INTERACTIVE PHONE COVER

ECEN 5023 Low Power Embedded Design Techniques

SHARANJEET SINGH MAGO
University of Colorado Boulder

Table of Contents

Overview

Nowadays the mobile phones are one of the most used gadgets and almost every person uses one on daily basis. Since most of the mobile phones present in the market have a touch interface, it might be interesting to see the touch interface on both the sides of the phone which gives better accessibility to the mobile phone.

The interactive phone cover communicates with the mobile phone to provide input using a capacitive touch pad on the back of the phone cover. The capacitive touch on the back of the phone cover gives the user multiple options to give input to the system which enhances the user experience. The cover interacts with the phone using Near Field Communication which also provides a medium to charge the device wirelessly. The touch interface provides additional features such as haptic feedback when the user reaches the end of the page while scrolling.

Purpose

The interactive phone cover helps the user to interact with the phone using a single hand and gives the user the complete view of the screen while surfing through the content on the screen. The user will be able interact with the phone using certain gestures like scrolling up or down, zooming in or out, and some other gestures like tap or certain shapes.

Technical Description

- The device has a microcontroller which acts as the central processing unit to interact with all the peripherals and communicate the data from the capacitive touch to the phone.
- The communication with the mobile phone will be done by NFC. NFC connects automatically in a fraction of a second.
- NFC also provides as a medium for energy harvesting.
- The device has an in-built battery to power the microcontroller and peripherals.

Use Case

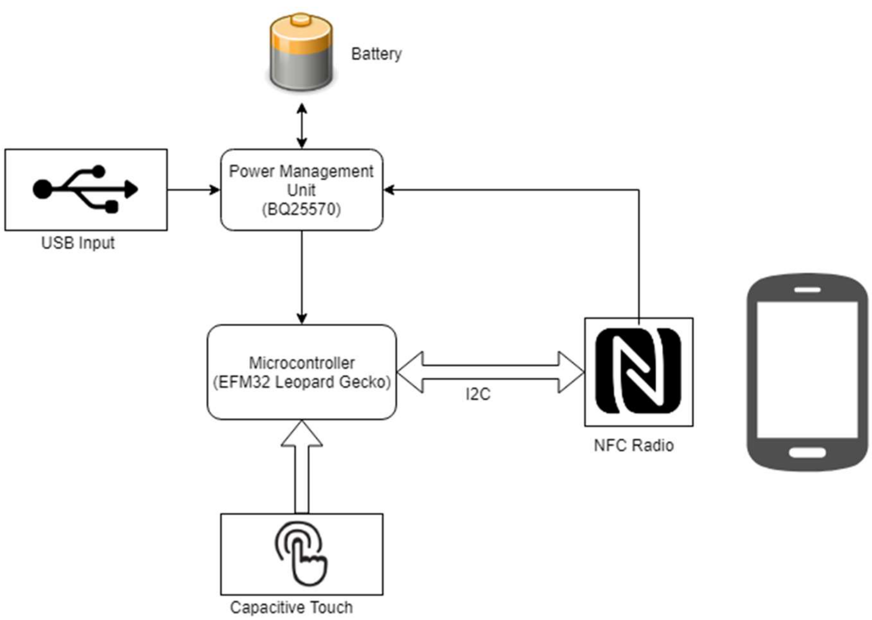
The touch input on the phone cover needs to be active all the time to detect any kind of touch input that the user might give. The NFC tag can be made to go into sleep mode and only wake up when the user gives any input from the touch pad.

The NFC will also be used for energy harvesting. And there will be a separate mode where the battery will be charged using the NFC.

Since the phone cover will be used along with a mobile phone, it should also have a battery life similar to that of the mobile phone. Therefore, the battery should run approximately 1 day.

To find out the On-Duty cycle, we can refer to the Screen-On time on our mobile phone as that is almost the same time when the user might want any kind of input from the phone cover. Therefore, we can assume the On-Duty cycle to be approximately 30%.

Hardware Block Diagram



Key Components

Leopard Gecko EFM 32

- The leopard gecko provides different energy modes which can be utilized to make the device run on low power.
- The leopard gecko has onboard capacitive touch buttons which helps to develop and test the firmware on the development board before the actual device. This also tells that capacitive touch buttons can be interfaced with leopard gecko.

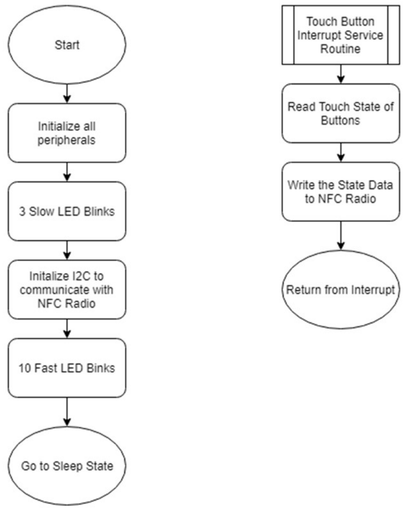
NFC Tag NT3H1101

- The phone cover is always in contact with the phone which is suitable to use near field connection.
- The NFC tag can also be used as an energy harvesting tool.
- The NFC tag communicates with MCU using I2C protocol.
- The voltage of operation is 1.7-3.6 v and it derives a current of 155 uA.

