Wake-up Harvester Design for Batteryless IoT System

DESIGN DOCUMENT

sdmay21-14

Henry Duwe

Jacob Bernardi, Edmund Duan, Douglas Zuercher, Kwanghum Park, Bryce Staver, Zacharias Komodromos

sdmay21-14@iastate.edu

https://git.ece.iastate.edu/sd/sdmay21-14

Executive Summary

Development Standards & Practices Used

As with any consumer hardware device, this project's device will be accompanied by multiple schematics and figures explaining each module. Testing will also be performed to verify the functionality and operation of the system. When designing the PCB, the team will follow standard design practices by creating a well-documented, clean, solderless board. RF design practices will also be followed, such as the creation of a solid ground plane on the top and bottom layers, stitching vias throughout the board, and enough shielding to protect against noise. Furthermore, critical traces and components will not be placed near elements which may introduce noise into the system, such as DC-DC regulators and switches.

A limited amount of software will also be necessary to complete this project. All software will be well-documented with both in-code comments and external documentation. Software will follow standard practices for the language and environment that it is written in, including limiting memory used.

IEEE standards relevant to project:

IEEE 211-2018 - IEEE Standard Definitions of Terms for Radio Wave Propagation

IEEE 149-1977 - IEEE Standard Test Procedures for Antennas

Summary of Requirements

The below list provides the requirements relevant to this project.

- Wireless IoT Device with Wake-Up System
- Wakes up to an RF signal trigger
- Capable of node-to-node communication for signal relaying
- Low power consumption

Applicable Courses from Iowa State University Curriculum

A number of courses taught at Iowa State University are applicable to this project. These courses include EE 201, EE 230, EE 321, EE 330, and EE 414.

New Skills/Knowledge acquired that was not taught in courses

Table of Contents

List of figures/tables/symbols/definitions (This should be the similar to the project plan)

1 Introduction

1.1 Acknowledgement

The senior design team would like to formally acknowledge and thank Henry Duwe, Assistant Professor at Iowa State University, for his technical advice on wake-up radios and low-power IoT devices.

1.2 Problem and Project Statement

Many Internet of Things devices are made to be wireless. In order for them to operate, they make use of some form of energy harvesting system. Once power is harvested, it is stored in an energy storage unit, which is then used to power the device with values neither constant nor of high magnitude. The result is then, when dealing with a wireless device, it is highly likely, if not guaranteed, that there will be times when the system is not powered. In order to help with power usage, a hysteresis loop is introduced in order to guarantee the device is not constantly cycling on and off when it is charged at the minimum power threshold. However, this hysteresis loop also causes inconsistency for when the device is active and inactive. Therefore, a method of ensuring that the device is only active at specific times is necessary.

The solution presented by our team to this problem is the creation of a device which utilizes a wake-up system to wait for a wake-up signal to be received. Upon reception, the wake-up system will interrupt the hysteresis loop and force the microcontroller (MCU) to activate, allowing the device to complete time-dependent tasks. This will create an efficient and feasible way of waking up low-power wireless IoT devices, allowing for greater ability for IoT devices to be synchronized and to perform tasks in unison if they have enough charge to be powered on and are in sleep-mode.

1.3 Operational Environment

The end-product is expected to be operated in indoor commercial environments, such as office buildings or shopping malls, where electro-magnetic noise is not of significant concern and where RF signals are prominent. The environment is expected to be clean and climate-controlled such that dust and humidity are unlikely to interfere with electronics.

1.4 Requirements

There are two types of requirements that our team considers to be of high priority: functional requirements and technical requirements.

Functional Requirements:

The wake-up harvester should activate only if there is enough charge to support turning on and maintaining the MCU, and if a synchronization signal has been sent. While the MCU is in low-power sleep mode, the wake-up IC should always be active, so that it can receive RF signals at any time to trigger the MCU back to its on-mode. This essentially means that if the MCU can be powered on, then the wake-up IC should be able to trigger it. If the IoT device does not have enough charge to support turning the MCU on, then the wake-up IC should not be able to turn the MCU on. The device as a whole has to be minimally sized and must operate wirelessly at a distance reasonable for IoT devices. Lastly, it must reliably be able to wake-up other devices like itself within a distance.

