Wireless Game Controller

TEAM NAME: DS – LITE
STUDENT NAME: LINFENG LI, VIKRANT WAJE

SEMESTER: FALL 2018
COURSE: LOW POWER EMBEDDED SYSTEM DESIGN
INSTRUCTOR: PROFESSOR KEITH GRAHAM

Contents

CHAPTER 1 : EXECUTIVE SUMMARY

In this report we share our findings and implementation specific details of the low power embedded system that we have designed. The embedded system designed is a “Wireless game controller” that can control a car-based video game.

The wireless controller has three primary sensors that are responsible for controlling the game, namely:
1) **An accelerometer** manufactured by Bosch is used to detect the orientation of the game controller which in turn is used to move the steering of the car in particular direction.

2) **Two Hall effect sensor** that are used to measure the magnetic field generated by the magnet attached to the linear input buttons. The generated magnetic field is measured using an Analog to Digital converter and used for accelerating or braking the car.

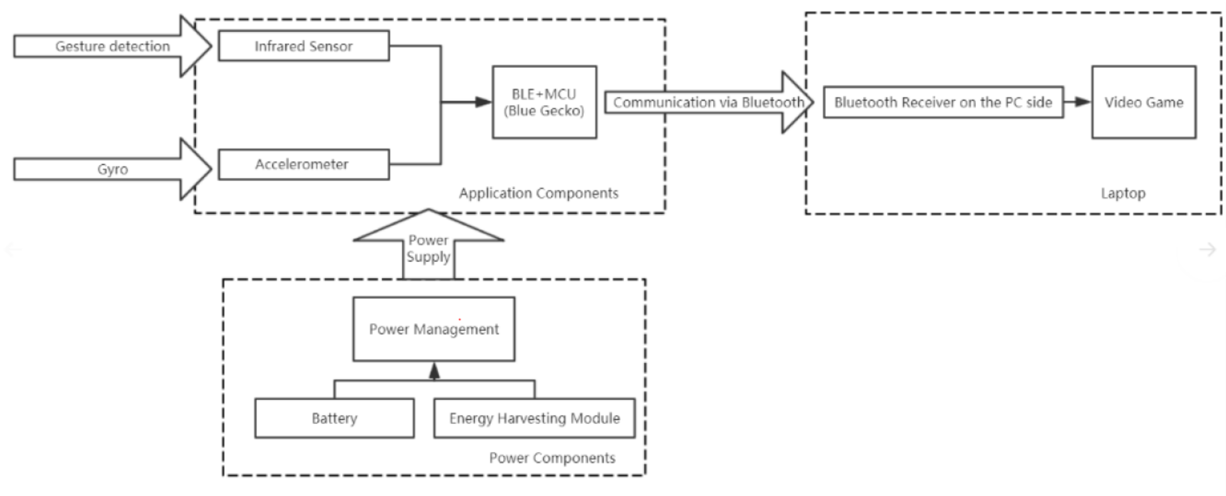
3) **A thermal image sensor** manufactured by Panasonic named as ‘Grideye’ is used to detect the gesture made by user. The thermal sensor has 64 pixel(8*8 array), which is programmed by user to be divided into four sections, wherein each section is mapped to different buttons.

4) **Six user buttons** that serve to provide user with an option to select various functions.

The wireless game controller can be powered using a **400mAh battery** capable of charged using two options namely:

1) **A USB charger**
2) **Wireless charging(Qi)**

The microcontroller that is used to control and monitor all the above-mentioned sensors and actuators is a **ARM cortex M4** architecture. The microcontroller is manufactured by **Silicon labs** which comes under the **EFR32BG1xx series**. The wireless game controller also uses **Bluetooth low Energy protocol** to transmit data wirelessly to client side. Note that we have used **BGM111 module** that comes with the EFR32BG1xx microcontroller and Bluetooth low energy module embedded inside it.



The overview of our project is represented in diagram shown above. It has three important sections namely Power management section, radio section and microcontroller and sensor section. The details of each section is given in Chapter 4: KEY COMPONENTS

CHAPTER 2 : OVERVIEW OF PROJECT

The project is based on wireless game controller that can control car-based video game. The game controller takes input from user in form of gestures, button press and tilting or changing the orientation of the game controller. The wireless game controller has been given a thought of making it close in performance to wireless game controller manufactured by tech companies such as Microsoft, Sony, Nintendo, etc.

The project has used wireless technology based on Bluetooth low energy to transmit data wirelessly. The controller is battery powered from 400mAh battery, the battery can be charged from energy harvesting mechanism such as wireless charging(Qi) or it can be powered using the USB charger. The 5V from USB charger or any other source of power is converted into 3.3V using a Low drop out voltage regulator. An under voltage integrated circuit is used to turn off the system when the voltage falls below 3.1 V. The sensors are interfaced to the microcontroller circuit using I2C interface and an Analog to Digital converter is used to convert the magnetic field generated by linear inputs into digital values.

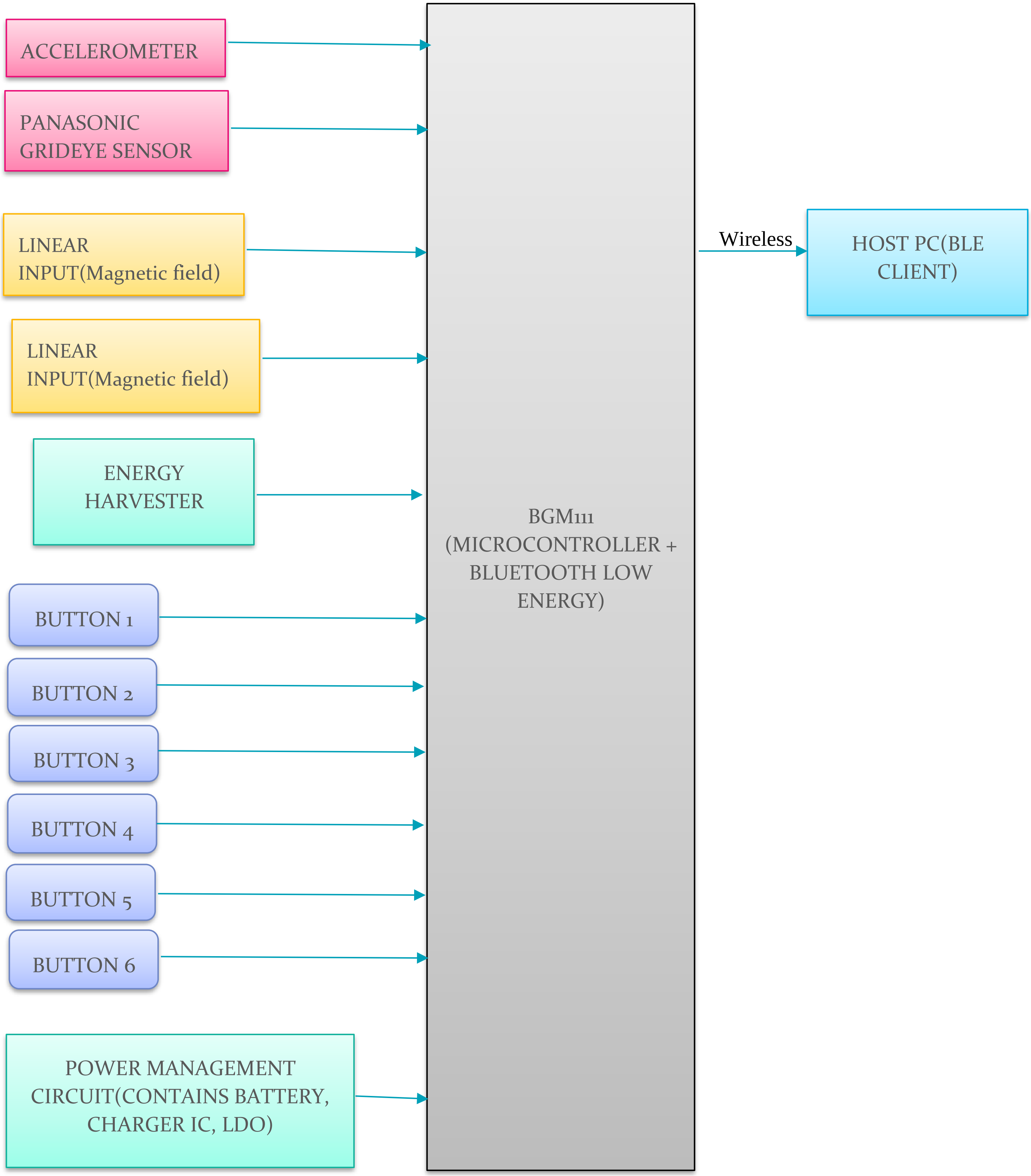
The project involves following development steps:

1) Selection of components: selecting microcontroller, sensors and actuators
2) Selection of Battery and Power management integrated circuit.
3) Drawing schematic of proposed design
4) Generating a Bill of materials of the selected components and ordering from Digikey.
5) Layout design of given circuit
6) Doing a design rule check and resolving all problems and doing a DFM check
7) Manufacturing the PCB from PCBway or any other fab manufacturer
8) Assembly of board
9) Designing validation plan
10) Testing the board after getting it and assembling it
11) Writing firmware for the board
12) Integrating the Hardware and Firmware
13) Optimizing the algorithms.

WHAT PROBLEM DOES THE PROJECT SOLVE?

- As such, there is no problem solved by our product since it is an entertainment-based product.
- The main objective of the project is to provide a way to user control the game wirelessly by means of gesture. Thus, it allows the user to use gesture to control the game.

CHAPTER 3: HARDWARE BLOCK DIAGRAM



CHAPTER 4: KEY COMPONENTS

The entire circuit is divided into three sections namely power management section, microcontroller and sensor section and radio section.

POWER MANAGEMENT SECTION:

1) **Wireless charger receiver(BQ51013)**: The input is connected to inductive coils which when placed on charging pad will provide some voltage that can be directed to charge the battery.
2) **LiPo battery charger (BQ24040)**: Provides a voltage output that can be used to charge the battery.
3) **Low Dropout voltage regulator(LTC3035)**: Responsible for providing a constant 3.3 V at output which can be then directed to microcontroller and sensors on the board.
4) **Under voltage protection circuit**(**LTC4365**): Provides a protection mechanism to switch off the voltage output in case input drops below 3.1 V
5) **400 mAh battery**: An energy storage elements which acts as an energy sourcing element for the circuit.

MICROCONTROLLER AND SENSOR SECTION:

1) **Microcontroller(BGM111)**: This is the microcontroller unit based on ARM cortex M4 architecture. This microcontroller is manufactured by Silicon Labs based on EFR32BG1xx series.
2) **Inertial measurement unit(BMI160)**: An IMU manufactured by Bosch sensor which has accelerometer, magnetometer and gyro meter embedded inside it. The accelerometer is specifically used for this project to detect the orientation.
3) **Hall effect sensors(DRV5055)**: The Hall effect sensor is used to detect magnetic field that is generated by linear inputs which can be used to accelerate or decelerate the car.

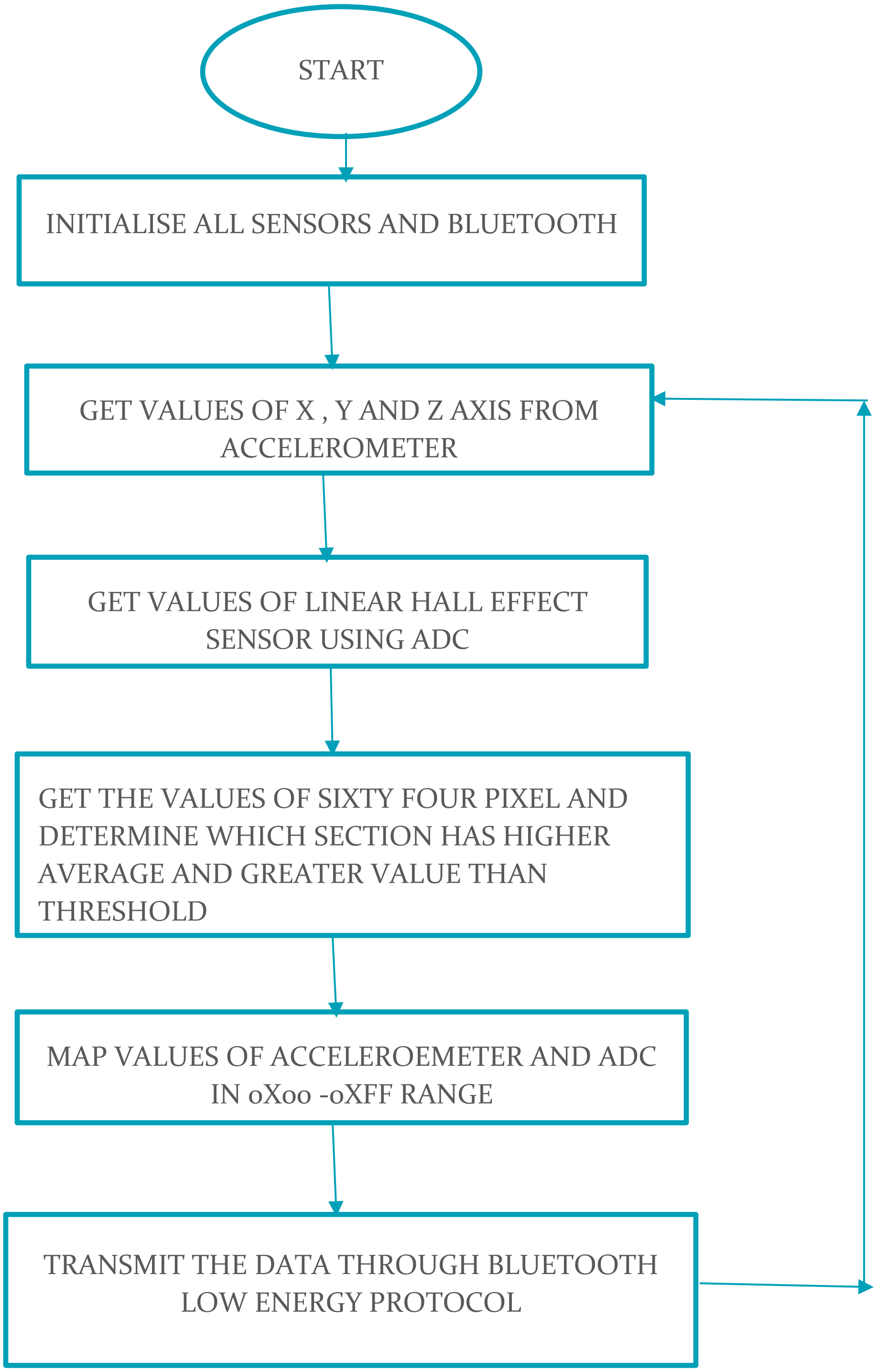
RADIO SECTION:

1) **Bluetooth low energy(BGM111):** The Bluetooth low energy module is used to transmit data wirelessly to the client laptop.

MISC SECTION:

1) **USB TO UART BRIDGE(CP2102):** This is used to enable UART functionality that can aid debugging
2) **Others:** Resistors, Capacitors, Op Amp, ESD Diodes, P MOSFET, jumpers, buttons

CHAPTER 5: SOFTWARE FLOW DIAGRAM/BLOCK DIAGRAM CHART



CHAPTER 6 : LIST OF COMMANDS

This a HID compliant controller based on HID on the GATT profile. Therefore, when the controller connects to the WINDOWS, the built-in DirectInput driver would recognize the device as a joystick automatically and parser the data accordingly.

