## 

## 

[**Problem statement:**](#_3yrealpd16ry) **2**

[**Project Proposal:**](#_3yrealpd16ry) **2**

[**Use Case:**](#_wwzbuu1ke7xn) **3**

[**Energy storage longevity requirements:**](#_52uzc4tiuiyq) **3**

[**Charge time Requirements:**](#_gxrwbsoj43qx) **4**

[**Hardware functional block diagram:**](#_ublng6i70uc8) **4**

[**List of key hardware components:**](#_11vdhlrngpmz) **5**

[**Software functional block diagram:**](#_ij9zmpk40ni7) **6**

[**High-Risk aspects:**](#_nak1fnrf1cop) **6**

[**Risk Mitigation plan:**](#_jvd3v1hcck6k) **6**

[**Project Update special topics:**](#_jxezkx5d0hj0) **8**

[**Energy use case:**](#_3ufsgbkumbro) **8**

[Calculations to specify the Energy Storage Device](#_ponbpmougf00) 8

[Energy Storage Device Justification:](#_3k9doulcknqo) 9

[Justification for MCU](#_1jcq5k7vxrq1) 9

[Justification for WiFi SoC](#_95c2zwv6g0m) 10

[Justification for PMU Solution](#_ecwszxj1rju4) 10

[Justification that voltage droops can be supported without failure](#_qg2gef6xkjjp) 10

[Proper engineering selection of PMU ceramic load capacitance](#_iaraq1avb5m4) 10

[Adequate ESD protection](#_pjr6ljyguxy4) 11

[Verification plan](#_i62gvu81h1t) 11

## 

## Problem statement:

As the Internet of Things continues to develop, both established security companies and newcomers are beginning to digitize home security via devices such as the Ring Doorbell[[1]](#footnote-0) or ADT Security Cameras[[2]](#footnote-1). Studies show that surveillance cameras can serve as a strong deterrent for crime[[3]](#footnote-2), and can help bring justice in the case that a crime occurs. With regards to the business itself, a cursory glance at the market forecast of home security shows[[4]](#footnote-3) that there is a huge amount of money to be made in security-specialized embedded systems. For these reasons, we believe that it would be both beneficial to the world and also lucrative to carve out a new niche in the field of wireless security sensor devices.

## 

## Project Proposal:

Although there are many competitors in this market (such as the Ring and ADT devices mentioned above ), we believe there are ways we can make our security node stand out. There are some basic features that represent a baseline successful device: at the very least, a security device must include a way to record video and audio, and then a way for a device owner to retrieve the captured data.

We will use an application to increase accessibility to past alerts and quick responsivity during an alert event. We have created a design that leverages renewable energy sources in order to free up our nodes from a power grid while still potentially not needing external recharges. Additionally, we have sensor technology to sense and log an intrusion event, as well as provide a timely alert to a user via our Android application.

## Use Case:

The state machine for our project is depicted below:

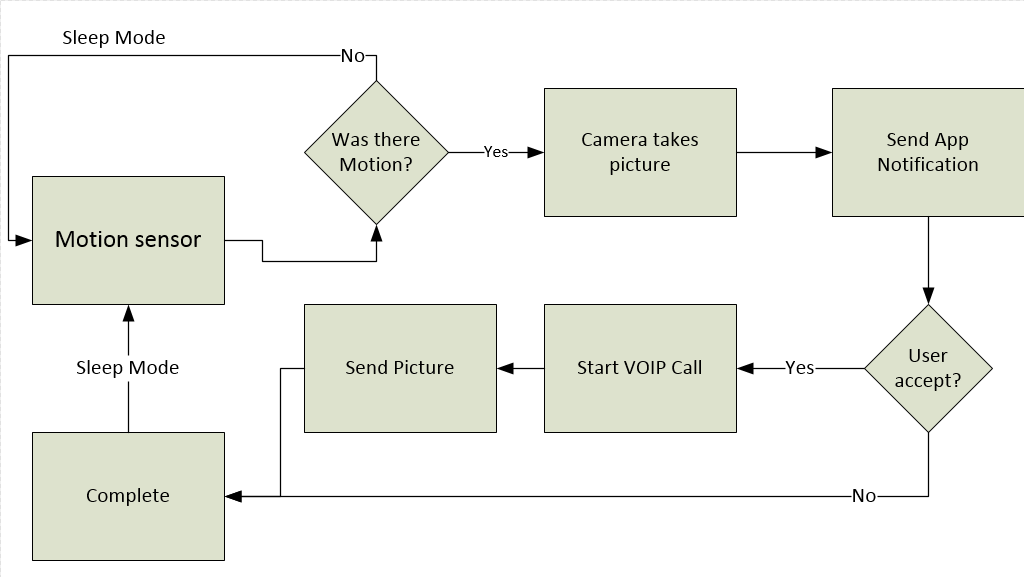


Figure 1: the flowchart that describes the PanoptiNode behavior.

