**, for providing us with excellent infrastructure that is required for the development of our project.

We would also like to thank **Dr.E.Poovammal,** Professor and Head of Department of Computer Science and Engineering, for providing us with conductive ambience for developing our project.

We would like to thank our **B.Tech Project Coordinator, Mrs. R.Annie Uthra,** Assistant Professor (S.G) Department of Computer Science and Engineering for her continuous support.

We would also like to express our sincere gratitude to our Project Guide, **Dr.E.Poovammal** Professor and Head of Department of Computer Science and Engineering for her able assistance, timely suggestions and guidance throughout the duration of this project

We also convey our gratitude to all other members in the project panel and those who have contributed to this project directly or indirectly.

CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVATIONS

LCD-Liquid Crystal Display

RTC-Real Time Clock

PDA- Personal digital assistant

LiPo- Lithium polymer

OS- Operating Systems

SD- Secure Digital

I^2C- Two wire interface

SPI- Serial Peripheral Interface

UART- Universal asynchronous receiver/transmitter

MCU- Microcontroller

CPU- Central Processing Unit

BCD- Binary Coded Decimal

I/O- Input/output

TX/RX- Transmit/Receive

API- Application programming interface

CHAPTER 1

INTRODUCTION

1.1 AREA OF RESEARCH

The area of the project can be described as Wearable Computing, or it can fall under the category of Future Technology for Computing.

1.2 INTRODUCTION

A wrist watch is an attractive form factor for a wearable computer. It has the advantage of always being with you; and it can be instantly viewed with the flick of the wrist. By comparison, devices such as pagers, cell phones, tablets, and PDAs are typically worn on belts or kept in pockets and need to be picked up and opened first before they can be accessed. One of the reasons of the success of the Palm was its moving to an instant-on paradigm. Wristwatches move us to the next step; to an instantly-viewable paradigm.

Because people generally keep watches on their wrists, watches are less likely to be misplaced compared to phones and pagers. For example a hip holster is not the most comfortable place to keep a cellular phone while sitting in a car and so people tend to keep them on the seat or dashboard and forget them when they leave the car in the parking lot.

It is also a common occurrence to get notifications and other such alerts on a phone or tablet during an event or meeting when it would be inappropriate to use the same. In such cases it would be very useful to be able to access the same on a watch which can be accessed more easily in order to act on such matters which may sometimes be very important.

The watch form factor requires a relatively small screen size, and there is not much room for input devices or batteries. The value of a wristwatch platform depends on finding good solutions to these issues. To interact with the watch, we need only one hand since the hand on which the watch is worn is practically useless for controlling input devices on the watch.

The increasing research into processors has resulted in the capability of interfacing a small processor with a small memory and a high resolution LCD screen effectively and efficiently. We intend to use this capability along with the internet capability of a phone to access and use various internet services.

There is plenty of interest in adding functions to watches in the industry. Our interest is not in so much in providing a watch, as in providing an open, extensible computing platform in a small form factor. Our objective is to understand the challenges in packaging, hardware design, power management, and embedded software.

1.3 EXISTING SYSTEMS

Several smart watches are available commercially today. Personal Information Management applications are provided on the Seiko RuputerTM [1], the onHand PCTTM [2], and the Timex DataLinkTM [3]. These watches pack an impressive amount of function but have low resolution displays. This limits the amount and type of data that can be displayed on their screens. Some of the above have cumbersome/confusing user interfaces with many buttons leading to limited adoption.

By having the capability of a high resolution LCD and a relatively powerful processor we can perform many more functions more efficiently. The increased display capability allows us to display much more information in a more elegant and beautiful manner. The increased processing power enables us to have more applications and better performance.

CHAPTER 2

BACKGROUND WORK

C. Narayanaswami and M.T.Raghunath propose [4] to create a small form factor device which can be used as a wearable computing device as a companion to another internet powered device. The objective of their work is to understand the various challenges involved in the design, power management and embedded software aspects of the device and to demonstrate whether such a device is feasible or not for widespread adoption. The researchers have analyzed various embedded devices at their disposal as well have carefully studies the various peripheral which are available and have selected those peripherals which satisfy their need for the power budget, processor usage, form factor and ease of use. They are using a monochrome display for showing the output, a low power ARM 7 processor and LiPo battery for powering the device, infrared for communication and roller wheel for input. They have shown their device to be the next step in the evolution of watches by moving watches form devices that only show the time to devices which can be used for personal information management. For communicating with the other devices the watch uses Wireless Markup Language and has created a suite of application for the device like clock faces, alarm, image viewer, calendar, to do list and small games such as Tetris. In the end they have successfully demonstrated that it is possible to create a device with the form factor of a watch and present data in timely fashion despite the various difficulties and problems associated with it.

CHAPTER 3

SYSTEM DESCRIPTION

3.1 PROBLEM DEFINITION

The aim of the project is to create a new Form Factor for wearable computing which displays the personal information of a user with internet connected apps along with standalone applications which uses Bluetooth for communication and making the device connect to intelligent sensors for home automation and start the development of an entire ecosystem of both hardware and software surrounding the app. The end goal is to release the designs along with the hardware, schematics and the software developed as Open Source to let the developers build on the work that we have done.

3.2 ARCHITECTURE

The Architecture (*Figure: 1*) is divided into five parts:

1. Smartwatch software architecture
2. Smartwatch hardware architecture
3. Automation circuit software architecture
4. Automation circuit hardware architecture
5. Android device

3.2.1 Smartwatch software architecture

The smart watch (Figure: 1) has an embedded operating system which is used to control the entire smart watch. It communicates with four other software modules which are:

i. Pushbuttons
ii. Real Time Clock
iii. LCD
iv. Bluetooth Module

The microcontroller polls the pushbuttons at the beginning of each processing cycle to check for inputs coming from them. At the startup the processor also loads the LCD

