LOW POWER EMBEDDED DESIGN
FINAL PROJECT REPORT

Team Name: WearTech

Team Mates:

Sanjana Kalyanappagol
saka2821@colorado.edu

Shekhar Satyanarayana
shsa5563@colorado.edu

Overview of the Project

A wearable ring/cap which is used to take short notes and control other devices through the movements of the user. The data is further sent to the application and trained using a Machine learning algorithm. The movements of the user are monitored using an accelerometer and it is also controlled using the tap sense of the accelerometer. The device is a prototype of a smart wearable technology which also obtains the ambient temperature and relays it to the application to display the ambient temperature.

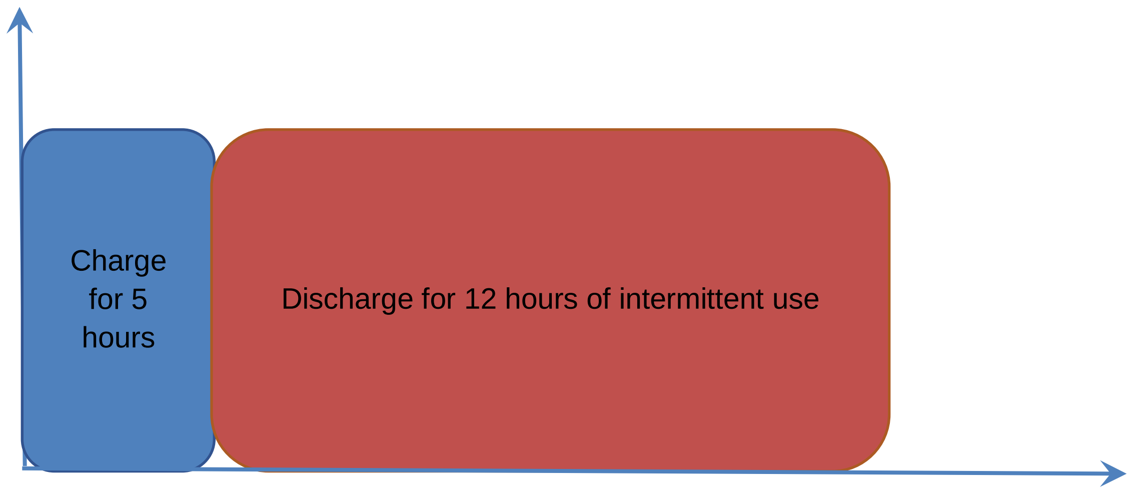
What is the problem that this project is attempting to solve? How is this project going to solve this problem?

A wearable ring/cap which is used to take short notes and control other devices through the movements defined in the Mobile App.
With the advent of technology, it is bane to open your mobile/laptop/book to take a note of something important that strikes you all of a sudden or take a phone number of someone you meet unexpectedly. Say you are in a harsh environment, what if you had a wearable device with you all the time as a look-out for such situations.
In the Corporate world, say you are in a meeting and you need to take notes, rather than using notes/pens and disturbing others you can easily take notes of all the important points you need. Say in an unanticipated meeting, wouldn't you feel more secure, safe and confident with our device always at help.
In the field of Software Development & maintenance, one can face a lot of issues/bugs to be fixed within due dates. And with that constantly in mind, Most of us get solutions to the same when we are involved in our day-to-day activities such as driving, eating, talking with a friend, etc. Hence we would end up looking for a pen & paper, if not we would waste time memorizing the solution we had just found unless we have this device around which can keep a record of every thought that you want to be saved at any point of time.
Now with the buzz of IoT we have seen a lot of devices which could communicate with our home appliances. We provide a solution with the new Bluetooth Mesh Network to control other home appliances in addition to the Ring/Cap a single device which acts as a note taker.

The end-applications to such a device is enormous, it can be used to define the movement of a wheelchair, or used by old people/patients to communicate.

Use case described

Use Case Model to charge energy storage device:

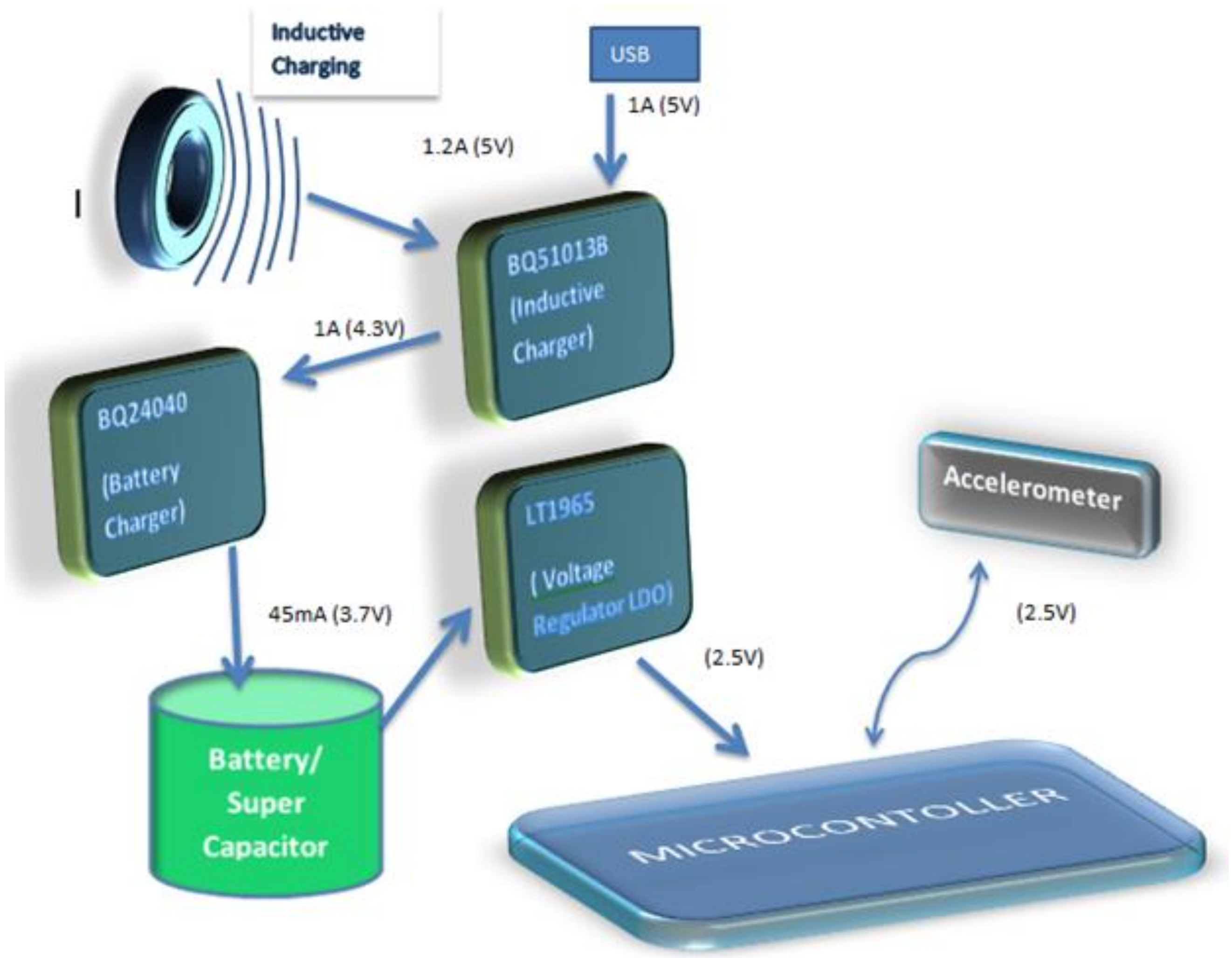


5hrs 12hrs Time -> (hrs)

Stage 1: The battery is of .2C standard charge capacity, thus to charge fully it will take 5 hours

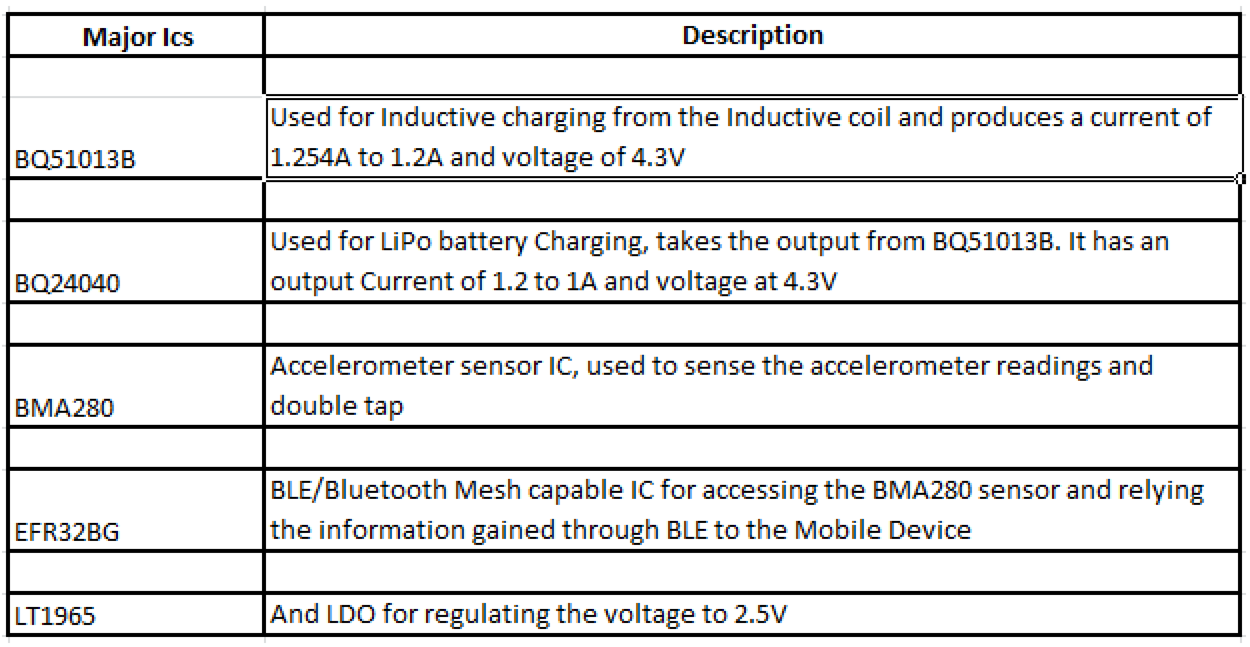
Stage 2: As per our use case, we need the device to be usable for at the most 12hrs.