Technical Requirements:

The device will operate when its energy storage unit has at least 1.02V, and the components should be low-power and operate from 0°C to 50°C. Since the wake-up radio has to be active while the IoT device has enough charge to support operation, the power consumption for the harvesting unit must be low, ideally in the micro-watt range. Additionally, the MCU unit should be capable of typical operations, while also being able to transmit signals.

1.5 Intended Users and Uses

The product is intended to be used by low-power, wireless IoT device manufacturers. The product is intended to be used specifically for devices which need to perform occasional computational tasks but do not need to be active at all times.

1.6 Assumptions and Limitations

Assumptions:

- Available RF energy to be harvested is in the magnitude of 250 mW or lower
- The IoT device will be in a network of nodes that are close enough to trigger each other
- Wireless
- The RF trigger should be a 16-bit digital packet
- The device can operate in commercial environments where electro-magnetic noise is not of significant concern

Limitations:

- The device's range where it can be triggered will be around 5 meters
- Not tested in high noise environments
- No physical tests (shaking, etc.)

- No ESD tests
- No test on humidity, dust aversion, temperature.
- Device lifespan not tested.

1.7 Expected End Product and Deliverables

The end-product is expected to be a low-power IoT device composed of a digital computing platform (a low-power MCU), supporting electronics for the MCU, an RF Harvester, and a Wake-Up system. The system will be designed to operate in commercial environments, such as office buildings, where electro-magnetic noise is not of significant concern and where RF signals are likely to be received.

The final product will be a low-power IoT device composed of a digital computing platform (MCU), a RF Harvester, a Wake-Up module and an energy storage unit. All of these modules will be assembled on a PCB with a suitable enclosure for protection from the environment (i.e. dust, low levels of moisture). Further details on each portion of the product are provided below.

The MCU will be programmable by the user, and can be used in any way the user sees fit and reasonable. The pins will be broken out so that the final product can interface with other boards, making the product like a device “shield.” It will be necessary for the device to at least have one pin configured as a digital input. When “woken up” by the Wake-Up system, the pin will be pulled high or low, allowing for the MCU to be aware that it was activated by a trigger instead of by receiving its minimum operating voltage. A secondary pin can also be used to make the Wake-Up system send a signal to other neighboring IoT nodes. There will be an output pin used to connect to the transmission antenna, along with a provided guide on how to program the functionality to transmit the trigger signal if the user intends to use the device to trigger another one.

The RF Harvester will not be something the user will be able to interface with or access. Any attempt to change the operation of the harvester may result in unexpected behaviour and possibly failure of the whole device.

The Wake-Up system is the module responsible for receiving and processing the input trigger. It is up to the user on how to generate the RF signal to trigger the device. A provided option is using one of these IoT devices to trigger another one. The device is meant to trigger when a number of specific frequencies are used in the analog trigger. This implementation prevents accidental triggering which could significantly drain the device's energy if it keeps cycling on and off. The trigger is not intended to be secure from external sources; meaning that if someone knows the frequency pattern being used, they will be able to trigger the IoT device if it has enough charge to support the device being active.

2 Project Plan

2.1 Task Decomposition

The main tasks for this project are detailed in the figure below.