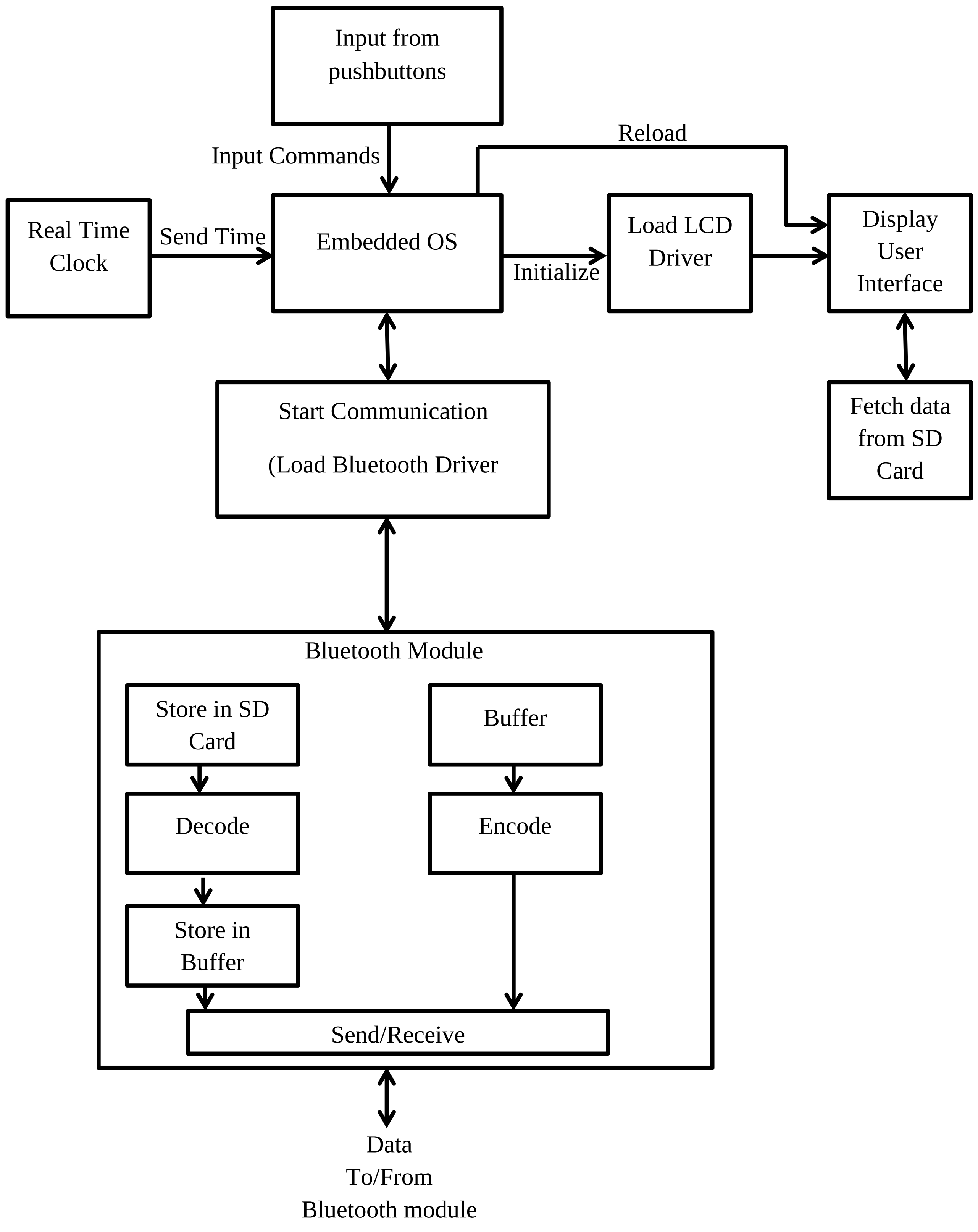


Figure 1: Smartwatch Hardware Architecture

driver and later only refreshes the data to be shown by the LCD which fetches the data from the SD card. When communication has to be established, the processor communicates with the Bluetooth module which depending on the whether it has to send or receive data performs that particular operation.

3.2.2 Smartwatch hardware architecture

The hardware (Figure: 2) on the smartwatch consist of mainly six different components:

i. ATMEGA 328 microcontroller
ii. Bluetooth Module(Seeedstudio bluetooth module)
iii. Pushbuttons
iv. SD card (MicroSD)
v. Real Time Clock(DS1307)
vi. LCD (Philips PCF8833)

The ATMEGA 328 microcontroller is used to control the entire smartwatch.

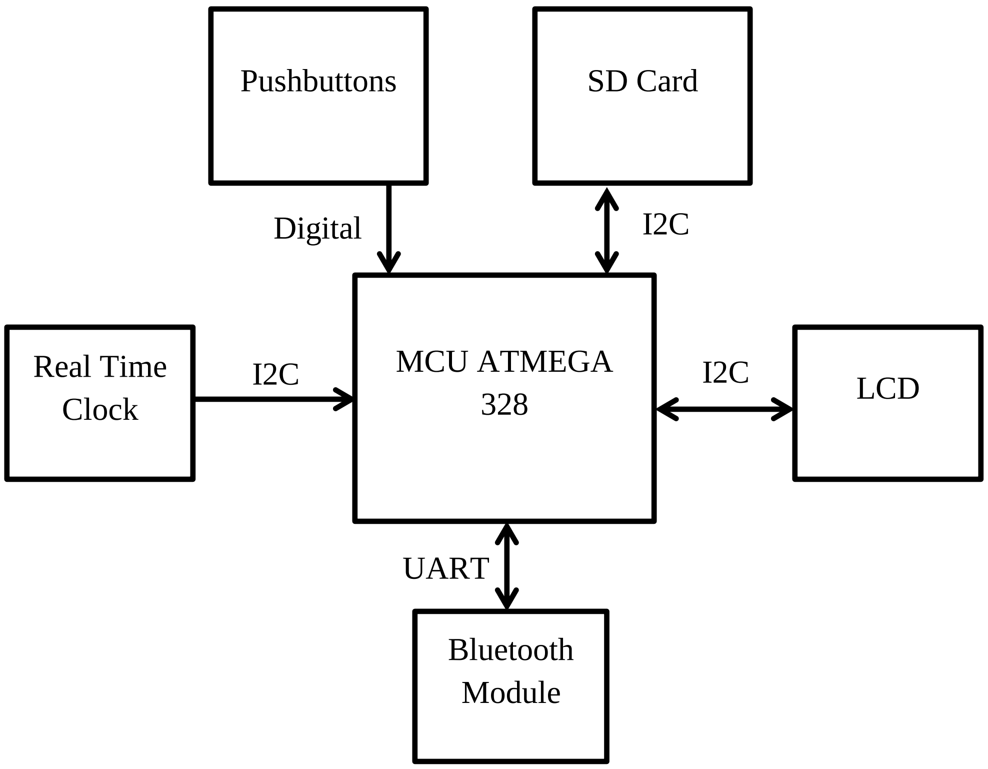


Figure 2: Smartwatch Hardware Architecture

It communicates with the peripheral via various protocols. It communicates with the real time clock, LCD and SD Card via I2C protocol and with the Bluetooth via UART and reads the digital inputs from the pushbuttons every time when polling.

3.2.3 Automation circuit software architecture

The Operating system on the hardware side of the automation circuit (Figure: 3) keeps on polling the Bluetooth module listening for commands or recognized Bluetooth devices and as soon as it receives the commands or finds the Bluetooth device it loads the profile for that Bluetooth device and controls the relays.