The main use case for this device is for a motion camera up near the roof of a house that draws power from a solar panel mounted on the roof and connected via a power cable. The expectation of events per day is between 2 and 3: Two events when homeowners leave to work and return, and occasionally more events during package deliveries, people coming over, or anything else. The device is expected to be largely self-sufficient, and with very little/nothing in the way of a direct user interface. It can relay a very basic status via an onboard LED or else all information from the device will be transmitted via packets.

## Energy storage longevity requirements:

Based upon the use case described above, we would like for our project to last as long as possible between charging events. It is reasonable to expect that a user would find a battery recharge irritating at the least, and likely compromising as it will force them to deactivate their intrusion detection node for perhaps a few hours to charge. Ideally the spotty nature of high-current events and the ability of the PMU to recharge the battery off of alternate energy sources could cause this device to never need a charge beyond that of the solar panel. However, we believe that being able to run for roughly 180 days on an expectation value of 2 events per day on average is sufficient for our project.

## Charge time Requirements:

Ideally the project will be constantly charging due to its trickle-charge PMU, large-capacity battery, and the fact that the device only occasionally draws full charge. This will allow the solar panel to be charging the LiPo most of the time, provided that the sun has a clear line to the panel. If for some reason the LiPo battery needs to be charged separately from the solar panel, it will likely take a few hours, as is common for similar LiPo batteries in phones, drones, or other devices. The logic for this rating is as follows: Our solar panel can output up to 1A in strong light- by back-of-the-napkin estimation, we could say that the panel on average could output ½ to ⅓ of that, so between 300-500mA. For most of the day, the board will be asleep and pulling far less than 500mA, so most of the day it would be charging the LiPo at the 100mA charge current limit. Therefore, the charge time would be most of the time, punctuated by short periods of great discharge lasting about 5-15 seconds or so. If there is a busy day where these events occur very frequently and/or the solar panel is unplugged/ unable to receive enough light, then the LiPo will eventually burn out and need to be unplugged and recharged for a few hours.

## Hardware functional block diagram:

Here is the hardware diagram for our project. It has been updated since the initial proposal to remove the GPS module, and the specific interfaces to the Pearl Gecko MCU have been enumerated. Basically there are several peripherals all attached via drivers to the microcontroller. The microcontroller is the foundation of the project, and it interfaces directly with every other peripheral.

