University of Colorado
Boulder

SELF-CHECKOUT SHOPPING CART

BY

DEEPESH SONIGRA
SATYA MEHTA
SIDDHANT JAJOO

UNDER THE GUIDANCE OF:

PROFESSOR TIMOTHY SCHERR, RANDALL SPALDING
UNIVERSITY OF COLORADO BOULDER

12/17/2019

TEAM NAME: ASHWATHAMA

TABLE OF CONTENTS

INTRODUCTION 3

PROJECT SOLVES THE FOLLOWING PROBLEMS 3

SOLUTIONS 3

FUNCTIONAL BLOCK DIAGRAM 4

BOARD BLOCK DIAGRAM 5

BOARD BLOCK DIAGRAM WITH PMU 5

SOFTWARE BLOCK DIAGRAM 6

KEY COMPONENTS 6

FEATURES 7

COMPONENT SELECTION 8

SPECIFICATIONS 10

PROCESSOR 11

- Programming and Debugging 11
- FIRMWARE ENVIRONMENT 12

EXTERNAL ENERGY SOURCE TO PROGRAM MCU 12

LIST OF FIRMWARE COMMANDS 13

POWER MANAGEMENT UNIT 14

- OVERVIEW 14
- C-RATE OF THE SPECIFIED LIPO BATTERY: 15
- I2C TIMING ANALYSIS 16
- PEAK DISCHARGE CURRENT 17
- RECHARGE TIME ANALYSIS 17
- BATTERY CYCLES 17
- CHARGE/DISCHARGE CYCLES 17
- BULK CAPACITANCE 19
- ACTUAL BOARD RESULTS FOR BUCK AND LDO 22
- ENERGY STORAGE ELEMENT 23
- Clock Generation Description: 24

USE CASE MODEL 27

ENERGY MODES 27
AVERAGE DC CURRENT MATH 28
PLANNED TEST POINTS 29
GANTT CHART 30
VERIFICATION REPORT 30
SIGNAL QUALITY ANALYSIS 30
SUMMARY 31
CHALLENGES FACED 32
LESSONS LEARNED 33
FUTURE SCOPE 34

INTRODUCTION

The "Self-Checkout Shopping Cart" is an innovative consumer purchasing product that is designed to help shoppers fast-track their shopping experience! The shopping cart has an inbuilt barcode scanner which can be used to scan the items to be purchased. The device communicates with the phone over the Bluetooth and bill is generated based on the items. Android app can be used for payment and faster checkout. With the advent of energy efficient devices and low power nodes, it has become imperative to design boards that consume low power which can last longer. To that end, we are designing nodes in order to consume minimal energy and address the issues mentioned below.

PROJECT SOLVES THE FOLLOWING PROBLEMS

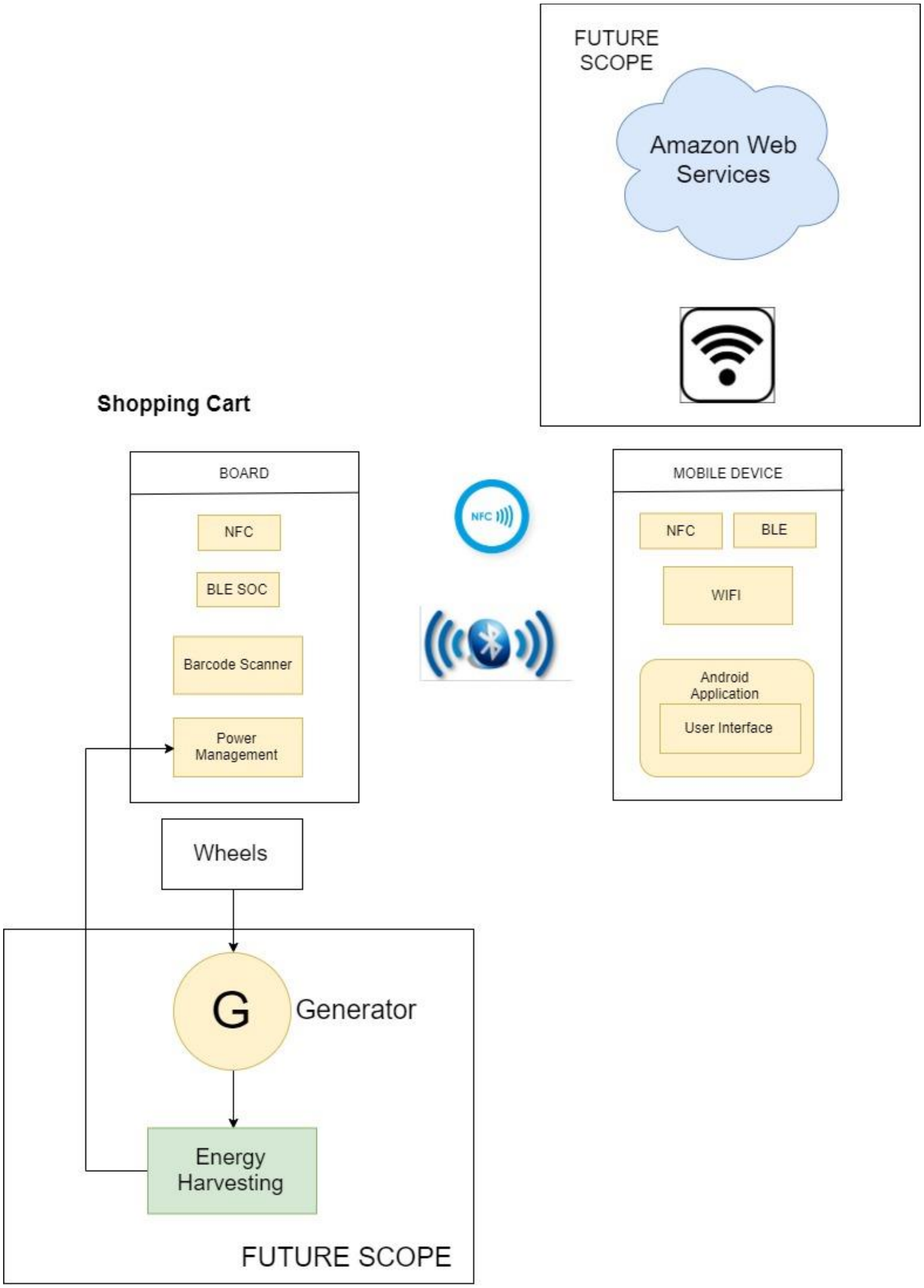
1. Customers usually get annoyed because of the long queues in the billing section of the huge shopping markets.
2. In addition to that keeping track of all the bills and budget is a very burdensome task.
3. Usage of lot of manpower in large supermarkets which can be expensive.
4. Stock management in supermarkets.

All these problems could be addressed by our "Self-Checkout Shopping Cart".

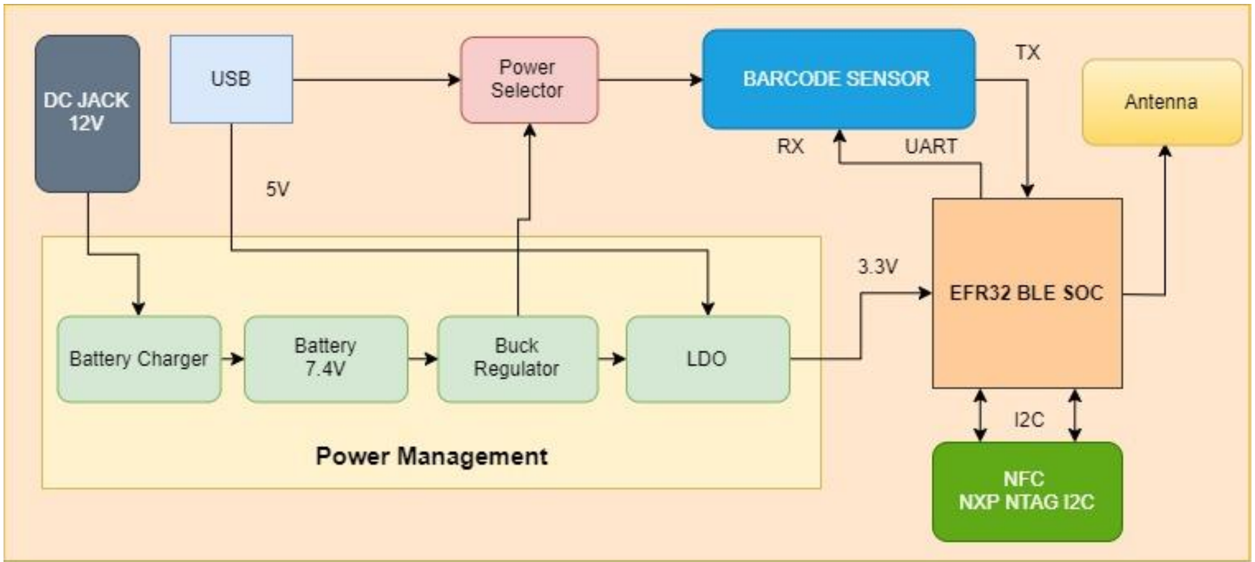
SOLUTIONS

1. Fast self-checkout saves time of customers and helps them buy items according to their budget.
2. Electronic bill is generated and saved in the cloud which makes it easy to keep track of all the bills and saves paper.
3. By letting customers handle their own scanning and bagging, workers can spend their time helping customers find what they need.
4. Better shopping experience for the customers and an innovative way for the sellers to attract customers.