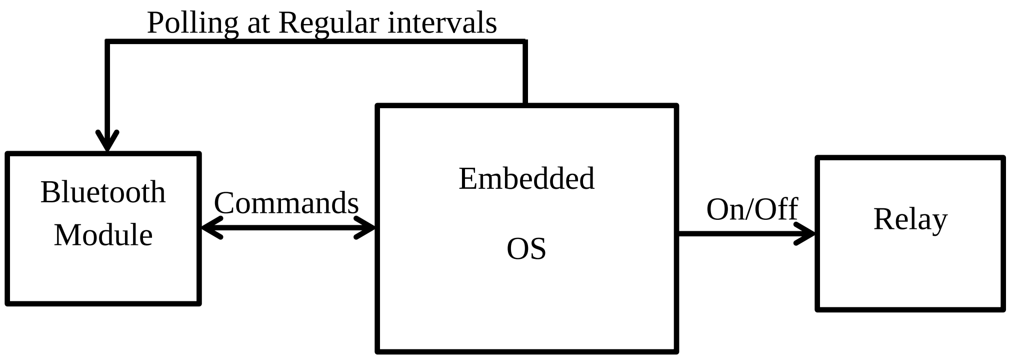


Figure 3: Automation circuit software architecture

3.2.4 Automation circuit hardware architecture

The automation circuit (Figure: 4) consists of a microcontroller which receives commands from the Bluetooth module[13] via UART protocol, interprets the command and sends the desired output to the relay via digital signals which switches it on or off and controls the home automation circuit.

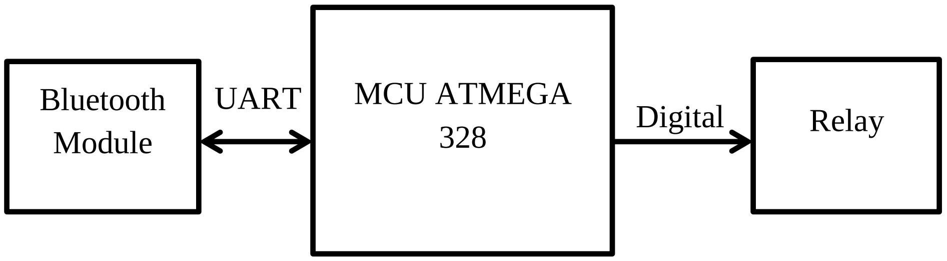


Figure 4: Automation circuit hardware architecture

3.2.5 Android Device

The Android [**5**] device (Figure 5) consists of a Bluetooth communication module which receives the various incoming requests and sends it to the parser. The parser in turn will interpret the request and summon the required module for its execution and reply. Upon obtaining the reply the module sends it to the generator which transforms the reply into the format specified by the protocol and sends it to the Bluetooth module which then sends it to the watch.

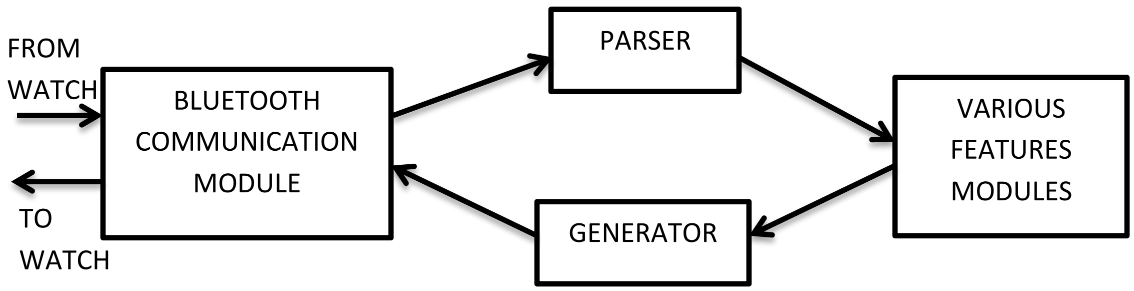


Figure 5: Architecture of Android App

3.3 METHODOLOGY

3.3.1 Smartwatch

The operating system in the smartwatch is a real-time embedded single-tasking event driven operating system. When the system boots, the Home screen is loaded by default and the processor keeps on polling the pushbuttons for input to come from. If a data processing task such as showing the twitter feed of mail is selected then the OS loads the particular routine to fetch the message from the SD Card, then parses it and display the same on the screen. In the background the processor polls the RTC every processing cycle to fetch the time and update the digital clock on the top of the screen. If the smartwatch is currently on the analog clock screen then it polls the clock every second and from the change in time updates the second, minute and hour hands. If the user wants to sync the data with the phone, the smartwatch switches on the Bluetooth module in Inquiry mode and waits for connection to establish. Once the connection is established the data transfer takes place and data is stored in SD Card and after the completion of the data transfer the device comes back to home screen.

3.3.2 Home Automation Circuit

When the home automation circuit boots, the microcontroller switches on the Bluetooth device in inquiry mode and keeps searching for the Bluetooth devices around it. When it finds one it looks for it in the recognized devices list in the controller, if it is there in the list the MCU loads its usage profile, i.e., the electronic circuits which were on for it the last time and loads the same and turns on or off the relays to control the electrical devices. This entire operation is repeated every 30 seconds and controls the devices till the time the watch is in the range of the home automation circuit.

3.3.3 Android Device

The Android application for the smart watch is intended to be run only once and then minimized to work in the background. Upon opening the application the user is presented with options to login in to the various features available on the smart watch. He can choose the features he intends to use by logging into them one by one. Also when the application is run for the first time it connects to the watch via the Bluetooth adapter. If the adapter is turned on the user will be asked for his permission before turning it on. The application can now accept the requests from the watch and send replies to the requests. Even upon minimizing the app the Bluetooth communication still goes on in the background and the requests are handled and the appropriate replies are sent.

3.4 SOFTWARE SPECIFICATIONS

For Smartwatch	
Operating System	Windows 7
Software	Arduino Eclipse with Pydev Python 3.3 Eagle 6.4 Processing 1.5.1
For Android	
Operating System	Android 2.3 (Gingerbread) or above
Software	Eclipse[6]
Software Packages	Android SDK [7] EasyFacebook Android SDK [8] Twitter4j [9] gmail-java-client-library [10]

Table 1: Software Specification

3.5 HARDWARE SPECIFICATIONS

Watch	Core 2 Duo(2.2 Ghz) 4GB RAM Bluetooth 2.0+
Phone	Bluetooth 2.0+ 800 MHz processor or above 256 MB RAM

Table 2: Hardware Specification

CHAPTER 4
IMPLEMENTATION

4.1 SMARTWATCH

The smartwatch has various peripherals interfaced[14] to the smartwatch are:

i. RTC(Real Time Clock)
ii. Pushbuttons
iii. Bluetooth Shield
iv. SD Card
v. LCD

Now we will discuss each of them in brief.

4.1.1 Real Time Clock

The real time clock[12] is used for time keeping on the smartwatch when the device is not powered on. The RTC module ((Figure: 6) uses a DS1307 chip for timekeeping which is connected to a 32kHz oscillator and the device is continuously powered by a coin cell. The communication between the RTC and the ATMEGA 328 MCU is carries out via SPI on analog pins 4 and 5 of the ATMEGA 328 MCU.