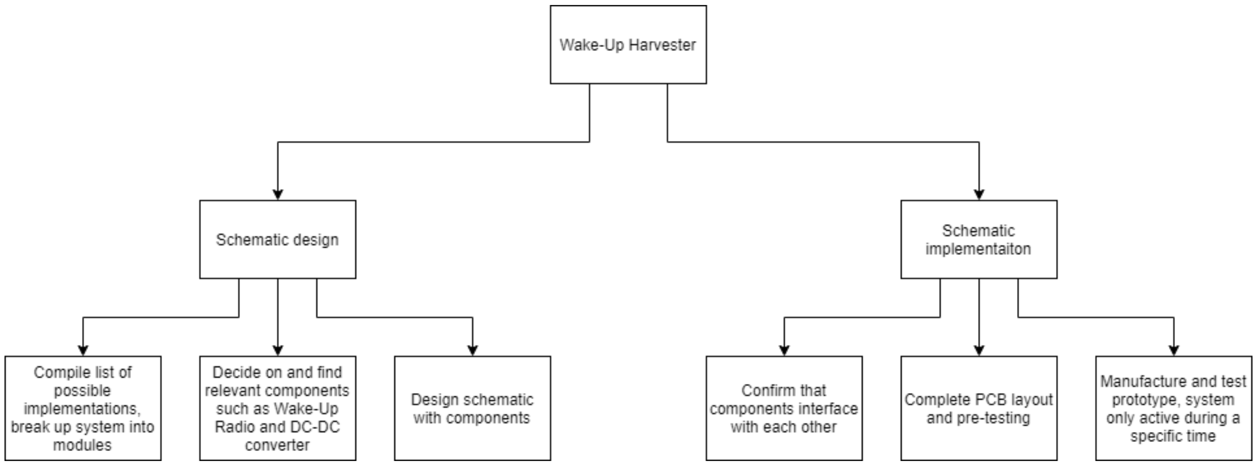

Figure 1: Task Decomposition for Project Design and Testing

2.2 Risks And Risk Management/Mitigation

The biggest risk associated with this project involves the system used to power the wake-up module, with a risk factor probability of around 0.3. The power source for the module will be a DC-DC boost converter that exclusively powers the wake-up module that takes a very low input voltage and boosts it up to a voltage that the module can operate at. The risk lies in the efficiency of the system since DC-DC boost converters are less efficient at low input voltages and low power applications. The team's method of minimizing risk is using a pre-made DC-DC boost converter which is made to be used for this purpose.

Another risk is the wake-up module itself, which is essentially a receiver and filter (risk factor probability of 0.2). While it is feasible for the team to make a custom module, it is very likely that it would add to the power consumption of the device, which is detrimental as it is necessary to keep power usage minimal. Beyond that, creating a high frequency filter with a precise enough band may prove expensive if attempted. For this reason, the team has opted for a RF wake-up IC that has these features built in and is optimized for low power consumption. In doing so, the team can almost guarantee the signal will be received and processed correctly and efficiently.

2.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

Milestone 1: A design concept that takes all requirements and assumptions into consideration with all initial bases covered on paper. Concept is peer-reviewed and approved by all team members.

Milestone 2: Creation of a low-level, detailed schematic for each module with generic parts. Simulations indicate that the system functions as intended.

Milestone 3: All components are chosen to spec. To pass this milestone, all devices necessary from the past milestone should be chosen to fit the design requirements. Based on guaranteed component specifications, the complete design should fall in a reasonable power consumption range. All devices which have SPICE models should be included in the previous milestones' simulation, and simulations should indicate that the system functions correctly.

Milestone 4: Completion of PCB layout and routing, with design rule checks passed. Peer-review and approval necessary for the milestone to be completed.

Milestone 5: PCB ordered, received, and populated. This milestone is complete when the board has been fully populated and is ready for testing.

Milestone 6: Testing of revision one is complete and changes necessary to ensure the proper operation of revision two have been identified.

Milestone 7: Revision two schematic revisions and simulations complete, along with board layout and routing.

Milestone 8: Second revision PCB ordered, received, and populated. This milestone is complete when the board has been fully populated and is ready for testing.

Milestone 9: Final revision tested and confirmed to be functional per required specifications.

2.4 Project Timeline/Schedule

WBS NUMBER	TASK TITLE	START DATE	DUE DATE	COMPLETE
1	Revision 1			
1.1	List of possible implementations	S1 W1	S1 W8	90%
1.1.1	High-level system design	S1 W8	S1 W11	20%
1.2	Low-level schematic complete	S1 W11	S1 W14	10%
1.3	Component selection	S1 W12	S1 W14	10%
1.4	Testing and schematic confirmation	Interim W1	Interim W2	0%
1.5	PCB Layout and routing	Interim W2	Interim W6	0%
1.6	Testing (rev. 1)	Interim W7	Iterim W8	0%
2	Revision 2			
2.1	Schematic updates	Interim W8	S2 W1	0%
2.2	Testing and schematic confirmation	S2 W2	S2 W4	0%
2.3	PCB Layout and routing	S2 W4	S2 W6	0%
2.4	Testing (rev. 2)	S2 W6	S2 W7	0%
3	Final Revision			
3.1	Schematic updates	S2 W7	S2 W9	0%
3.2	Testing and schematic confirmation	S2 W9	S2 W10	0%
3.2.1	PCB Layout and routing	S2 W10	S2 W11	0%
3.2.2	Testing (final)	S2 W11	S2 W13	0%