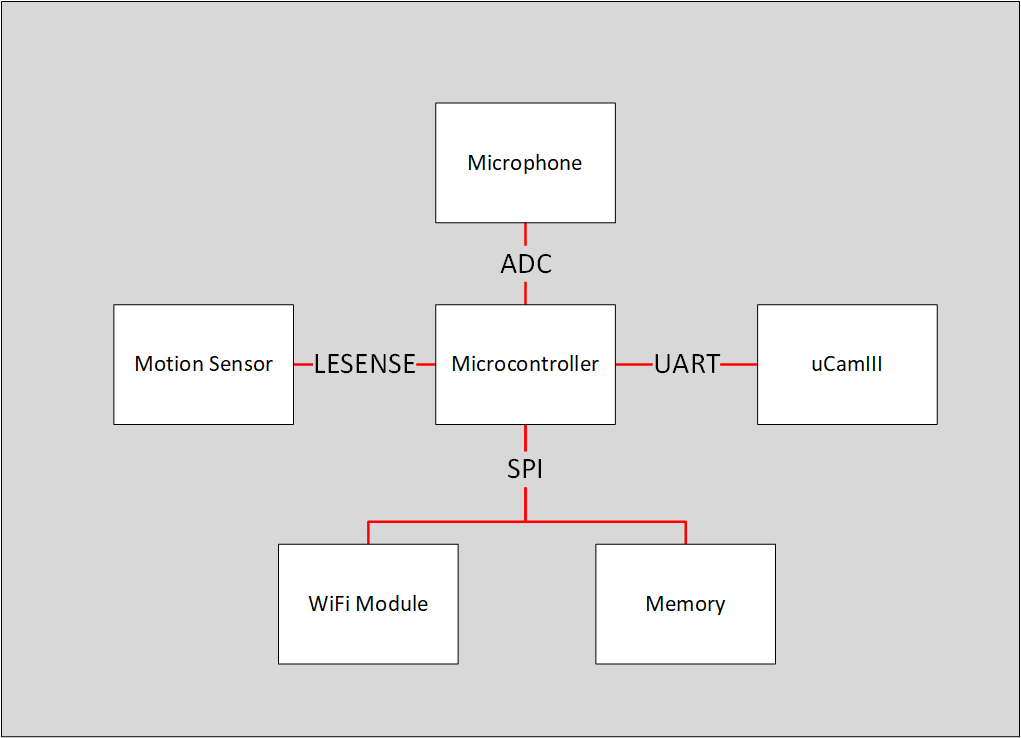


Figure 2: The PanoptiNode’s peripherals and interfaces to the MCU

Another important facet of our hardware design is our power tree. Our board draws power from a 3.7V LiPo