Hardware functional block diagram



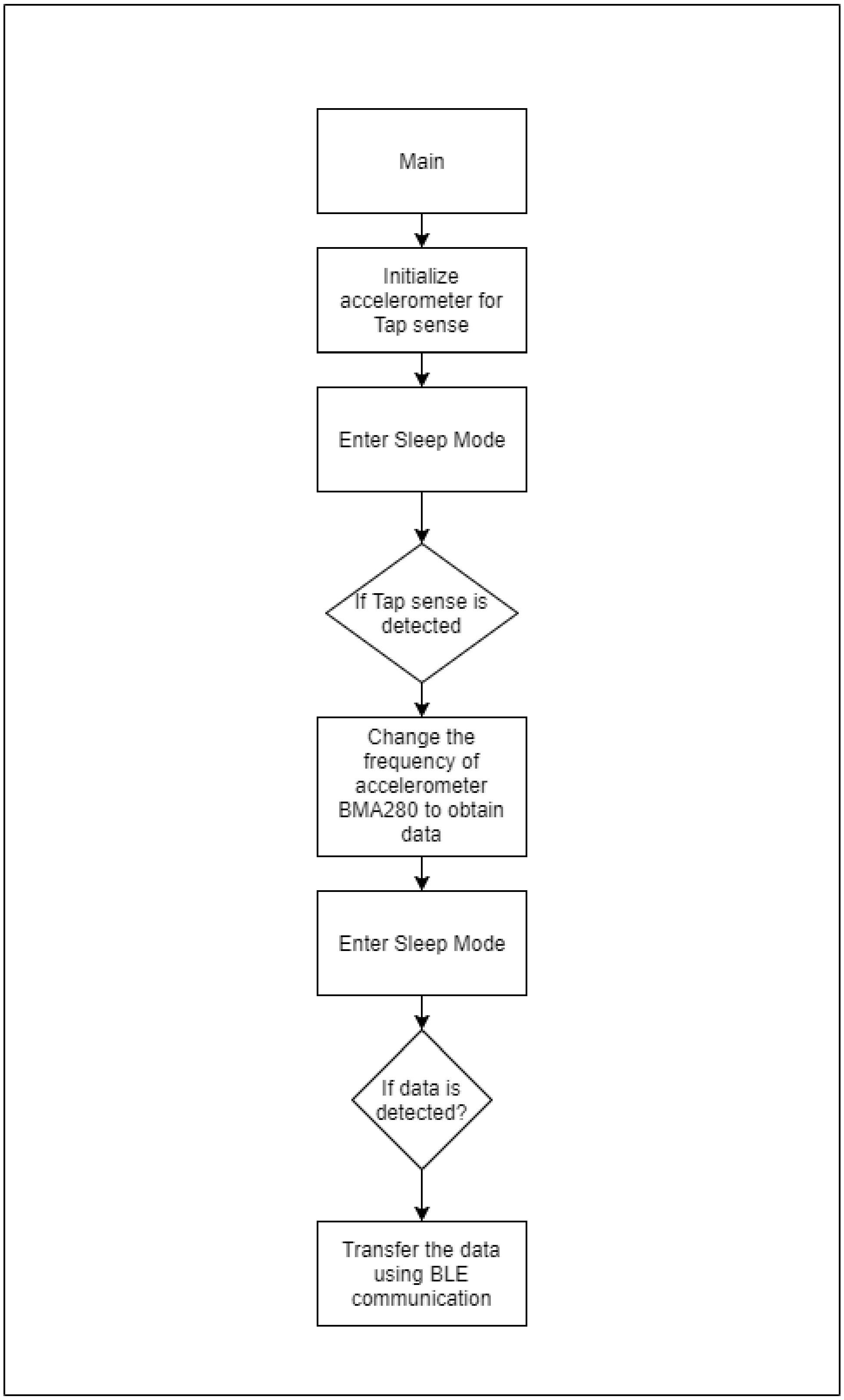
Key hardware components used

Battery	**GMB401215-45mAh**
PMU IC	**LT1965**
Processor	**EFR32BG13– f1024**
Inductive Charging IC	**BQ5103B**
Battery Charger IC	**BQ24040**
Sensor	**BMA280 - Accelerometer**
Sensor	**BMA280 - Tap Sense**
Sensor	**BMA280 – Temperature Sensor**
Switch	**TPS1120DR**



Major Ics	Description
BQ51013B	Used for Inductive charging from the Inductive coil and produces a current of 1.254A to 1.2A and voltage of 4.3V
BQ24040	Used for LiPo battery Charging, takes the output from BQ51013B. It has an output Current of 1.2 to 1A and voltage at 4.3V
BMA280	Accelerometer sensor IC, used to sense the accelerometer readings and double tap
EFR32BG	BLE/Bluetooth Mesh capable IC for accessing the BMA280 sensor and relying the information gained through BLE to the Mobile Device
LT1965	And LDO for regulating the voltage to 2.5V

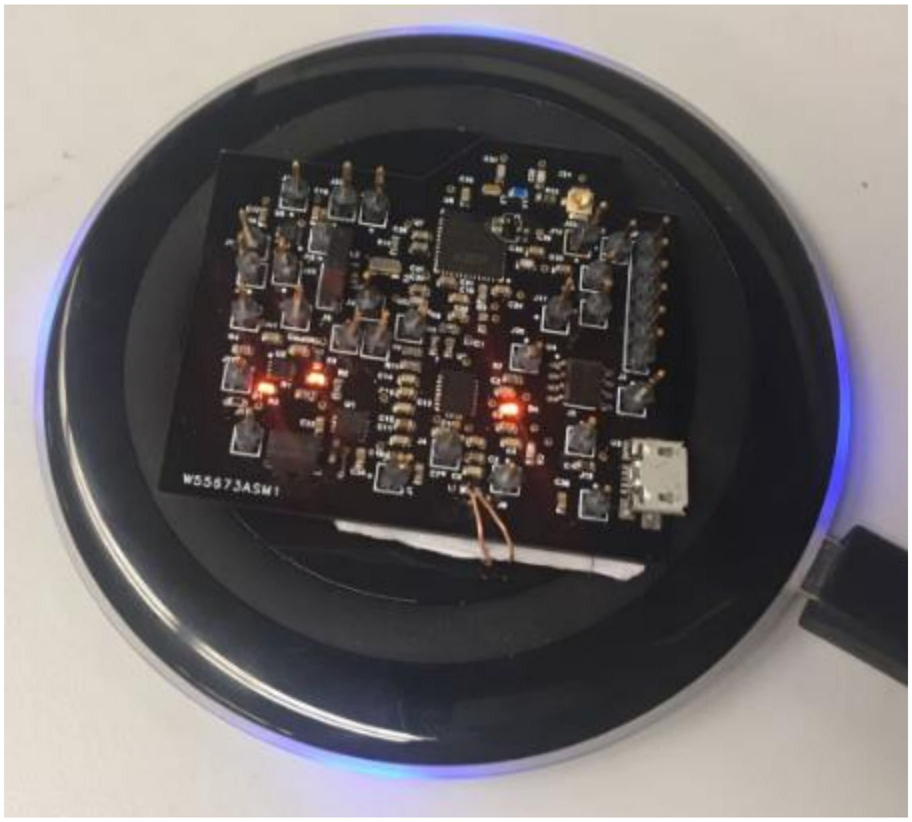
Functional Description (Software functional block diagram)


Main
Initialize accelerometer for Tap sense
Enter Sleep Mode
If Tap sense is detected
Change the frequency of accelerometer BMA280 to obtain data
Enter Sleep Mode
If data is detected?
Transfer the data using BLE communication

So how to use the Device ?

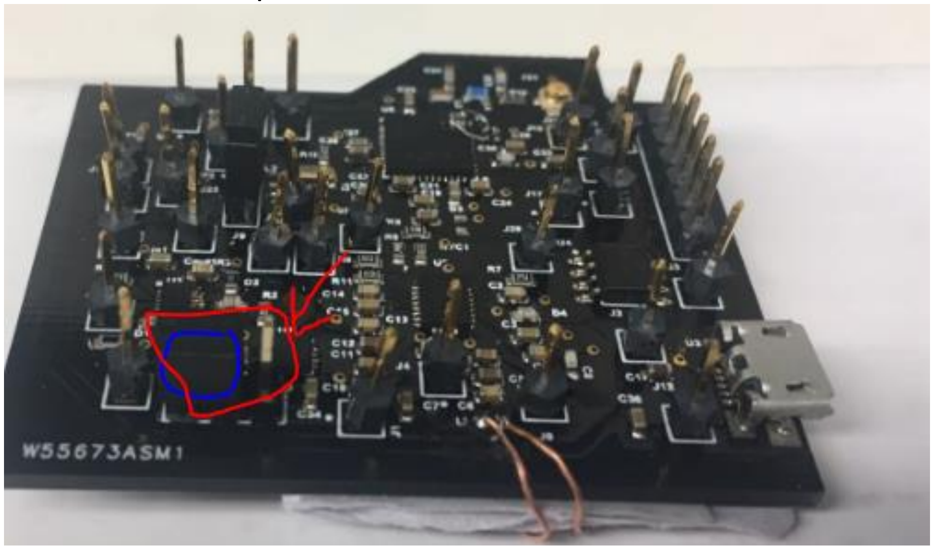
1. Charge the Device through Inductive or USB charging

2. Once the charge is done, it is indicated by an LED

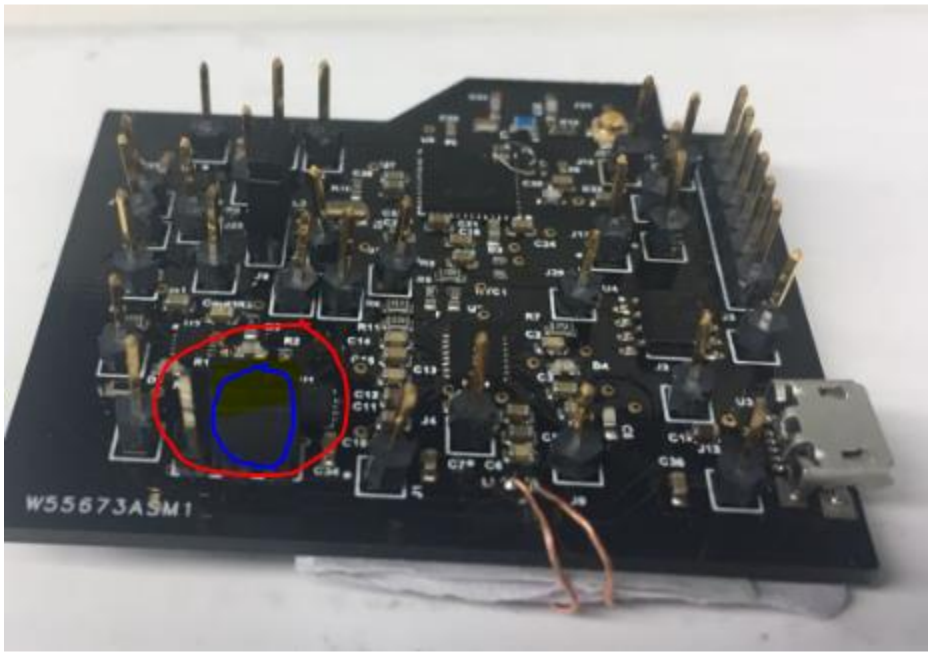


- Indication of Charge complete.