For communication via I^{2}C we use the default address of the RTC i.e., 0x68 and start communication on that address

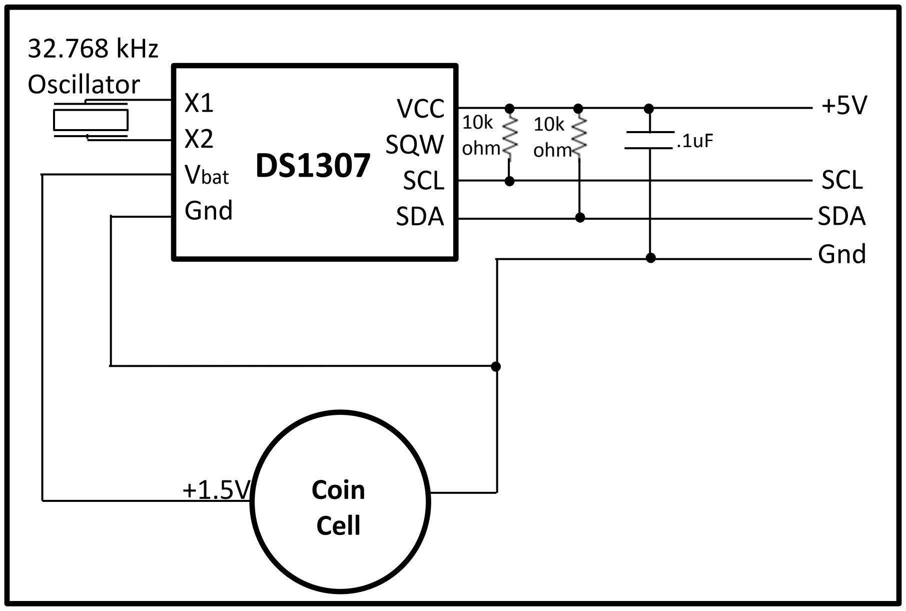


Figure 6: RTC Circuit

. After the communication is started we receive 7 BCD bytes which are converted to get the Time and the Date from the RTC and are used by the MCU for timekeeping.

4.1.2 Pushbuttons

The smartwatch receives input from pushbuttons (Figure 7) three connected on one side and one on the other of the display. They are used for giving input for Up, Down, Select and back actions.

When the pushbutton is open (unpressed) there is no connection between the two legs of the pushbutton, so the pin is connected to ground (through the pull-down resistor) and we read a LOW. When the button is closed (pressed), it makes a connection between its two legs, connecting the pin to voltage, so that we read a HIGH. (The pin is still connected to ground, but the resistor resists the flow of current, so the path of least resistance is to +5V.)

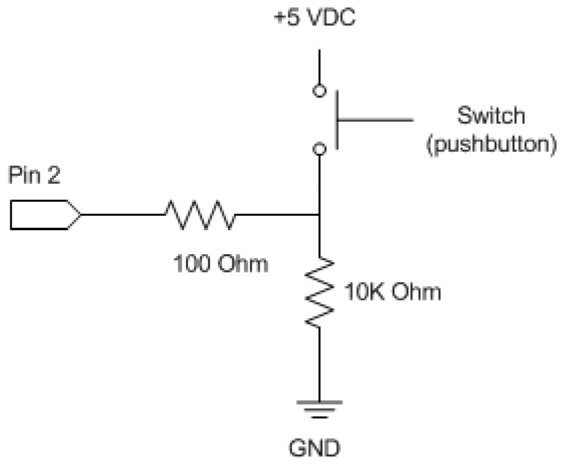


Figure 7: Pushbutton Schematic

4.1.3 LCD Display

The output device which is used in the Smartwatch is an LCD Display Philips PCF8833 which was widely used in Nokia 6100[11].

The important specifications for this display are as follows:

- 132 x 132 pixels
- 12-bit color rendition (4 bits red, 4-bits green, 4-bits blue)
- 3.3 volts
- 9-bit SPI serial interface (clock/data signals)

The Nokia 6100 display has 132 x 132 pixels; each one with 12-bit color (4 bits RED, 4 bits GREEN and 4 bits BLUE). The normal orientation is shown in figure 8.

To communicate with the device we send 9 bits to the display serially, the ninth bit indicates if a command byte or a data byte is being transmitted. The ninth bit (command or data) is clocked out first and is LOW to indicate a command byte or HIGH to indicate a data byte.

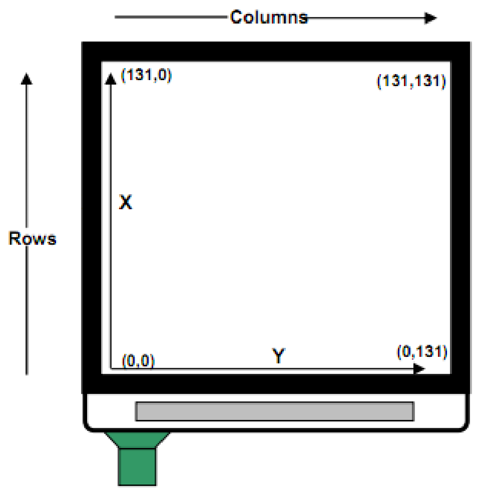


Figure 8: Orientation of the LCD

The Philips PCF8833 controller has a 17424 word memory (132 x 132), where each word is 12 bits (4-bit color each for red, green and blue). We address it by specifying the address of the desired pixel with the Page Address Set command (rows) and the Column Address Set command (columns) (Figure 9).

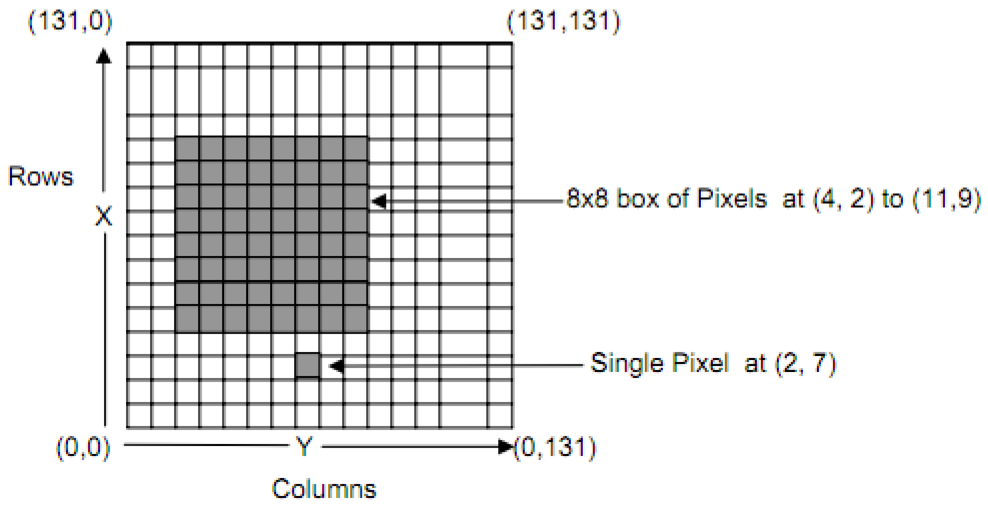


Figure 9: Philips PCF8833 Pixel Memory

We can now use the SPI channel to communicate with the display to print the data we want by sending a SPI Command followed by the SPI Data. Using this various functions like setPixel(), setCircle (), setChar(), setStr(), setLine(), setdLine(),setRect() are created and used to create the user interface on the LCD.