Capacitive buttons AT42QT1012

- The capacitive buttons are used as the input for the device.
- The capacitive buttons can be used at EM2 mode with the leopard gecko.
- The voltage of operation the buttons is 1.8-5.5 v.
- The capacitive buttons draw a current of 124uA.

Software Flow Chart



List of Commands

When the system is powered up it follows the following sequence

1. 3 slow LED blinks indicate that all the peripherals have been initialized
2. 10 fast LED blinks indicate that the MCU is able to write to the NFC Radio
3. Now the system waits for user input from the capacitive touch buttons.
4. Whenever a touch is detected on the capacitive touch buttons, the MCU reads the states of the buttons and writes that to the NFC Radio.

Planned development schedule and when tasks were completed

Week 1

- Coming up with an idea for project

Week 2

- I worked on studying the basic energy modes that the Leopard Gecko supports.
- Interfaced the NFC NTAG I2C development kit to see how it works.
- Shortlisted the different type of capacitive buttons for the input and went ahead with a normal breakout one for now.

Week 3

- Selection of Energy Storage Element.
- Selection of PMU.
- Calculations of currents at different energy modes.
- Introductory coding for Leopard Gecko.

Week 4

- Symbol selection for components
- Basic schematic planning
- PCB Layout Assignment
- Introductory coding for leopard gecko and NFC Tag

Week 5

- Symbol selection for components
- Started schematic design
- Introductory coding for leopard gecko and NFC Tag

Week 6

- Schematic design for Power Circuit
- Debugger Circuit
- Introductory coding for leopard gecko and NFC Tag

Week 7

- Schematic design
- Selecting ESD diodes

Week 8

- Schematic Review
- Start the layout of the board

Week 9

- Schematic Review
- Layout of the board
- Bills of material ready

Week 10

- Layout of the board
- Layout Review
- Ordering of components

Week 11

- Placing the order for the PCB
- Order remaining components

Week 12

- Start the assembly of the board
- Verify the circuit

Week 13

- Firmware Development
- Testing

Week 14

- Integration of all components
- Testing

How target microcontroller/SoC will be programmed

Features added to the schematic to enable programming of the MCU

- Debug connector (ARM 20-pin Debug Header)

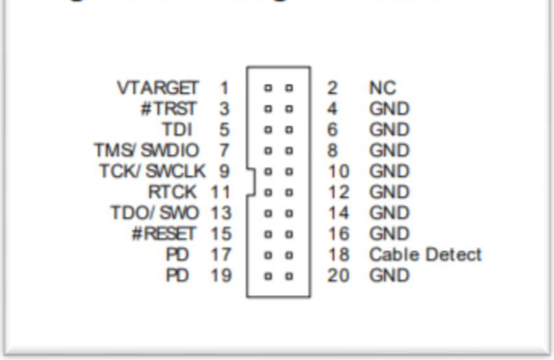


Figure 9.3. Debug Connector

The following pins will be connected to the MCU

SWDIO (Pin7) - PF1 on MCU

SWCLK (Pin 9) - PF0

SWO (Pin 13) - PF2 or PC15

RESET (Pin 15) - RESETn

Process to program the target board

- Programming the MCU (target board) can be done by connecting the board to another Leopard Gecko board (debugger board). The debugger board is connected to the PC.
- The debug mode on the debugger board is set to OUT.
- Move the switch to the AEM or Debug position.
- Launch the Simplicity Studio V4 on the PC and select the debugger from the device list on the launcher page.
- On the launcher page, find Debug Mode and click change.
- On the Adaptor configuration tab, Change the Debug Mode to OUT.
- On the Device Hardware Tab, In the Target part, type the part number of the external device.
- Make a new project in simplicity studio and type the code.
- Compile the code and check for any errors.
- Build the code to get the hex file and download it to the target board.