3. Since this is prototype version, (Charging and Discharging are isolated mechanisms) we have to manually connect the battery to the MCU Circuit with a simple connector



Charging



Discharging

4. Now the user can go ahead and start using the device,
    a. A double tap on the device starts the Character recognition
    b. Since this is a connection oriented BLE, you have to open the App and connect to it, and once the characters are recognised the BLE app seamlessly stars the Machine Learning App which can further start the recognition
    c. The user can train any character by switching the “Training” button and for Recognition, toggle the switch to “Recognition”. If the Character Matches, it will display “A Match” - if not it will not. Since this is a prototype version, the user is provided with the information about the X,Y,Z distances, if distance of any of these two axes is <0.1 means that it recognised the character. This App for now is only tested for a single plane of the character written in air.
5. After the required functionality is completed the User can switch off the character recognition by “Double Tapping” again.

-The BLE MESH:

BLE MESH had to be integrated with the BLE, but the current sdks provided by the Silicon Labs are not compatible with this. Thus we implemented a simple BLE Mesh example. In our project we have used 3 MESH boards, one of them is the switch and the others have an LED attached to it.

1. To Use the BLE MESH, User has to have the MESH App.
2. Click on the Provision , scan and click on the devices scanned. Wait untill it goes green

3. Select the Settings option in each of the devices and connect it to a MESH group with appropriate options user wants.
4. Now the user can control the LEDs from the MESH app and even from the Board configured as a switch.

List of commands:

BMA280 Configuration Commands:

BMA280_RegisterRead(USART_TypeDef *usart, uint8_t address)

BMA280_RegisterWrite(USART_TypeDef *usart, uint8_t address, uint8_t data)

BLE :

```
 /* Create the temperature measurement characteristic in htmTempBuffer and store its length */
 length = htmProcMsg(htmTempBuffer);

 /* Send indication of the temperature in htmTempBuffer to all "listening" clients.
  * This enables the Health Thermometer in the Blue Gecko app to display the temperature.
  *  0xFF as connection ID will send indications to all connections. */

 gecko_cmd_gatt_server_send_characteristic_notification(
   htmClientConnection, gattdb_temp_measurement, length, htmTempBuffer);
```

Android App:

1. BLE APP

/*To Send the values of the Accelerometer from BLE APP to the Machine Learning APP*/

```
Intent sendIntent = new Intent();
sendIntent.setAction(Intent.ACTION_SEND);
sendIntent.putExtra(Intent.EXTRA_TEXT, temperatureReading.getX() + "," + temperatureReading.getY() + "," + temperatureReading.getZ());
//Toast.makeText(HealthThermometerActivityFragment.this.getActivity() , "D(X:" + String.valueOf(Math.abs(temperatureReading.getX())) + ", Y:" + String.valueOf(Math.abs(temperatureReading.getY())) + ", Z:" + String.valueOf(Math.abs(temperatureReading.getZ()))+":)", Toast.LENGTH_LONG).show();
//Toast td = Toast.makeText(HealthThermometerActivityFragment.this.getActivity() , "D(X:" , Toast.LENGTH_LONG);

sendIntent.addFlags(Intent.FLAG_ACTIVITY_REORDER_TO_FRONT);
sendIntent.setType("text/plain");
```

```
startActivity(sendIntent);
refreshUi();
```

2. Machine Learning Algorithm (Dynamic Time Warpping):

```
final DTW      lDTW      = new DTW();
// Iterate the Histories.
for(int i = 0; i < 3; i++) {
  // Fetch the Primitive Histories for this Axis.
  final                    float[]                    lTraining              =
MainActivity.primitive(MainActivity.this.getTrainingHistory()[i]);
  final                    float[]                    lRecognition           =
MainActivity.primitive(MainActivity.this.getRecognitionHistory()[i]);
  // Calculate the distance using Dynamic Time Warping.
                lAverages[i] = lDTW.compute(lRecognition, lTraining).getDistance();
```

What aspects of the project are considered high risk? How is this risk planned to be mitigated?

Risk factors:

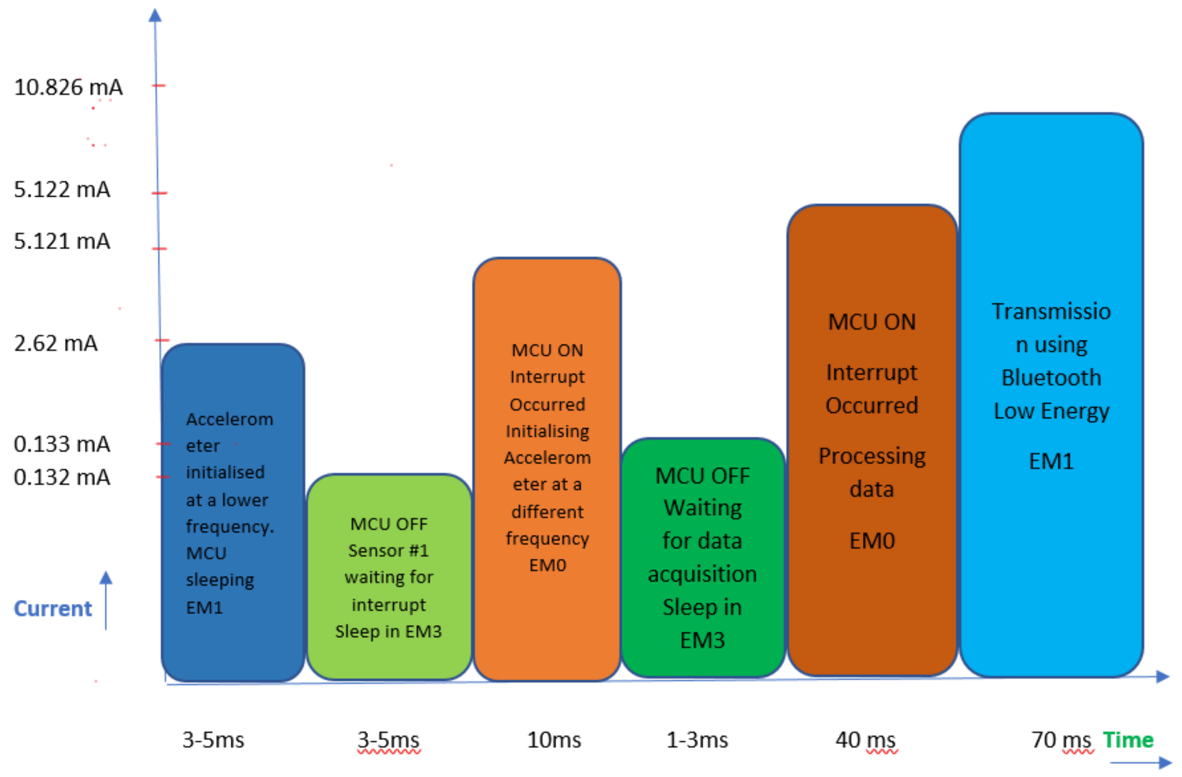
1. We might end up consuming more current because we are using accelerometer (BMA280) to both turn on the entire device and for data acquisition.
2. Our project involves inductive charging for the battery which is a new concept for us and we are not sure if we will be able to get the appropriate energy requirements from inductive charging as required by our application.
3. The issue with accelerometer data acquisition is that we might not know how much data is enough /less to compute the algorithm on the data and detect the pattern or the command
4. The Machine learning algorithms always need more information about the data to differentiate different patterns accurately. And for such high data handling it needs more high powered CPUs, which would help in computing the algorithms on the data in few milliseconds
5. The main problem is the time it takes to capture the data and send the data to either Phone/Cloud => process the data through FTT/any other algorithms to make sense out of the data and pass it through the pattern detector and predict a most accurate command. - All these have to happen within few miliseconds, if not the product can fail

6. The Bluetooth mesh network is a new technology which needs a lot of time to understand and troubleshoot the issue faced.

Mitigation:

1. We are currently using accelerometer (BMA280) in low power 2 mode with 0.5 ms of sleep duration. So, we can consider reducing the current consumption either by increasing the sleep duration or using the accelerometer in a mode with lesser current than low power 2 mode.
2. We will focus on referring more documents and articles on inductive charging to get the required energy requirements. If we are beyond the schedule and still not successful with inductive charging, then we will consider switching to an alternative power source such as USB charging.
3. To know how much data is enough/ less from the accelerometer, we have found out a simple solution to configure a simple app from the mobile (all mobiles have a accelerometer and a gyroscope).
   We take the values of the mobile-accelerometer through a simple javascript application, populate the data to a csv-file/ populate the data in realtime to the algorithm we are generating. Thus before the firmware development we will have a clear idea on how much data would be required for acquisition.
4. We are trying to get a simpler lightweight Machine learning algorithm - which can run on a smartphone. If we cannot find the exact configuration we need, we will try dumping the data to any cloud platform to do the work for us - say AWS/IBM Watson etc..
5. Since it is considered as a prototype, even with a max delay to recognise the commands, the minimal success would be to achieve a command. Thus if the commands are recognised we can go ahead and start mitigating the delay.
6. We will focus on configuring only 1 or 2 devices with Bluetooth mesh network to make a simple network by referring documentations and receiving guidance from people who have worked on Bluetooth mesh network. If we are unable to achieve Bluetooth mesh network in the required time then we will consider using ZigBee/WiFi.