battery as well as a ~6V solar panel, and then powers our MCU and peripherals off of a 3.3V and 5V output rail as shown below:

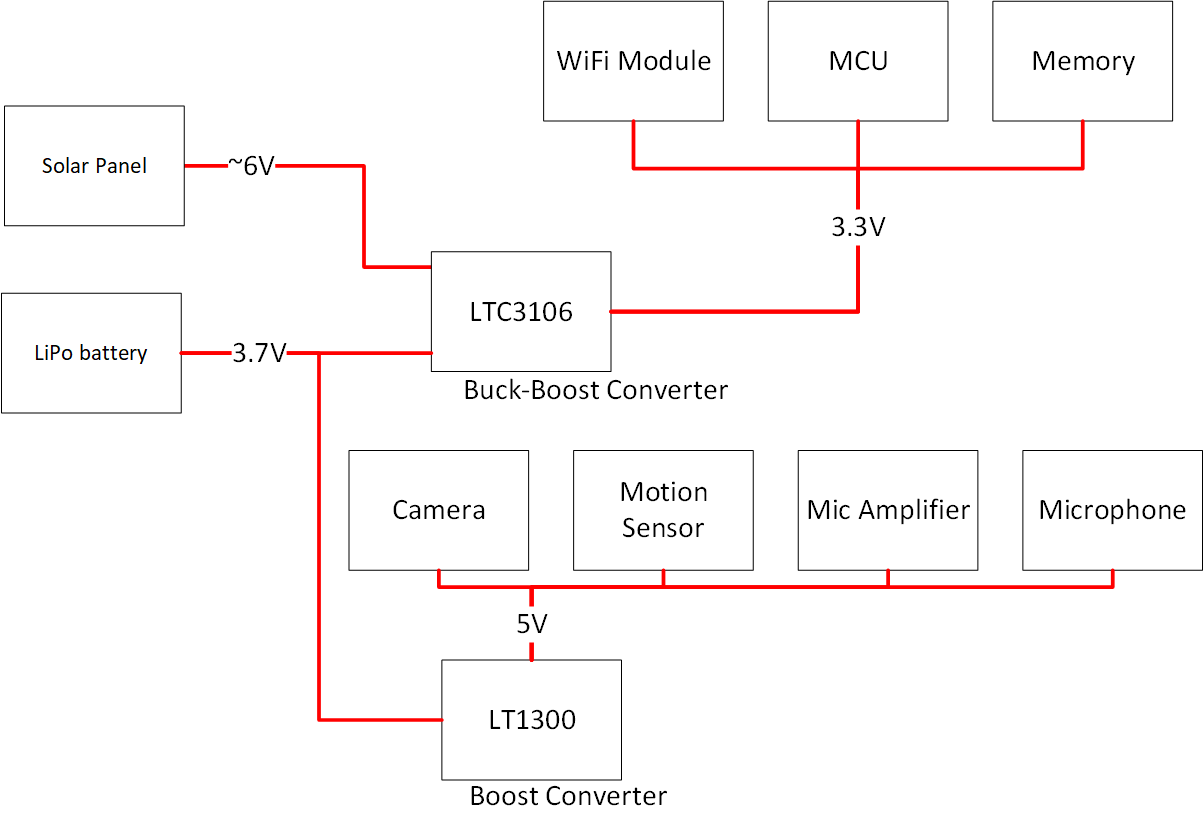
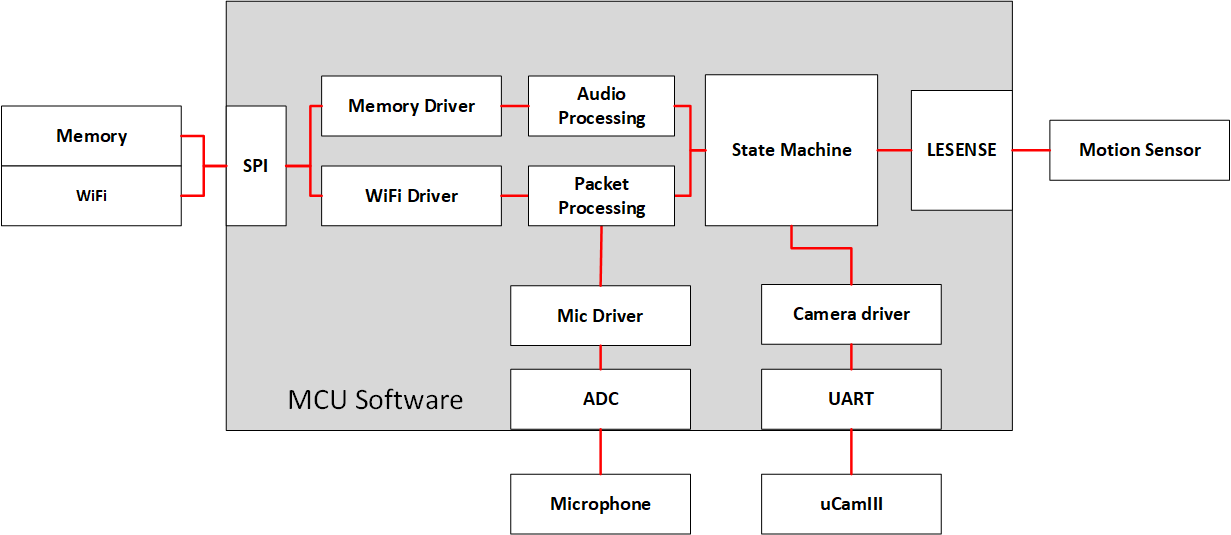