According to the HID report descriptor, the report should be six bytes and set as a characteristic that could be notified. Upon the recognition of the controller, the DirectInput driver would automatically subscribe this report(characteristic), and the controller then activate all of the peripherals and start off transmit data by notification of the corresponding report characteristic.

Based on the report descriptor, the data contained in a six-byte report and each one has following meaning:

Data[0]: X axis; one byte length; logical range: 0~255.

Data[1]: Y axis; one byte length; logical range: 0~255.

Data[2]: L trigger; one byte length; logical range: 0~255.

Data[3]: R trigger; one byte length; logical range: 0~255.

Data[4]: Lower 4 bits: each one for a mapped grid eye buttons.

Data[4] higher 4 bits and Data[5] lower 2 bits: each for one physical button on the bottom of the board.

Data[5] higher 6 bits: constant, should not be touched.

CHAPTER 7: PLANNED DEVELOPMENT SCHEDULE AND WHEN TASKS WERE COMPLETED

WEEK 1:

- Selection of project idea

WEEK 2:

- Selection of components
- Figured out system level features in detail
- Decided the operation modes and behaviour in each mode(use case model)
- Specification of energy harvesting and sstorage elements.

WEEK 3:

- Figured out the components- pmu, supercapacitor vs battery, wireless charging
- Installed simplicity studio and started to get familiar with blue gecko board
- Updated the use case model(bar graph)

WEEK 4:

- Finalised pmu solution
- Started writing firmware with blue gecko and got familiar with it
- Programmed letimer to blink led and observed current consumption in energy mode em0, em1, em2
- Analysed the risky part of project and developed back up plans to mitigate that
- Observed the timing characteristics (hold time, setup time) of sensors used in the project
- Looked for schematic symbols on snapeda website and verified few of them
- Verified c rate and battery solution

WEEK 5:

- Bulk capacitance selection
- Started writing firmware with Blue gecko to get familiar with it
- Started with I2C driver firmware for Blue gecko
- Debugging feature on Board

WEEK 6 & WEEK 7:

- Started schematics for Power Supply side
- Started schematics for interconnection between MCU and sensors
- Investigated more about Debugging feature on board

WEEK 8:

- Completed the schematics

- Reviewed our schematic from Teaching Assistant
- Started with Firmware for Panasonic Grideye sensor.
- Completed with selection of ESD and finalized the number of buttons, switches and I/O devices.

WEEK 9 & WEEK 10:

- Started layout(Almost done)
- Tested the Evaluation kit(Wireless Charging IC, Wireless receiver and Hall Effect Sensor)

WEEK 11 & WEEK 12:

- Completed layout and layout review
- Programmed the Panasonic Grideye to write and read
- Started the firmware for UART communication
- Developed a verification plan for hardware and software elements

WEEK 13:

- Ordered the Board and the components
- Workjng firmware that can read 64 pixels from Panasonic Grideye
- Paired the Bluetooth module with computer/laptop

WEEK 14:

- Assembled the board
- Tested the board
- Tested individual firmware module

WEEK 15:

- Started with integration of peripheral with Bluetooth
- Completed the HID application

WEEK 16:

- Integrated Hardware, Software and firmware
- Test the whole system
- Gave project Demo

CHAPTER 8: HOW THE TARGET MICROCONTROLLER OR SOC WILL BE PROGRAMMED

Features, Components to enable debugging

Figure 1 debugging connector of the LEOPARD gecko EVAL board

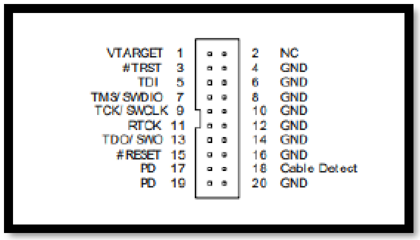


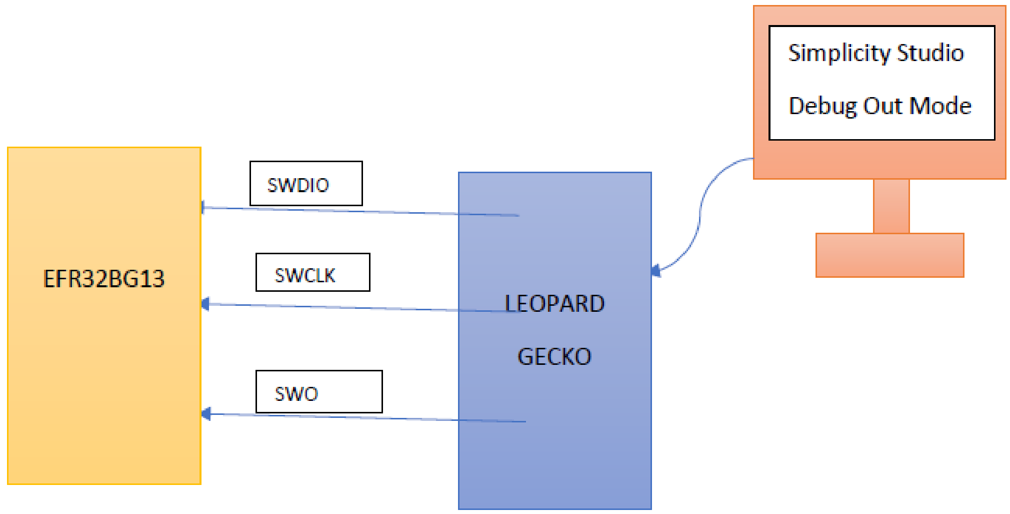
Figure 2 debugging modes description of the LEOPARD gecko EVAL board

Mode	Description
Debug MCU	In this mode the on-board debugger is connected to EFM32 on the EFM32LG-STK3600.
Debug IN	In this mode the on-board debugger is disconnected, and an external debugger can be connected to debug the EFM32 on the EFM32LG-STK3600.
Debug OUT	In this mode the on-board debugger can be used to debug an EFM32 mounted in your own application.

Figure 3 default pinout setting of the SWDIO and SWCLK

DBG_SWCLKTCK	0: PF0									Debug-interface Serial Wire clock input and JTAG Test Clock. Note that this function is enabled to the pin out of reset, and has a built-in pull down.

DBG_SWDIOTMS	0: PF1									Debug-interface Serial Wire data input / output and JTAG Test Mode Select. Note that this function is enabled to the pin out of reset, and has a built-in pull up.



We will bring the SWDIO, SWCLK and GND to a connector and then connect this signals to the LEOPARD gecko evaluation board. With the LEOPARD gecko evaluation board being set to DEBUG OUT mode, we can load a program or debug the microcontroller

CHAPTER 9 : CURRENT PROFILE OVERTIME BASED ON APPLICATION USAGE

USE CASE AND BAR GRAPH:

Generally, we propose four major operation modes as following below:

Sleep mode: Bluetooth module and Grid Eye sensor shut down; IMU runs in a standby mode which can be triggered by movement and send an interrupt signal to activate the whole system.

Active mode: Grid Eye and IMU have all function running. MCU gets one set of data from sensors, process, and parse them into the string command that would be sent via Bluetooth.

Transmit mode: Bluetooth turns on and sends the command. Whether shut the sensors down is yet to be decided.

Suspend mode(optional): Shut down the sensors and Bluetooth module. Suspend the MCU (running in a slightly lower energy mode) for a certain period of time and then goes back to active mode again. This approach could realize a configurable command rate and save energy when high command rate is not required.

