# CS-308-2016 Final Report

Smart Wallet


Team Members

Shivam Garg (12D020036)

Aditya Nambiar (12D070012)

Mihir Kulkarni (12D020007)

Siddhartha Dutta (120040005)

Team Leader: Mihir Kulkarni

# Table of Contents

# 1. Introduction

We spend and receive money from our wallets all the time, be it bank accounts, lending to or borrowing from friends, spending at outings, or whatever else. Keeping track of all of these is hard, and oftentimes, we're left wondering where we've spent and borrowed all our money at the end of the month.

The Smart Wallet is the solution to this problem. It is a plug and play device for your wallet which will help you keep a record of your cash flow transactions, allowing you to analyse how your spending and lending patterns vary with time and location.

Furthermore, it can act as a calculator when you need to do fast calculations while making any transactions. Moreover, it also has a theft detection system which will send you an alert on your mobile if your wallet is stolen.

# 2. Problem Statement

The aim of the project is to build a plug and play module for your wallet which can be used to

1. keep track of all cash flow transactions
2. Work as a calculator
3. Act as a theft detection system for your wallet.

The device will be placed inside your wallet. Whenever you make a transaction, you can perform calculations on this device. Suppose you paid Rs 100 for a hostel canteen treat. You will enter 100 on the keypad and press the **Send** button.

This data will be sent to the Smart Wallet app on your smartphone, which will store the amount paid (Rs 100) along with the time (using RTC) and location (using mobile GPS) of the transaction.

At the end of, say, a week, you can open the app to see what, where and when you've done your spending and lending. The app also allows you to add additional details such as the other party's name.

If you forget to take your wallet while going outside, or if your wallet gets stolen, the app recognises that the wallet is sufficiently far away from the smarphone, and immediately sends you a notification.

# 3. Requirements

## ■ 3.1 Functional Requirements

- Should be able to take input from keypad
- Should be able to display input on LCD
- Should be able to perform arithmetic operations and display output on LCD
- Should be able to keep track of current time
- Should be able to send data from to app using bluetooth module
- App should be able to show the data, along with time and location.
- Should be able to edit transaction details in app.

## ■ 3.2 Non-Functional Requirements

- Theft detection/alert system should work properly
- Storage: The wallet should be able to buffer data for as long as is necessary and send it to the app once connectivity is strong
- Time of transaction should be accurate

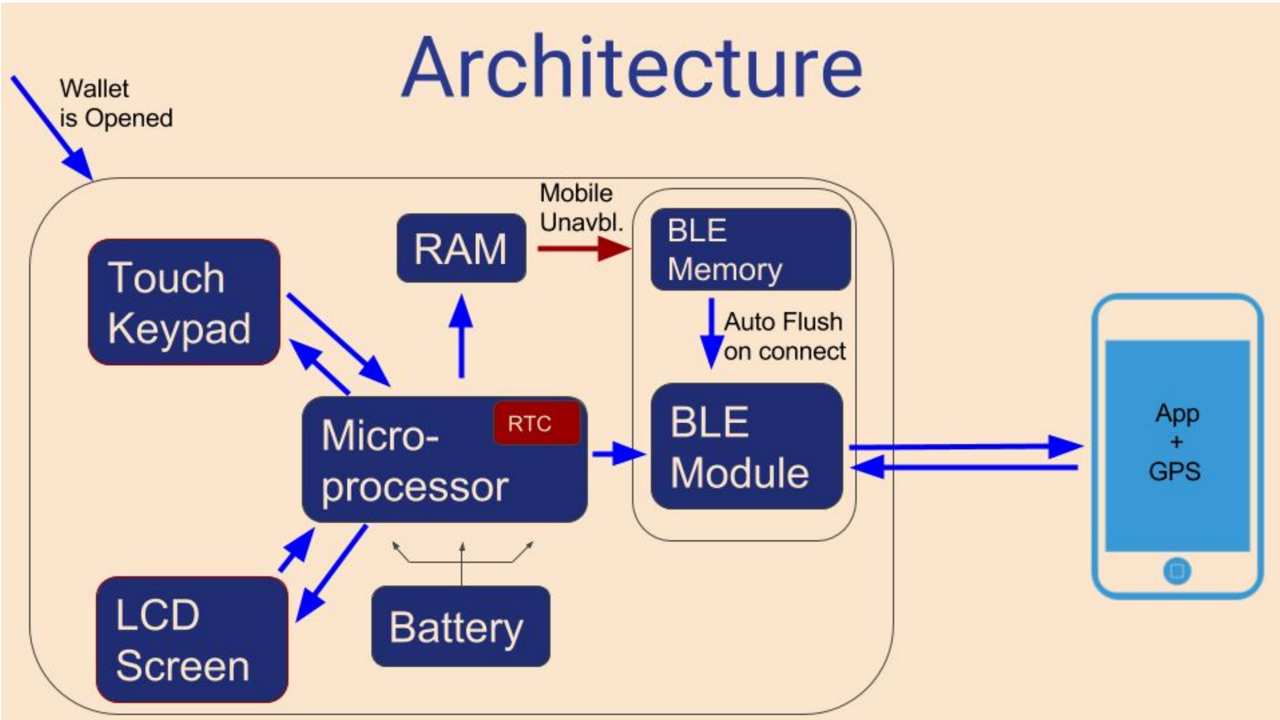
## ■ 3.3 Harwdare Requirements

- Microcontroller ( we used the Tiva C Series TM4C123G LaunchPad)
- Matrix Keypad
- Bluetooth (BLE) module
- LCD output screen