Figure 2: Project Timeline for Fall 2020-Spring 2021

2.5 Project Tracking Procedures

Our team is using Git, Google Drive, Confluence and Discord in order to track the progress of our project. Git will be used later in the process for version control and maintenance of our hardware design files (PCB and schematic implementations). This tool is being used because it provides an easy way to have version control and to have a centralized place where all members of the team will be able to pull the current working versions from. Google Drive is used to store agendas, meeting notes, images, as well as any collaborative assignments done in EE 491. It provides the team with in-browser tools to be able to complete these documents collaboratively while working in a virtual environment. Confluence is a centralized place to keep notes, documentation and links to all of our devices. It is also a method of implementing SCRUM since we can use its tools to organize ourselves both time and action-wise. Finally, Discord is used for quick communication and sharing general thoughts regarding meetings and the overall project plan. It provides a way for us to organize different aspects into separate channels, join calls and share our screens through a simple process.

2.6 Personnel Effort Requirements

Task	Number of people	Number of hours required (cumulative among members)
Understand what the project is about, how the device for project work and what components are in the device	6 members	20 hours
Research on technologies to use and list the options to choose the best one by comparing the pros and cons	6 members	40 hours
Split up the components to research the wake-up system	3 members	20 hours
Work on Rev. 1 (Design, assembly and test)	6 members	70 hours
Work on Rev. 2 (Design, assembly and test)	6 members	40 hours
Work on final rev. (Design, assembly and test)	6 members	20 hours
Weekly meeting with advisor and client to report the project status and plan	6 members	1 hour per a week

Weekly group meeting to discuss about the project and check if everyone is on the right track	6 members	2-4 hours per a week
Write a Bi-weekly report to update the project status and explain about the plan for next week	6 members	1.5 hours per week

Figure 3: Decomposition of project tasks and effort requirements

2.7 Other Resource Requirements

To complete this project we will first and foremost need access to a PCB Design software. In our case, we will be using Altium Designer, which is provided by the university. In addition to this tool, we will need a custom PCB designed by the team, which can be purchased and received within 2 weeks. In addition to the board, we will need a variety of IC components, some of which include a Powercast RF harvester, a Wake-Up IC, an MCU, antennas and a few smaller components that we will populate the board with.

2.8 Financial Requirements

The different components used in this project will not exceed $150. On top of this, we will need to pay for the PCB itself for which the price can vary from vendor to vendor. A good estimate for the PCB is around $200.

3 Design

3.1 Previous Work And Literature

Although we are the first senior design team to take on a Wake-Up harvester, the concept of Wake-Up radios has existed before, despite being relatively recent. Wake-Up radios are described as devices that are active only when they receive a specific signal or trigger and many have been made in the past few years, an example of which is listed in section 6.2. In our system, we are using a premade component for our wake-up radio in addition to premade components for our power harvester, MCU, and DC-DC regulator. The datasheets for these components are listed in section 6.2.

3.2 Design Thinking

When defining the scope of this project, we initially believed that we would have to modify the harvester itself. However, in the "define" stage, it became more apparent that we would not have access to the controller on board to activate the harvester's output. After we revised our design, we then moved to the "ideate" phase, in which we began looking for low-power components that could work in the system such as a Wake-Up radio and a DC-DC regulator. The components we could find drove our design, since there is a small market for low enough power components, which meant our design was fairly constrained.

3.3 Proposed Design

A number of solutions to the problem of creating IoT devices that can perform computational tasks while operating on a low-power or inconsistent power supply already exist, with all solutions focusing on ways in which power can be conserved when not needed. The most popular way of conserving power is by placing the computing device, typically some form of a microcontroller, in a sleep state until an internal watchdog timer triggers and reactivates the device. In this fashion, the MCU can be set to use minimal power for the vast majority of the time, only consuming significant amounts of power during the intervals where it activates and performs computational tasks.

While this solution is undoubtedly efficient, it is only appropriate for IoT devices that do not need to network with other devices or are networking with systems that are always active (for example, an IoT device that periodically downloads data from a server). For devices that need to communicate periodically with other periodic devices, this solution is not suitable since it depends on devices re-activating at precisely the same times. For small meshes, this may be achievable by occasionally ensuring that both devices are properly synchronized, but the reliability of this solution decreases dramatically for meshes containing more than a trivial number of nodes.