4.1.4 Bluetooth Module

The Bluetooth Shield integrates a Serial Bluetooth module. It is used with ATMEGA 328 for transparent wireless serial communication. We are using two pins D0 and D1 i.e., Hardware Serial Port to communicate with Bluetooth module.

Some features of the module are:

- Low Power 1.8V Operation ,1.8 to 3.6V I/O
- PIO control
- UART interface with programmable baud rate
- Integrated antenna
- Default Baud rate: 38400, Data bits:8, Stop bit:1,Parity:No parity

The Bluetooth module works with the microcontroller by connecting to the hardware Serial port and using it for data transfer (Figure 10).

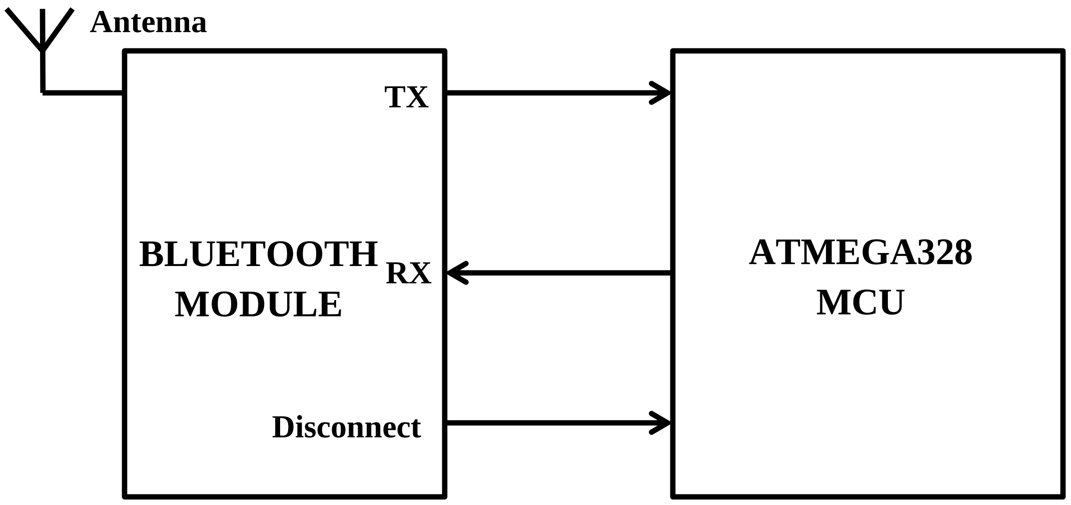


Figure 10: Bluetooth Module Working Sketch

The message request and response sequence used in the Bluetooth module to connect and start data transfer is shown in Figure 11.

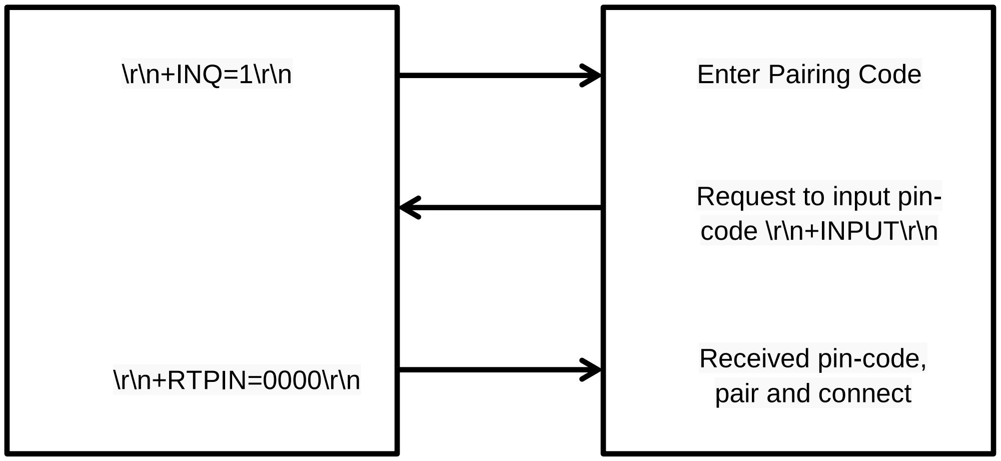

\r\n+INQ=1\r\n
Enter Pairing Code
Request to input pin-code \r\n+INPUT\r\n
\r\n+RTPIN=0000\r\n
Received pin-code, pair and connect

Figure 11: Command Sequence to establish Bluetooth communication

4.1.5 Sequence of operation of Smartwatch

The smartwatch works on 4 levels (Figure 12) to keep the entire working as simple as possible. As soon as the device boots up, it loads the home screen which is on the first level. From there the user has option to select a menu and go to a sublevel i.e., detailed description of a menu like Twitter, Mail etc. or open the analog watch which will do the timekeeping and uses the input from RTC.

From level 2 the user can open the level 3, which has the detailed description of the message in level 2 or go back to the home screen.