Figure 3: the power tree for the PanoptiNode showing voltage input and output for each IC.

## List of key hardware components:

1. MCU - EFM32PG12B500F1024GM48
2. Flash Memory - SST25VF010A
3. Motion Sensor - AMN24112
4. Camera - uCamIII
5. WiFi Module - WGM110A1MV2R
6. Electret Microphone - CMA-4544PF-W
7. Electret Microphone Amplifier - MAX9814
8. PMUs - LTC3106 and LTC1300
9. Energy Harvester - PRT-13783
10. Energy Storage Unit - LiPo Battery: PRT-13855

## Software functional block diagram:

Figure 4: The software diagram for the project including all necessary modules.

## High-Risk aspects:

The high-risk aspects of the project are the following

* Getting the PMU to work/ re-charging the LiPo from the solar panel.
* Learning and effectively using the WiFi’s BGAPI to send VOIP packets
* VOIP audio specification/ requirements - streaming audio reasonably to the app.
* Using a memory chip/ added complexity from this device
* Network connectivity of the device/ number of boxes required.
* The voltage and current within all tolerances for all peripherals is tricky.

## Risk Mitigation plan:

Here are the mitigations to the above, in a one-to-one mapping from the points above

* We have decoupled the “digital power” from the power sources for both of our power rails, allowing us to power and program our board while we continue to debug the PMU. If all else fails, we can redo the PMU with a breakout board and just pass the power rails back via the ISO digital power rails.
* We will attempt to defuse the learning time for the WiFi BGAPI by quickly getting the board to flash and interacting with the wifi module, then giving this driver development a high priority. Additionally, we will keep a possibility in our back pocket of just flashing the microcontroller on the WGM110A and hard-coding a few shorthand messages for it to interpret if the BGAPI proves too challenging to master.
* We will attempt to use other sources and sinks at first to figure out how to solve the VOIP problem. We will look for an AWS service that can provide a SIP server, and then just attempt an app-to-app transmission or something similar to learn more about the protocol before we need to implement sending from the PanoptiNode.
* Similar to the WiFi module, we will attempt to defuse the learning time for the memory device by quickly getting the board to flash and interacting with the memory chip, then giving this driver development a high priority.
* We have largely tamed this risk by using AWS services and stubbing out functions/ using simple tests and integrations to ensure connectivity up and down the chain. Connectivity is currently verified from AWS all the way to the app, now the only thing needed is the PanoptiNode client- which will only need to implement the simplest of clients.
* We will ensure proper voltage and current by verifying digital power rails and PMU outputs separately using a DMM and/or oscilloscope before using the PMU to power the entire board. When using digital power, we will ensure that the current is limited and the voltage is extremely close to the nominal voltage (3.3V and 5V) for each rail. When using the PMU circuit, we will observe that the voltages are proper and current output are reasonable again using a DMM before un-isolating the traces. We have further reduced this risk by carefully selecting parts whose tolerances are within the output limits of our PMU ICs.

## 

## 

## 

## Project Update special topics:

## Energy use case:

There is one main use-case for the project- it is a largely-scripted interaction that occurs after motion is detected. The device wakes up, sends a push notification, and (assuming the user chooses to investigate the intrusion), takes a photo to send and starts up streaming audio from the microphone. The device should have an extremely short duty cycle, as intrusions may not come for days on end (depending on the placement of the device), and then calls will likely not go on for more than a minute. The vast majority of energy expenditure in this project will come when the user is streaming photos and audio from the node, as shown in the figure below.

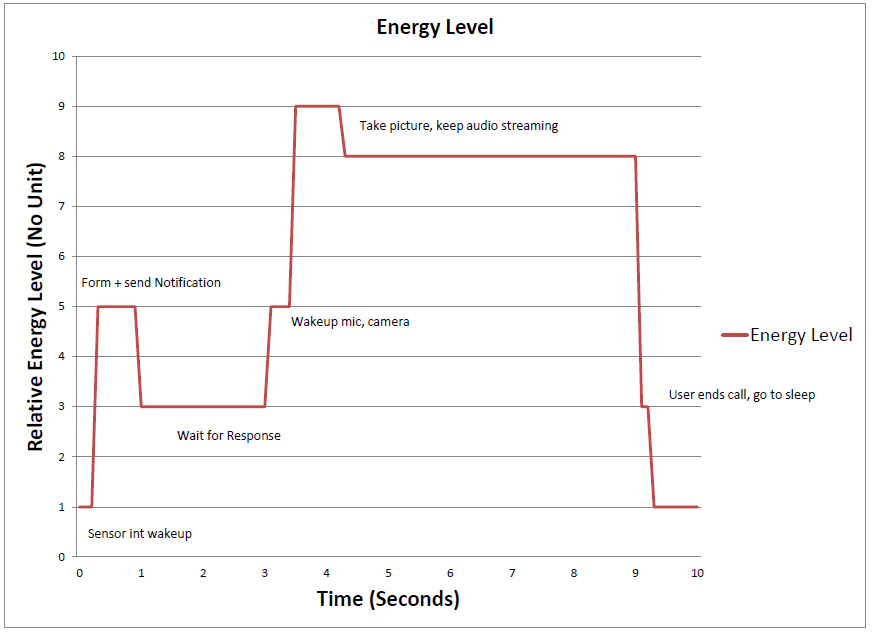


Figure 5: Relative current consumption across a wakeup event on a scale of 1-10 based on number and current draw of awakened peripherals.

### 

### Calculations to specify the Energy Storage Device

▪ Average current / energy capacity calculation

The average power consumption for our device was about 13.5 mAh for an average use case of 3 events per day. We have included C Rate calculation in our energy requirement excel sheet (see attached with the document), which is around 8.8e-4 C. To calculate this, we used the following logic: if a C-rate of 1 implies total battery discharge in 1 hour, then our C-rate corresponds to the average current expenditure in an hour divided by our battery capacity in mAh. At our calculated C-rate, the battery can last up to 3532 hours without recharging. We are aiming our product power life to be roughly around 4 months without recharging and therefore the C Rate and power calculation coupled with our energy harvesting system more than justifies our battery selection of 2000 mAh. In fact, if the estimated number of events per day drops to two (which seems reasonable), we can support 6months between charges. Unfortunately, our battery part does not provide much of a datasheet at all, so we are unable to find the battery’s discharge curve.