Energy use mode model



Stage 1:
Configuring the Accelerometer for double tap sense (3-5ms)
State 2:
MCU is off, Sensor #1 i.e Accelerometer tap sense is On and it is waiting for an interrupt to switch on the entire device. The device stays in this state for about 3-5 ms after the user has used touch sense to switch on the device. It uses SPI serial port interface which works in EM3 for Blue Gecko and also, we wait for an interrupt in EM3 energy mode.
State 3: MCU becomes on when an interrupt occurs, hence it will be in EM0 mode. In the interrupt service routine (Accelerometer data acquisition) is switched on. This state will take about 10 ms to handle the interrupt and configure the Accelerometer to accept the Data acquisition at higher frequency.
State 4:
In this state, MCU is off while Accelerometer is waiting for an interrupt i.e it is waiting to receive data from the user. As use case considered is when user switches

on the device and immediately writes the data, this state will approximately take 1-3 ms.
State 5:
In this state, MCU becomes on when an interrupt occurs i.e when data is received from the user and the data is processed. This state will approximately take 40 ms to handle the interrupt and process the data.
State 6:
In this state, the data is transmitted to the mobile phone using BLE and it is interpreted to display the character received. This will take about 70 ms and operates in EM1 for transmission.

Calculations to specify the Energy Storage Device
▪ Average current / energy capacity calculation

<u>Determining the average weighted power out of battery using Blue Gecko:</u>

On duty cycle:
Assuming that the blue gecko is given a input voltage of ~2.5v and ~1.9v is an output voltage.
Blue Gecko's efficiency at ~2.5v input to ~1.9v out is 2.5v/1.9v = 75.75%
Since it cannot be a 75.75% ,
Weighted average power out of battery = (On duty cycle * On time current * 2.5v) / 75.75% = (0.2 * 10.826mA * 2.5v) / 75.75%
Weighted average power out of battery = 71.70 uW

Off duty cycle:
Blue Gecko efficiency at ~3.3 input to ~2.5v out is 2.5v/3.3v = 75.75%
Weighted average power out of battery = (Off duty cycle * Off time current* 2.5v) / 75.75% = (98.8 * 0.133mA * 2.5v) / 75.75%
Weighted average power out of battery = 433.68 uW

Total average power out = Weighted on duty cycle average power + Weighted off duty cycle average power
= 71.70 uW + 433.68 uW
= 505.38uW

Average energy the battery must provide on the use case:
= 12 hrs of use * 60 mins * 60 secs * 505.38uW
= 21.83J

Peak currents and Peak current step functions that will need to be considered

BQ51013B has an inductive charging: the max current from the coil can be 1A
USB : the max current from the USB is 1A

The MAX current the BQ24040 (Battery Charger) & BQ51013B (Inductive charging) can handle is 1.245A

The Energy Harvester input doesn't require any current limiting circuit.

Engineering reasons why the particular Energy Storage Device has been selected
▪ Number of recharge cycles
▪ Provide supporting documentation that the use case number of recharge cycles can be achieved.

As per our calculation, we need a battery with the following specifications:
Average energy 21.83J
Vout from the Battery is ≈ 3.3v
Max current required 10.826mA (all on - MCU, Sensors, BLE)
If the product is targeted to be used for 1.5years , 5 days a week. We have 390 recharge cycles.
With weighted average power = (0.2% *10.826mA) + (99.8% * 0.133mA) = 15.439mA
15.439mA * 12 (->12 hours) = 1.83mAh
For rechargeable cells the battery nominal capacity should be 10x capacity.
Thus total =>18.3mAh

We need a battery with 18.3mAh and 21.83J
Thus selected: https://www.powerstream.com/lip/GMB401215 45mAh_With%20PCM_A0.pdf

Above calculations substantiates the choice of the Energy source to be Battery and not Super-Capacitor.
Batteries provide consistent voltage (3.7 V) - which is within the voltage range of the components we are using in our application. Since our application does not require any bursts of current and we need our device to discharge the power for a long time - we choose Battery.

Why is the particular MCU / SoC / Radio being proposed?

EFR32BG13 is chosen since it has the supported sdk for Bluetooth Mesh

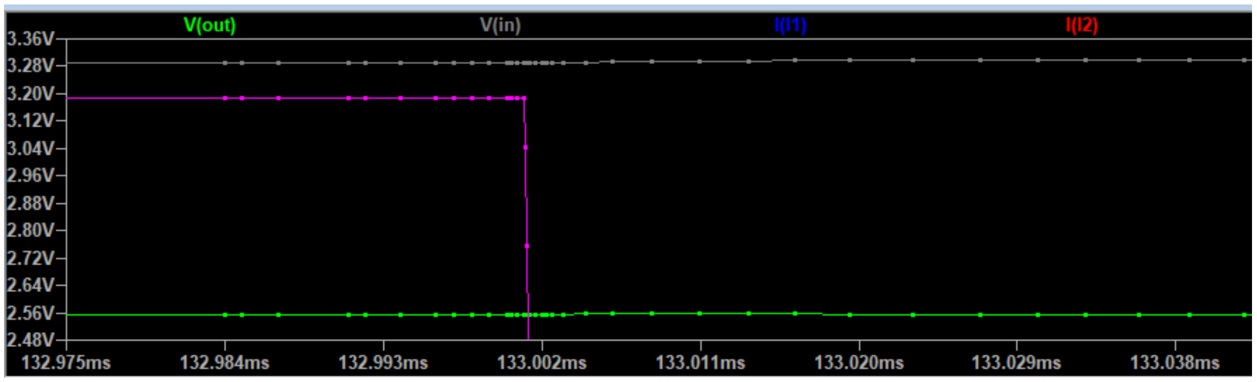
Why is the particular PMU solution being proposed?

We researched for Inductive Charging, BQ51013B is the best available TI IC for inductive charging which also supports alternative USB charge mechanism.
BQ24040 is required to charge the LiPo battery from the BQ51013B.
LT1965 is chosen to regulate the voltage at 2.5V, which is provided to the MCU.

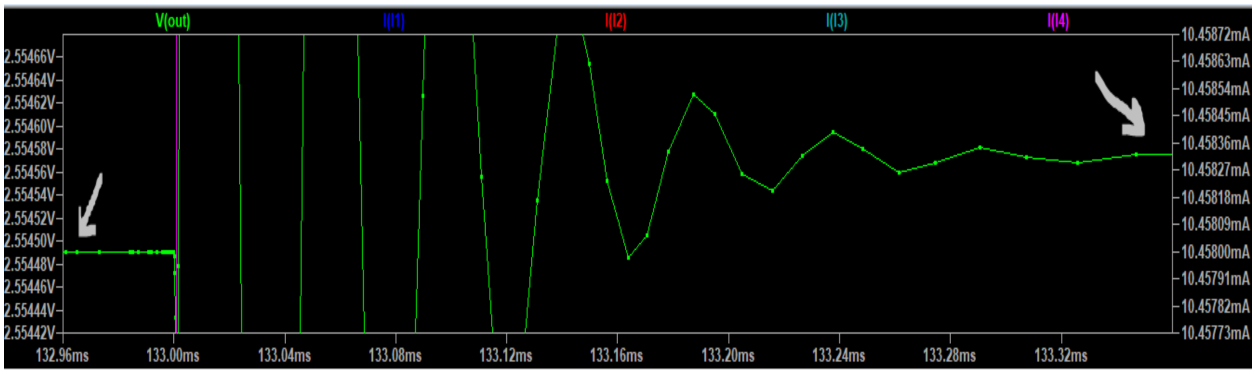
Supporting documentation that voltage droops due to possible current delta functions of this low energy mode device have been engineered not to result in a failure

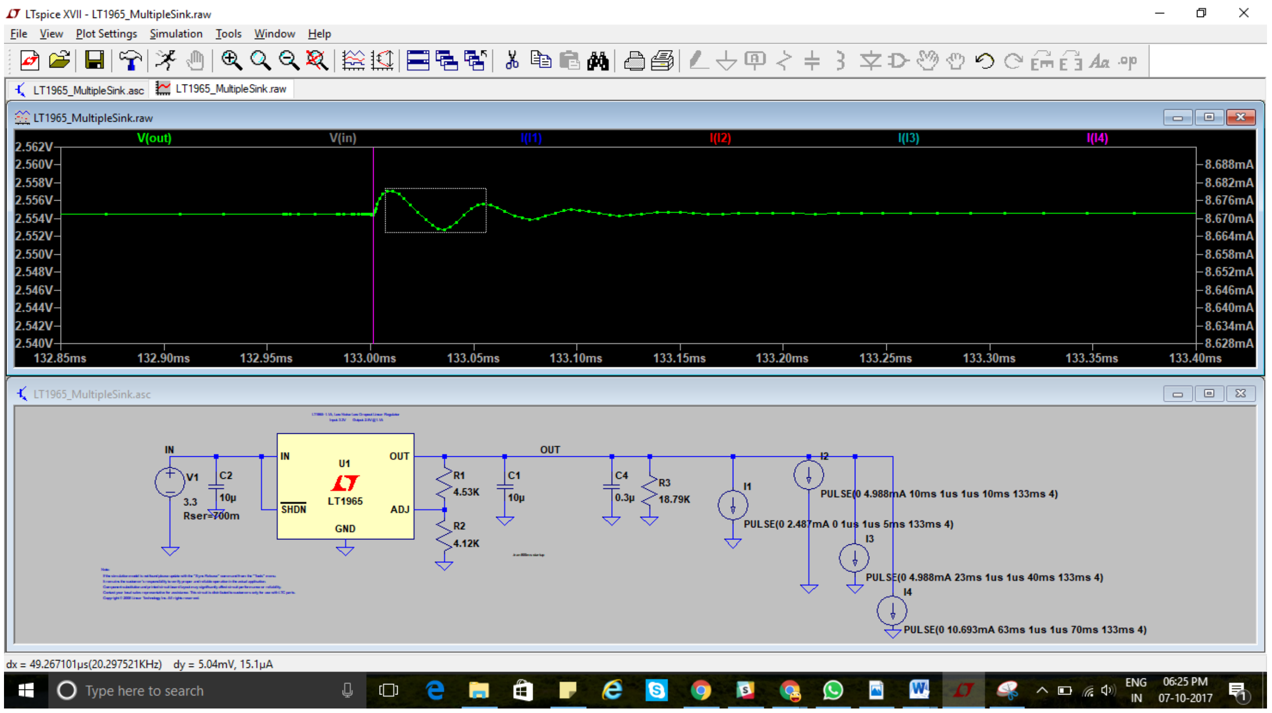
a.At Vin = min

Yes, the voltage was increased of about 0.1mV. The Supply Voltage was about 3.28V to the PMU IC from the battery.
The PMU IC would be giving approximately 2.5V output.