Following is the chart of consumption of current in different energy modes by different peripherals

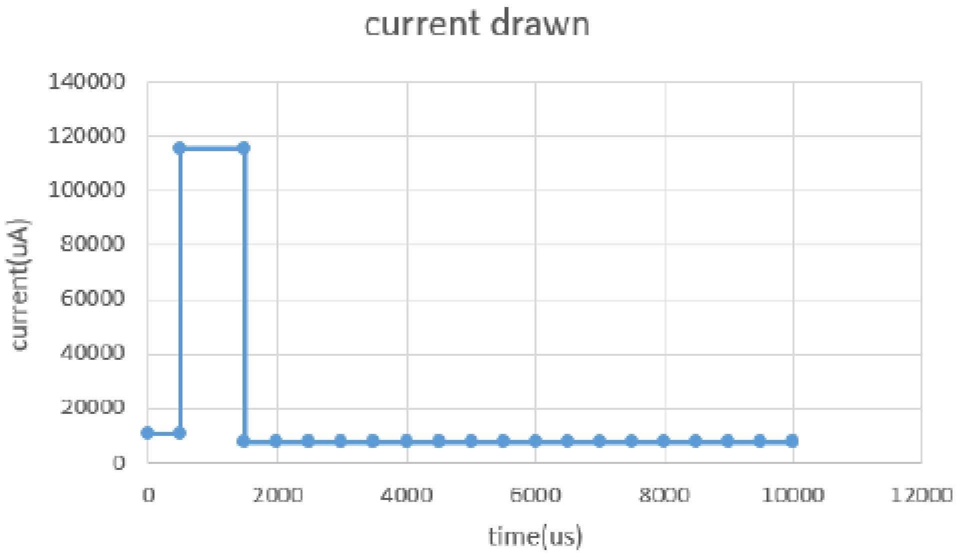
Peripherals Active	Sleep Mode	Transmit Mode	Active Mode	Suspend Mode
MCU + Bluetooth	1.3 uA	110000 uA	5120 uA	3040 uA
IMU	20 uA	850 uA	850 uA	3uA
Infrared Sensor	200uA	4500 uA	4500 uA	4500 uA
Total	221.3 uA	115350 uA	10470 uA	7543 uA

Detailed illustration

MCU:

- • **Sleep**: 1.3uA. (using EM2 mode)
- • **Transmit** :8.5mA (assuming a 2MB/s data rate, 50 Bytes per command packet and command updating rate of 100 per second would result a 25us transmit session)
- • **Active** : 5.120 mA (working at 40MHz without DCDC and EM0 mode)
- • **Suspend Mode**: 3 mA(Working at 40 MHz without DCDC and EM1 mode)

Transmit mode duty cycle: 25us/10ms =0.25%


current drawn
140000
120000
100000
80000
60000
40000
20000
0
current(uA)
0
2000
4000
6000
8000
10000
12000
time(us)

CHAPTER 10 : ENERGY STORAGE ELEMENT SELECTED AND SELECTION DOCUMENTATION

The selection of battery vs super-capacitor is based on the use case and bar graph(previous section)

Choice of battery vs Super Capacitor

- After analysis, we have proceeded to go with battery. Since we are looking for energy storage element that can provide constant energy, we have decided to go ahead with battery. The output of battery will be given to some form of converter(usually a step down converter) which will step down the voltage to a range that will be within the operating range of sensors.
- The converter that we have planed to use is a LDO since it will be effective against any noise,transients although somewhat less efficient than other converter.
- Also, since we are designing a wireless game controller, we want the converter to be as small as possible so that it does not occupy much space on the PCB. Switched mode power supply occupy too much of space because of use of capacitors and inductors while a LDO has a very small footprint.
- Power dissipation = (input voltage – output voltage) × load current

Some calculations:

Super-Capacitor

dv= Resistive voltage(ESR) + Capacitor voltage(VC)

dv= (i*R) + (i* dt /C)

dv = i*(R + dt/C)

dv/i=R + dt/C

(dv/I -R)/dt = 1/C

C = dt/(dv/I - R)

Assume R is neglected

C = (Iavg * dt)/((Vwork - Vmin))

= (6 * 3600) *(18.47mA) /(3.7 – 3.3))
= 997.38 F
Approximately equal to 1000 F which is almost difficult to find unless we connect the normal capacitors in parallel

Battery

The Average current for the circuit is

= (Current in Transmit mode * Duty Cycle of Transmit mode) +(Current in Active mode * Duty cycle of Active mode) + (Current in Suspend Mode * Duty Cycle of Suspend Mode)

=((115.35mA * 1 ms) + (10.47 mA * 8.5 ms) +(7.543 mA *0.5ms))/(10 ms)

=18.47 mA

We need to provide energy for hours, since the capacity of battery is measured in mAh, From the table above, the current consumed in total across all mode comes out to be 18.47mA. So, if we want to sustain our battery for about 5 hours, the calculation will be as follows:
Capacity = (Average Current * hours the battery should supply current)/(Worst Case Efficiency of LDO)

Capacity = (18.47 mA * 6)/(3.3/4.2) = 141mAh.

Thus, we need a battery of (141mAh + some buffer) to sustain for 6 hours. Since we haven't taken current consumed by other parts, this figure is just a rough estimate.

CHAPTER 11: PMU SIMULATION RESULTS/ SUMMARY

Operation voltage of components:

Parameter	Infrared Sensor	MCU+ Bluetooth	Gyroscope/Accelerometer
Minimum Voltage	3.0V	1.85V	1.7V
Maximum Voltage	3.6V	3.8V	3.6V

Simulation background:

- Battery supply voltage range: Vmax: 4.0V; Vmin: 3.4V.
- Use internal impendence measured at 1KHz of 200m ohms for the approximation of the battery DC internal resistance.
- 1µF capacitor (typical required value referred in the datasheet)at the output.
- Rising and falling time set to 10µs based on the wake up time from EM3 listed in the Blue

  Gecko's datasheet

Figure 1 battery's internal impedance @1KHz

6.2	Internal Impedance	≤200mΩ	Measure cells using an alternate current impedance meter at 1kHz .

The circuit we are designing consist of four major region that we need to analyses namely:

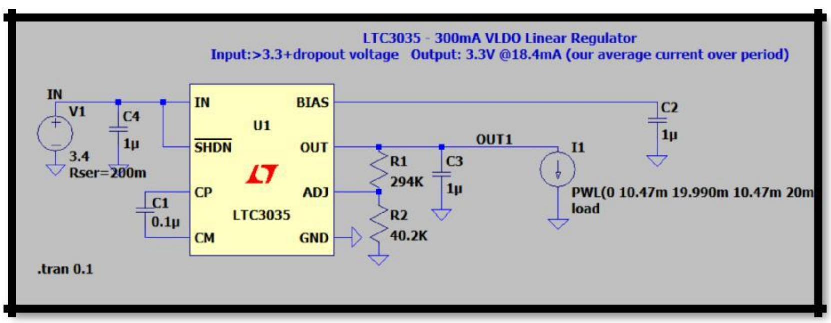
- Sleep(MODE 0) to Active (MODE 1)
- Active (MODE 1) to transmit(MODE 2)
- Transmit (MODE 2) to suspend(MODE 3)
- Suspend (MODE 3) to Sleep(MODE 0)

Output load model, current and its voltage rating of starting energy mode:

- The starting energy mode of our circuit is sleep mode
- In sleep mode, our circuit consumes current of about 221.3 uA
- Hence, the circuit will remain in sleep mode until there is any activity which can interrupt and wake it up
- After waken up, the system would run into the following operation pattern.
- Transition pattern(period of 10ms): Active mode(0.5ms) -> Transmit mode(1ms) -> suspend mode(8.5ms) -> active mode

Simulation model

Figure 2 simulation model with 1µF on the output

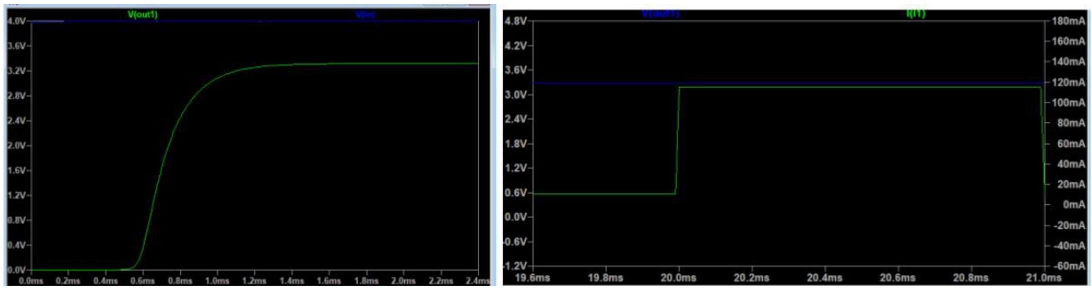


Q1 & Q2:

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

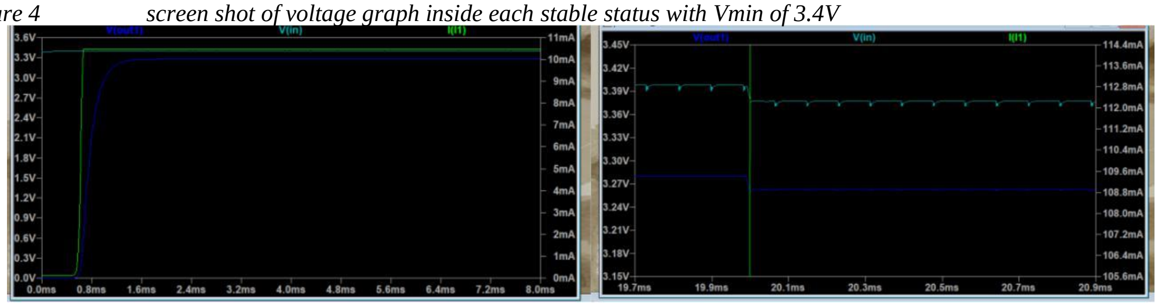
Ans: At Vmax: There is no voltage ripple at the output before the step load function throughout all operation mode (Screen shot of the same are attached in this report).

Figure 3 screen shot of voltage graph inside each stable status with Vmax of 4.0V



At Vmin: There is no voltage ripple at the output before the step load function except for the transmit mode with 115mA. The ripple in the transmit mode has a amplitude 3mV(from 3.260V to 3.263V) and last for 0.006ms.

Figure 4 screen shot of voltage graph inside each stable status with Vmin of 3.4V



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

At Vmin of 3.4V: Yes. The screenshot for the voltage ripple calculation are attached in this report and all of the ripple meet the requirement of the operation voltage range of our components.

At Vmax of 4.0V: Yes. The screenshot for the voltage ripple calculation are attached in this report and all of the ripple meet the requirement of the operation voltage range of our components.

QUESTION 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?

Ans: At Vmin: Yes, the output voltage dips when the dynamic step load is added(increasing current load) and the output voltage spikes up when the step load current is reduced(decreasing current load).

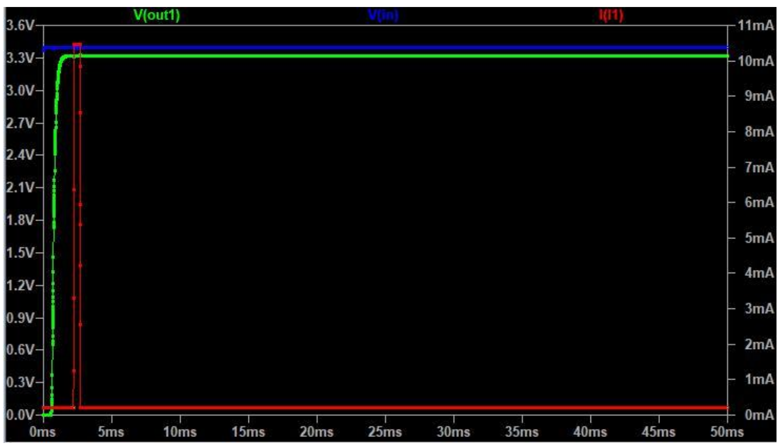
At Vmax: Yes, the output voltage dips when the dynamic step load is added(increasing current load) and the output voltage spikes up when the step load current is reduced(decreasing current load).

Minimum output voltage of power supply when voltage dips: See the section below(screenshots) to get the values

<u>For Mode 0(sleep mode) to Mode 1(Active mode) transition(increasing current region):</u>

<u>At VINmin = 3.4 V</u>

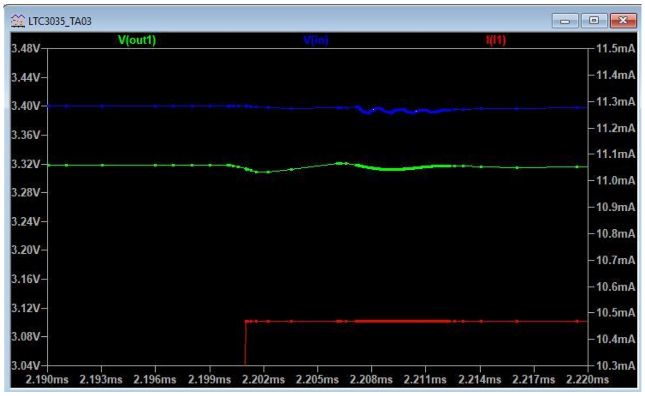
Figure 5 simulation for Mode 0(sleep mode) to Mode 1(Active mode) transition(increasing current region) with Vmin



Voltage ripple in output voltage= 3.308V V to 3.318 V

Minimum Output Voltage of VDD when voltage is drooping = 3.308V

Figure 6 *closer look for the previous graph in the transition region*



At VINmax = 4.0 V

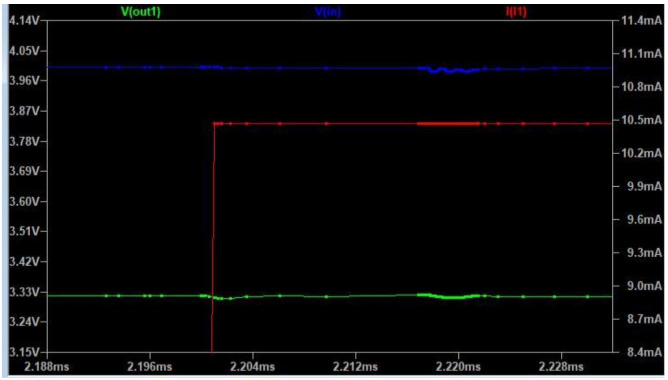
Figure 7 *simulation for Mode 0(sleep mode) to Mode 1(Active mode) transition(increasing current region) with Vmax*

Voltage ripple in output voltage = 3.309V to 3.318V

Minimum Output Voltage of VDD when voltage is drooping = 3.309V

Taking a more closer look:

Figure 8 closer look for the previous graph in the transition region

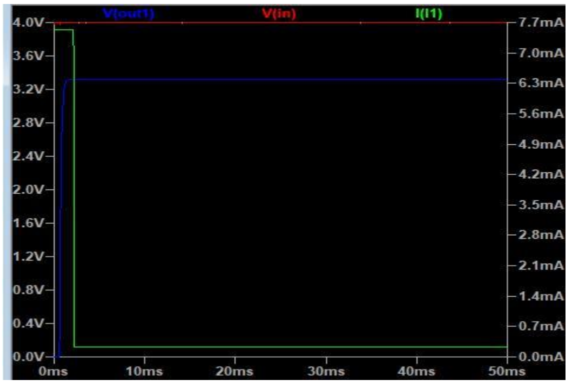


Ans for the Q3 &Q4: All of the ripple mentioned above meet the requirement of the components operating voltage range.