FUNCTIONAL BLOCK DIAGRAM

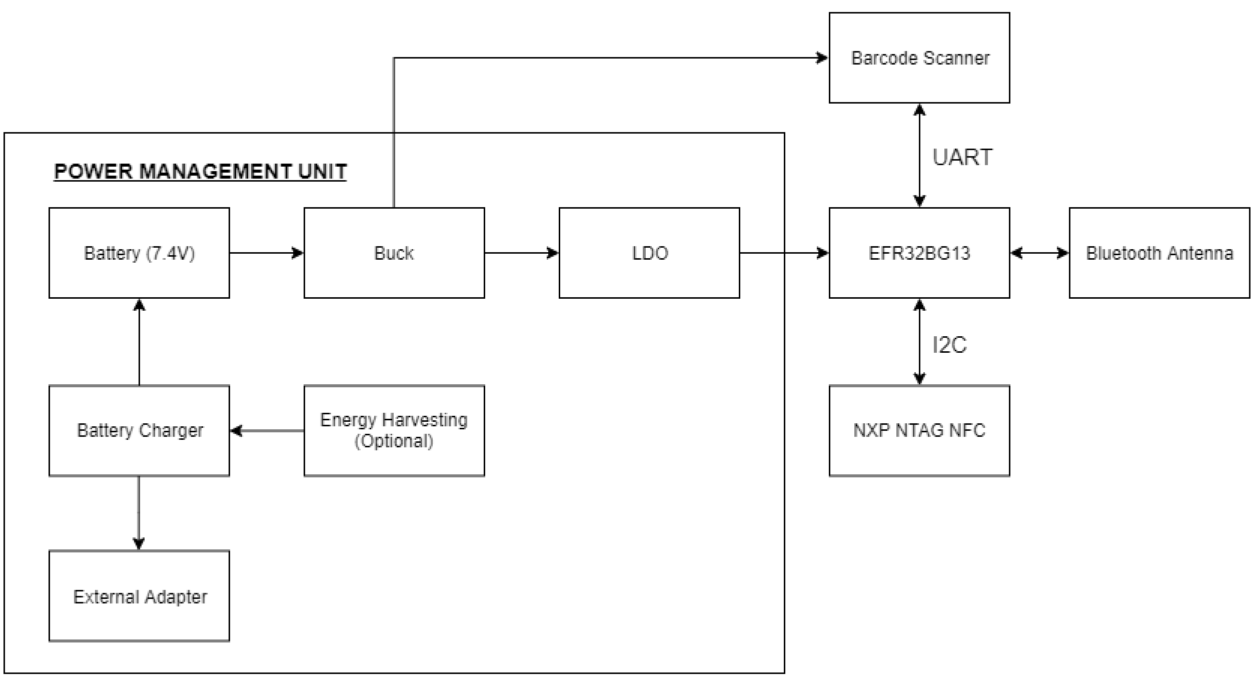


BOARD BLOCK DIAGRAM

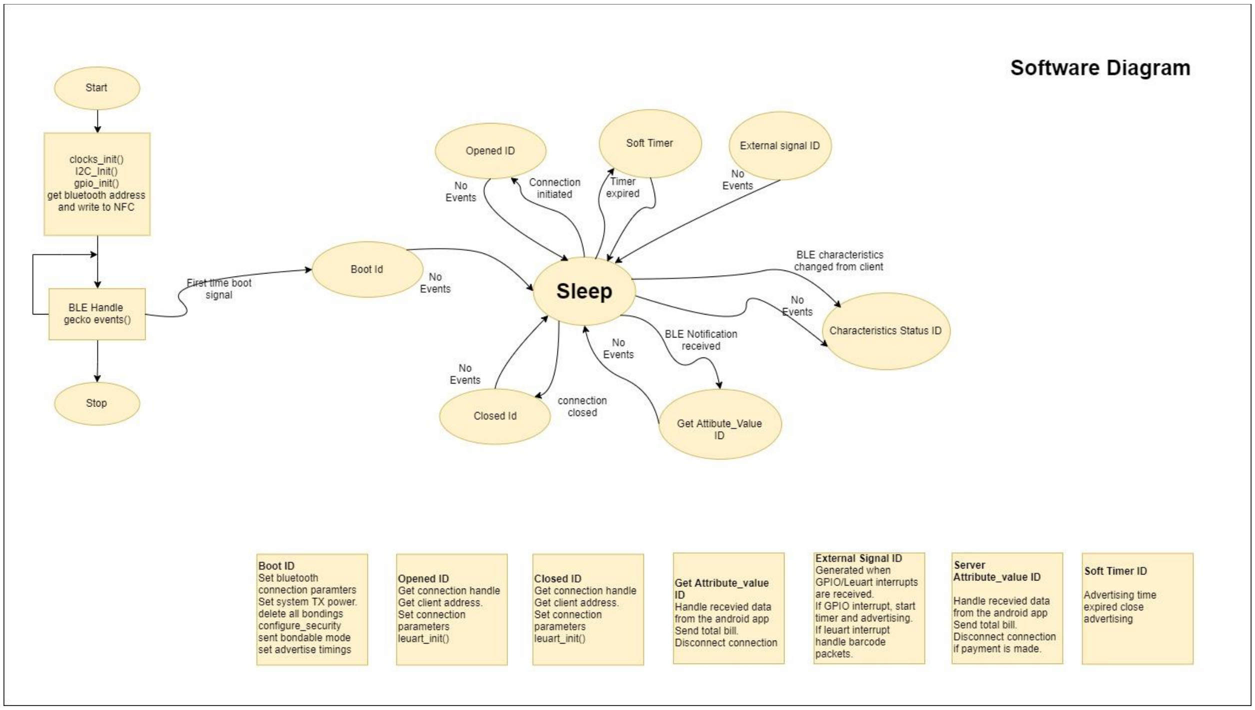


SELF-CHECKOUT SHOPPING CART

BOARD BLOCK DIAGRAM WITH PMU



SOFTWARE BLOCK DIAGRAM



KEY COMPONENTS

- Battery Management Circuit
- Buck Regulator
- LDO Regulator
- SOC (EFR32BG13)
- Barcode Sensor
- NXP NTAG NFC
- Debugger
- Antenna
- Android Application

FEATURES

- The device connects with an android application using Bluetooth and Bluetooth authentication/pairing is done using NFC for faster and secure connection.
- The Barcode Sensor scans QR codes/Barcodes to obtain the information of the product including the name and cost using Low Energy UART peripheral on EFR32BG13.
- The scanned item information is sent to the android application using BLE with the help of an onboard designed antenna.
- The Android application is capable to display all the scanned items and total price of the items in the cart.
- The Barcode scanner works on 5V and the NFC module works on 3.3 V which is obtained by using Buck and LDO respectively.
- The device is battery operated with recharging capability using a battery management circuit provided on the circuit. The Battery management circuit is powered by a 12V adapter in order to provide constant power to charge the 7.4V battery.
- A debugger is used in order to obtain debug printfs (VCOMM) for debugging purposes and to flash code on the processor.
- Several test points and isolation points have been provided on the board for debugging purposes.
- The board can also be powered through other sources via USB.

COMPONENT SELECTION

1. **Processor - Blue Gecko EFR32BG13:** This is a processor (SOC) sold by Silicon Labs, which has Bluetooth stack built into it with peripherals such as I2C, SPI, UART and 31 GPIO pins. This processor would serve as a controller for the entire system and would also be responsible for executing commands related to the Bluetooth stack. The SOC supports low energy modes and has provisions for load power management (i.e EM0, EM1, EM2, EM3, EM4) thus making it the appropriate choice for a low power project.
   The EFR32BG13 Blue Gecko Bluetooth Low Energy SOC Family Data sheet can be found here - https://www.silabs.com/documents/public/data-sheets/efr32bg13-datasheet.pdf
   Reference Manual: https://www.silabs.com/documents/public/reference-manuals/efr32xg13-rm.pdf
   Mouser Link to purchase: Purchase here
   The Blue gecko to be used for this project is: EFR32BG13P632F512GM48-D.
2. **NFC Module - NXP NTAG NFC Module:** This NFC module would be used for hands-free quick Bluetooth pairing authentication between the cart and the mobile phone. The interface used for NFC module is I2C which is available in the processor selected above. This module can interact with an NFC device as well as unpowered NFC chips such as tags, stickers, key fobs and cards which do not require batteries. This module also has the NFC energy harvesting capability if required. The module also consists of different modes for energy saving purposes.
   Digikey Link to purchase: Purchase here
   The datasheet for this module can be found here:
   https://www.nxp.com/docs/en/data-sheet/NT3H2111_2211.pdf
3. **Sensor - Barcode Scanner:** This barcode scanner was chosen because it gave us the lowest sleep mode current in comparison to all the other bar code scanners. It consists of a USB and UART interface. The module has different modes such as sleep, standby and scanning modes for load power management. There are very small amounts of portable barcode scanners available in the market which consume low current and most of them support UART or RS232 interface. Thus, we did not have much flexibility in choosing an interface for the barcode scanner.
   Link to purchase: Purchase here
   The user manual/datasheet for this module can be found here:
   https://www.waveshare.com/w/upload/3/3c/Barcode_Scanner_Module_User_Manual_EN.pdf
4. **Energy Storage Element: Battery - Lipo Lithium Ion battery:** Since our sensor operates on 5V, it made sense to have a battery whose voltage level is more than 5V. The battery has a 1000mAh capacity. This battery is a dual cell battery which consists of two 3.7V

cells in series. This battery specification is subject to modifications as per the power calculations in subsequent weeks.

Sparkfun Link to purchase: Purchase here

Battery Manual:

https://cdn.sparkfun.com/datasheets/Prototyping/Lithium%20Ion%20Battery%20MSDS.pdf

5. **Power Management IC:**

   a. Buck Converter (TPS560430XFDBVR) -

   The battery specified above has a maximum voltage rating of 7.4 voltage and can drop till a minimum of 6 volts for the system to remain operational. The Barcode Scanner sensor would require a rail voltage of 5V. Thus, it is required to have a buck converter and not a buck-boost converter since the voltage would never drop below 5V. The buck converter would be able to provide a rail voltage of 5V at battery voltages above 5V.

   Digikey Link to purchase: Purchase here

   Datasheet:

   http://www.ti.com/general/docs/suppproductinfo.tsp?distId=10&gotoUrl=http%3A%2F%2Fwww.ti.com%2Flit%2Fgpn%2Ftps560430

   b. Low Dropout Voltage Regulator (LDO) (LM117) -

   The NXP NTAG - NFC reader and the processor have an operating voltage 0V-4.6V and 1.8V-3.8V respectively. Since the minimum operating voltage for the battery is 4V, it was apt to use an LDO or a buck converter instead of a buck-boost converter to generate a rail voltage of 3.3V. Voltage dropout consideration is not required in this case as there is a wide gap between the minimum operating battery voltage and the rail voltage required by barcode sensor. In terms of efficiency for our application, the LDO provides a respectable efficiency as compared to a buck converter. Since the LDO provides the smallest solution footprint and is the cheapest option, LDO was selected in order to obtain a 3.3V for the barcode sensor.