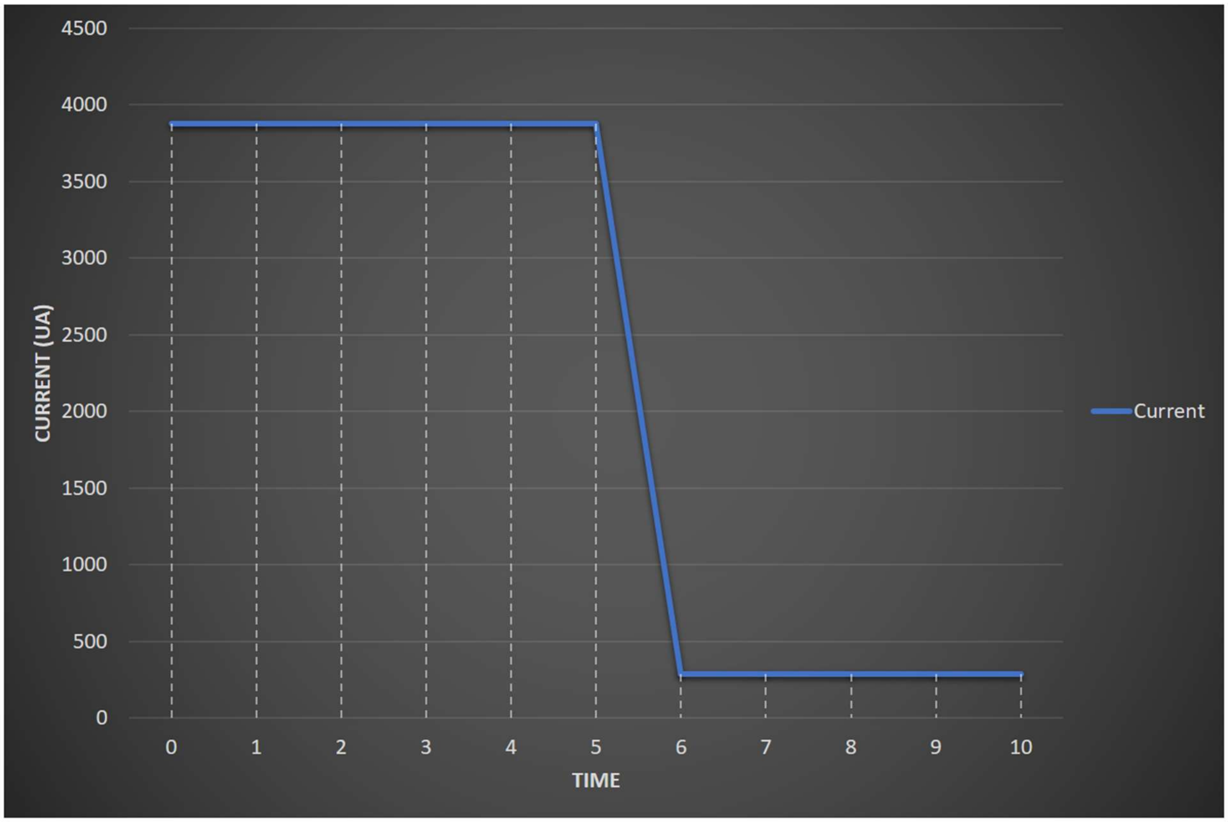
Current Profile and Energy Modes

There will be 2 energy modes for the operation of this device

1. Active Mode
   In the active mode, the MCU runs in the EM0 energy mode. The CPU is active in this mode. All the peripherals also run in this mode.
2. Sleep Mode
   In the sleep mode, the MCU runs in the EM2 energy mode. The capacitive touch sensors can operate in this energy mode. And thus, will be used to wake up the CPU and switch to active mode.

Peripheral	Current in sleep mode	Current in active mode
NFC Tag	155 uA	155 uA
Capacitive Touch Input	124 uA	124 uA
MCU	4 uA	3600 uA
Total	283 uA	3879 uA = 3.789 mA

The device will stay 50% of the time in active mode and 50% of the time in sleep mode. This is assumed based on the usage of our mobile phones. The screen on time of my mobile is approximately equal to 50% and since the device will be used as a phone accessory, the device will need to be in the active mode for 50% time as well.



Plot of current vs time assuming the system runs for 10 time periods

Current = 0.5 * 283 uA + 0.5 * 3879 uA = 2081 uA = 2.081 mA

Energy Storage Element Selection

Our mobile phones have an approximate battery life of 1 day or 24 hours. So we would expect our device to run for at least 1 day.

Required Capacity for battery = 2.081 mA * 24h = 49.944 mAh

Required Capacity for Super Cap

$I = C * dV/dT$

$2.081 \times 0.001 = C * (V_{work} – V_{min})/dT$

$2.081 \times 0.001 = C * (3.3 – 2.0)/(24*60*60)$

$C = 138.306461$ F

Since a capacitor with this much big value is not available, we choose to go ahead with a battery as a power supply.

Battery Selection

Type : Polymer Li-ion Rechargeable Battery

Model : DTP603443

Specification : 3.7V / 850mAh

Let us keep a margin for our battery capacity and let us assume our required capacity to be 100 mAh.

The battery selected has a capacity of 850 mAh.

So our device can run on this battery for 8-9 days. Or Approximately 1 week.