<u>For Mode 3(suspend) to Mode 0(sleep mode) transition(decreasing current region):</u>

<u>For VINmax = 4.0 V</u>

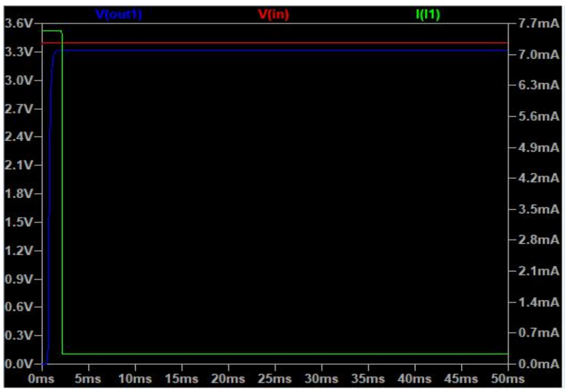
Figure 9 simulation for Mode 3(suspend mode) to Mode 0(sleep mode) transition(decreasing current region) with Vmax



Voltage ripple in output voltage = 3.322V to 3.317V

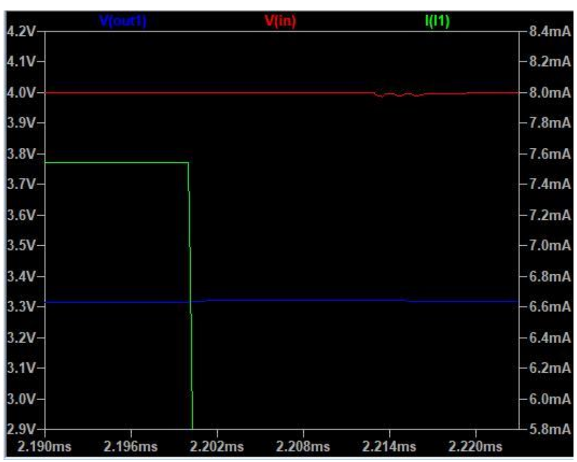
Minimum Output Voltage of VDD when voltage is drooping = 3. 3.317V

Figure 10 closer look for the previous graph in the transition region



<u>For VINmax = 3.4 V</u>

Figure 11 simulation for Mode 3(suspend mode) to Mode 0(sleep mode) transition(decreasing current region) with Vmax



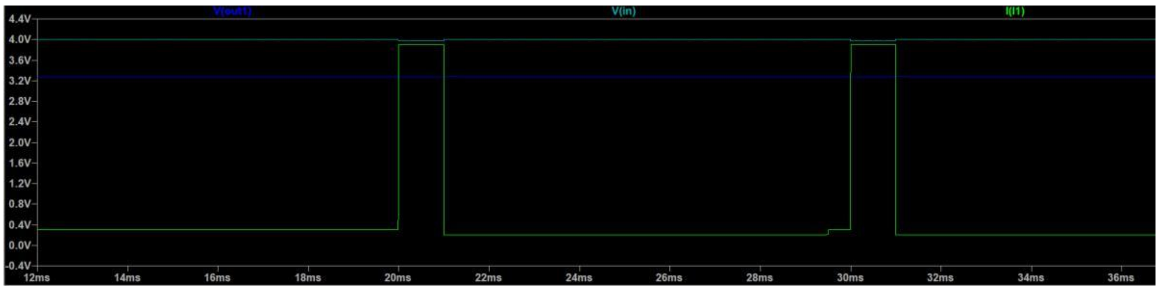
Voltage ripple in output voltage = 3.322V to 3.317V

Minimum Output Voltage of VDD when voltage is drooping = 3. 3.317V

Ans for the Q3 &Q4: All of the ripple mentioned above meet the requirement of the components operating voltage range.

<u>After waken up, it would run into an operating pattern as mentioned above.</u>

Figure 12 the simulation for the whole operating pattern



<u>For Mode 1(active) to Mode 2(transmit mode) transition (increasing current region):</u>

<u>For VINmax = 4.0 V</u>

Figure 13 simulation for Mode 1(active) to Mode 2(transmit mode) transition (increasing current region) with Vmax

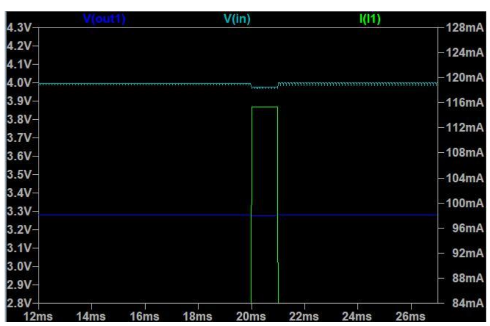
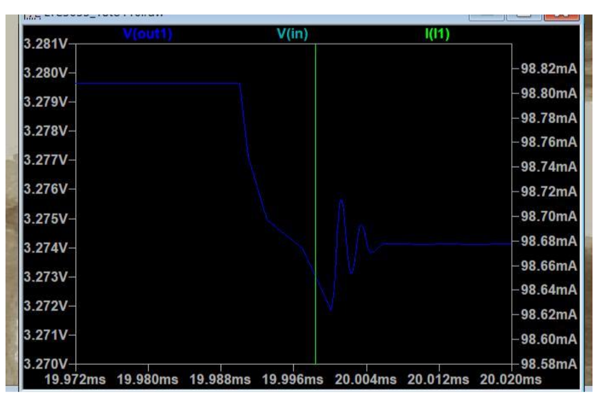


Figure 14 closer look for the previous graph in the transition region



Voltage ripple in output voltage = 3.272V to 3.276V, stabilized at 3.274V **Minimum Output Voltage** of VDD when voltage is drooping = 3.272V

<u>For VINmin = 3.4 V</u>

Figure 15 simulation for Mode 1(active) to Mode 2(transmit mode) transition (increasing current region) with Vmin

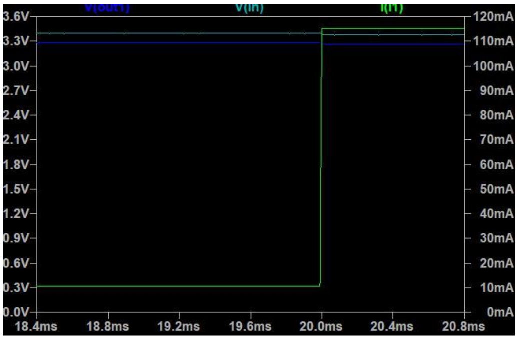


Figure 16 closer look for the previous graph in the transition region

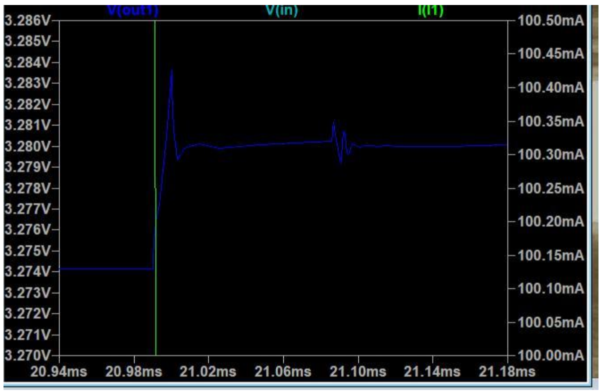
Voltage ripple in output voltage = 3.259V to 3.262V, stabilized at 3.262V **Minimum Output Voltage** of VDD when voltage is drooping = 3.259V

<u>For Mode 2 (Transmit mode) to Mode 3 (Suspend mode) (decreasing current region):</u>

<u>For VINmax = 4.0 V</u>

Figure 17 simulation for Mode 2(Transmit) to Mode 3(suspend mode) transition (decreasing current region) with Vmax

Figure 18 closer look for the previous graph in the transition region



Voltage ripple in output voltage = 3.279V to 3.283V, stabilized at 3.280V **Minimum Output Voltage** of VDD when voltage is drooping = 3.279V

<u>For VINmin = 3.4 V</u>

Figure 19 simulation for Mode 2(Transmit) to Mode 3(suspend mode) transition (decreasing current region) with Vmin