Note: The silver line indicates the stabilized voltage change





dx = 49.267101µs(20.297521KHz) dy = 5.04mV, 15.1µA

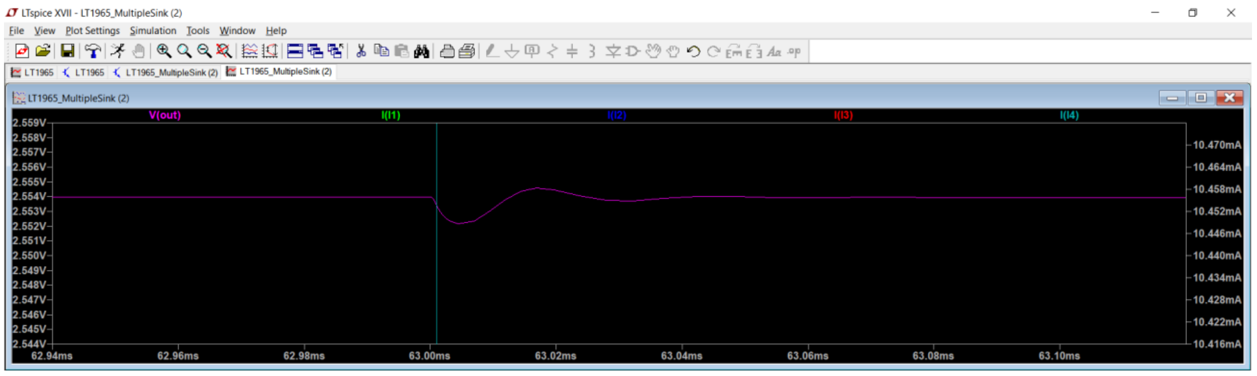
5mV is the ripple which is within the specifications of 0.5Vp-p

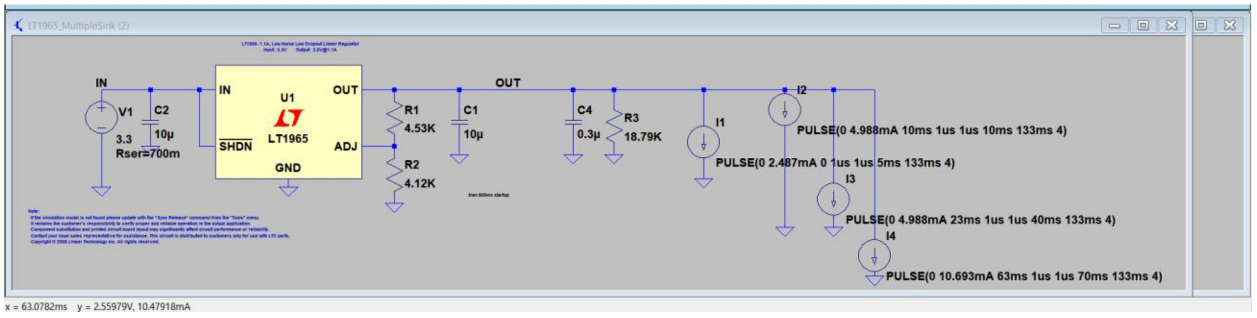
When the dynamic load is added there is a dip in the voltage, the min value is 2.553V, the appropriate voltage supplied by the battery is 3.296V
The Stabilized output voltage of the battery before the dynamic load was 2.5544V
Thus, there is a variation of 2.5544-2.553 = 0.0014V => 1.4mV which is within the specifications.

Dynamic current step up load:

Yes, there was a voltage dip with dynamic step up load.

Below screenshot shows the voltage dip when there is dynamic current step up load i.e when the load switches from EM0- Accelerometer interrupt to EM1-BLE transmission.




LT1965_MultipleSink (2)
IN
V1
3.3
Rser=700m
C2
10µ
IN
U1
LT1965
SHDN
GND
OUT
ADJ
R1
4.53K
R2
4.12K
C1
10µ
OUT
C4
0.3µ
R3
18.79K
I1
PULSE(0 2.487mA 0 1us 1us 5ms 133ms 4)
I2
PULSE(0 4.988mA 10ms 1us 1us 10ms 133ms 4)
I3
PULSE(0 4.988mA 23ms 1us 1us 40ms 133ms 4)
I4
PULSE(0 10.693mA 63ms 1us 1us 70ms 133ms 4)
x = 63.0782ms y = 2.55979V, 10.47918mA

Minimum out voltage is 2.552167 V

Therefore, voltage droop = 1.83 mV < 0.5Vp-p

Hence, this is within the IC specifications

The stabilized voltage in EM0 before dynamic step up load = 2.55399 V

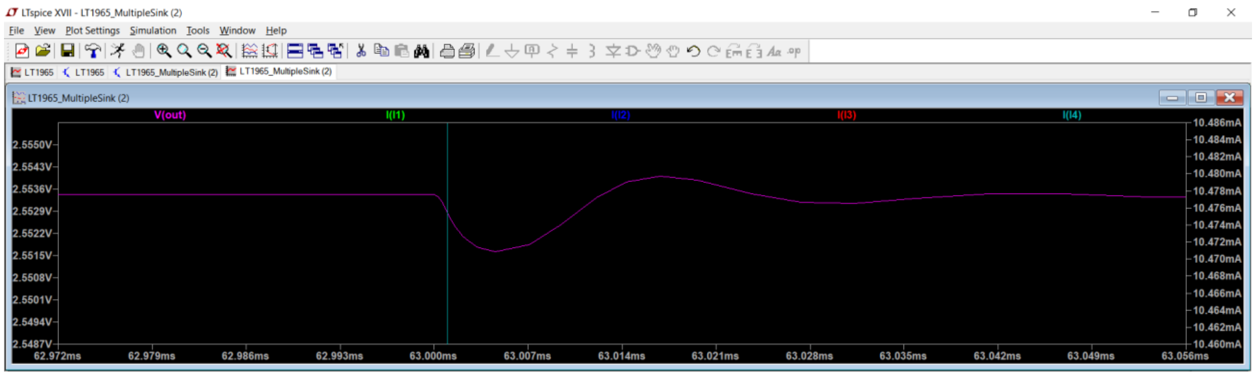
The stabilized voltage in EM1 after dynamic step up load = 2.55394 V

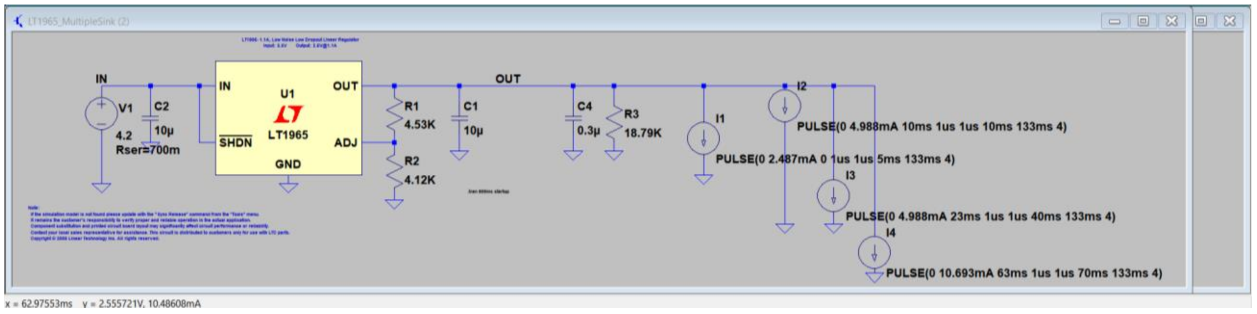
b. At Vin = max i.e Vin = 4.2 V

Dynamic current step up function load:

Yes, there was a voltage dip with dynamic step up load.

Below screenshot shows the voltage dip when there is dynamic current step up load i.ewhen the load switches from EM0- Accelerometer interrupt to EM1-BLE transmission.