   Digikey Link to purchase: Purchase here

   Datasheet:

   http://www.ti.com/general/docs/suppproductinfo.tsp?distId=10&gotoUrl=http%3A%2F%2Fwww.ti.com%2Flit%2Fgpn%2Flm1117

   c. Battery Charge Management IC (MCP73213) -

   The MCP73213 is a highly integrated Li-Ion battery charge management controller for use in space-limited and cost-sensitive applications. This IC would be required in order to charge the Lipo battery using energy harvesting methods or using USB. Since charging requires a constant current and voltage across the

battery, it is imperative to use an IC which provides regulated voltage and current. In addition to this, the 7.4V battery would require voltage greater than 7.4V for charging it. This IC would convert the 5V supplied by the USB to Output Voltage options such as 8.20V, 8.4V, 8.7V, 8.8V. This IC support fast charging mode which can charge 7.4V battery by sourcing current upto 1100mA which can fully charge our battery in an hour.
Digikey Link to purchase: Purchase here
Datasheet: http://ww1.microchip.com/downloads/en/devicedoc/20002190c.pdf

SPECIFICATIONS

1. Processor (SOC) - Blue Gecko EFR32BG13P732F512GM48-D, Operating Voltage: 1.8V-3.8V freq band: 2.4 Ghz@19 dBm, Flash: 512kB, RAM: 64kB, GPIO: 31, Operating Range: -40°C to 85°C
2. NFC Module: Working Voltage: Baud Rate: 115200 bps, Operating Voltage: 0V-4.6V, Operating range: 2 to 5 cm, Size: 11cm x 5cm
3. Sensor-Barcode Scanner: Operating Voltage: 5V, Operating Temperature: 0°C - 50°C, Size: 53.3mm x 21.4mm
4. Buck Converter: Input Voltage Range: 4V to 36V, Output Voltage: 2.5V to 9V, 2A output current in buck mode, Size: 2.90mm x 1.60mm
5. Low Dropout Voltage Regulator (LDO): Output Voltage: 3.3V, Output Current: 800mA, Size: 6.50mm x 3.50mm
6. Battery Charge Management IC (MCP73213): Battery Charge Voltage Option: 8.20V, 8.4V, 8.7V, 8.8V, fast charge current: 130mA -1100mA, Size: 3mm x 3mm.
7. Battery - Weight: 85g, Size: 70mm x 35mm x 18mm
8. Dimensions: 120mm x 50mm (Approx)
9. Battery: 7V Battery (1) - Rechargeable
10. Wireless Range: 60 meters /180ft
11. Temperature Range: 0 - 50°C
12. Temperature Accuracy: Typical: ±0.3°C /± 0.5°F, Max: ±0.5°C /±0.9°F3
13. Humidity Range: 0~99%RH
14. Humidity Accuracy (25°C/ 77°F, 20%~80%RH): Typical: ±3%RH, Max: ±4.5%RH
15. Warranty: 2-3 years.

PROCESSOR

Programming and Debugging

The silicon labs MCU and wireless starter kits provides a powerful development and debug environment. Debug capabilities include serial wire debug, JTAG, C2 interface, embedded trace microcell (EDM), advanced energy monitoring (AEM), packet trace interface (PTI), virtual comp port and virtual UART. All these features can be included by a 20 pin debug connector and a 20 pin simplicity connector provided on a WSTK can be used. For space constraint design, a mini simplicity connector is being used.

The target microcontroller/SoC will be programmed using this debug connector with the help of Silicon labs IDE – Simplicity Studio and a development board.

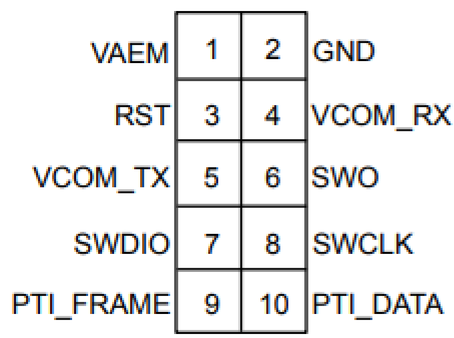


Figure 4.2. Mini Simplicity Connector Pin-Out

The mini simplicity connector has the following capabilities:

- SWD
- AEM
- PTI
- VCOMM

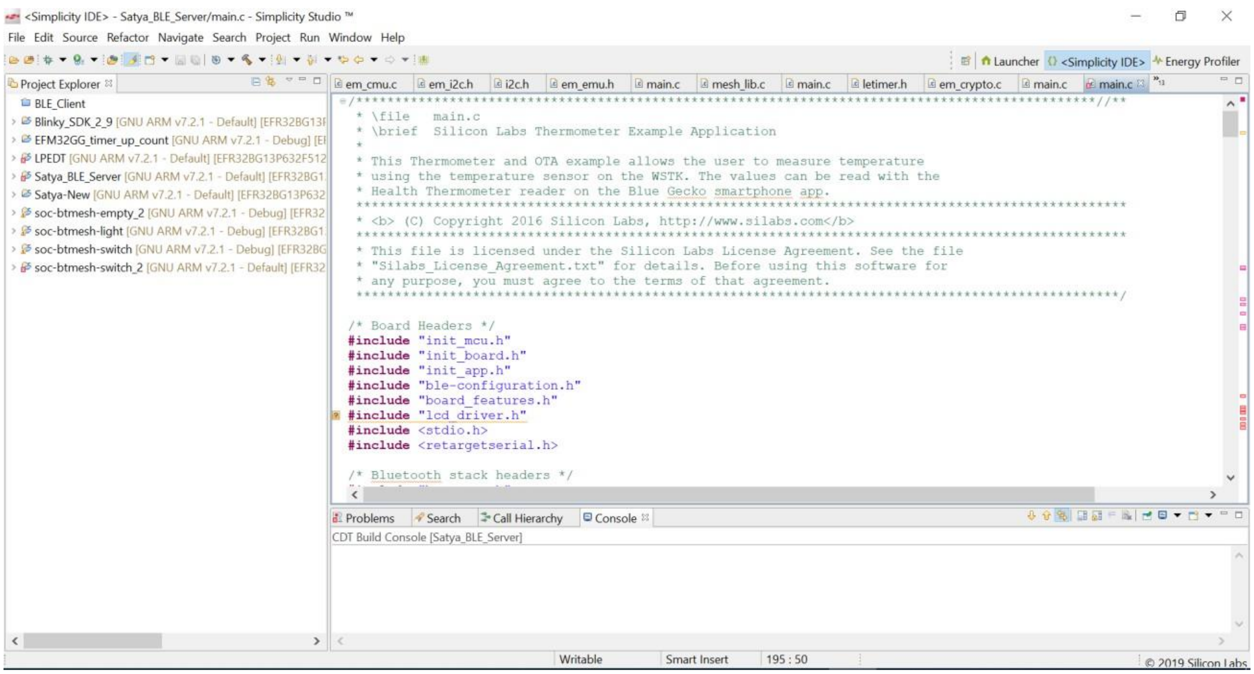
Feature	20-pin Standard ARM Cortex Debug+ETM Connector	20-pin Simplicity Connector	Simplicity Debug Adapter Board Interfaces (Standard or Tag-Connect 10-pin cable)			Tag-Connect 6-pin Interface
			Mini Simplicity Connector	Cortex Debug Connector	ISA3 Packet Trace Port Connector[1]	
SWD (serial wire debug)	X		X	X	X	X
JTAG	X			X	X	
C2	X			X		
ETM (embedded trace module)	X					
AEM (advanced energy monitoring)		X	X			
PTI (packet trace interface)		X	X		X	
VCOM (virtual COM port)		X	X			
Virtual UART	X		X	X	X	X

Note:
1. ISA3 Packet Trace Port Connector interface is not yet supported by the WSTK.

The above image is the connector to be used for programming and debugging.

FIRMWARE ENVIRONMENT

- Simplicity Studio
- Blue Gecko
- NXP NTAG Explorer Kit
- Barcode Sensor
- MCP73213 Batter Charger Evaluation Board
- TPS560430 Evaluation Board



EXTERNAL ENERGY SOURCE TO PROGRAM MCU

Since WSTK is used in order to program the board using the debug connector, the debug connector does supply power from the development board. So technically, external energy source is being used in order to program the code. In order to achieve this, we have isolated the PMU to prevent backflow current.

But there is also a provision to program the code using the PMU of our embedded device by just switching the switch to BATT mode on the development board. This would prevent the development board to source energy to our embedded device.

LIST OF FIRMWARE COMMANDS

Firmware API:

- bt_connection_init() - This function initializes the bluetooth connection.
  The function prints the server address, deletes all previous bondings, sets into bonding mode, sets the transmit power to zero, configures and starts advertising.
- barcode_packet_create() - This function creates a barcode packet structure as per the data received from the leuart_buffer on pop() and returns the payload size to a pointer passed as a parameter.
- gpio_init() - This function initializes the GPIO pins for NFC interrupts.
- leuart_init() - Initialization function for LEUART0. This function enables the required clock and GPIO peripherals.
- leuart_buffer_push() - This function pushes the data received over UART into the Circular Buffer. i.e leuart_circbuff. Note: This function must be enclosed in CORE_AtomicDisableIrq() and CORE_AtomicEnableIrq().
- leuart_buffer_pop() - This function pops and returns the data from the Circular Buffer. i.e leuart_circbuff. Note: The function return -1 when the buffer is empty. The received data needs to be typecasted to a signed integer to print the -1 value.
- i2c_init() - This function is used to initialize I2C0 peripheral. PortC 10 SCL, PortC 11 SDA is used.
- i2c_write_poll() - This function is a polling write driver for NXP NTAG I2C NFC Device.
  16 Bytes are written in a single I2C write transfer.
- i2c_read_poll() - This function is a polling read driver for NXP NTAG I2C NFC Device.
  16 Bytes are read in a single I2C read transfer.
- i2c_disable() - This function is used to disable the I2C peripheral.
- leuart_disable() - This function is used to disable the LEUART0 peripheral.