In this way he can navigate through the whole watch

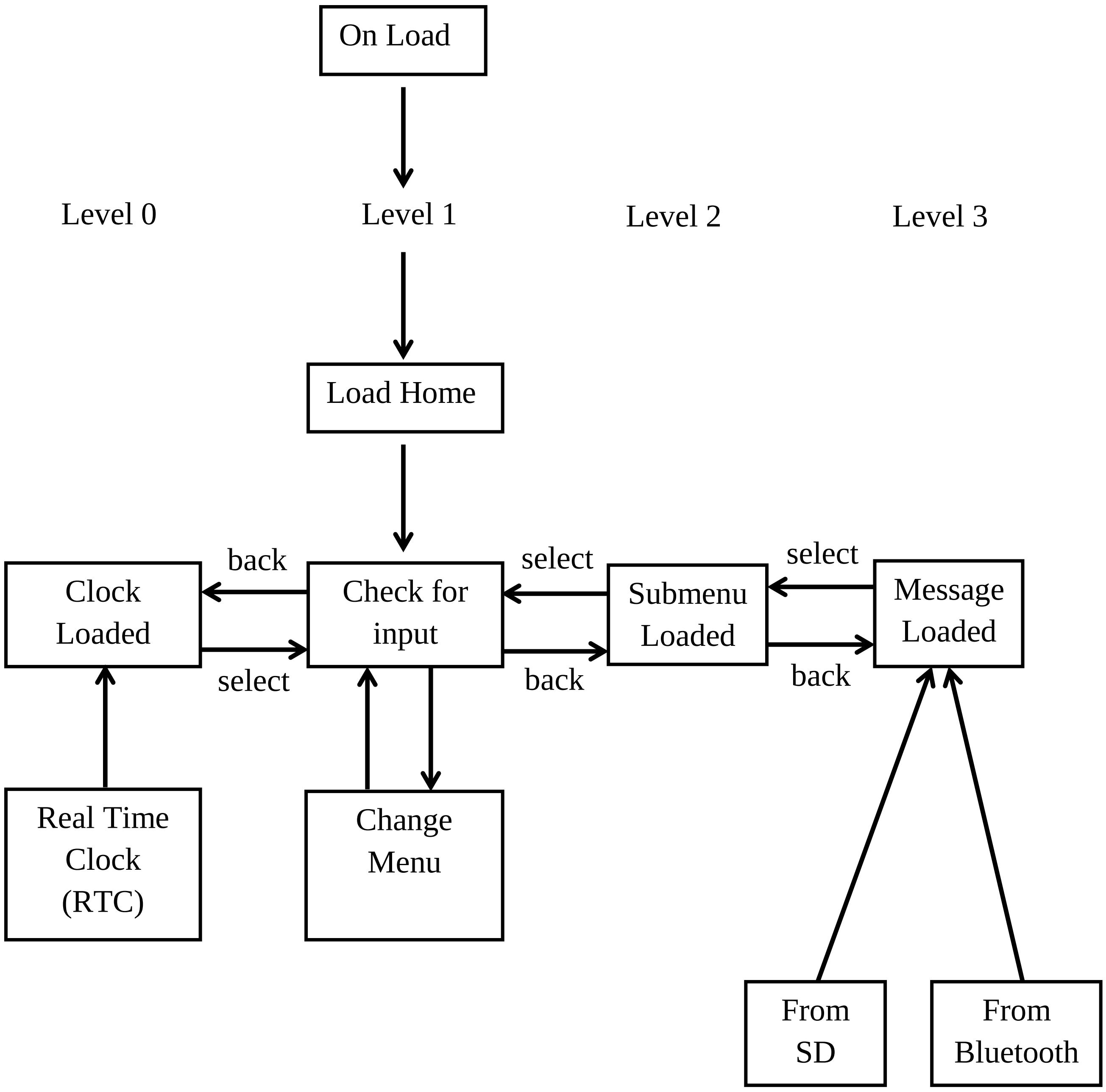


Figure 12: Sequence of operation of Smartwatch

4.2 ANDROID MODULES

Name	**Modules**
Bluetooth Communication	• Receives data and requests from the watch • Sends replies to the watch upon receiving the requests.
Parser	• Interprets the requests coming in from the watch according to the protocol and activates the appropriate module.
Facebook	• Enables access to Facebook feed and notifications.
Twitter	• Enables access to Twitter home feed.
Gmail	• Enables access to user’s inbox and any other mailboxes.
Generator	• Generates the reply according to the protocol after upon receiving the data from the appropriate module.

Table 3: Android Modules

ALGORITHMS FOR PROPOSED MODULES

4.2.1 Algorithm for Bluetooth Module

- Check whether Bluetooth Adapter is switched on.
- If the Bluetooth adapter is off.
    - Ask the user to turn it on.
- If the phone is not paired with the watch.
    - Pair the phone with the watch.
- Connect to the watch using serial communication.
- Begin listening for data from watch.
- Upon receiving data send it to the parser.
- Receive data to be sent from generator.

- Send data to watch

4.2.2 Algorithm for Parser

- Accept the request data from the Bluetooth Module.
- If the data is a request
    - Check the feature ID to establish what feature the request is targeting.
- If the data is a reply
    - Parse the reply and send it to the appropriate module based on the feature ID.
- If the data is an acknowledgement.
    - If data is acknowledgement request
        - Send acknowledgement reply.
    - Else
        - Accept the acknowledgement.

4.2.3 Algorithm for Twitter Module

- If Twitter application is installed
    - Login using the Twitter application.
- Else
    - Login using the built-in browser.
- Use the OAuth 2.0 protocol to request for user's access token and store it in the shared preferences.
- Upon receiving the request for Twitter home feed use the Twitter4j library functions to get the home feed of the user.
- Store the feed in String format.
- Send the data to the generator.

4.2.4 Algorithm for Facebook Module

- If Facebook application is installed
    - Login to Facebook using the Facebook application.

- Else
    - Login to Facebook using built-in browser.
- Request access token for user_about_me and manage_notifications permissions.
- Store the access token in the shared preferences.
- Upon receiving the request for Facebook notifications make a call to the Facebook Graph API using the access token stored earlier.
- Store the API reply in String format.
- Send the data to the generator.

4.2.5 Algorithm for Gmail Module

- Access the Accounts Manager to get the default Gmail account of the user.
- On obtaining the credentials of the account make a request to the Gmail API for an access token to read the user's mail.
- Store the access token in the shared preferences.
- Upon receiving the request for mail use the access token and the gmail-java-client-library to read the contents of the user's inbox or other requested labels.
- Store the reply in String format.
- Send the data to the generator.

4.2.6 Algorithm for Generator

- Accept the data from the various application feature modules.
- Check for the feature number and generate the required string data according to the specified protocol.
- Send the generated string to the Bluetooth module to forward it to the watch.

CHAPTER 5

RESULT ANALYSIS

SMARTWATCH PROTOTYPE

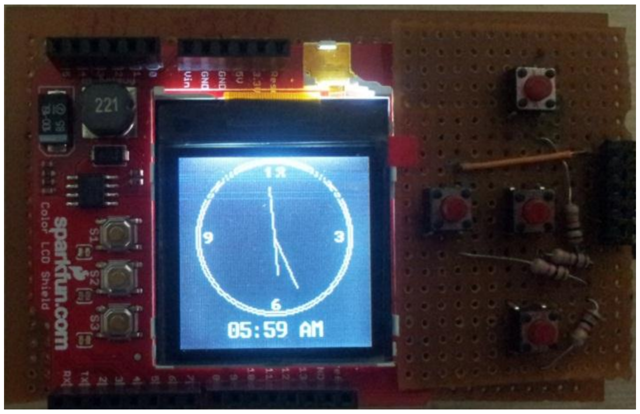


Figure 13: Smartwatch Prototype Board

The smartwatch prototype board which is used to demonstrate the feasibility of the concept showing an analog clock in operation on level 0 of the sequence of operation.

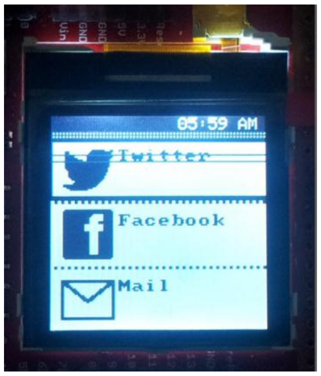
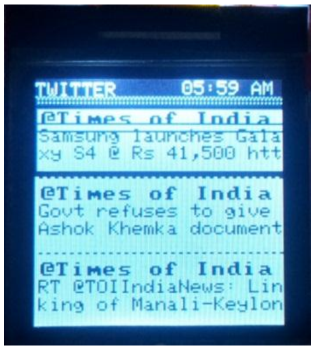


Figure 14: Home Screen

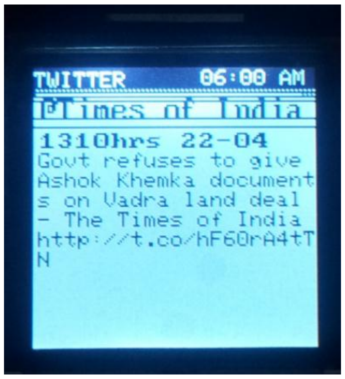
The home screen which is the main menu and is on level 1 from where various applications can be selected and opened and is the default screen when the device boots up.



```
TWITTER        05:59 AM
@Times of India
Samsung launches Gala
xy S4 @ Rs 41,500 htt
@Times of India
Govt refuses to give
Ashok Khemka document
@Times of India
RT @TOIIndiaNews: Lin
king of Manali-Keylon
```