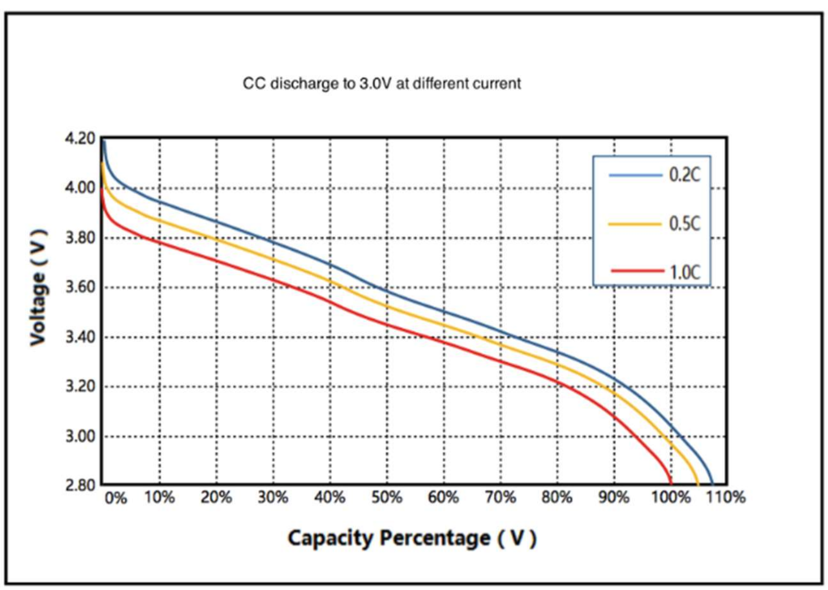
The battery is good for 300 recharge cycles. Which gives us a battery life of 300 weeks = 300 / 52 years = 5 years 9 months.

C Rate and Battery Solution

C-rate of the battery: 0.2C

Peak Discharge Rate: 1C

Standard Charge	Charge with 0.2C(170mA) up to Limited Voltage , Charge with limited Voltage up to end-of-charge current.
Standard Discharge	Using 0.2C(170mA) constant current discharge to the Discharge Cut-off Voltage.
Maximum Continuous Charge Current	1C (850mA)
Maximum Continuous Discharge Current	1C (850mA)



Discharge at different Current

Lowest Nominal Voltage of the battery: 2.8V

Discharge Cutoff Voltage of the battery: 3.0-3.2V

The circuit can operate in the voltage range: 1.8 – 3.3V

The LDO can give output in the voltage range: 1.8 – 3.3V

The LDO input voltage range is: 1.8 – 5.5V

The battery operates in the voltage range: 4.2 – 2.8V

For this project, the output of the LDO is fixed at 3.0V

So, the operational range of the battery is 3.0 – 4.2V

The dropout voltage for the LDO is 105mV, which might cause the output voltage of the LDO to drop at around 2.9V. But this does not cause any problems in operation as the circuit can operate till voltages as low as 1.8V.

PMU Selection

Device : TPS797 LDO

I out (max) : 0.05 A = 50mA

Vin (min) : 1.8V

Vin (max) : 5.5V

Vout (min) : 1.8V

Vout (max) : 3.3V

Operation voltage for different components

NFC Tag : 1.7 – 3.6 V

Capacitive Touch : 1.8 – 5.5 V

MCU : 1.98 – 3.8 V

Peak Current Requirement : 3.879 mA

The voltage and current requirements are satisfied by the chosen LDO.

We use LDO as our Power Management Unit. LDO Provides output with low noise.

This device provides a current of 50mA. And peak current required for our device is 3.789 mA.

Change in PMU Unit

BQ25570 Boost Charger and Buck Convertor will be used instead of TPS797 LDO.

This change has been made because the BQ25570 provides an inbuild charging circuit for the battery along with a buck convertor to give the power output.

PMU simulation results

PMU Circuit Specifications

Vin = min = 3.1V

Vin = max = 4.2V

Vout = 3.0V

Current in sleep mode Isleep = 283uA

Current in active mode Iactive = 3.789mA

1. What is the voltage ripple of your design before the step function load?

    a. At Vin = min

        No voltage ripple.

    b. At Vin = max

        No voltage ripple.

2. Does the voltage and its ripple before the step load meet the IC specifications of your circuits in terms of specified ripple or minimum voltage of the ICs?

    a. At Vin = min

        Yes, the output voltage is 3.0V which is the desired output voltage.

    b. At Vin = max

        Yes, the output voltage is 3.0V which is the desired output voltage.

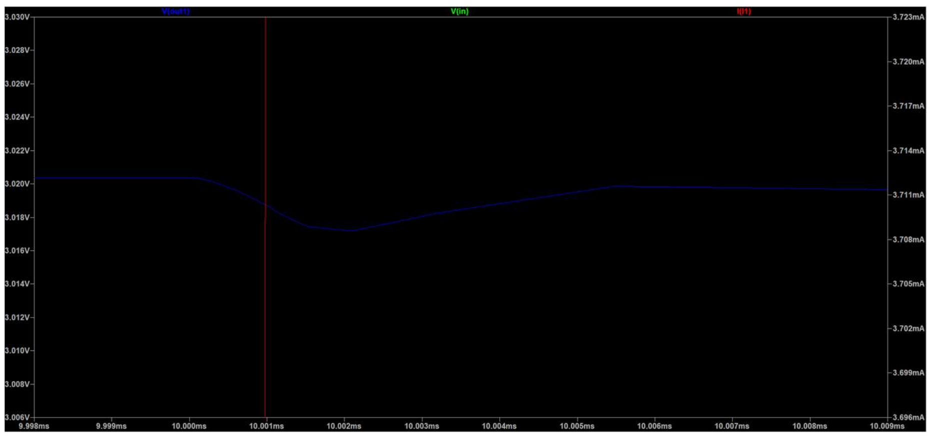
3. Does the output voltage dip when the dynamic step load is added? What is this minimum out voltage of the power supply or Vdd of the system when the power supply voltage is drooping?