USER INSTRUCTIONS:

1. Place the mobile phone with NFC enabled and consisting of the android application provided near the NFC module on the embedded device.
2. The mobile phone will then automatically connect to the device over Bluetooth.
3. Scan as many products as the user wants using the onboard barcode sensor.
4. Finally, press the Bill button on the android application to generate the bill followed by pressing the Payment button in order to proceed to payment.

POWER MANAGEMENT UNIT

OVERVIEW

The Power management unit would consist of a circuitry to supply regulated power supply since the barcode scanner requires a rail voltage of 5V, NXP NTAG requires a voltage of 0V-4.6V and processor requires a voltage between 1.8V-3.6V. Thus, NXP NTAG and the processor have wide voltage inputs. The operating voltages of all the ICs have been specified in the specifications section. In order to generate all the rail voltages according to different sensors and processor input voltage requirements, buck and LDO is required.

The battery specified above has a maximum voltage rating of 7.4V and can drop till a minimum of 6 volts for the system to remain operational. The Barcode Scanner sensor would require a rail voltage of 5V. Thus, it is required to have a buck converter in order to improve the battery life of the system. The buck converter would be able to provide a rail voltage of 5V at battery voltages above 5V.

LDO would be required for NXP NTAG - NFC and the processor as their operating voltages are predominantly below 4V. The LDO would be attached in series with the Buck converter in order to convert the 5V obtained from the Buck converter to 3.3V for the NXP NTAG and the EFR32BG13 processor.

There were two possible options to obtain the 5V and 3.3V from the power supply:

1. Having the buck and LDO in parallel and generating the rail voltages independently.
2. Having Buck and LDO in series so that the LDO generates a rail voltage of 3.3V following the buck converter which generates 5V.

The first option dissipated a lot of heat as compared to the second one. Reducing the voltage is done at the expense of heat, the LDO drop from 7.4V to 3.3V proved to be a lot expensive in terms of power dissipation. The efficiency of LDO was not even 50% which implied that the heat dissipation would require a large heat sink as well. Assuming the efficiency of Buck to be 80%, we did come up with some numbers to justify the series connection. The calculations are done for constant power mode, so we assume current values to demonstrate the selection.

1. **In case of parallel connection:**

   In case of Buck (Output Voltage = 5V):

   Assuming buck output voltage of 5V and a current of 2mA, Power output required = V*I = 5*2 = 10uW.

   Assuming 80% efficiency for the buck converter, the input power required would be 10/0.8 = 12.5uW. Thus 12.5uW = 7.4V x 1.6 mA.

Thus, extra power required would be 12.5 – 10.0 = 2.5 uW.

Now for LDO (Output Voltage = 3.3V):
Assuming the current to be 3.5mA, Power output required = V*I = 3.3V * 3.5mA = 11.55 uW.
Similarly Input Power = 7.4V * 3.5 mA = 25.9 uW.
Thus, extra power required would be 25.9 – 11.55 = 14.35 uW.

Thus, the total power lost in heat would be 14.35 + 2.5 = 16.85 uW.

2. **In case of series Connection:**
   In case of Buck (Output Voltage = 5V):

   Assuming buck output voltage of 5V and a current of (2 + 3.5)mA, Power Output required = V*I = 5*5.5 = 27.5uW.
   Assuming 80% efficiency for the buck converter, the input power required would be 27.5/0.8 = 34.35uW. This 34.35uW = 7.4V x 4.64 mA. 4.64mA would be the input current.
   Thus, extra power required would be 34.35 – 27.5 = 6.85 uW.

   Now for LDO (Output Voltage = 3.3V):
   Assuming the current to be 3.5mA, Power output required = 3.3V * 3.5mA = 11.55 uW.
   Similarly Input Power = 5V * 3.5 mA = 17.5 uW.
   Thus, extra power required would be 17.5 – 11.55 = 5.95 uW.

   Thus, the total power lost in heat would be 6.85 + 5.95 = 12.8 uW.

As you can see the power lost in case of parallel connection is 16.85 uW and in case of series connection is 12.8 uW. This proves that the power lost in series connection is less than the parallel connection. In addition to this, the total input power required is 38.4 uW and 34.35 uW for parallel and series connection respectively.
Thus, it is clear that series connection is better than the parallel connection in terms of power dissipation.

C-RATE OF THE SPECIFIED LIPO BATTERY:

C-Rate is used to express the magnitude of the charge or discharge current; expressed as a multiple of the battery rated capacity multiplied by the current. In general, the charge or discharge current is expressed as a multiple of C.

C-Rate = Average current/Battery Capacity = 9.08mA / 1000 mah = 0.0908 C

Peak discharge rate out of battery:
Peak Current/Battery Capacity = 138.195 (Mode 4)/1000 = 0.138 C

Based on our lithium battery discharge curve, the lowest nominal voltage is 7.2V.
The battery cut off voltage of our circuit is 6V.
This nominal voltage will require a buck only solution.
6.1V will be programmed as the cut off voltage from the battery of the PMU circuit.
The difference between the nominal and programmed voltage is due to the voltage drop across the buck converter.
The battery specified above has a maximum voltage rating of 7.4 voltage and can drop till a minimum of 6 volts for the system to remain operational. The Barcode scanner sensor would require a rail voltage of 5V. Thus, it is required to have a buck converter and not a buck-boost converter since the voltage would never drop below 5V. The buck converter would be able to provide a rail voltage of 5V at battery voltages above 5V.

I2C TIMING ANALYSIS

I2C timing table as found in the NXP NTAG datasheet:

Symbol	Parameter	Conditions	Standard-mode		Fast-mode		Fast-mode Plus		Unit
			Min	Max	Min	Max	Min	Max	
f_{SCL}	SCL clock frequency		0	100	0	400	0	1000	kHz
$t_{HD;STA}$	hold time (repeated) START condition	After this period, the first clock pulse is generated.	4.0	-	0.6	-	0.26	-	µs
t_{LOW}	LOW period of the SCL clock		4.7	-	1.3	-	0.5	-	µs
t_{HIGH}	HIGH period of the SCL clock		4.0	-	0.6	-	0.26	-	µs
$t_{SU;STA}$	set-up time for a repeated START condition		4.7	-	0.6	-	0.26	-	µs
$t_{HD;DAT}$	data hold time[2]	CBUS compatible masters (see Remark in Section 4.1)	5.0	-	-	-	-	-	µs
		I^2C-bus devices	0[3]	-[4]	0[3]	-[4]	0	-	µs
$t_{SU;DAT}$	data set-up time		250	-	100[5]	-	50	-	ns
t_r	rise time of both SDA and SCL signals		-	1000	20	300	-	120	ns
t_f	fall time of both SDA and SCL signals[3][6][7][8]		-	300	20 × (V_{DD} / 5.5 V)	300	20 × (V_{DD} / 5.5 V)[9]	120[8]	ns
$t_{SU;STO}$	set-up time for STOP condition		4.0	-	0.6	-	0.26	-	µs
t_{BUF}	bus free time between a STOP and START condition		4.7	-	1.3	-	0.5	-	µs
C_b	capacitive load for each bus line[10]		-	400	-	400	-	550	pF
$t_{VD;DAT}$	data valid time[11]		-	3.45[4]	-	0.9[4]	-	0.45[4]	µs
$t_{VD;ACK}$	data valid acknowledge time[12]		-	3.45[4]	-	0.9[4]	-	0.45[4]	µs
V_{nL}	noise margin at the LOW level	for each connected device (including hysteresis)	$0.1V_{DD}$	-	$0.1V_{DD}$	-	$0.1V_{DD}$	-	V
V_{nH}	noise margin at the HIGH level	for each connected device (including hysteresis)	$0.2V_{DD}$	-	$0.2V_{DD}$	-	$0.2V_{DD}$	-	V

Time required for 1 I2C transmission for 1 packet = tsetup_start + tdata/ack + trise + tfall + tdata/ack + tdata_hold + tsetup_stop = (4.7 + 3.45 + 1.3 + 3.45 + 5 + 4) us = 21.9 us.

For UART, the baudrate chosen is 9600 since we would be using the LEUART peripheral of EFR32BG13 which runs at a baudrate of 9600.

PEAK DISCHARGE CURRENT

The battery chosen is capable of providing the peak discharge current as specified in the datasheet which is 0.5C which is equal to 500 mA and our peak current is 170mA.

RECHARGE TIME ANALYSIS

Standard charge time for charging cells to their rated capacity is 2.5 hours at a charging voltage of 4.20 V/cell and charging current of 1C.

1. Charge characteristics
1-[1] Charge characteristics
Figure 3 shows the charging voltage, charging current, and charging capacity when charging under constant-voltage, constant-current conditions (maximum charging voltage 4.2V, maximum charging current 720mA, ambient temperature 23℃).

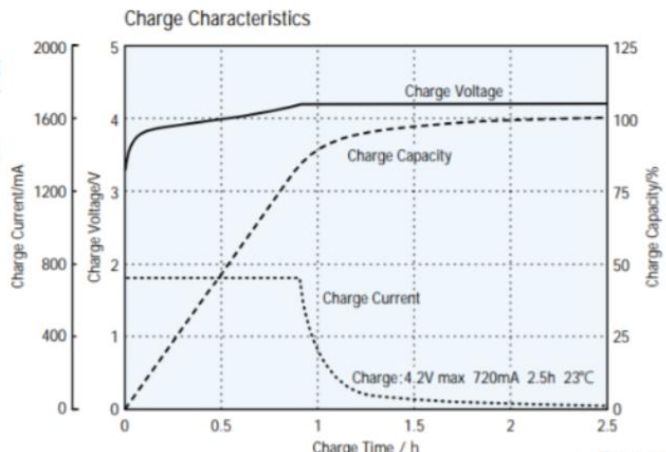


Figure 3

Shows for single cell. We will charge with 8.4V and 500mA current constant.

BATTERY CYCLES

Battery should last more than 500 charge cycles if used under given charge/discharge specifications.

CHARGE/DISCHARGE CYCLES

We are planning to use 80% (800mAh) of the battery storage per discharge cycle. The voltage drops to around 3V/Cell that is 6V at 800mAh discharge capacity.

2. Discharge characteristics

2-[1] Discharge characteristics on load

Figure 6 shows changes in the battery voltage for constant-current discharge at an ambient temperature of 23℃, with the discharge current at 145mA, 360mA, 720mA and 1440mA.