It is clear that a more appropriate solution is a system that shifts the responsibility of reactivating the MCU from the MCU's internal watchdog timer to a low-power device capable of networking with other devices, a wake-up radio or always-on low-power radio transceiver being examples. With these devices, the MCU can be configured to only re-activate when a set of packets on a designated carrier frequency are received, allowing for device activation to occur on the receiving of a signal external to the node itself. In theory, this also introduces the ability for a node to wake up neighboring nodes by reconfiguring its transceiver to transmit packets, allowing the opportunity for the mesh size to not be limited by the range of the original transmitter.

Shifting the responsibility of waking the MCU from the MCU itself to another component creates an additional opportunity, being the opportunity to *completely* unpower the MCU when not in use. This saves additional power since the MCU is no longer even maintaining basic bare-bones operations, and in theory helps get some of the power back that is being spent to maintain the operation of the always-on transceiver. This optimization was heavily explored by the team, but ultimately the decision was made to not use this power-saving strategy for one primary reason: while more efficient, it significantly reduces the number of tasks that can be performed by the system. For example, the user may decide that they want the MCU to still reactivate automatically after a period of time, whether or not it received an external trigger, so that it can complete another portion of its task. Completely unpowering the MCU removes this ability, limiting the system's uses. It was thus decided to not unpower the device and to instead continue just placing the device into a sleep mode.

The final design proposed by the team is a design that incorporates three primary components: an RF harvester, a low-power always-on transceiver, and a low-power MCU. The RF harvester will harvest and store energy from background RF signals on a capacitor, which will then be used by the harvester's in-chip boost regulator to power the MCU. The capacitor will also interface with an additional low-power high-efficiency boost regulator, which will power the always-on transceiver and necessary circuitry for the transceiver.

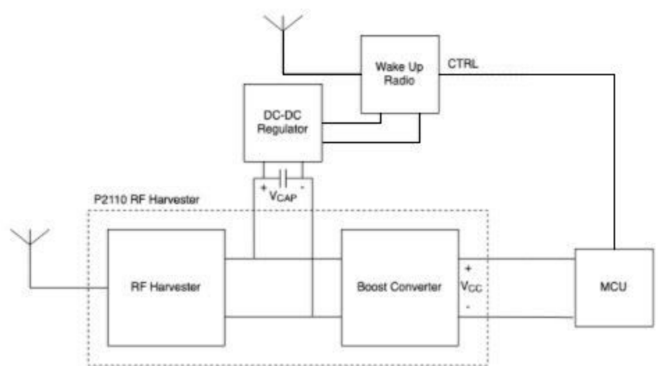

Figure 4: Proposed Design

The transceiver and MCU will be connected in such a way that the two devices can communicate with each other through SPI, allowing the MCU to reconfigure the transceiver at will and allowing for the transceiver to wake the MCU. During normal operation, the MCU will be in a sleep state where only functions such as interrupts handlers and watchdog timers are active. The transceiver will be in a wake-on radio mode, waking up at intervals to briefly check if a carrier signal is present. When a carrier signal is present and a valid packet of data is detected, the transceiver will trigger the MCU's external interrupt, waking the device from sleep mode and activating it. The MCU will complete its tasks and re-enter sleep mode.

3.4 Technology Considerations

The greatest technological consideration that must be considered for the system is the power consumption of each node. Because each node is a device that harvests its own power, there are two effective limitations on power supply: how much the nodes can harvest in a given period and how much the nodes will consume in a period. The necessity for the nodes to be wireless means that these challenges are unavoidable.

To consider the topic of how much the nodes can harvest in a given period of time, one must first look at the harvester that the customer encouraged the team to use, as it is the source of the power used by each node. Unfortunately, the harvester in question, the Powercast P2110B, does not provide detailed information on its typical harvesting capacity and capabilities. It instead refers to stating its maximum capacity and capabilities. As a result of this, additional testing is needed before harvesting considerations can be made.

While the consideration of how much the nodes will harvest in a period is defined by the capabilities of the harvester being used, the measurement of how much the nodes will consume in a period of time is not. This evaluation depends on the power consumption of the components in each node and on the size of the capacitor we decide to use. The larger the capacitor, the more storage capacity the node will have and thus the longer the nodes can keep their on-board MCUs active. The downside is that it will take longer to charge and thus may keep the system out of commission for longer. On the other hand, a smaller capacitor will charge quickly but won't be able to support the system for long, especially on high power operations.