▪ Peak currents and Peak current step functions

The peak discharge rate of the circuit is about 280 mA per intrusion event while the microphone and WiFi module are transmitting. The recommended charge and discharge rate for this battery is 0.2C= 500mA, which is well above our needs, and our PMU and step-up converter each support at least 280mA given our specifications. We later discovered that the WiFi module prefers a direct connection to the battery for its current-hogging RF components, rather than through a PMU or converter- we have since incorporated this direct battery connection into our PCB so that the large >250mA current draw of the WiFi module will not strain the LTC3106 directly and thus lower the strain on our PMU.

### Energy Storage Device Justification:

▪ Number of recharge cycles

As discussed above, we have very low power requirement and we have selected a 2200 mAh battery to power the device keeping longevity of the device into consideration. Hence, for an average use case the requirement of chargings is once per 2 months. But having a good energy harvester. It will self charge if sunlight is available and battery voltage threshold is below 3.3V

▪ Recharge cycle use-case justification/ supporting documents

Rather than requiring a complete removal and charging, we expect to have a constant charge/recharge cycle as mentioned in various entries above. In the nominal case, we expect that the constant charging of the battery from the solar panel during the day will be enough to offset the total mAh usage for a given day, and thus the board will be self-powered without need for a removal and recharge-cycle. It can be expected that occasional, “greater-than-average” wakeup time may cause the battery to fully discharge over days- but the 2000mAh capacity battery that we have selected should allow the board to be powered for days on end with even a larger number of average events per day.

### Justification for MCU

After analyzing a number of processors, we have finalized our decision on the Silicon Labs Pearl Gecko series. It has 4 USARTS which can be configured as UART or SPI as dictated by our requirements. Apart from that, it also has sufficient number of I/O pins needed by our project. The initial impetus to change from our choice of Giant Gecko last week was the possibility of using the ADC and LEUART in EM3- the power benefits were obvious, but there was an open question as to whether we could work with the reduced number of serial interfaces and I/O pins- we believe we can survive with the 48-pin package, and so we have chosen this microcontroller for the project.

### Justification for WiFi SoC

The WGM110A (AKA Wizard Gecko) provides a savings of $40 from the TI alternative, an on-chip antenna, a slightly lower TX current draw, and most importantly a SPI-based “BGAPI” protocol for configuring and interfacing with the WiFi module purely over serial communications. This brings much simplicity to our development as we can directly call the APIs using SPI and transmit and receive data.

### Justification for PMU Solution

We found the LTC3106, a buck-boost battery manager that allows primary and secondary battery with charging from the energy-harvester, accepts input of up to 6V and can output 3.3V with almost 500mA of current, and can programmatically limit charging when the battery gets above a certain voltage.

With our worst case current usage of ~300mA, this seem to be perfect fit for our use case.

Since we have some peripherals like camera and microphone that require 5V input power, we decided to add another step-up converter called an LT1300 which is able to convert 3.3V rail to 5V output and apparently source enough current to power our camera and microphone through our peak current needs.

### Justification that voltage droops can be supported without failure

We have selected ICs that have voltage droop protection built into them: The LT3106 has a low- and high-voltage dropout setting which will power down the chip. Additionally, this chip has a PGOOD line tied to the MCU’s reset to hold the chip in reset until the power is stable. The LTC3106 outputs a rail of 3.3V that promises to be between 3.22 and 3.40- within the tolerances of the devices it feeds.

As for the LT1300 Boost-Converter, it accepts input voltage as low as 1.8V (well below the shutoff current within the LiPo’s protection circuit) and accepts as high a maximum voltage of 10V. The LT1300 IC has a large enough range to accept the entire conceivable range of the battery, and promises to output a 5V rail between 4.8 and 5.2- again well within the tolerances of the devices it feeds.

### Proper engineering selection of PMU ceramic load capacitance

For LTC 3106, based on the Application circuit (Fig 16) in the datasheet and simulation of the PMU circuit on the LTSpice simulation tool, we selected 47uF as the ceramic load capacitance. On our simulation we were getting output voltage of about 3.3V and 400mA current for 3.7V battery input. Similarly for LT1300 we performed simulation and got the desired result of 5V for 0.1 uF ceramic cap coupled with 100 uF electrolytic capacitor

### Adequate ESD protection

Since we are not using any USB ports or buttons and also during schematic review we didn’t encounter any special need for any ESD protection circuit. According to the feedback from update 6, the professor had stated that “If the signal is within your PCB, no ESD device is normally required”. As we have no external devices beyond our power supplies, we fit this criteria.