Figure 15: Twitter Submenu

The submenu for twitter is loaded when the user selects the twitter menu on level 1. It shows the preview of 3 of the most recent tweets that were synchronized the last time

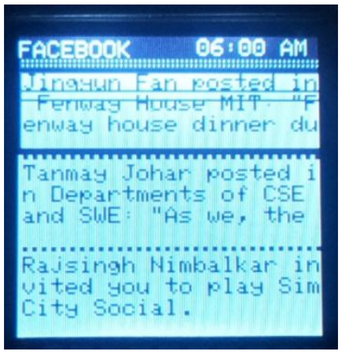

TWITTER 06:00 AM

Times of India

1310hrs 22-04
Govt refuses to give Ashok Khemka documents on Vadra land deal - The Times of India http://t.co/hF60rA4tTN

Figure 14: Twitter Full Message

The entire twitter message is loaded in level 3 when a particular tweet is selected in level 2 and shown to the user.


FACEBOOK 06:00 AM

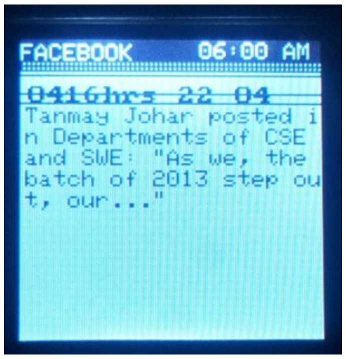
Jingyun Fan posted in Fenway House MIT: "Fenway house dinner du

Tanmay Johar posted in Departments of CSE and SWE: "As we, the

Rajsingh Nimbalkar invited you to play SimCity Social.

Figure 17: Facebook Submenu

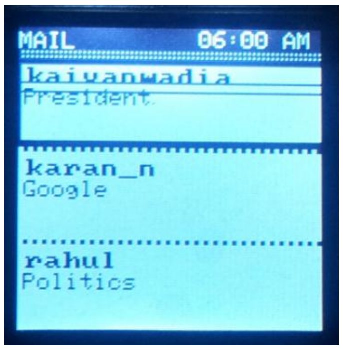
The submenu for Facebook is loaded when the user selects the twitter menu on level 1. It shows the preview of 3 of the most recent tweets that were synchronized the last time


FACEBOOK 06:00 AM

0416hrs 22-04
Tanmay Johar posted in Departments of CSE and SWE: "As we, the batch of 2013 step out, our..."

Figure 15: Facebook *Full Message*

The entire Facebook message is loaded in level 3 when a particular message is selected in level 2 and shown to the user.


MAIL 06:00 AM

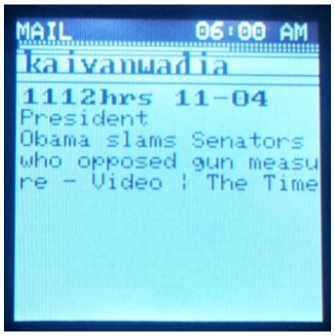
kaivanwadia
President

karan_n
Google

rahul
Politics

Figure 19: Mail Submenu

The submenu for Mail is loaded when the user selects the twitter menu on level 1. It shows the preview of 3 of the most recent tweets that were synchronized the last time



MAIL 06:00 AM
kaivanwadia
1112hrs 11-04
President
Obama slams Senators
who opposed gun measu
re - Video | The Time

Figure 20: Mail full message

The entire Mail is loaded in level 3 when a particular mail is selected in level 2 and shown to the user.

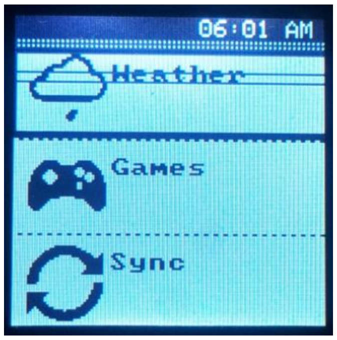
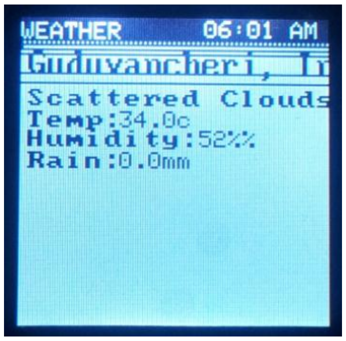


Figure 21: Second Menu Screen

The second screen in the main menu and is on level 1 from where various applications can be selected and opened.



WEATHER 06:01 AM
Guduvancheri, In
Scattered Clouds
Temp:34.0c
Humidity:52%%
Rain:0.0mm

Figure 22: Weather page

The weather in detail is loaded when the user selects the weather submenu on level 1.

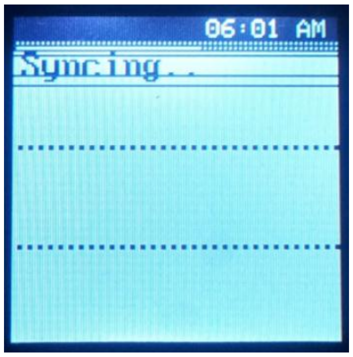


Figure 2316: Sync Screen

The sync screen is loaded when the user selects the Sync submenu on level 1 and fetches the personalized user data from the mobile phone.

SCREENSHOTS

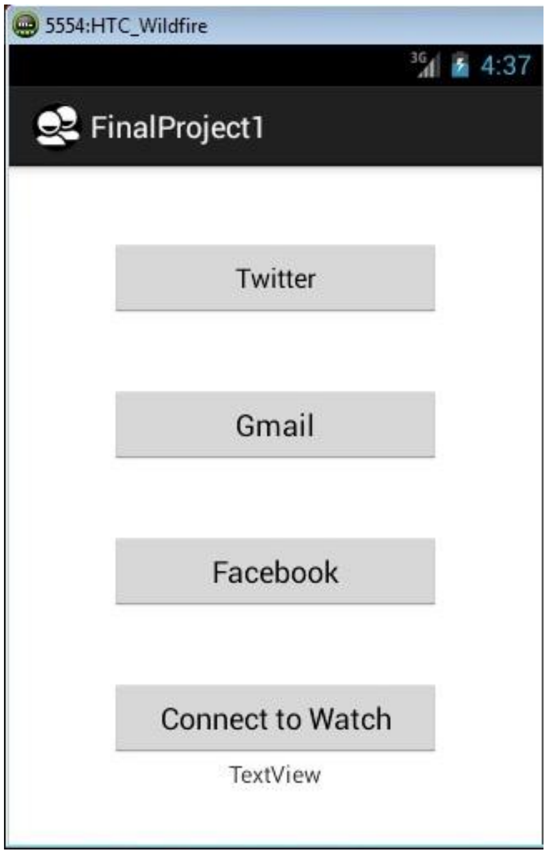
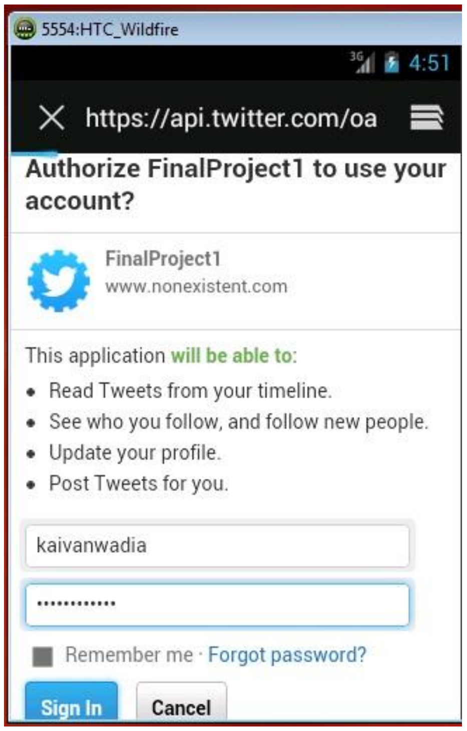