The voltage dips if the current increases with the step and the voltage rises if the current decreases with the step.

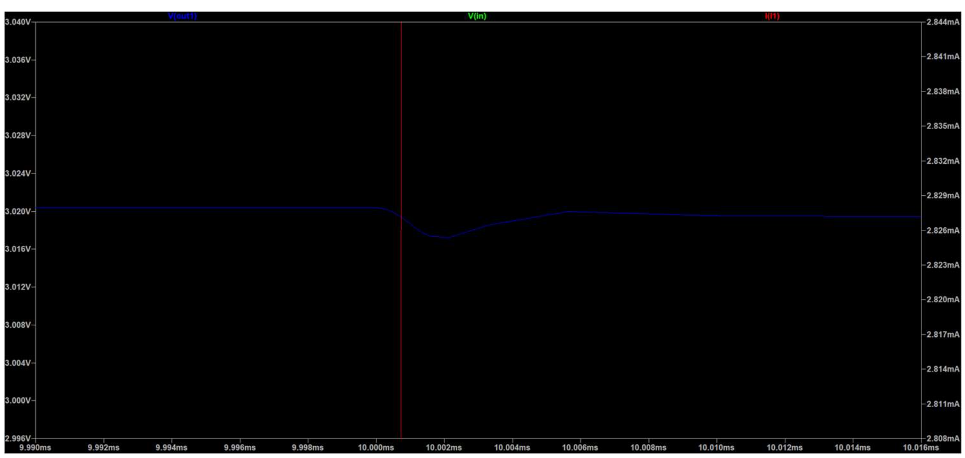
Vout = 3.02035V

Dynamic Step Up (Sleep to Active Mode)

a. At Vin = min
3.0171V



b. At Vin = max
3.0171V

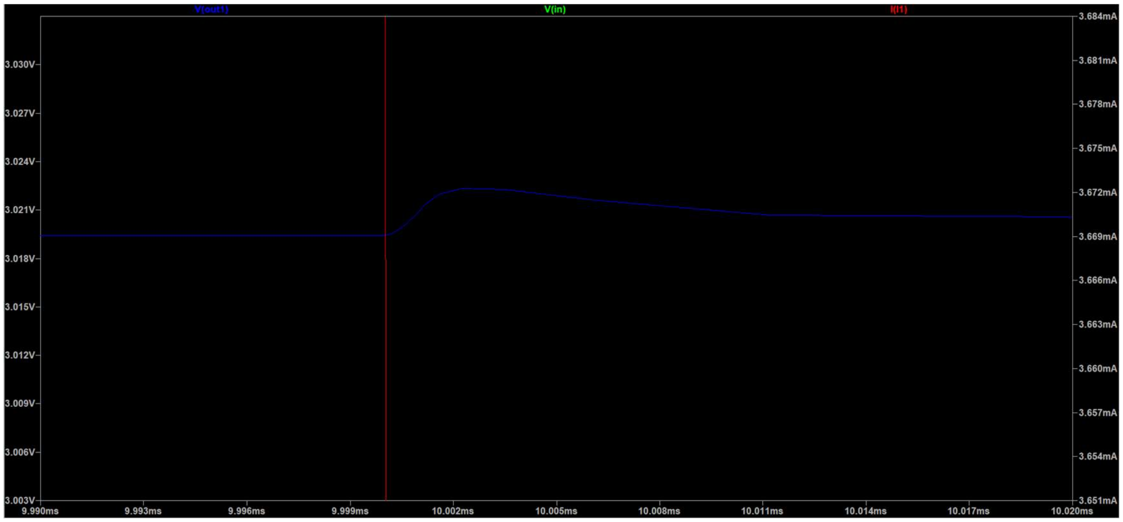


	Vout min	Vripple = Vout - Vmin
Vin = min	3.0171V	3.25mV
Vin = max	3.0171V	3.25mV

Dynamic Step Down (Active to Sleep Mode)