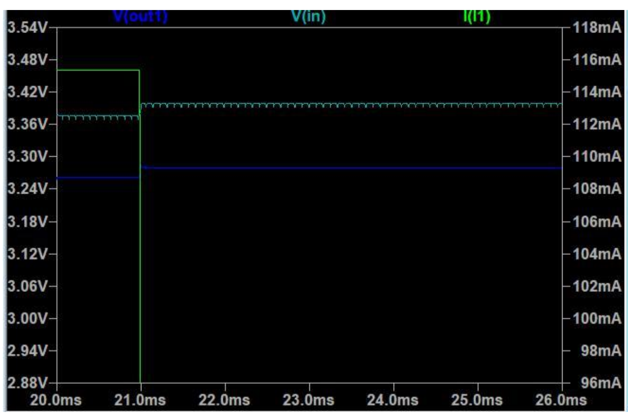
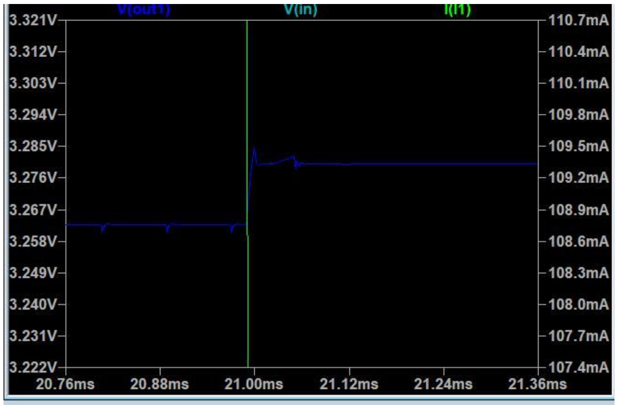


Figure 20 closer look for the previous graph in the transition region



Voltage ripple in output voltage = 3.278V to 3.283V, stabilized at 3.280V **Minimum Output Voltage** of VDD when voltage is drooping = 3.278V

Ans for the Q3 &Q4: All of the ripple mentioned above meet the requirement of the components operating voltage range.

CHAPTER 12: BULK OR LARGE DECOUPLING CAPACITOR SELECTION AND BACK UP DATA

NOTE: BACKUP DATA IS PROVIDED IN PMU SIMULATION (PREVIOUS SECTION)

Does the minimum voltage due to current load step meet the IC specifications of your circuit in terms of specified ripple or minimum voltage of IC's?

Ans: **At Vmin:** Yes, the minimum voltage due to current load step meets the IC specifications in terms of ripple, but to get a more cleaner output for the convenience of the potential use of ADC, we have decided to add a capacitor of 50uF that can eliminate ripple voltage.

At Vmax: Yes, the minimum voltage due to current load step meets the IC specifications in terms of ripple, but to get a more cleaner output for the convenience of the potential use of ADC, we have decided to add a capacitor of 50uF that can eliminate ripple voltage.

<u>Simulation with an extra 50µF capacitor.</u>

Figure 21 simulation model with 1µF on the output

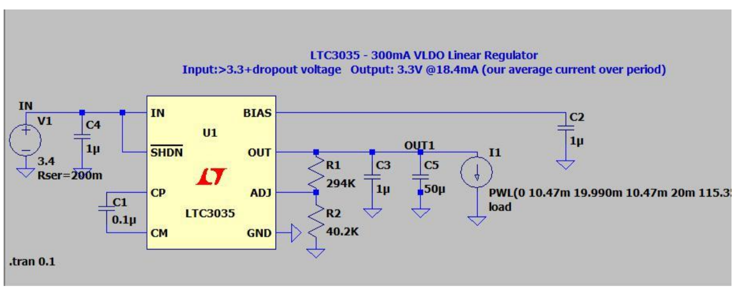
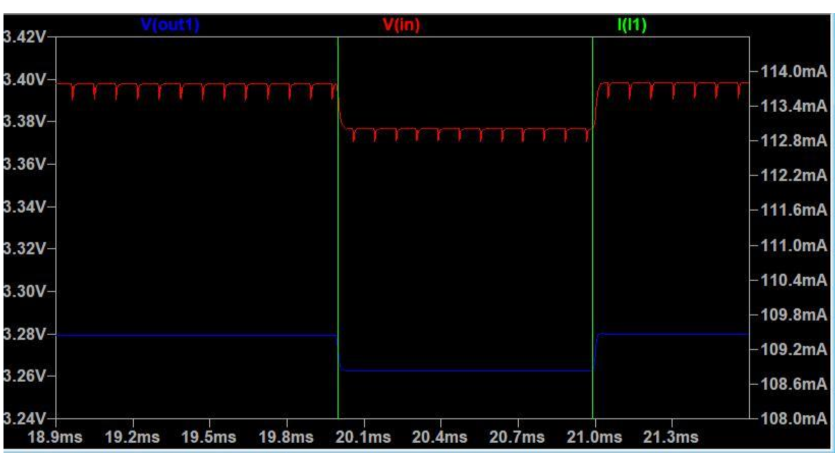
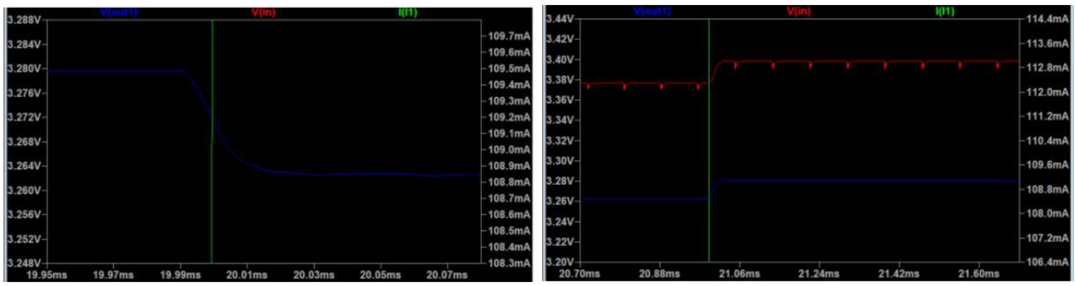


Figure 22 simulation for the transmit mode with Vmin of 3.4V



We can see that the 3mV ripple during the transmit mode with the 3.4V input has been eliminated.

Figure 23 transition region with minimum input



We can also find that the transition region has been smoothed out.

So after going through all the analysis, we have planned to use a 1uF ceramic capacitor and another 50uF electrolytic capacitor.

<u>Ceramic capacitor of 1uF of LDO:</u>

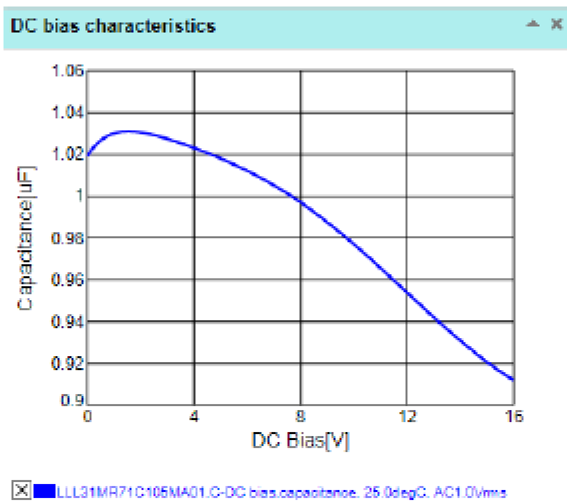
After going through Murata, and looking at the various Graphs which showed change of capacitance with respect to DC Bias, we have chose the following which shows less change in capacitance

Also, it has low ESL

Product Information

Item	Value
Part Number	LLL31MR71C105MA01
Capacitance	1uF
Rated Voltage(V)	16
Temperature	X7R
Size Code	1632M/0612
T size(mm Max.)	1.25
Cap. Tolerance	+/-20%
Public Standard	EIA
SRF (MHz)	9.09

Graph of Bias Voltage vs Capacitance:



Aluminium Electrolytic Capacitor of 50uF of LDO:

According to the "Aluminum Electrolytic Capacitors General technical information" from TDK™, we know that, the capacitance tolerance is the range within which the actual capacitance may deviate from the specific rated capacitance. The rated capacitance is the AC capacitance value for which the capacitor has been designed and which is indicated upon it, which may have a higher value for the DC voltage. So we just need to leave the margin for the tolerance of the rated capacitance. Then we decide to use a 68µF.