Figure 24: Android Application Home Screen

This is the home screen of the Android application which is displayed when the application is run for the first time. It displays all the features that are available and require signing in. It also has additional functionality to connect to the watch.


5554:HTC_Wildfire

4:51

https://api.twitter.com/oa

Authorize FinalProject1 to use your account?

FinalProject1
www.nonexistent.com

This application will be able to:

- Read Tweets from your timeline.
- See who you follow, and follow new people.
- Update your profile.
- Post Tweets for you.

kaivanwadia

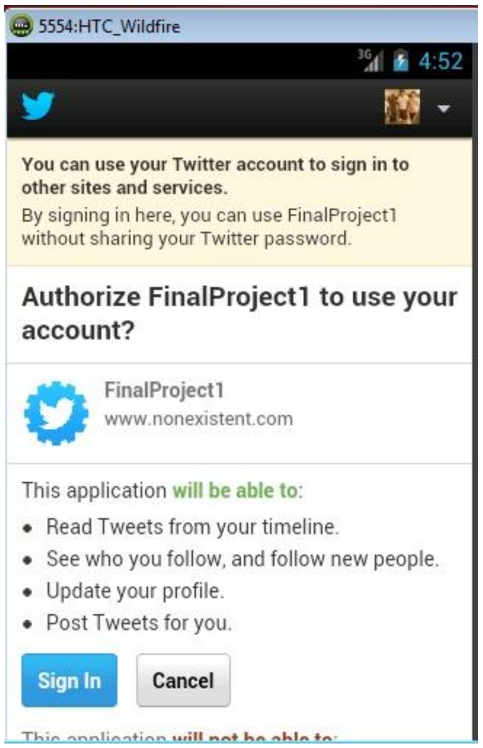
••••••••••••

Remember me · Forgot password?

Sign In Cancel

Figure 25: Signing into a feature

This screen shows the login screen for the Twitter feature. The user needs to login and grant access to his twitter feed to be able to access it on the watch.


5554:HTC_Wildfire

4:52

You can use your Twitter account to sign in to other sites and services.
By signing in here, you can use FinalProject1 without sharing your Twitter password.

Authorize FinalProject1 to use your account?

FinalProject1
www.nonexistent.com

This application will be able to:

- Read Tweets from your timeline.
- See who you follow, and follow new people.
- Update your profile.
- Post Tweets for you.

Sign In Cancel

This application will not be able to:

Figure 26: Granting access to twitter

This screen shows the authorization page of the Twitter feature. The user has to authorize the application to access its home feed.

CHAPTER 6

PERFORMANCE ANALYSIS

After completion of the prototype we ran some benchmark tests and found the following results:

	Min	**Avg**	**Max**
Computation Cycle time (micro seconds)	416	1492	9360
No of Instruction per cycle	6656	23872	149760
Computation cycles per second	2404	670	106

Table 4: Processor usage and computation time

	Max	**Usage**	**Free**
Flash (bytes)	30720	30138	582
RAM (bytes)	2048	1694	354

Table 5: Memory Usage

(in mA)	**Min**	**Max**
ATMEL ATMEGA 328	40	300
RTC(DS 1307)	.002	1.5
Bluetooth Module	3	100
SD Card	.159	200
LCD Display	2	53
Total	45.161	654.5

Table 6: Power Budget

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 Conclusion

Our aim was to create a prototype of a smartwatch capable of communicating with the world around it using existing technologies and demonstrate the feasibility that such a device can perform well in and we have been successful in creating the prototype as well as have been able to make the device connect to the internet via an internet connected device.

In the process of creating the device we had encountered various bottlenecks as well as problems related to the hardware as well as the software, most of which have been solved but some problems were also identified due to the hardware such as even though the hardware is capable of processing the data at a very fast space the internal storage is not enough to accommodate a all the software suites hence a more powerful processor such as the Atmel ATMEGA 2560 would be much more desirable and the Bluetooth module that we are using is a Class 0 device which is not identified by many other devices because of an inherent bug in the Broadcom Bluetooth stack. The only solution to these problems is the replacement of the hardware with better performing hardware.

The current system has a stable operating system within it and is capable of working as a standalone device with a few software features which we had planned about. The device is able to retrieve personal data from the internet, store it, parse it and display it when required. The device performing satisfactorily as a smartwatch and has demonstrated the off the shelf components and open source hardware and software products can indeed be used to create a device such as a smartwatch.

7.2 Future Work

In this age of technological growth and innovation the form factor of the devices is shrinking day by day and for a smartwatch the device form factor and the user interface matters a lot. The next step for us would be to refine the software end of the product as well as create a miniaturized PCB free from the problems found in the course of the project. After the PCB is ready and the software is stable we would release the software as well as the schematics to the smartwatch to the open source community so that we can let it grow and reach the next level.

REFERENCES

[1] Seiko RuputerTM - http://sv.ruputer.com/english/

[2] onHandTM PC, the world's smallest computer - http://www.onhandpc.com/

[3] Timex DatalinkTM Watch - http://www.timex.com/html/data_link.html/

[4] C. Narayanaswami and M.T.Raghunath at The Fourth International Symposium on Wearable Computers ,2000 *"Application design for a smart watch with a high resolution display"*

[5] Android - http://www.android.com/

[6] Eclipse - http://www.eclipse.org/

[7] Android SDK – http://developer.android.com/sdk/

[8] Easy Facebook SDK - http://www.easyfacebookandroidsdk.com/

[9] Twitter4j Library - http://twitter4j.org/en/index.html

[10] Google-api-java-client - https://code.google.com/p/google-api-java-client/wiki/Android

[11] Nokia 6100 LCD Display Driver by James P. Lynch

[12]Adafruit forums http://learn.adafruit.com/ds1307

[13] Seeedstudio Forums http://www.seeedstudio.com/forum/

[14]Arduino http://www.arduino.org