To mitigate the issue, and maximize the energy storage while decreasing the off time, our team needed to choose very low power components and try to decrease consumption as much as possible. During the design phase we had a decision to make on how to power the trigger receiver module of the device. It could either be powered from the Powercast harvester or the capacitor itself. Using the option of powering off the Powercast as a base case, we looked into methods of improving the consumption on paper. This led to the solution of using an external DC-DC converter that had a higher efficiency at low voltages compared to the Powercast in order to power the trigger receiver module.

In concurrence with minimizing consumption, the MCU module of the device has transmission capabilities. However, its capabilities are limited to a small number of transmission protocols. This prompted us to explore other options, and in doing so we found transmitters that would operate with lower consumption rates compared to the MCU when transmitting. Having a need for both transmission and receiving, we could now spec a transceiving device as our trigger module that would act either as a receiver or as a transmitter with a higher efficiency than the base case of using the MCU, as well as not adding to the total number of components used.

3.5 Design Analysis

Each iteration of the design process had an issue that significantly changed the overall design. Starting from the first design, we assumed there was a way to bypass the harvester module's hysteresis loop to force it on during its off cycle. While it looks like a good plan in theory, after looking into the datasheet and trying to get in touch with the manufacturer, we concluded that it was not possible to have access to the on board DC-DC converter to trigger it on.

For the second design idea, its simplicity is what was most attractive. The biggest issue with this design is that the MCU would not be off when the trigger is received. Specifically, since the onboard harvester's hysteresis loop is not something we have access to, the trigger won't change the operation if the MCU is active. On the other hand, if the MCU is off, it is because the Powercast DC-DC converter is off, meaning that it is in the charging state. If the trigger connected the MCU to the Powercast output, it would still not turn on. Thus this design was redundant and didn't meet the goals and requirements of the final device.

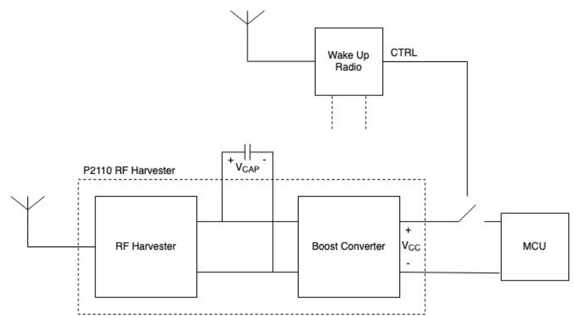

Figure 5: Design Option 2 - Rejected

Finally, the current design is to make use of the MCU's sleep mode. With this method, the trigger would take the MCU out of sleep mode when triggered. What this ends up achieving is, if the user has intensive operations running on the MCU which require it to be on, it will operate the device as usual. On the other hand, if the device is idle, instead of consuming power while not accomplishing anything, it can go into sleep mode. If the user has time sensitive operations that are internal to the device, the MCU is fully capable of getting itself back to normal operation. On the other hand, if the user has time sensitive operations that don't rely on an individual device, they can use an external wake-up signal to trigger the MCU on. What this ultimately achieves is a longer lifecycle for the device, and a way of externally triggering it on.