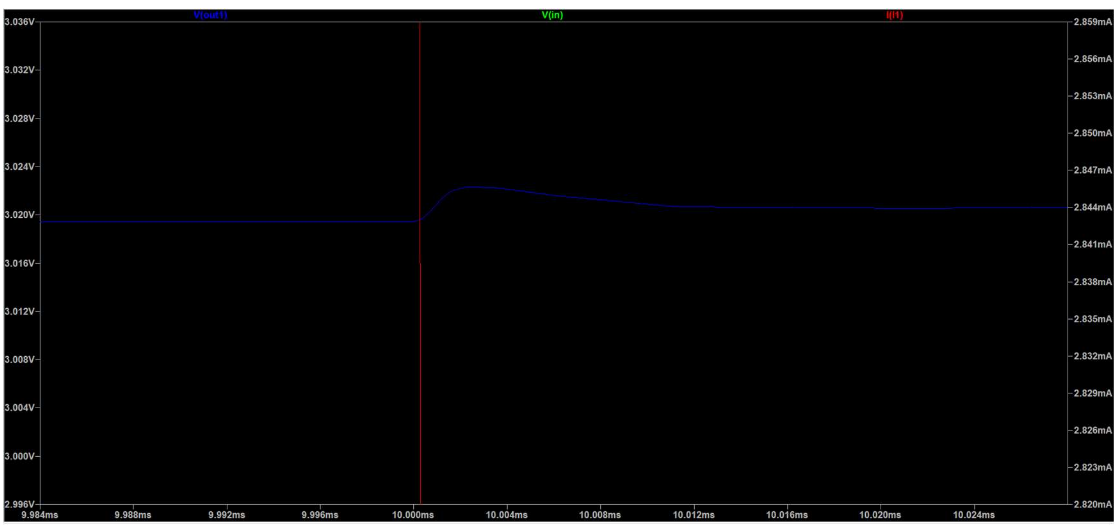
a. At Vin = min

3.02235V



b. At Vin = max

3.02231V



	Vout max	Vripple = Vout - Vmin
Vin = min	3.02235V	2.00mV
Vin = max	3.02231V	1.96mV

4. Does this minimum voltage due to the current load step meet the IC specifications of your circuit in terms of specified ripple or minimum voltage of the ICs?

a. At Vin = min

i. If not, add additional capacitance on the output, and rerun to determine if the voltage droop can be reduced to meet the specifications of the ICs

ii. Or, change the regulated output setting

iii. Please provide screen shots of each simulation and specify the change in output capacitance from one simulation to the next

b. At Vin = max

i. If not, add additional capacitance on the output, and rerun to determine if the voltage droop can be reduced to meet the specifications of the ICs

ii. Please provide screen shots of each simulation and specify the change in output capacitance from one simulation to the next

Yes, the output satisfies the IC specifications for the circuit.

5. If changes were made to meet the system requirements, you must re-simulate and provide data for both before and after the circuit change

6. Does the design meet your system and component requirements?

Yes, the design meets the system and component requirements.

Decoupling capacitor selection and back up data

Based on PMU/Capacitance simulation assignment, the ripple is very less and does not require a bulk capacitor as the voltage output satisfies the requirement.

The recommended capacitor of 1uF is connected by default which will be used for the circuit.

Capacitor details

Digi-key Part Number: 1276-6524-1-ND

Description: CAP CER 1UF 16V X7R 0603

1µF ±20% 16V Ceramic Capacitor X7R 0603 (1608 Metric)

As per the datasheet instructions, we use X7R capacitor because it has better stability across temperature and bias voltage.

Will an external source be required to program the MCU?

Alternative Charging Method / Backup solution to energy harvester (NFC).

USB Charging input, this will also serve as the jump start power supply

Maximum charging current allowed by the PMU Circuitry

285 mA

Maximum charging current allowed by the energy storage unit

850 mA

Standard charging current of the energy storage unit

170mA

What will the maximum current of Jump start power source be set to?

The input of the PMU can take maximum input power of 510mW.

At 5V, the input can take maximum current of 0.510/5 = 0.102A = 102 mA.

Therefore, the maximum current of the Jump start power source should be set to 102 mA at 5V.

Where will the jump start power and ground signals connect to?

Input of PMU (VIN_DC and VSS of PMU IC)

Current for programming of the MCU flash

7 mA for both Erase and Write

Current supply from energy storage element and PMU

Energy Storage element : 850 mA (Standard Discharge current: 170mA)

PMU : 110 mA

The PMU limits the current supply to the MCU. Therefore, the current that can be provided is **110mA**.

Connection points to enable external power to MCU Portion of the board

A jumper will be placed between the power and the MCU portion of the board to supply the external power to the MCU.

Test Points

Signals with test points

Power Circuit

- Ground
- Vin
- VBat
- Vout

MCU Circuit

- Ground
- Vdd
- I2C SDA (NFC)
- I2C SCL (NFC)
- FD (NFC)
- Vout (NFC), for energy harvesting
- OUT Pin from Touch Sensor

Should there have been more test points?