CHAPTER 13: WILL AN EXTERNAL ENERGY SOURCE BE REQUIRED TO PROGRAM MCU

No, as explained in programming MCU section, only five signals are required from external debugger, out of which Vtarget signal provides 3.3 v required fro programming. Failure to connect Vtarget signal gives target voltage too low warning and is not able to connect to the board.

PLANNED TEST POINTS

1) Power signals: GND and VCC
2) I2C interface signal: SDA and SCL
3) USB interface: USB Transmit(USB Tx) and USB Receive(USB Rx).
4) AD/EN signal of Wireless Receiver
5) RECT signal of Wireless Receiver

SHOULD THERE HAVE BEEN MORE TEST POINTS

From our experience, we felt that there is no more need of adding more test points since the signal values and plot can be actually obtained by probing the passive components such as capacitor and resistor across the points which you want to test.

CHAPTER 14: COMPLETE DETAILED VERIFICATION REPORT(SPREADSHEET)

See the validation report attached as separate excel file

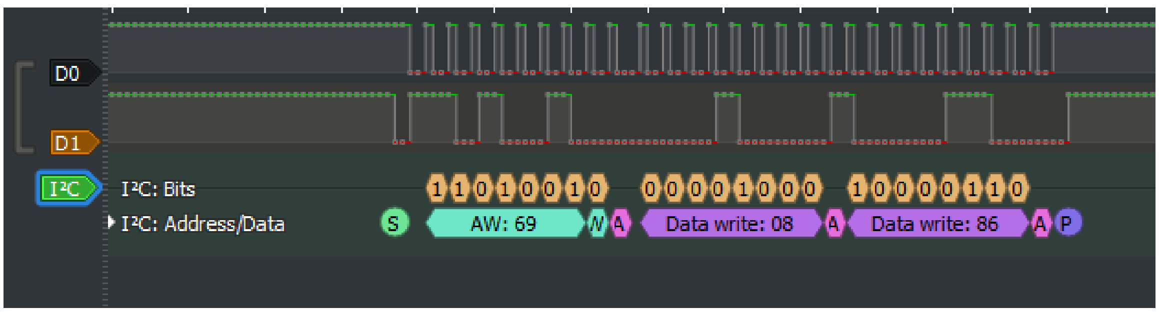
CHAPTER 15: SIGNAL QUALITY ANALYSIS OF KEY SIGNALS: INCLUDE ALL BUSES SUCH AS I2C AND HIGH SPEED SIGNALS

I2C SIGNALS: SCREENSHOTS FROM LOGIC ANALYZER

1) **PANASONIC GRIDEYE**

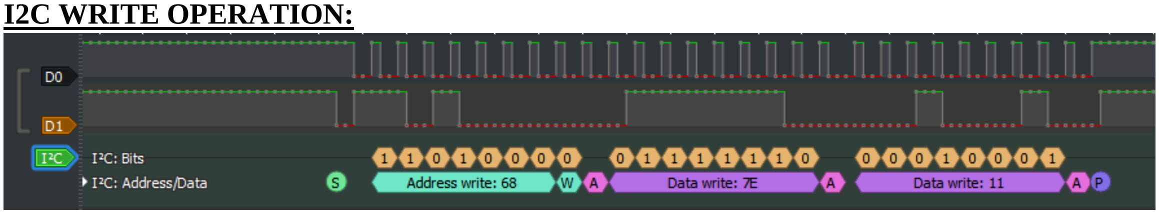
I2C READ OPERATION:

I2C WRITE OPERATION:



2) **BMI160**

I2C WRITE OPERATION:



I2C READ OPERATION:

CHAPTER 16: DIFFICULTIES ENCOUNTERED IN PROJECT

1) We forgot to order the debug header to be soldered on our PCB for debugging and loading the program. The workaround involved using male to female header wires soldered sufficient enough to program the board
2) There were some passive components such as capacitors which were back ordered from Digikey. To fix that issue, we used capacitor values that were close enough to the desired value and our circuit did worked successfully.
3) Since printf function was not supported by the BGM11 module soldered on the board, we had to redirect the UART_putchar() function to sprintf() function to successfully output the characters on the serial terminal
4) At one point, I accidentally erased the preloaded bootloader present on the chip. The solution that we applied was to flash the .s37 file which seemed to fixed that issue.
5) Also, while writing the firmware for BMI160 IMU sensor, the values of accelerometer seemed to change very frequently even though it appeared to be in a stable position. After reading through the datasheet and application note, I adjusted the sensitivity and range parameter of BMI160 sensor fine enough to keep stable readings.
6) For the Panasonic Grideye sensor, the sensor seemed to stuck in loop while initialization due to the delay that was inserted between two simultaneous read operations. The solution was to insert a delay that was large enough to allow the sensor to stabilize. The workaround seemed to work pretty good as the sensor no longer got stuck in the loop.
7) When we fixed the board to the controller, due to some obstruction ,the ADC value did not went down full scale and only went half way through. To fix that problem, we had to recheck what value the ADC gave under no press condition and note its maximum and minimum voltage value. We then mapped the new range to ADC and it worked.

CHAPTER 17: SUMMARY OF FUNCTIONALITY OF FINAL PROJECT

The project that we implemented looked complicated in first pass considering the hardware and software blocks it needed to work. However, we got all the basic blocks functional and were able to get the product work successfully as desired.

The one thing that needed more work was the Panasonic Grideye which was unable to detect the gesture effectively. So, if we had more time, we would have definitely improved the algorithm that was used to detect gesture in Panasonic Grideye.

Apart from this, we got the IMU and ADC working which were able to control the steering and acceleration/deceleration of car. The buttons were interfaced properly and were properly programmed and mapped to different functions.

Another thing that I think needed more work was addition of energy modes to further lower the power consumption of wireless game controller. Initially, we had planned to use interrupt mechanism from IMU to generate significant motion interrupt to notify end of inactive period. Addition of all the above parameters would have definitely lowered power consumption.

CHAPTER 18: FIVE LESSONS LEARNED NOT TAUGHT IN LECTURE BUT LEARNED THROUGH DOING THE PROJECT

FIRMWARE LESSONS:

- Always use a logic/protocol analyzer when debugging I2C or SPI interface, they often come handy to know which part of transaction the protocol is stuck at
- It is better to write firmware beforehand on evaluation kit before porting the firmware on your actual printed circuit board. In that way, you get to know that atleast your firmware is working properly. Any incorrect behavior can then be targeted in resolving hardware bugs without having to worry about firmware
- Always write UART driver for any microcontroller before proceeding to software debugging. UART helps us to get continuous data (fastens debugging) rather than to stall program flow by use of breakpoints.
- Lastly, try to test every function that you write before using it in another function. This way, you gain some confidence and helps you to know which function is actually trying to break the firmware.

HARDWARE LESSONS:

- Although this was my first course in PCB design, I learnt it is better to review the schematic and layout from someone else. Sometimes, there are issues that might not catch your attention in first glance but can be caught with by another person.
- Always verify the footprint that you obtain from third party website like SnapEDA. There is no guarantee that the footprint and symbol that you are importing is correct and matches the datasheet.
- Always perform test on evaluation kit like measuring different voltage and current value. This will give you some values that you can expect while validating your printed circuit board.

MISC LESSONS:

- Try to always have a backup plan so that it can be executed when things do not work as expected
- It is better to divide the work equally if working on a team. This will help you to reach your goal faster and leave some time for debugging.
- Try not to make any last minute changes (mostly for hardware layout).That is because last minute changes are mostly not verified for worst case conditions and may give you inadvertent results.