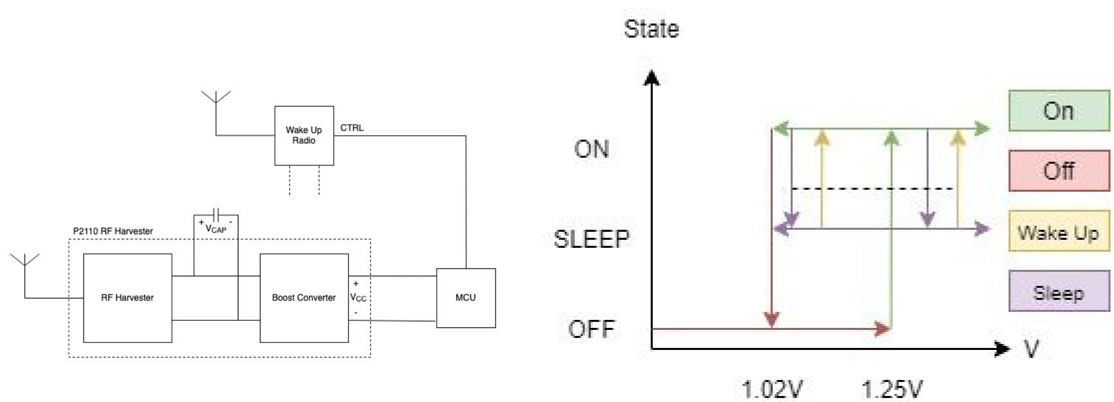

Figure 6: Design Option 3 - Current

3.6 Development Process

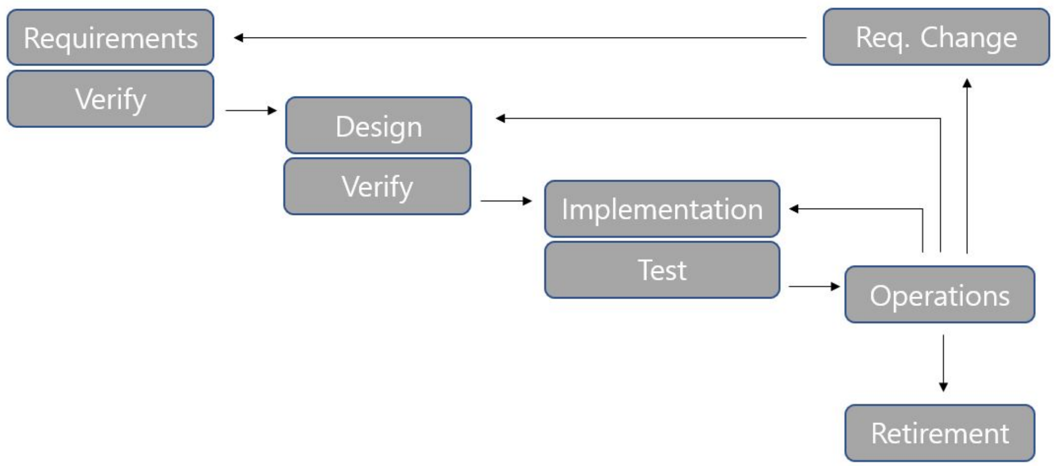

Figure 7: Modified version of waterfall model

We decided to choose the development called “Modified version of waterfall model.” The original waterfall model has the disadvantage of not allowing repetition. Therefore, it is inflexible and monolithic to the requirements. We have to design the high-level systems like MCU, Wake-up radio, and DC-DC regulator, and we are required to change the design of them whenever there is a problem with testing or verifying processes. For our project, we have to combine all these parts together to make sure they are working well together. Because of this, we have to test and change it often. However, the original waterfall model is not flexible, because it cannot proceed to the next stage until the previous stage is completed. To solve this disadvantage, we chose the modified model. This model has more flexible features for documentation and for analysis of changing requirements.

Also, when we get access to our components and have space for testing, we will move into a test-driven design process. The reason for this is we rely at first on on-paper material to get as many details of the project planned out, and as soon as we have access to work with materials in hand we will start using our results from module-oriented testing to optimize and refine the overall design. The benefit of this is we won’t have to backtrack as much in the processes if something fails, and it also provides the opportunity for a multi-front approach where failure or setbacks in one does not affect the others.

3.7 Design Plan

To design and build our project, we are going to use components available in the market. We first had to research for the reasonable Powercast RF harvester, a Wake-Up IC, an MCU, and DC-DC regulator. Our project is intended to be used by wireless IoT device manufacturers with low power consumption. Considering the requirements, we had to find the right components which can be run by low power.

For our design, we are going to use Altium Designer. Altium Designer is an electronic circuit design software that provides the entire process of PCB development such as circuits and simulations in an integrated development environment. The whole process of development can be developed efficiently because it can be carried out through a single program.

We set the workspace to design the footprint and high-level system of components to test if our chosen components work. Our project needs to be verified and tested, because one part may not be operated well by another and we need to make sure that all the components are connected and work well.

4 Testing

4.1 Unit Testing

The main parts of this project that will be tested in isolation will include the RF harvester, the wake-up transceiver, and the MCU. Much of this testing will be done using a power supply so that we can take our time debugging and not rely on the intermittent power the RF harvester would provide. To start we will be testing the RF harvester on its own so that we can ensure that it can take in an RF input and produce a boosted DC output. In addition to ensuring that it operates correctly, this will be an opportunity to understand what amount of power we can expect to collect when this device is in an office setting. This information will be valuable in confirming that the components selected will be viable with this power source.