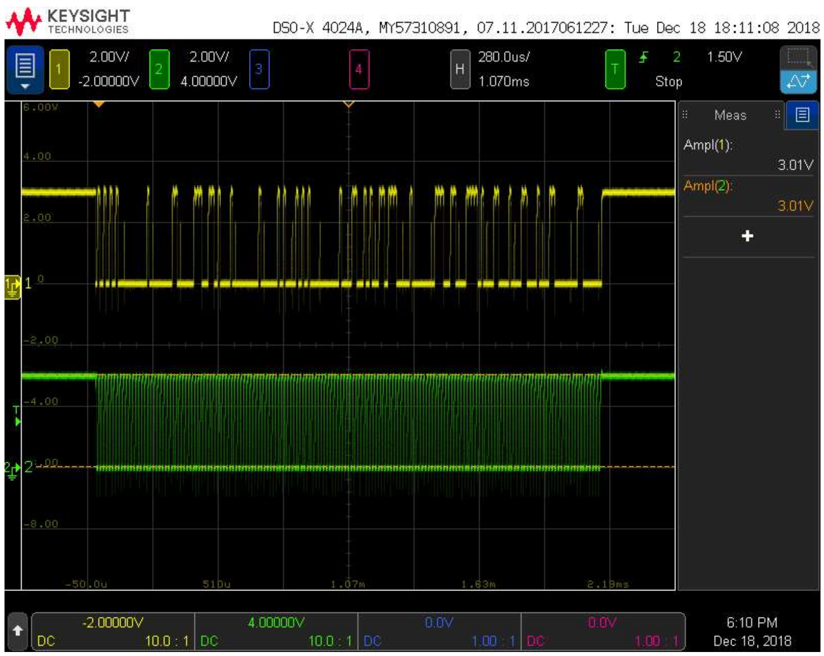
The test points that were placed on the board were sufficient to test and debug the board.

Verification Report

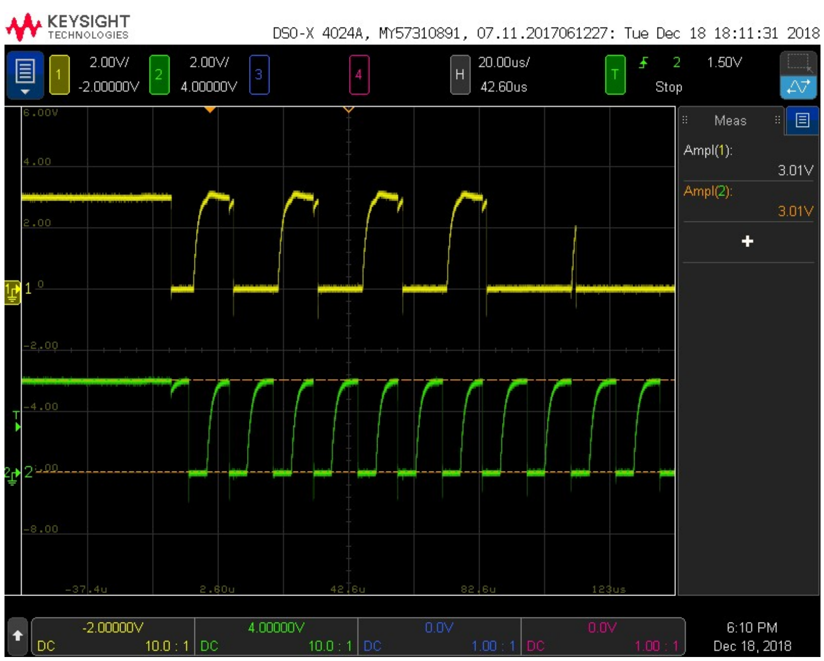
Detailed Verification Report has been attached as an Excel Sheet

Signal Quality Analysis of Key Signals

I2C Signals for communication with NFC Radio (The green signal is SCL and The yellow signal is SDA)