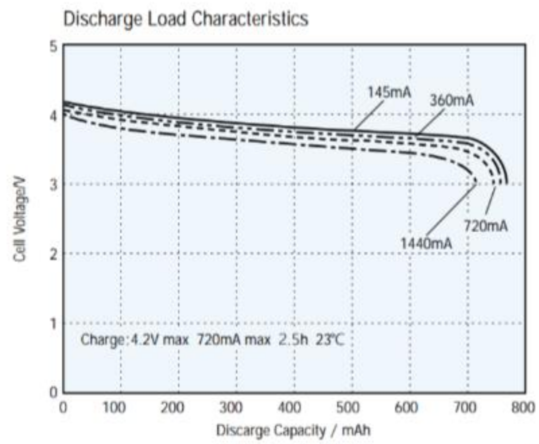


Figure 6

As charge/discharge cycles are repeated, the battery capacity (ability to hold a charge) gradually declines. However, when batteries are charged and discharged under the conditions recommended by Sony, they can be used for 500 or more charge/discharge cycles. The maximum voltage for charging is 8.4 V, and the cutoff voltage in discharge is 5 V (for hard carbon batteries) and 6.0 V (for graphite batteries with cobalt oxide cathode).

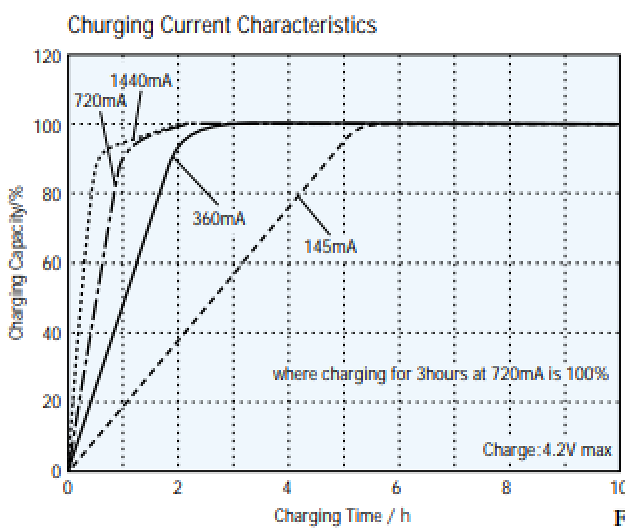


Figure 4

Maximum charging voltage is 4.2V and maximum charging current is 720mA. Battery's peak current is 1.7 A and the peak current required by our system (Mode 4) is 138mA.

(Provide by LDO)

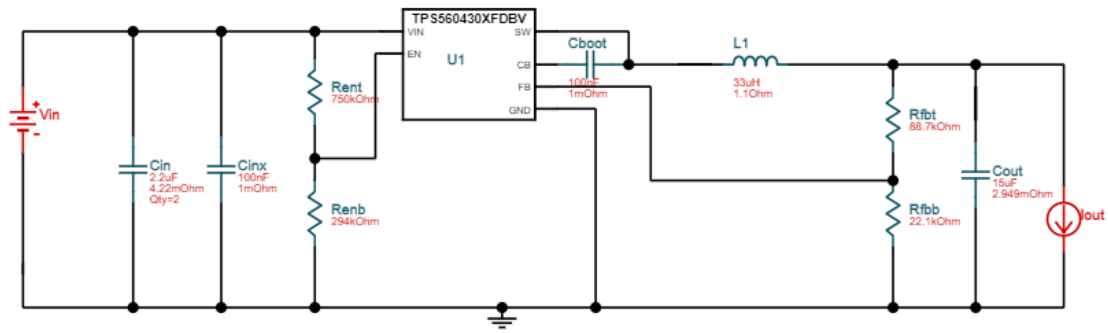
Maximum Minimum Operating Voltage: 1.8V

Minimum Maximum Operating Voltage: 3.8V

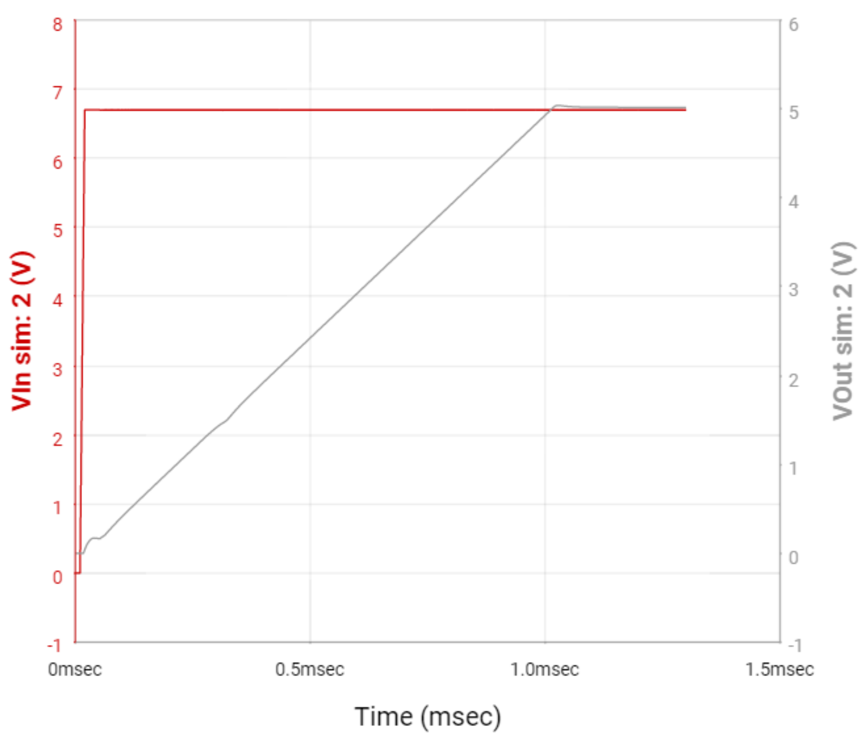
The minimum voltage required for the system to function would be 5V since the barcode scanner only works if it is provided a rail voltage of 5V. This would be provided by the buck converter.

BULK CAPACITANCE

1. Buck Regulator

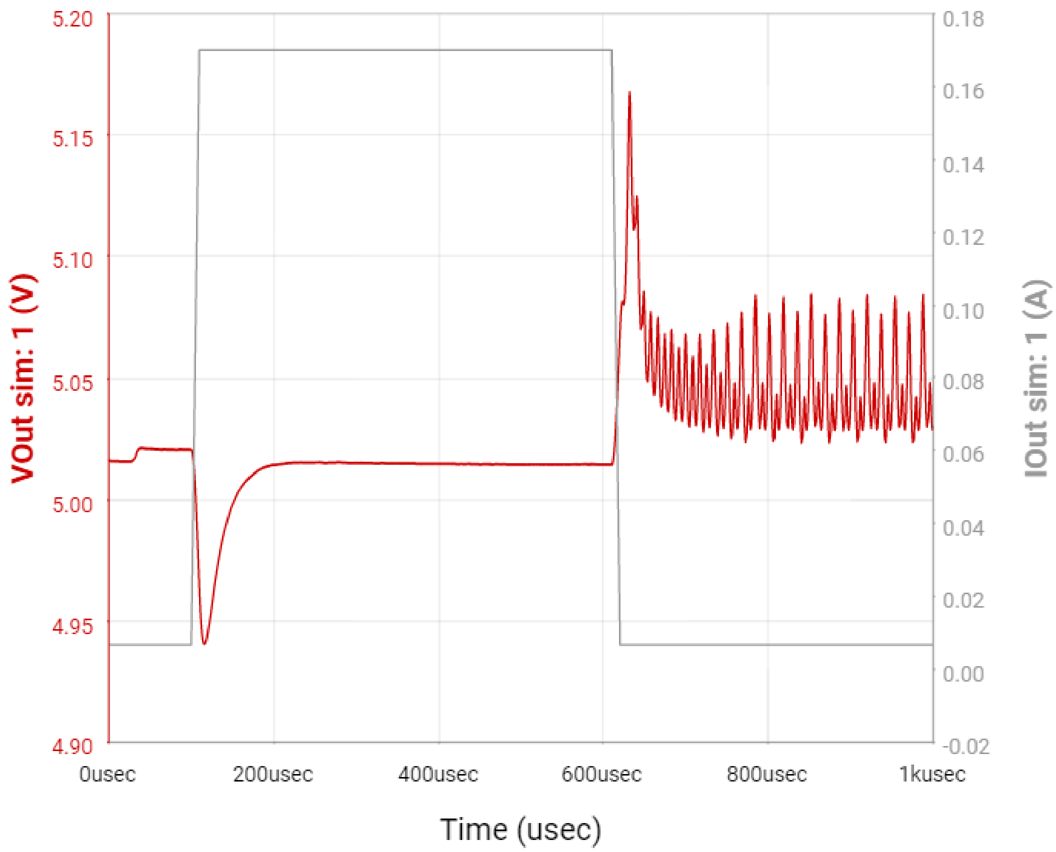


Above image is the circuit for buck regulator



The above image is the startup graph i.e when the system boots.

LOAD TRANSIENT:



Performance Summary

Sim ID	Vout Maximum	Vout Minimum	Undershoot Settle Time	Overshoot Settle Time	Undershoot	Overshoot
1	5.17 V	4.94 V	0.08 ms	0.08 ms	0.07 V	0.13 V

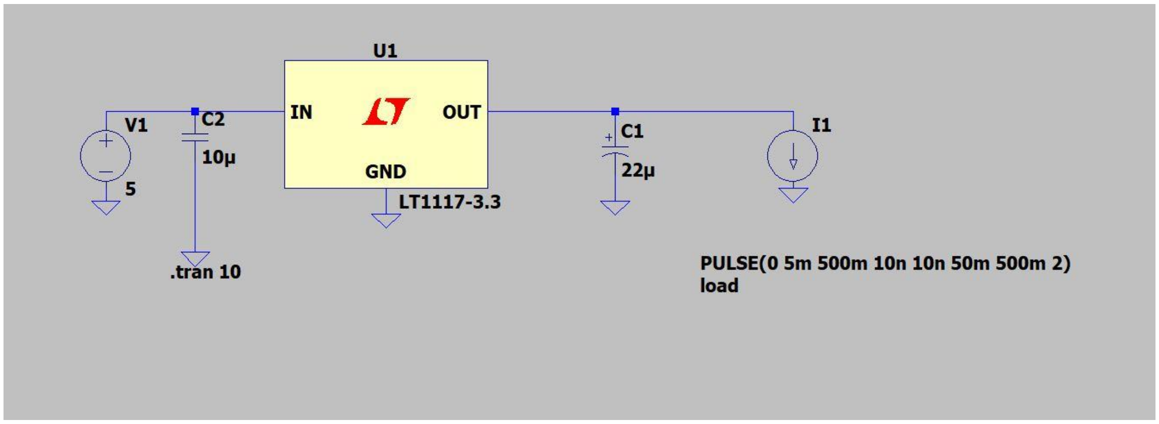
The minimum Vout is 4.94V which is well above the minimum requirement of 4.8V of the sensor.