## ■ 3.4 Software Requirements

■ Android app to interface with BLE
■ Programs to run and interface the hardware components with each other

# 4. System Design

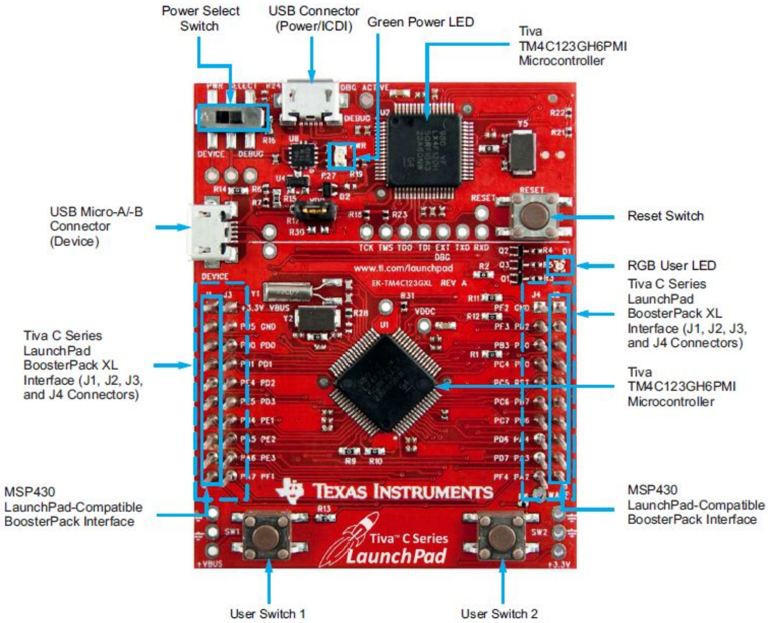



## System Design

**4x4 matrix Keypad**: This has keys for entering numbers and basic arithmetic computations, with buttons to signal what has been sent and what has been lent. this interfaces with the microcontroller.


**LCD**: We use JHD162A LCD display. It takes the keypad input from the microcontroller GPIO and displays it on screen.

**Microcontroller**: We use Tiva C Series TM4C123G microcontroller.  The microcontroller connects to the keypad and the LCD screen, interfacing between them. All numbers input on the matrix keypad are stored in its memory(until computation is over), and all calculations are also performed by the TIVA microcontroller.



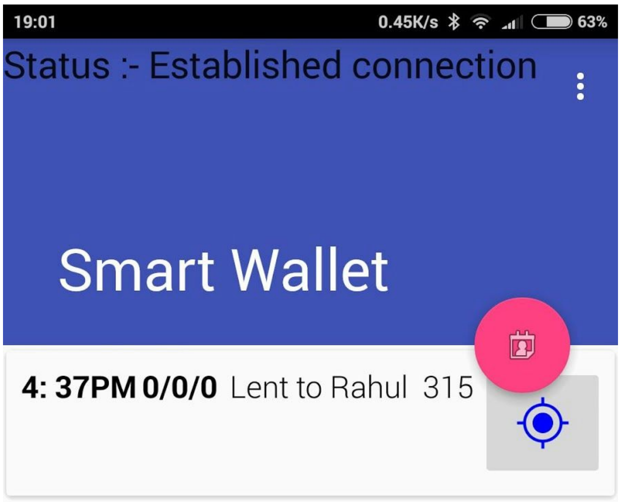

**Tiva C Series TM4C123G LaunchPad Evaluation Board**

**BLE**: We use HC-05 bluetooth module. The BLE module interfaces with the microcontroller UART. Once **Send** is pressed, if it is possible to interface with the smartphone app at that point in time, the BLE sends the amount immediately along with a timestamp (taken using the RTC). Else, this data is buffered inside the BLE, and automatically flushed as soon as a connection is established


.

**App**: The app receives data from the BLE, and adds an entry in a list it maintains, detailing the amount, timestamp and location (obtained via GPS) of the transaction executed. It allows you to add/edit details once the transaction is done.

If the Bluetooth connection is lost (due to the wallet being too far from the phone, or otherwise), the app notifies you and makes the phone vibrate.


19:01
0.45K/s
63%
Status :- Established connection
Smart Wallet
4: 37PM 0/0/0 Lent to Rahul 315

**The device**


1
2
3
A
4
5
6
B
7
8
9
C
*
0
#
D

# 5. Working of the System and Test results

As detailed in the code, the matrix keypad uses a mechanism that polls each row to see if a button has been on that row has been pressed (the pin for that row is LOW), before writing HIGH to each column and checking which column caused the row to go back to HIGH. This requires timing delays, and we introduce a three state FSM to be able to recognise input (long and short keypresses) at speed.