PULSE(0 4.988mA 10ms 1us 1us 10ms 133ms 4)
PULSE(0 2.487mA 0 1us 1us 5ms 133ms 4)
PULSE(0 4.988mA 23ms 1us 1us 40ms 133ms 4)
PULSE(0 10.693mA 63ms 1us 1us 70ms 133ms 4)

x = 62.97553ms y = 2.555721V, 10.48608mA

Minimum out voltage is 2.5516 V

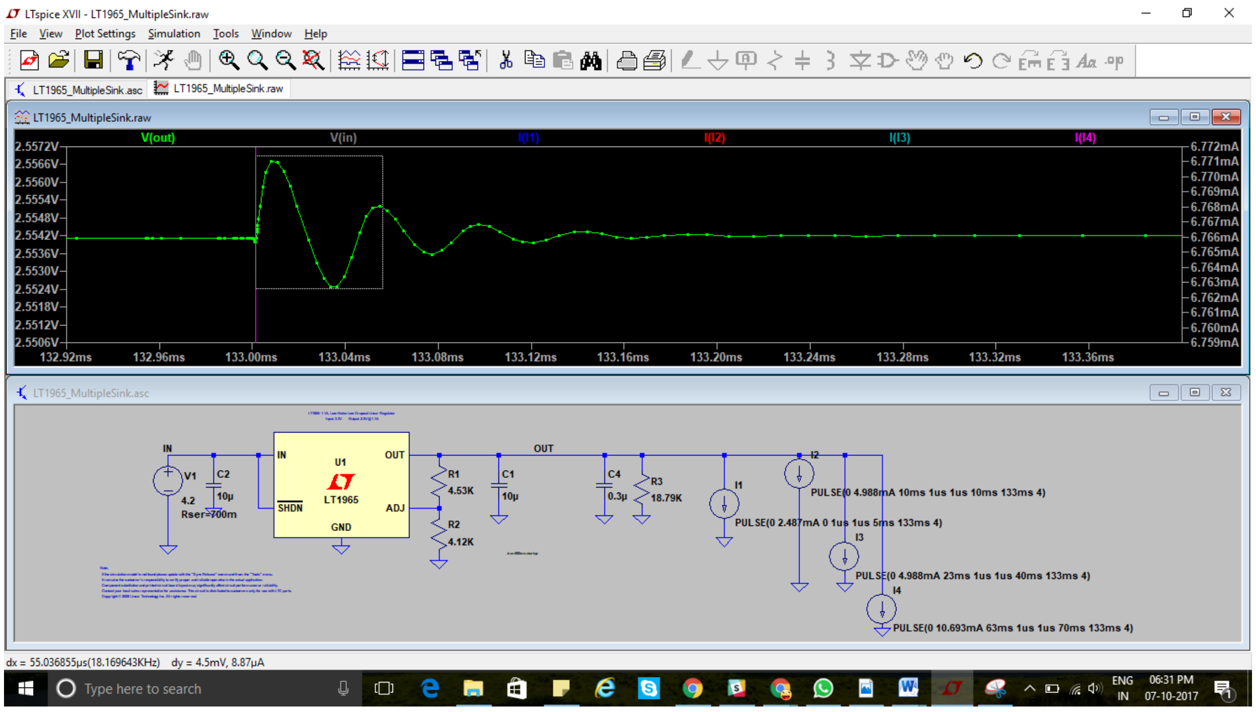
Therefore, voltage droop = 1.8 mV < 0.5Vp-p

Hence, this is within the IC specifications

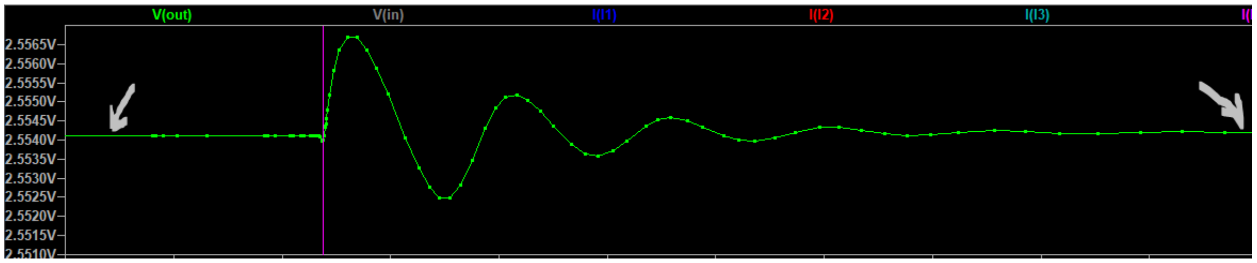
The stabilized voltage in EM0 before dynamic step up load = 2.5534 V

The stabilized voltage in EM1 after dynamic step up load = 2.5533 V

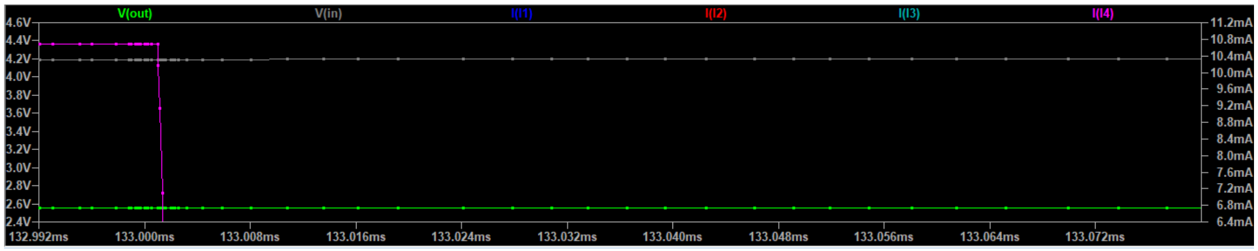
For Step Down Load : Vin: Max: 4.2V

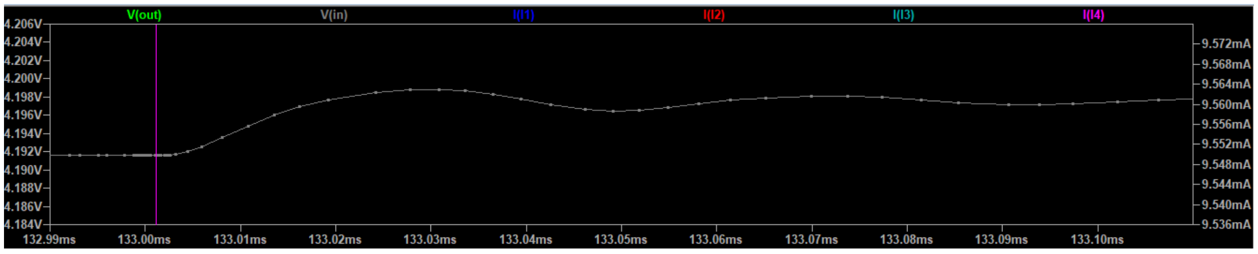

LTspice XVII - LT1965_MultipleSink.raw
File View Plot Settings Simulation Tools Window Help
V(out)
V(in)
I(I1)
I(I2)
I(I3)
I(I4)
IN
OUT
U1
LT1965
SHDN
ADJ
GND
V1
4.2
Rser=700m
C2
10µ
R1
4.53K
R2
4.12K
C1
10µ
C4
0.3µ
R3
18.79K
I1
I2
I3
I4
PULSE(0 4.988mA 10ms 1us 1us 10ms 133ms 4)
PULSE(0 2.487mA 0 1us 1us 5ms 133ms 4)
PULSE(0 4.988mA 23ms 1us 1us 40ms 133ms 4)
PULSE(0 10.693mA 63ms 1us 1us 70ms 133ms 4)
dx = 55.036855µs(18.169643KHz) dy = 4.5mV, 8.87µA

The ripple voltage is about 4.5mV, there is a voltage raise of about 1mV indicated by the arrows for step down function.



Vdd:



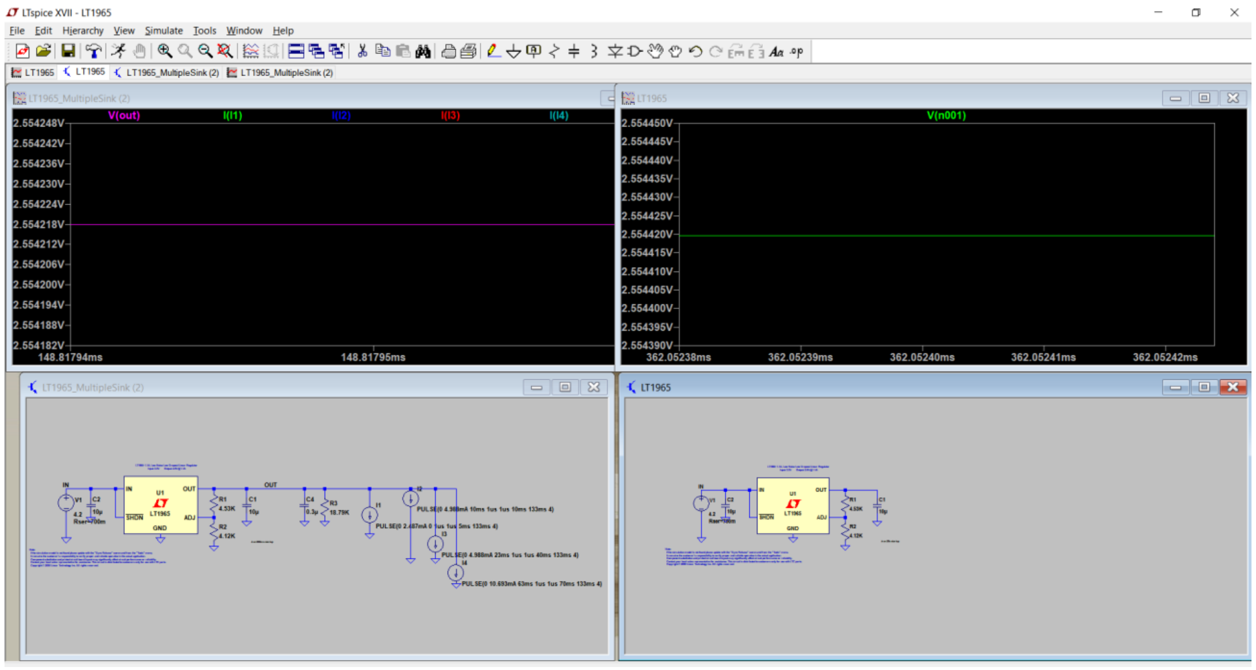


The Vdd was increased to about 5mV to 4.198v

When the dynamic load is added, a voltage dropped to about 2.5522V, the stabilised voltage before the variation was 2.55407V

The difference of about 1.87 mV
Which is within the specifications of 0.5Vp-p ripple.

Simulation for with & without Dynamic load:



The circuits as shown above:

Voltage when there is no dynamic load: 2.5542 V (Image on the right)

Voltage when there is dynamic load: 2.5542 V (Image on the left)

Hence there is no change in the voltage when a dynamic load is added to the circuit.

The min voltage when the power supply is drooping is 2.53 V which is within the specifications of the IC.