The maximum Vout is 5.17 V which will be handled by the onchip AMS117 regulator in the barcode sensor.

Buck Capacitance = 15 uF (Part Number - KCM55LR71E156KH01)

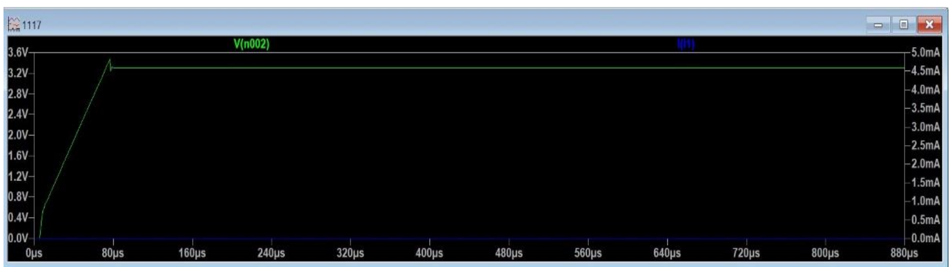
At DC bias 5V, capacitance is 15.64 uF.

2. LDO Regulator



Our Vin = min = max = 5V since the LDO is in series with the buck regulator.

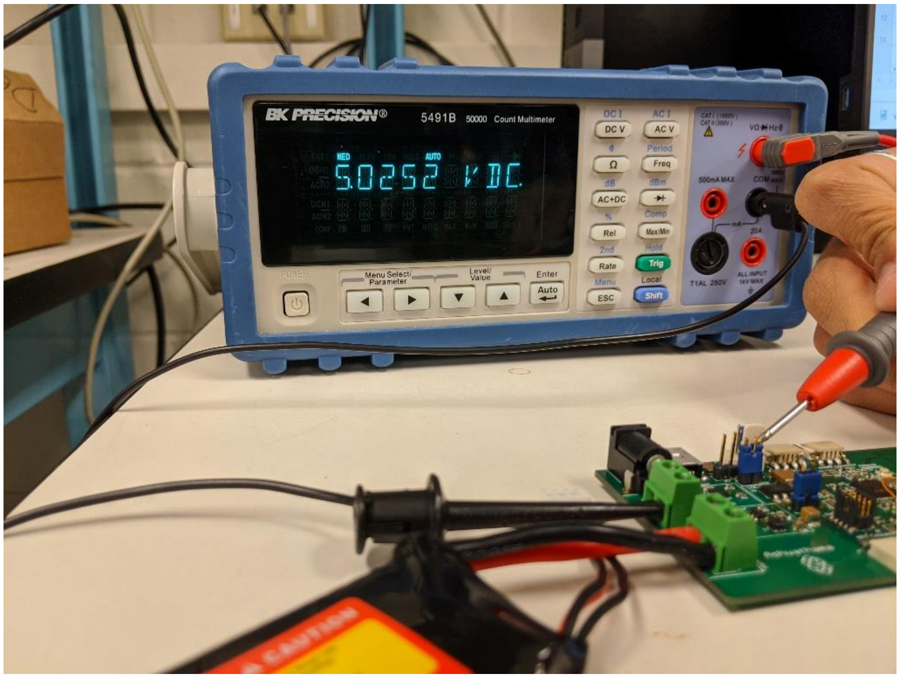
At Vin = 5V



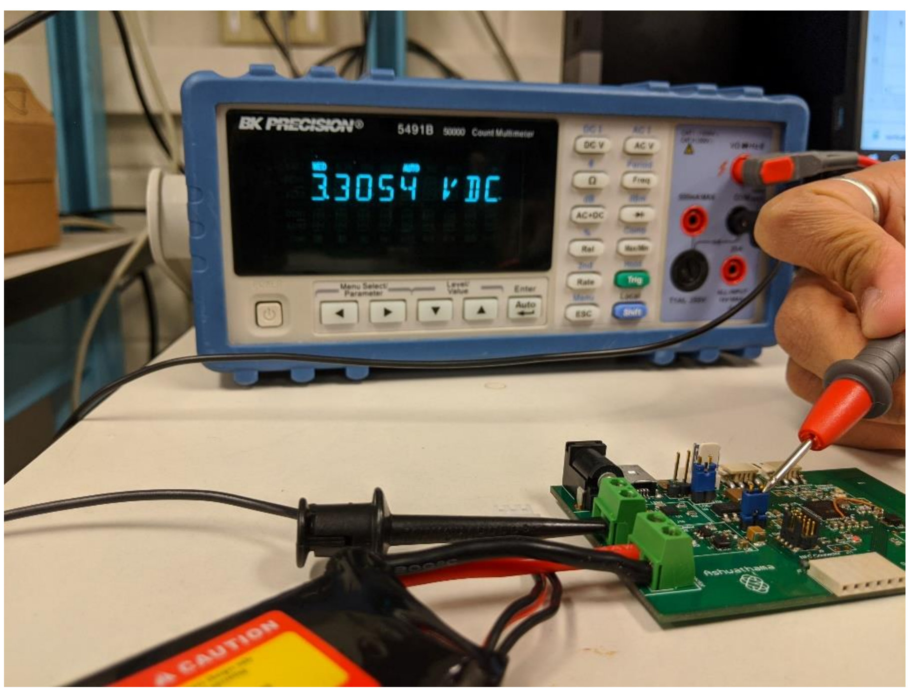
Capacitance chosen = 22 uF (Part Number - GRM32ER71C226KEA8)
At DC bias 3.3V, capacitance is 22.61 uF.

ACTUAL BOARD RESULTS FOR BUCK AND LDO

Buck Output Voltage is 5 Volts as expected.



LDO Output Voltage is 3.3 Volts as expected.

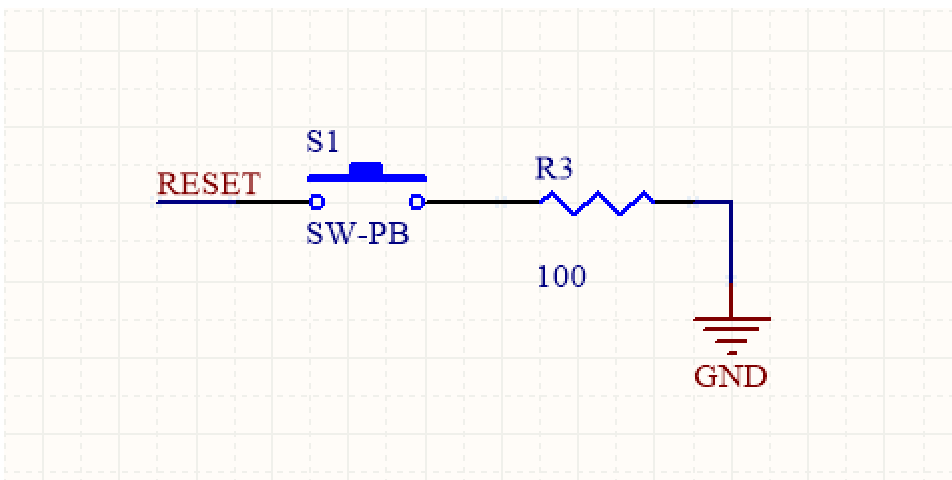


ENERGY STORAGE ELEMENT

The Energy Storage Element used would be a battery as specified in the component selection section.

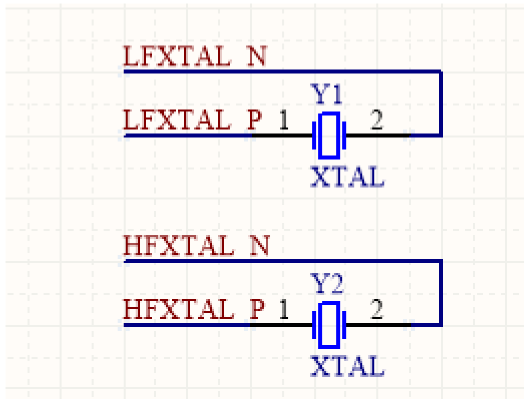
COMPONENT	ENERGY MODE	CURRENT(mA)
NXP NTAG – NFC	I2C (Idle Mode)	0.195
	I2C (Active Mode)	0.24
Barcode Scanner	Sleep Mode	2
	Active Mode	135
Processor - EFR32BG13	EM2 (I2C)	0.002
	EM1 (UART)	3
	EM0 (Radio on)	10