The input, once entered, is received by the microcontroller, which does on-the-fly computation on it and stores the result in a variable. At the same time, every input is being displayed on the LCD screen. When we press **Send**, the final computation, if any, is performed, and the number, along with timestamp and information regarding whether the transaction is a credit or debit, is prepared to be sent via the BLE. If there is an active connection, the BLE sends this data to the app directly. Else, this data is queued in the BLE memory, where all of it is flushed to the app as soon as a connection is established.

The app receives the data, and adds an item in a list it maintains, which details the time, location and value of a transaction. Location is obtained as soon as this data is received via GPS. It is possible to edit these details later.

If there is a connection loss, this is detected by the smartphone's bluetooth module (which maintains connection with paired devices). The module then informs the app about this connection loss. The app sends a notification when this happens.

**Testing**

Matrix keypad:

We checked for correctness of input of digits, arithmetic operations, and send/receive buttons (which requires state maintenance in the microcontroller). We discovered during testing that one requires pull-down registers to implement this, which TIVA didn't have, so we had to insert them ourselves. We tested speed of this by varying the polling delays and introducing a state machine for enabling faster input.

LCD:

Tested for correctness of display of digits without delay on the screen, and old digits vanishing when new data is to be input. We had earlier used two interconnected TIVA pins for the LCD and the keypad, which caused issues in interfacing the two.

BLE:

1. Connection establishment
2. Breaking of connection when distances were large, what distance this happened at

3. Correctness of data transmitted

App:

1. We tested correctness of all data(amount, given/taken, timestamp) sent from the hardware to the app
2. GPS location checking
3. Correctness of editing details
4. Persistence of data after closing the app
5. Attempted sending of data without connection, leading to buffering and sending of data once connection was established. We were successfully able to buffer 10-15 transactions
6. Buzzing and notification on connection loss (on going too far from hardware, ~8 metres)

# 6. Discussion of System

a) What all components of your project worked as per plan?

### Functional Requirements

- Input from keypad: After much debugging, delay optimization, and introducing an FSM for key-press state, this did work as per plan
- Output on LCD, and refreshing this with speed
- Arithmetic operations
- Sending data from to app using bluetooth module
- App shows the data, along with time and location.
- Should be able to edit transaction details in app.

### Non-Functional Requirements

- Theft detection/alert system should work properly
- Storage: The wallet should be able to buffer data for as long as is necessary and send it to the app once connectivity is strong
- Time of transaction should be accurate: This works upto a certain limitation, because in our prototype, the RTC is turned on only once we connect the module to a laptop

b) What we added more than discussed in SRS?

- Used the TIVA Board’s real time clock to send timestamps, to get an accurate transaction time, as compared to doing so in the mobile
- Using BLE memory
- Delay optimization and state machine introduced into the keypad to ensure a seamless input experience
- Presented the product in an attractive encasing, hiding wires and enabling easy use.

c) Changes made in plan from SRS:

- Implemented own calculator rather than reverse-engineering an existing one: This is because the voltage thresholds on the existing calculators were low, and we weren't able to get robustness using it.
- Used the memory in the BLE instead of using an external Flash Memory: Given that we had the BLE existing already.

## 7. Future Work

- Analytics to give the user a clearer picture of their transcations over a period of time, and recommending when and where to spend to cut down costs
- Using analytics from various users to predict aresa where tbefts are likliers
- Notification when cash is low (and possibly, when an ATM is nearby)
- Making the product thinner and more user friendly (we currently only have a prototype)

## 8. Conclusions

This is a simple embedded system that interfaces with a mobile app to track user transactions over a time period. Given that keeping track of expenses is essential for all of us, this is something that a lot of people can find use for. We aimed to create something easy to use and helpful for a lot of people, and the app+smart wallet combo we've come up with seems to do just that.

We've detailed what can be done ahead in the previous section. However, location and time based tracking of spending patterns seems to be something very amenable to large scale analysis, and could greatly help users in gaining more information about their spending patterns and what to do ahead.

## 9. References

[1]TIVA Workbook
https://www.cse.iitb.ac.in/~erts/html_pages/Resources/Tiva/TM4C123G_LaunchPad_Workshop_Workbook.pdf

[2] Using the matrix keypad
http://embedjournal.com/interface-4x4-matrix-keypad-with-microcontroller/

[3] Bluetooth
http://www.tec.reutlingen-university.de/uploads/media/DatenblattHC-05_BT-Modul.pdf

[4] LCD

http://www.binaryupdates.com/wp-content/uploads/JHD162A-LCD-with-LPC2148-ARM7.pdf

[5] TIVA datasheet

http://www.ti.com/lit/ug/spmu296/spmu296.pdf

[6] Android

http://www.tutorialspoint.com/android/