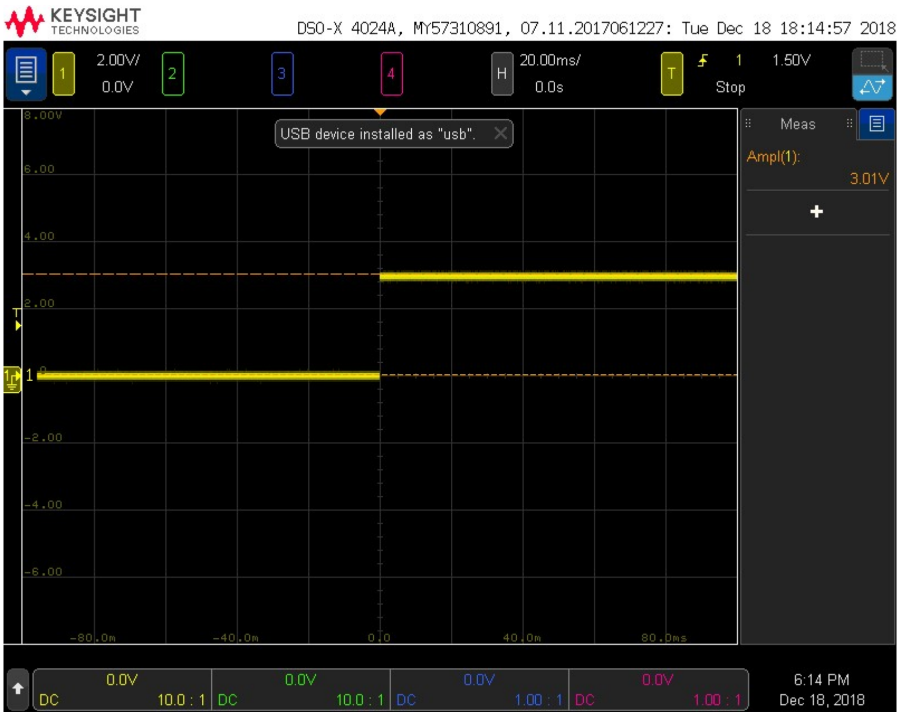


Zoomed out view of the signals

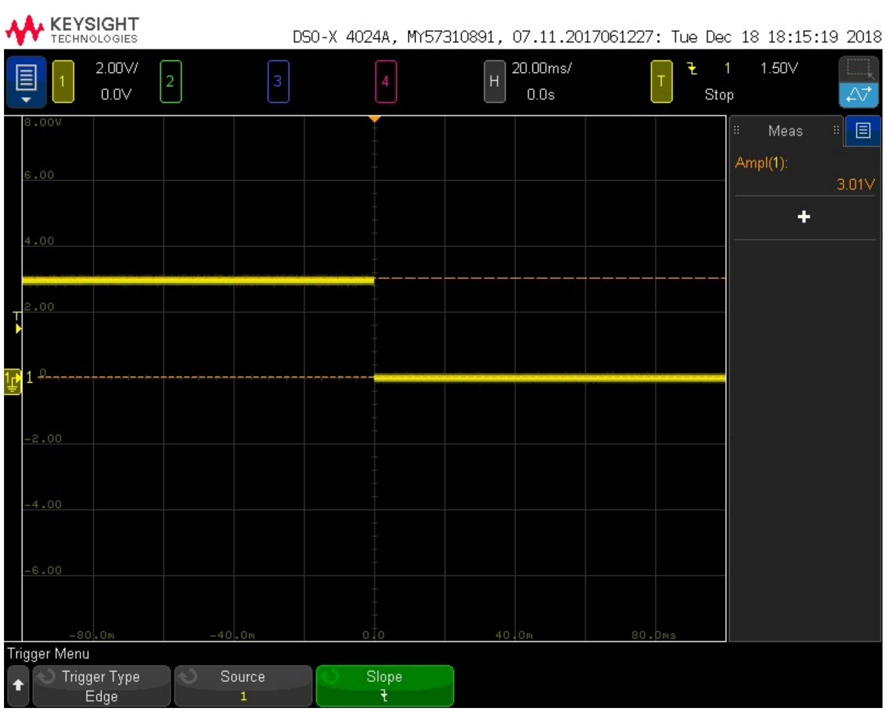


Zoomed In view of the signals

GPIO Read Signals from Touch Buttons



Rising Edge Signal



Falling Edge Signal

What were the difficulties encountered on the project?

1. While creating the footprint for the MCU, I had to create the ground pad for connection beneath the MCU. I placed the vias but missed placing a solder pad and realized this when the board came back after fabrication.

   Solution: Placed a little bit more solder paste than required at the vias and then did the reflow.

2. The dimensions of the Reset button didn't match the dimensions of the footprint.
   Solution: Found a SMD reset button with similar dimensions and bent its pins to make it through hole.

3. I accidently damaged the NFC Radio by supplying 5V instead of 3.3V.
   Solution: Professor Keith Graham had some extra NFC Radios.

4. My initial component for Power Management Unit was LDO but then I realized that I need to have something for charging circuit for the battery.

   Solution: Replaced the LDO with BQ 25570 buck-boost convertor.

Summary of functionality of final project

The final project has 2 main parts,

1. The Power Management Unit
2. The MCU

The Power Management Unit has one power input port which can be connected to USB or NFC, one port for battery and the third output port to power the digital plane. The PMU can function in the following ways

1. The battery is connected, and the output is given to the digital plane (No USB or NFC)
2. The battery is connected, USB connected to charge the battery, and the output is given to the digital plane.
3. The battery is connected, NFC connected to charge the battery, no output for the digital plane as NFC does not provide enough current.
4. Battery disconnected, USB connected, and output given to digital plane.

The MCU section of the board is responsible to collect data from the touch buttons and write it to the NFC antenna. When the board is powered up it follows the following sequence

1. 3 slow LED blinks indicate that all the peripherals have been initialized
2. 10 fast LED blinks indicate that the MCU is able to write to the NFC Radio
3. Now the system waits for user input from the capacitive touch buttons.
4. Whenever a touch is detected on the capacitive touch buttons, the MCU reads the states of the buttons and writes that to the NFC Radio.

Lessons Learned

1. Using logic analyzers to probe any high-speed signals helps a lot in debugging.
2. ENIG finishing for PCB boards provides better corrosion resistance, and better shelf life as compared to HASL.
3. Initially perform testing using evaluation kits before working on actual layouts.
4. The datasheet for BQ25570 states that input voltage can be as high as 5.5V but the component heats up if input voltage is more than 4.5V.
5. Verify component footprints which are taken from online sources.
6. A lot of routing when doing the board layout is reduced by effectively placing the components.