Proper engineering selection of the required bulk or PMU ceramic load capacitance

The value of Decoupling Cap is calculated as the project's circuit doesn't require any Bulk load capacitance (Thus not taken into consideration)

Decoupling capacitors:

Blue Gecko: According to hardware considerations, we have considered a decoupling capacitor (ceramic capacitor) of value 0.1 uF
Ref:https://www.silabs.com/documents/public/application-notes/an0002.1-efr32-efm32-series-1-hardware-design-considerations.pdf

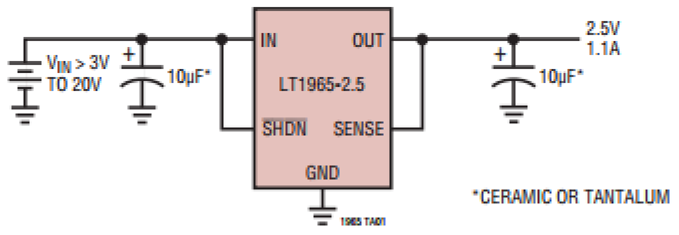
BMA280: We have considered a decoupling two capacitors of 100 nF(0.1 Uf each).
Ref:http://www.mouser.com/ds/2/783/BST-BMA280-DS000-11_published-786496.pdf

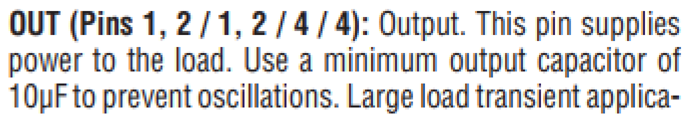
Hence total capacitance = 0.3 uF

Thus, the capacitance for the power plane is the same value as the decoupling capacitor of 0.3uF.

According to the Data sheet of LT1965:

3.3V to 2.5V Regulator




OUT (Pins 1, 2 / 1, 2 / 4 / 4): Output. This pin supplies power to the load. Use a minimum output capacitor of 10µF to prevent oscillations. Large load transient applica-

And also ceramic capacitor of type X5R is preferred according to the data sheet of PMU and for the decoupling capacitors for MCU and accelerometer sensor.
Ref:https://www.silabs.com/documents/public/application-notes/an0002.1-efr32-efm32-series-1-hardware-design-considerations.pdf
http://cds.linear.com/docs/en/datasheet/1965fb.pdf

Considering the DC-Bias characteristics, we have selected the below part-number for the load capacitor which would suffice with the condition given in the datasheet (min of 10uF for the load capacitor)

We have a decoupling capacitor of 0.3 uF. Considering the DC-Bias characteristics, we have selected the below part-number for the load capactior which would suffice the requirement.

We have a decoupling capacitor of 0.1 uF each at each volatage terminal of BMA280 and PMU.

Selected part number to give a capacitance of 0.1UF at 2.5V which is our working voltage is :
http://psearch.en.murata.com/capacitor/product/GRM188R71C104KA01%23.html
Ref: http://cds.linear.com/docs/en/datasheet/1965fb.pdf

ESD protection has been engineered on these user accessible lines to ensure a long-life product

A TVS DIODE 5VWM 12.5VC SC88 is selected for the power lines from the USB, the inductive charging does not need any ESD diodes.
Since the BMA 280 is on the board, we do not need any ESD diodes for the communication.

Planned Development Schedule:

Task	**Date to be finished**	**Date it was finished**	**Status**
MCU Selection	9/16/2017	9/16/2017	Done
Sensor Selection	9/16/2017	9/16/2017	Done
Battery Selection	9/23/2017	9/23/2017	Done
Radio Communication Selection	9/23/2017	9/23/2017	Done
PMU Selection	9/23/2017	9/23/2017	Done
Energy Charger Selection	9/23/2017	9/23/2017	Done
Current profiler	9/30/2017	9/30/2017	Done
Energy Spike Calculation	9/30/2017	9/30/2017	Done
Voltage Droop Simulation	10/8/2017	10/8/2017	Done
Schematics	10/28/2017	11/4/2017	Done
Foot print generation	10/28/2017	11/4/2017	Done
Layout	11/4/2017	11/11/2017	Done
Ordering parts and board	11/11/2017	11/20/2017	Done
Assembly of the board	11/27/2017	12/1/2017	Done
Testing of the board	12/10/2017	12/12/2017	Done
Firmware for tap sense	12/12/2017	12/15/2017	Done

Firmware for accelerometer data	12/12/2017	12/15/2017	Done
Firmware for temperature	12/16/2017	12/16/2017	Done
BLE code development	12/14/2017	12/14/2017	Done
Custom Application development	11/15/2017	12/15/2017	Done
Machine learning algorithm for data interpretation	12/15/2017	12/15/2017	Done
BLE mesh implementation	12/16/2017	12/16/2107	Done
Integration of the code	12/15/2107	12/16/2017	Done
Testing of firmware	12/15/2017	12/16/2017	Done

Verification plan :

To be verified	Definition of passing	Date test performed	Tested by	Measured result	Passed ?
Supply voltage to the MCU should be within the minimum and maximum specification	The voltage should be within 1.8V to 3.3 V	2/12/17	Sanjana	2.5 V - Measured using multieter	Passed
Supply voltage to the BMA280 sensor using SPI interface should be within the minimum and maximum specifications	The voltage should be within 1.2V to 3.6 V	2/12/17	Sanjana	2.5 V - Measured using multieter	Passed
Signal quality of SPI communication line	Check correct signal level transition on the oscilloscope	14/12/17	Sanjana	Signal level at 2.5 V - Measured using multieter	Pased
Firmware Block of BMA280 using SPI interface: The	The interrupt should get triggered whenever the	15/12/17	Sanjana	Verified the signal transiti	Passed

MCU is receiving correct data from the sensor	sensor senses data. The MCU is receiving the correct accelerometer reading from the sensor- verify clock too. Verify Chip Select of SPI goes low while transferring data			on on a logic analyser and interrupt signal was also verified on the IDE in debug mode	
The radio is transmitting data correctly	Verify at the receiver’s end the data received	14/12/17	Shekhar	Tested with soc-smarphone example code, to verify its connection and working on BLE.	Passed
Supply voltage from the USB	The voltage should be within 4.3 to 5 V	3/12/17	Shekhar	5V obtained which	Passed

				is within the limits	
Supply current from the Inductive coil	50mA to 1A	11/12/2017	Shekhar	1A - Measured with ammeter in series	Passed
The Current out from the Inductive Charger – BQ51013B	700mA to 1.2A	11/12/2017	Shekhar	700-800 mA - Measured with ammeter in series	Passed
The Current input to the BQ24040	700mA to 1.245 A	11/12/2017	Shekhar	800 mA - Measured with ammeter in series	Passed
The Voltage at the RECT pin	Must be within 4.3 to 5V	10/12/2017	Shekhar	4.3V - Measured using multimeter	Passed
The voltage at the BAT	3.5 to 4.2V	10/12/2017	Shekhar	4.2 V - Measured with	Passed

Pin/VOUT from BQ24040				ammeter in series	
The current at the BAT Pin/VOUT BQ24040	45mA +/- 10%	11/12/2017	Shekhar	22 mA – 45 mA - Measured with ammeter in series	Passed
The voltage input to the LDO LT1965	1.8 to 20 V	7/12/2017	Sanjana	3.7V - Measured with ammeter in series	Passed
The voltage output from the LDO to MCU	2.5V exact	7/12/2017	Sanjana	2.5V - Measured with ammeter in series	Passed

Difficulties encountered in the project:

One of the foremost difficulty encountered is that both of us, as a team had very little experience in hardware and this is the first class we have got an opportunity to learn and design a hardware of our choice. It took us a while to adapt to Altium software and get the best design for our board.

As we had proposed to work on BLE mesh, we had to make the right choice of the MCU board because most of the MCUs in Blue Gecko family support BLE mesh but all of them are yet to support the APIs for BLE mesh except EFR32BG13 boards. This information is not available on any of the silicon labs site and we had to confirm this only by contacting silicon labs and discussing with peers who have already worked on BLE mesh.

Implementation of Machine Learning algorithm for character interpretation is one of the difficulties encountered. We had to do a lot of study and research to choose an optimised algorithm that trains the data from the accelerometer and interprets the character accordingly. The machine algorithm is classified into classification and regression type of algorithm, While, choosing one of the algorithms and doing a research, we found out that regression type of algorithm is better in real time for alphabet recognition. Support Vector Machine(SVM) is a widely used regression type of algorithm, but the problem with SVM is that, it uses huge amount of data sets, and dataset is not common i.e the dataset is not same for different accelerometers. Each device has a different dataset and it consumes huge amount of time. Hence an alternative to SVM algorithm is Dynamic time warping algorithm. In this algorithm, it calculates the Euclidian's distance and generates matrix depending on X, Y, Z axis array values. Since this calculation is done in real time, the training and recognition of any devices in real-time is accomplished without any predefined dataset.

In our project, we are obtaining data using the accelerometer and train the dataset accordingly. We faced a difficulty while reading the required data from the accelerometer because BMA280 does not provide us the resolution required to train the dataset accurately. We tried giving a resolution of 2g and a maximum frequency(2Khz) but it still failed to provide us the accurate data. So even the maximum frequency was less for our application.

Unable to send an array of values to BLE: As maximum payload for BLE is 20 (length) we were unable to send an array of values to BLE application hence we had limit by sending each acceleration data of X,Y,Z axis and the train the it accordingly.

One of the other difficulties we faced is while developing the firmware block for BMA280. Firstly, it took a lot of time to figure out that the dev kit is not getting detected because of the version of SDK we had downloaded. The simplicity studio 4 which we had downloaded did not have BLE 2.6 SDK and hence it was not supporting the kit. Later on, after developing the firmware block on the example code, it gave a lot of linker errors and which consumed lot of time to be solved. This process was not as easy as using leopard gecko with simplicity studio 3, hence setup was one of the difficulty.
LCD interfering in BMA280(SPI) communication: The LCD functionality was causing a problem in working of BMA280. The RXDouble register that we were reading while doing a register read from BMA280 always gave a 0XFF instead of reading an appropriate value. After commenting LCD functionality, we were able to read appropriate values hence we were able to read correct interrupt status to differentiate between single tap, double tap and accelerometer reading. Even while integrating with BLE, we faced the same problem, after realising that the BLE code was causing a problem, we put a register read after every command to find the solution of the problem. We realised that in the init() function, there is a hardware init() function that has commands for LCD support, after commenting that, we got our BMA280 working appropriately with BLE.