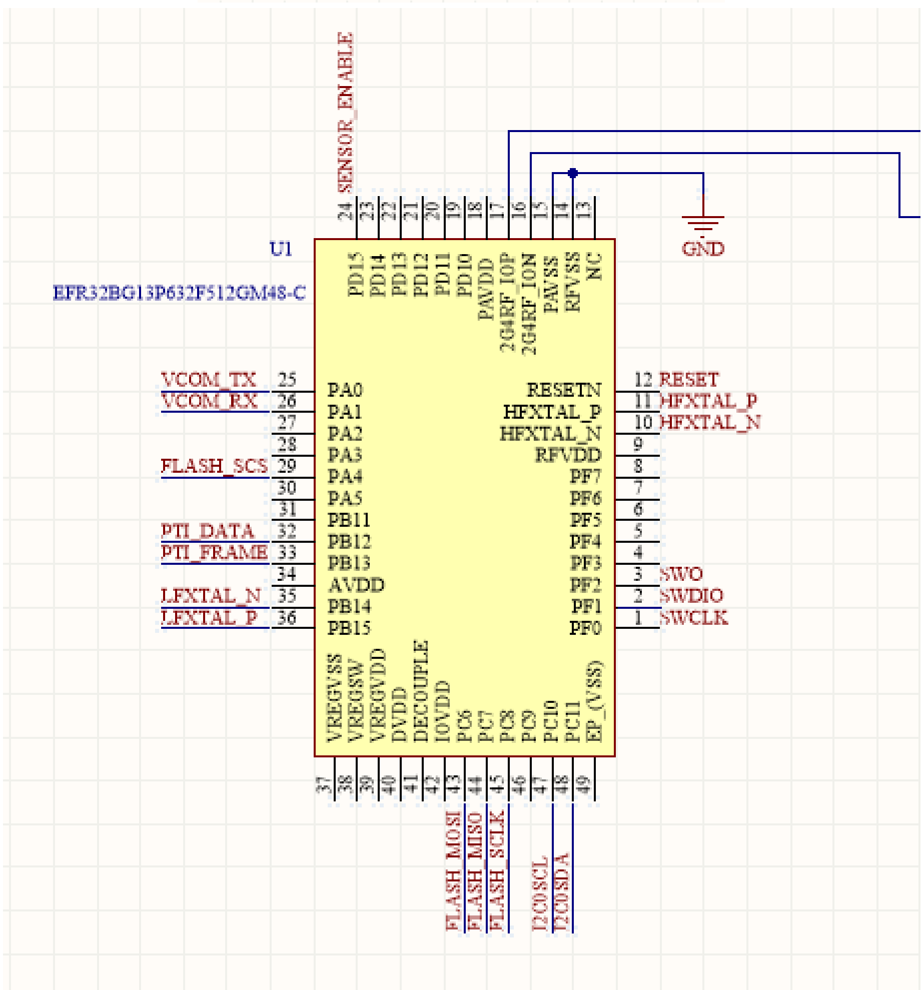
Reset Circuit Description:



Taking the schematic design of silicon labs as a reference, the above image is the design for our reset pin. This pin is internally pulled up to AVDD. To apply an external reset source to this pin, it is required to only drive this pin low during reset, and let the internal pull-up ensure that reset is released.

Clock Generation Description:





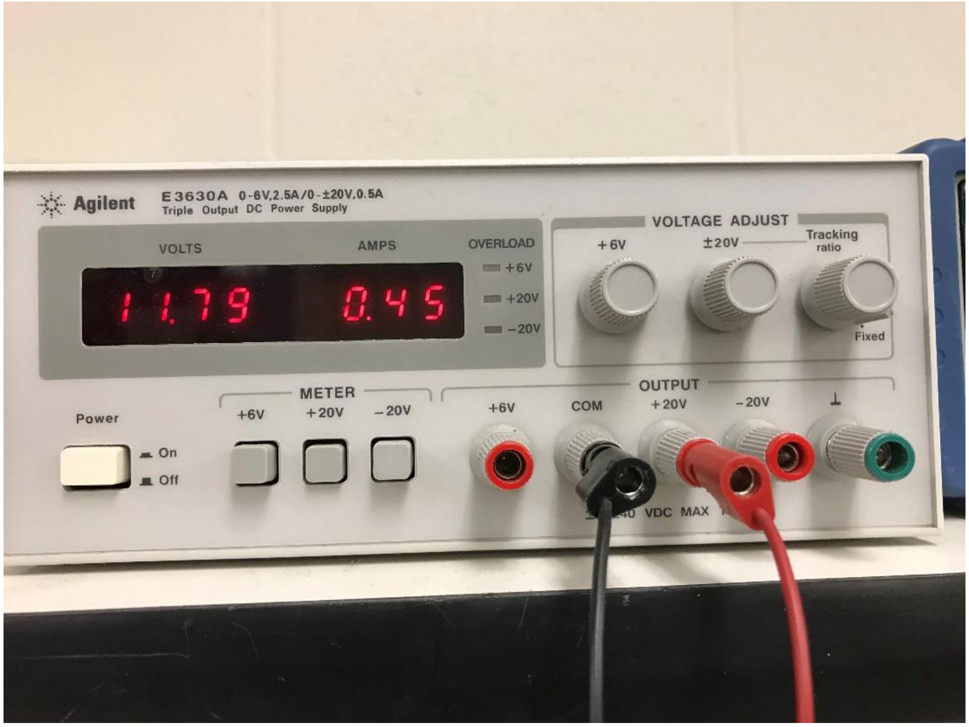
The two oscillators are namely: the Low Frequency Crystal Oscillator (LFXO) and the High Frequency Crystal Oscillator (HFXO). These oscillators require an external clock or crystal connected to the crystal oscillator pins of the device. However, no external crystal load capacitors are required as they contain on-chip tunable load capacitors. The LFXO supports crystals with a nominal frequency of 32.768 kHz, while the HFXO supports 38.4 MHz. Both

the high and low frequency clock sources can be used simultaneously.
LFXO part number: DST210A 1TJG125DP1A0012
LFXO load capacitance: 12pF
HFXO part number:
1ZZNAE38400AB0A HFXO load
capacitance: 10pF

What is the maximum charging current allowed by the PMU circuitry?

ANS: Our battery has isolated charging pins and discharging pins. Hence our PMU has no interference with the charging current. Our battery management provides constant current/constant voltage to charge the battery. We have kept constant current as 500mA. It is possible to configure our constant current by changing the value of a single resistor.

Charging Current has been verified as per the image below on the board:



How much current will the programming of the MCU flash require?

- Write Current – 3.5mA
- Erase current – 2mA
- Supply voltage – 1.6V to 3.6V

How much current will the energy storage element and the PMU be able to provide?

ANS: PMU is configured to provide currents upto 200mA.

What are the connection points to enable external power to digital / MCU portion of the board?

ANS: A switch would be provided in order to switch between an external supply and the onboard battery. The external supply would be a 5V USB port which will be connected to the Sensor and LDO. The Sensor has an onboard regulator, hence there is no requirement of using a regulator to regulate the voltage from the USB port.

What is the maximum charging current allowed by the energy storage unit?

Ans: 1C mA = 1000mA.

What will be the maximum current of the JUMP START power source?

Ans: As per the battery specifications, we require 2-3 volts and few microamps per cell in order to JUMP START the battery.

Where will the with JUMP START and power ground signals connect to?

Ans: JUMP START and power signals will be connected to the input of THE PMU IC.

USE CASE MODEL

ENERGY MODES

1. **Mode 1 - Deep Sleep Mode**

   In deep sleep mode, the barcode sensor is powered down, NXP NTAG and the processor are in sleep mode. The idea behind this principle is that the device shouldn’t start scanning until it is connected to the phone. The power to the sensor is GPIO controlled, providing us the capability to turn if off when not in use.

   NFC sleep current = 195uA
   EFR32BG13 Sleep mode = 2uA
   Total current in this mode = 197uA

2. **Mode 2 – Connection Initialization**

   The NFC tag is triggered shifting into Active mode as specified in the table above using the Android app on the phone. This trigger will be used to turn on the radio and start Bluetooth advertising thus, switching the processor in EM0 mode (radio on). The authentication of Bluetooth is done using NFC. During this mode the sensor is powered down.

   NFC active mode = 240 uA
   EFR32BG13 radio on = 10mA
   Total current in this mode = 0.240mA + 10mA = 10.24mA
   After this mode the NFC would always be in idle state.

3. **Mode 3 – Connected Mode**

   In this mode the barcode sensor is in sleep mode. The processor is in EM1 mode waiting for a successful read. Both the sensor and processor are in sleep mode.

   Barcode Sensor Sleep Mode = 2mA
   EFR32BG13 EM1 Mode = 3mA
   NFC Sleep Mode = 195uA
   Total current = 2mA + 3mA + 0.195mA = 5.195mA

4. **Mode 4 – Product Scanning mode**

   In this mode the barcode sensor scans the items and transfers data to the board using I2C interface.

   Barcode sensor Active Mode = 135mA
   EFR32BG13 EM1 Mode = 3mA
   NFC Sleep Mode = 195uA
   Total Current = 135mA + 3mA = 138.195mA

5. **Mode 5 – Transmission Mode**
   In this mode the Bluetooth radio communicates with the android app to send over the details of scanned items
   EFR32BG13 radio on = 10mA
   Sensor Sleep Mode = 2mA
   NFC Sleep Mode = 195uA
   Total current = 10mA +2mA = 12.195mA

AVERAGE DC CURRENT MATH

All the duration times considered in the following calculations are worst case scenarios.

1. **Connection Initialization Mode**
   Time duration = 2 seconds
   Considering 1 connection per hour with a consumption of 26mA current per connection
   Total Current Consumed = 26mA

2. **Product Scanning Mode**
   Assuming the barcode scanner will successfully scan an item barcode in 1 second and considering 100 successful scans per hour with a consumption of 137mA current per scan.
   Total Current Consumed = 138.195mA * 100 = 13819.5mA.