### 

### Verification plan

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test # | To be verified | Definition of passing | Tested By | Measured result | Date performed | Passed? |
| 1 | 5V does not drop below min value | Over a 30sec oscilloscope measurement of 5V line with board powered on, it does not drop below 4.8 |  |  |  |  |
| 2 | 5V does not go above max value | Over a 30sec oscilloscope measurement of 5V line with board powered on, it does not rise above 5.2 |  |  |  |  |
| 3 | 3.3V does not drop below min value | Over a 30sec oscilloscope measurement of 3.3V line with board powered on, it does not drop below 3.2V |  |  |  |  |
| 4 | 3.3V does not go above max value | Over a 30sec oscilloscope measurement of 3.3V line with board powered on, it does not rise above 3.4V |  |  |  |  |
| 5 | Signal quality for each communication bus is accurate | Attach logic analyser to the signal test points to check the signals and also send “Hello World” message over UART and SPI to host and see if logic analyser receives it correctly |  |  |  |  |
| 6 | Energy Storage Element Charging | Power on board with both battery and solar panel plugged in, observe VSTOR Pin sources negative current to the battery. |  |  |  |  |
| 7 | Verify Mic output | Check if voltage at the output pin goes above the set threshold of 2 volts when sound occurs and indicating the event by toggling onboard led |  |  |  |  |
| 8 | Verify Motion sensor output | Tape-measure out a line 8 feet beyond motion sensor, ensure that person walking 8 feet away can cause motion sensor output to trigger our EFM32PG12 comparator |  |  |  |  |
| 9 | Verify system receiving enough power from Energy storage unit | Observe full cycle of state machine including TX/RX from WiFi module and uCam photo without brownout. |  |  |  |  |
| 10 | Verify radio can transmit and be receive | Send a string to public echo server and see if receives it correctly by printing it out |  |  |  |  |
| 11 | Verify radio can negotiate a WiFi connection | Have our device attempt to get an IP, scan our subnet for the device |  |  |  |  |
| 12 | Verify Energy Harvesting can charge the energy storage element to the target use case specification | Measure current out of solar panel - ensure nominal voltage is 5V and nominal current is > 80mA |  |  |  |  |
| 13 | Verify the energy storage element can provide the energy to the  target use case specification while not being assisted by external  energy source | Run device off of battery with solar panel unplugged at barrel jack input and ensure full software flow occurs. |  |  |  |  |
| 14 | Verify System receiving Power Good | Check Power Good Test point and see if it stays high after boot up |  |  |  |  |
| 15 | Microphone is sufficiently clear | Standing 2 feet away, choose 5 random words from dictionary- aim to understand at least ⅘ of them. |  |  |  |  |
| 16 | Camera responds to UART commands | Ability to send and receive UART commands in the app. |  |  |  |  |
| 17 | Camera image is of acceptable quality | Capture the image and send it the host and check if the image is clear and can differentiate objects. |  |  |  |  |
| 18 | VOIP call maintained | Be able to maintain a 10 second one-way VOIP call to a target recipient. |  |  |  |  |
| 19 | Timely Notification Alerts | Generate an intrusion event, then ensure user is notified within 15 seconds of the event. |  |  |  |  |
| 20 | Read/write Picture from Memory | Send the captured image and store it in memory and read the memory to check if image is correct and clear. |  |  |  |  |
| 21 | No false positives from motion sensor | Leave motion sensor for awhile, ensure sense pin stays relatively stable over 10min with no stimuli |  |  |  |  |
| 22 | Use all possible sleep modes | Ability to place EFM32PG, WiFi module, Camera in sleep mode. |  |  |  |  |
| 23 | MCU working correctly | Check if MCU is recognized by the debugger, if it can be programmed by flashing a blink led code. |  |  |  |  |

Figure 6: Verification table as we plan to use at the end of the project.

1. https://ring.com [↑](#footnote-ref-0)
2. https://www.adt.com/video-surveillance [↑](#footnote-ref-1)
3. https://news.rutgers.edu/news-release/rutgers-study-finds-alarm-systems-are-valuable-crime-fighting-

   tool/20090205#.Whds60qnFPY [↑](#footnote-ref-2)
4. https://www.marketsandmarkets.com/PressReleases/home-security.asp [↑](#footnote-ref-3)