Difficulties faced with HFXO oscillator: We had a mistake in schematics of the HFXO oscillator because of which our code was not running properly. Even while testing the BMA280 accelerometer code, though we did not require HFXO for its functionality, the code was getting stuck in a while loop, waiting for readybit of HFXO to be set in one of its init() functions and if we commented that, then the an error use to pop up displaying the error caused due to power loss and same repeated. So until we fixed the HFXO connection, we could not check any of our code on the board.

BLE & Bluetooth mesh together: Initially we had proposed a method which integrated BLE and BLE mesh together in an application. But with time, we realised that they cannot be made to work together, both BLE and BLE mesh have their own events and characteristics which cannot be integrated.

Battery and coil reorder: We intended to use a lipo battery of 45mAh rating and the one we needed was available only on of the websites which needed us to wait for their quote to order the component. Hence it delayed ordering of our battery but instead we had to order another battery from sparkfun with differ specifications to support our application and same applies to the coil. Hence this put us at risk until we tested the components on our board.

Breakpoint in BLE: If we keep a break Point in the BLE advertisement or GATT service char send, the BLE application does not work because it disconnects the device from the application while doing a step by step debugging. Hence it is very hard to understand what actually happens while doing a BLE communication and tweak in our code into already available example code.

Resistor Issues: According to the Use case model, the Battery capacity should be 45mAH. But during the component-order state, the part was not in stock. Thus the ISET resistors value in the Battery Charger circuit (BQ24040) had to be changed. According to the Circuit we had ordered 4k Ohms resistor which should be providing a 45mA of charge current to the Battery. Since we could not find a 45mAH battery – we ordered a 40mAH and 110mAH from sparkfun. The issue was that the standard Charger Current of both the 40mAH and 110mAH is 0.2C, a max of 22mA is the standard charge current. But the resistors were offering a 45mA of current. So, we had to change the resistor to 12K.

Battery Datasheet Issue: The Battery datasheet for the battery was not available in the Sparkfun website as well, we even called them but in-vain. We needed the data sheet to get the information about the Standard Charge and Discharge Current. Once we got the battery we saw the specs on the Battery and we looked for any datasheet with that part-number. We got one from Alibaba.com from which we calculated the ISET resistors.

The Application diagram provided in the Inductive Charging was improper. The capacitor values were wrong and the Part numbers specified in the DataSheet for the P-to-P channel switch was not present. We had to approach peers who have worked on Inductive Charging to make sense out of the Application Diagram.

Thermal pads in BQ51013B: The BQ51013B had a thermal pad footprint which had smallest distance between the pad and thermal pad. This difference could not be solved, and even the DFM check gave an error. Thus we had to change the thermal pad of polygon shape. The extended thermal pad was changed as extra pads on the IC.

Android SDK Issue: The BLE app and the Machine learning app were developed separately. Both the applications were of different SDK, and this should not be an issue to merge the APPs but the problem was never known. Could not merge both the applications due to the SDK issues. The solution to this was to get the Machine Learning App from BLE app whenever we receive any information about the change in accelerometer. This was achieved by Intent Messages.

Intent Messaging in Android always starts the second App as a new instance, due to which whenever we got new data from the BLE app, it used to create a new instance of the Machine Learning APP. The issue with this is that the Machine Learning algorithm was not saving all the values. To overcome this I used a flag and a change in the Manifest file to set only a single instance of the APP.

Faced difficulty in measuring Discharging/Charging of Battery (in the units of Current/Voltage). Found out how to discharge/charge battery and supercap.

Hardware Issues:

HFXO Crystal:

Root cause of the hardware issue:

The schematic connection to the HFXO oscillator was wrong. The oscillator we have generally used in the past is two pin oscillator and this was the first time we used a 4 pin crystal. We ended up interchanging a connection to hot with a connection to ground, because of which we could not run any code with HFXO Enabled on our MCU.

Solution to Hardware issue:

After a wide range of proposed-fixes for this solution, the pad of the footprint of the Crystal came out due to overheating the board. And this is when the solution for a DeadBug was introduced. For one of our boards we had to solder wires from the MCU pins but the other board was intact. The basic Idea behind the solution is this: To remove the crystal from the board, placed it on top of the MCU using hot glue and made wired connections to the correct pads. Etch and remove the wrong trace, and connect the appropriate pins of the MCU to the Crystal.
One more information we found out was that we don't have to connect the Sensor pin and the Ground Pins, since we were not using any temperature sensor for the Crystal. This made it easier to apply the above mentioned solution.
This way we fixed this issue and we were able to test our BMA280 code and BLE code on our board.

ESD Diode:

Root cause of the hardware issue:
Choice of the ESD diodes were incorrect. Even though the clamping voltage was enough. There were some reasons that they were conducting below the clamping voltage.

Solution to Hardware issue:
We removed the ESD diode from our design. Tested our design without the ESD diode and only when we were assured, this would not cause any more problems, we moved forward to the next step.

Battery and Resistor for the Battery circuit:
Root cause of the hardware issue:
According to the Use case model, the Battery capacity should be 45mAH. But during the component-order state, the part was not in stock. Thus the ISET resistors value in the Battery Charger circuit (BQ24040) had to be changed. According to the Circuit we had ordered 4k Ohms resistor which should be providing a 45mA of charge current to the Battery. Since we could not find a 45mAH battery – we ordered a 40mAH and 110mAH from sparkfun. The issue was that the standard Charger Current of both the 40mAH and 110mAH is 0.2C, a max of 22mA is the standard charge current. But the resistors were offering a 45mA of current.

Solution to Hardware issue:
Hence, we had to change the resistor to 12K which provided us a lesser required current.

Lack of LEDs and extra sockets on unused GPIO pins in the MCU circuit for BMA80:
Root cause of the hardware issue:
When we are trying to test double tap and single tap interrupt in normal run mode, it is hard to identify if the interrupt is triggered unlike in debug mode we can put a breakpoint and read the register and confirm the interrupt status. As the accelerometer is not sensitive, the double tap has to be tapped in a specific way for it to be triggered. Hence it was hard to confirm if there was a double tap because of no LEDs or extra sockets in the MCU design to confirm the interrupt by making these GPIO pins high.

Solution to Hardware issue:
We solved this issue using software by passing a string, indicating , double tap to the BLE application to let us know that there was a double tap interrupt.

Lack of LEDs to test BLE Mesh:

Root cause of the hardware issue:

As above we did not have a LED to test BLE mesh on our board nor sockets for the GPIO pins

Solution to Hardware issue:

We connected an external LED to the GPIO pin socket of BMA280 power pin for which we are doing load power management. First, we connected it with a 1k resistor but the LED was not glowing. Then we realised as our voltage is only 2.5V, we removed the resistor and it provided enough current for the LED to glow and hence we could test the mesh application on our board.

BMA280 Sensor Issue:

Root cause of the hardware issue:

Since our goal was to gain the smallest wearable device, we searched for sensors with integrated functionalities. But the issue is that even though it is equipped with multiple functionalities it is not effective. To recognise a character in the Machine Learning, we need a lot of data points, with the BMA280 the max frequency achievable is 2Khz (Unfiltered - data) meaning that it even recognises the noise, But the maximum resolution achievable is 2g- with just 16bits of acceleration bits. The Tap sense on the other hand has to be with least possible resolution say at least a 16g with less frequency. If both the TAP and Accelerometer data is being taken at same frequency the accuracy with which the user uses the device will be completely undetermined.

With High frequency and more resolution, the tap sense has to be very fast that it recognises it as double tap, even with slightest latency it is considered as a data rather than tap sense.

With least frequency and less resolution, the accelerometer values will never be taken, and we end up getting very few data points - which is never enough for the machine learning algorithm.

Solution to Hardware issue:

This is the issue with the selection of the integrated sensor, we have no solution for the implementation now, but in future work, we can implement different effective sensors for each functionality

Functionality of the Final Project:

Two layer PCB design with optimum component density and desired functionality of the sensors are designed and brought up. We have tried achieving all the functionalities as proposed

The prototype uses a battery to provide power to our device. The battery is charged using inductive charging (wireless charging) which is a trending technology in the present day to charge cell phones.
We have alternate USB charging to charge the device in case of failure of inductive charging as a back-up charging circuit.
Our prototype works at 2.5 V (as compared to 3.3 V) which reduces the energy consumption.
We have used GPIO pin to power our sensor for load power management to achieve low power.
The sensor use SPI interface to interface with the MCU.
Double tap sense of the sensor enables the entire device for use.
After the tap, the user movements are recorded by the accelerometer sensor, it is sent to the application and it is trained by the machine learning algorithm to interpret the data and display it on the application.
In the mean while, temperature sensor , obtains the ambient temperature and sends it to the application and displays the same.

Wow Factor:

- With this product, we did not restrict ourselves to just the complex-Hardware implementation, the challenge flows even to the Software part of the project. We have implemented Machine Learning to recognise the characters by the accelerometer data being received from the accelerometer through the BLE. Our product shares equal complexity in both Hardware and Software and we have achieved both

- Another wow factor is that we have obtained a great density of the board, and as both of us being beginners in this domain, we are very proud to obtain this density and obtain the smallest size of the board in the class.

- ➢ Studying on Bluetooth Mesh which is a new technology and try to implement a simple functionality on our board.

- ➢ Attempting inductive charging which is not very common energy harvesting technique used for this class and getting it to work according to our project requirement is what we would consider as another wow factor.Inductive charging is one of the trending technology being used in the present day and we are glad to have got an opportunity to work on it.

Future Work

1. To use different effective sensors to achieve modularity and ease of use in different functionalities.
2. To test the machine learning algorithms in different planes and make it more robust in regression-based recognition
3. Once Silicon releases a mode to integration of BLE sdks with BLE MESH sdk, we can define each character recognised to a command and send the commands through BLE MESH. Since we already developed all three APPs and the Back-end code for the functionality, integrating BLE MESH with BLE will and should not take much time to implement.

References

1. https://datascience.stackexchange.com/questions/8732/algorithm-for-gesture-classification-in-a-wearable
2. https://powerbyproxi.com/wireless-charging/
3. https://www.youtube.com/watch?v=mRAWrXePAw0
4. https://www.technologyreview.com/s/601461/wireless-charging-is-actually-charging-ahead/
5. Bluetooth Mesh Networking/ An Introduction to Developers by Martin Woolley
6. **http://www.lipolbattery.com/lithium%20polymer%20battery.html**
7. **https://milliamps-watts.appspot.com/**