3. **Transmission Mode**
   Assuming 1 second per transmission considering failed attempts and retries (worst case).
   Considering 100 successful transmissions per hour with a consumption of 12mA current per transmission.
   Current Consumed = 12.195mA * 100 = 1219.5mA

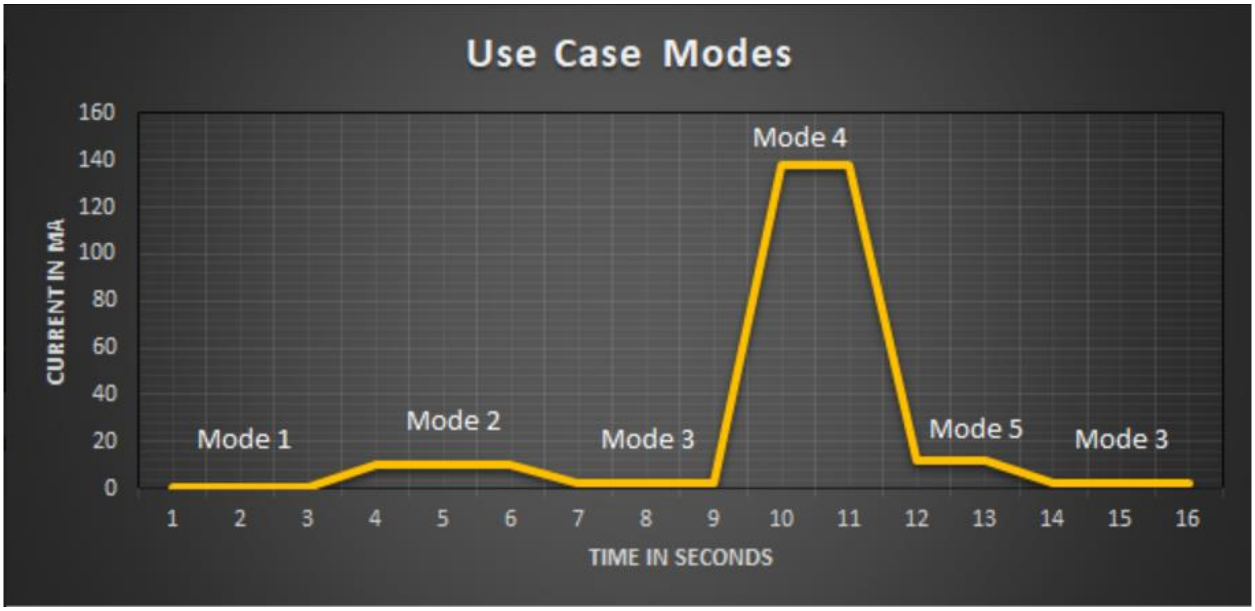
4. **Connected Mode**
   Being in connected mode without any data transmission for the remaining time i.e 1 hr – time consumed in previous modes = 3397 seconds.

Current Consumed = 3398 * 5.195 = 17647mA

Average DC current in one hour = Total Current consumed in all modes/3600
= 9.08mA
Battery Capacity chosen = 1000mAh
Total operating hours in single full charge = Battery Capacity/9.08mA
= 1000/9.08 = 110.13 hours

Approximately 4-5 days of continuous operation for 24 hours with 100 scans per hours.

These calculations are for worst case scenarios. In order to have better continuous days of operation, better results can be achieved by reducing the current consumption in connected mode. In addition to this, energy harvesting methods can also improve the operating hours of the product.



The above image represents our use case model broken into generic time slots showing current consumption for different phases of the system. The above graph is just an approximate representation of the energy modes that the entire system goes through.

PLANNED TEST POINTS

Test Points:

1. VBATT+
2. 5V
3. 3V3
4. SCL, SDA
5. UART TX, UART RX
6. RESET

1. Should have been there more test points?
   Ans: VUSB test point should have been added.

GANTT CHART

Gantt Chart has been attached with the submission files.

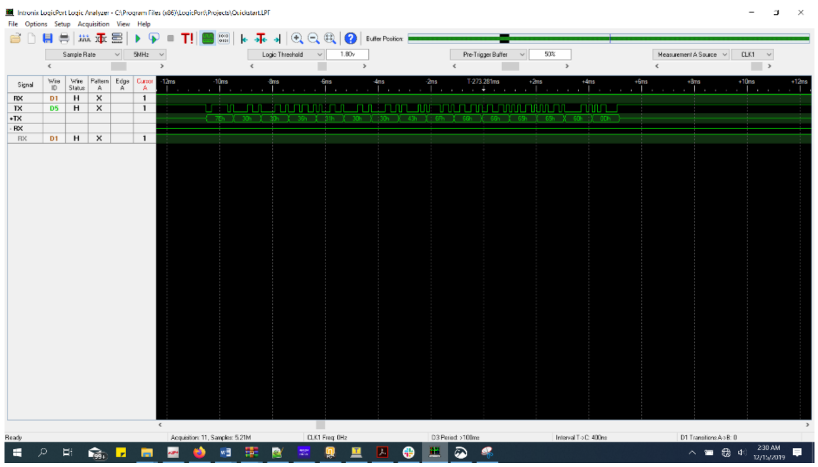
VERIFICATION REPORT

The verification report has been attached with the submission in a verification and validation.xlsx file.

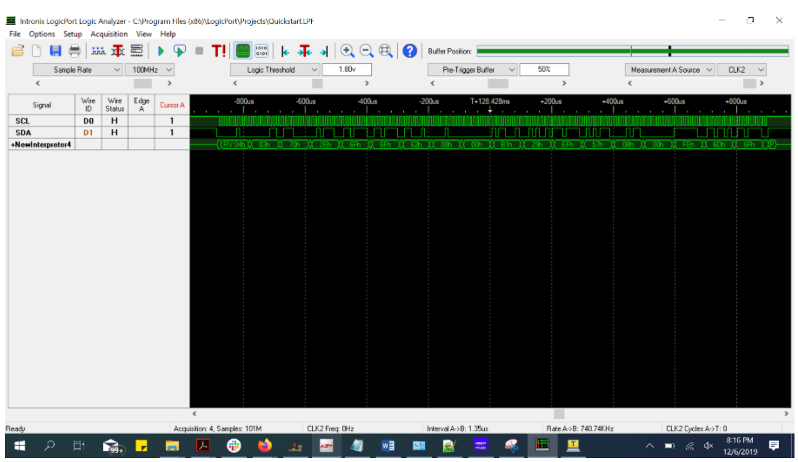
SIGNAL QUALITY ANALYSIS

The two peripherals I2C and UART were analyzed using logic analyzer. The data sent was interpreted on a logic analyzer and verified

UART Peripheral -



I2C Peripheral –



SUMMARY

This project is a prototype for self-checkout shopping cart. The project consists of a PCB designed with a Power management Unit including a Battery Recharging Circuit, Buck Regulator and LDO regulator to recharge the given battery, obtain 5V and 3V respectively. The Board has been provided with a 12V DC Jack to recharge the attached battery and a micro USB to supply power to the board in case of absence of the battery. Appropriate Isolation points have been provided in order select the power source as battery or USB. The Board also has a number of test points to isolate and test the outputs of individual circuits such as Battery Management circuit, Buck regulator and the LDO circuit.

A set of GPIO pins have been provided in order to connect the barcode sensor and the NFC module. A set of GPIO pins have also been provided for future use. These extra pins have no use in the current project. The reset pin is provided which resets the EFR32BG13. This reset pin is active low. In addition to this, a debug connector is provided in order to have debug prints on serial console and to flash firmware on the board.

Finally, the processor (EFR32BG13) with the antenna is used for Bluetooth communication. An android application is also designed to receive and display data from the embedded device.

The NFC module is used to obtain an interrupt which signifies that the user wants to connect to the embedded device. On placing the mobile phone near to the NFC module, the Bluetooth connection process is initiated automatically, and a connection is established between the mobile phone and the device. Further, the user then only needs to use the barcode sensor to scan the items in order to obtain a list of items and their cost on the android application in the connected phone. Then in order to generate the bill, the user simply needs to press the Bill button and then press the Payment button in order to proceed to checkout.

CHALLENGES FACED

1. Firstly, getting an innovative project idea which could satisfy all requirements and also serve as an end to end product.
2. Learning and working with PCB designing tool - Altium for the first time, developing skills, learning various concepts.
3. Understanding the antenna matching circuit, required ground vias and deploy on our board.
4. Designing power management circuit as we had to design it for 2 regulated power supplies (5V and 3V3).
5. Setting Up NFC for the first time and to write the data into the Tag using an I2C interface.
6. Understanding how to interface the Barcode scanner as the datasheet was not clear and detailed. Also configuring the barcode sensor into sleep mode required a lot of trials.
7. Delayed by 2 weeks for the PCB bringup due to reordering our PCB.
8. Understanding the working of the debugger and make work at first place on the board.
9. Process of understanding the do’s and don'ts while reflowing our board as we ruined one of our boards by putting it into reflow oven three times
10. Reworking tracks was a challenging task as components were too small to hand solder and required a lot of patience and concentration.
11. Writing firmware which is robust and generalized so it can be ported into any application.
12. Creating an Android application from scratch and understanding the working of MIT App inventor.

LESSONS LEARNED

Deepesh

1. Setting Reflow profiles and realizing the importance of doing reflow just once and not consuming too much time to place board in the machine as it could lead to cooling down of the machine.
2. Always having extra GPIO pins on the board that gives you the capability to interface components you might have missed during designing the board.
3. Creating Custom Design Rules for the layout based on the requirement of our board components and signals
4. Component selection based on the application design. For example, we chose EFR32BG13 because Blue Gecko pins are multiplexed, each pin capable of performing multiple alternate functions, giving us the freedom to use different pin functionality based on the Layout.
5. Create Android apps using official Android Studio rather than using any other sources because it can be very difficult to debug your application during runtime errors.

Satya

1. Studying the allowable temperature reflow profiles for all the components and setting profiles most suited for all the components on our boards.
2. Learnt that the ground plane acts as a low impedance current return path for all components in the circuit and care should be taken for that high speed or high current signals have a good return path.
3. Creating Custom BLE Gatt Service and Characteristics, generating UUID and setting notification and indication parameters for communicating with Android App.
4. Realized the importance of having firm headers and connectors on the board as the slight external force could rip off the pad which could lead to unsolvable problems.
5. We should not get too ambitious and always have enough time dedicated for integrating software and hardware and resolve bugs.
6. Always follow and update your Gantt chart.

Siddhant

1. Grouping pins of ICs based on the functionality for better readability rather than placing it based on their actual footprint.
2. Developing skills to read datasheet thoroughly which involves understanding electrical characteristics, thermal characteristics, mechanical characteristics, design parameters, size and footprint.
3. Verifying schematic and footprints of components as mentioned in the datasheet involving width of components, pitch of the pins and thermal padding.
4. Not placing the ground and VCC test points next to each other as it could cause a short while testing.
5. Learnt to break down the importance of the various tasks and not to predict that most difficult tasks in the project consumes more time and easy tasks would take less time.

FUTURE SCOPE

- The product data can be maintained in a cloud server instead of sending all the data to the android application from the embedded device.
- Energy harvesting system can be incorporated by converting kinetic energy of the trolley wheels to electrical energy.
- Budget Alert feature where the app notifies the user about budget limit crossed.
- Load cell can be incorporated for the items which has to be purchased in weight rather than in quantity.
- We can analyze the data for stock and asset management.
- Secured automatic payment through the android application.