Another component that will be unit tested is the wake-up radio system that utilizes a sub-GHz frequency transceiver to send and receive a “wake-up signal.” The purpose of testing this device on its own is to confirm that the power drawn is less than the average power that the power harvested will be. If the transceiver draws less than the average power harvested, we can be confident that the system will be able to receive and send a “wake-up signal” at any given time and that the harvester will not limit its functionality.

Lastly, we will be testing the MCU’s functionality in isolation before combining it with the rest of the system. The main things that we want to test include making sure that we can send a “wake-up” interrupt to the MCU and that it can put itself into sleep mode when not performing computations. The interrupt will be coming from the sub-GHz transceiver but before interfacing the two systems we will make sure that our MCU program can recognize an interrupt and leave sleep mode to complete its task. Note: The task that the MCU performs is not in the scope of this project. The last thing that the MCU needs to do is once that task is completed it should take the proper steps to safely re-enter sleep mode.

4.2 Interface Testing

Once the proper unit tests have been completed, we will be ready to begin interfacing the MCU, RF Harvester, and Wake-up Radio system to work together as an RF Harvesting IoT device. One of the most important interface tests will be making sure that the Wake-Up radio system can send and receive the "wake-up signal." This means that we will be using two separate radio systems where one will act as the transmitter and one as the receiver. The purpose of this is to ensure our custom "wake-up signal" can be transmitted and received between devices, which was a main feature desired by the client. Related to this test, we will also need to check that once the wake-up signal has been received, the transceiver can send another signal to the MCU that will act as the interrupt to leave the sleep state.

The other important interface test is to confirm that the RF power harvester can support the wake-up radio system at any given time. This would mean making sure that the wake-up radio system isn't intermittently turning on and off because it constantly drains the capacitor the harvester is connected to. Assuming that the harvester is able to support the wake-up radio, the other thing to test would be that the harvester's boost converter can turn on and supply 3.3V to the MCU. If there are no issues supplying the necessary power to the other components, we can be confident that the RF harvester will be able to consistently support the entire device.

4.3 Acceptance Testing

To demonstrate that the design requirements for the final product are all met, an overall test of the system will be performed. Multiple nodes will be set up throughout a room such that each periodically turns on and performs a simple task before entering a sleep mode which will be exited automatically after some delay on the devices' internal watchdog timers. The purpose of this is to demonstrate that the MCU can activate itself periodically, a requirement that customers may need, and so a complicated task is unnecessary; driving an LED to be active for a few seconds will be sufficient. The nodes will be allowed to run for several hours to ensure that they are deactivating and not extraneously draining the charge available to each node. Throughout the experiment (near the beginning, middle, and end), the nodes will be triggered by the primary transmitter. If the nodes are operating as expected, they will all simultaneously activate and perform their task. Doing this multiple times throughout the lifetime of the experiment will prove that the nodes are fully functional even after running for a long duration of time and there has been enough time for the charges on each node to deplete dramatically.

Once confirming that all nodes are able to wake-up from the central transmitter, the experiment will then be repeated, but with the nodes spaced farther and not all located around the transmitter. If nodes outside the range of the transmitter are able to activate upon the transmitter being activated, this will prove that the nodes are capable of forwarding the wake-up signal to other neighboring nodes.

4.4 Results

– List and explain any and all results obtained so far during the testing phase

- Include failures and successes
- Explain what you learned and how you are planning to change the design iteratively as you progress with your project
- If you are including figures, please include captions and cite it in the text

5 Implementation

Describe any (preliminary) implementation plan for the next semester for your proposed design in 3.3.

6 Closing Material

6.1 Conclusion

Summarize the work you have done so far. Briefly re-iterate your goals. Then, re-iterate the best plan of action (or solution) to achieving your goals and indicate why this surpasses all other possible solutions tested.

6.2 References

List technical references and related work / market survey references. Do professional citation style (ex. IEEE).

"CC1352R," Texas Instruments, May-2020. [Online]. Available: https://www.ti.com/product/CC1352R. [Accessed: 23-Oct-2020].

L. Wilhelmsson and D. Sundman, "Wake-Up Radio – A key component of IoT?," Ericsson.com, 18-Dec-2017. [Online]. Available: https://www.ericsson.com/en/blog/2017/12/wake-up-radio--a-key-component-of-iot. [Accessed: 23-Oct-2020].

"Low Input Voltage Boost Converter," Texas Instruments, Dec-2014. [Online]. Available: https://www.ti.com/lit/ds/symlink/tps61202.pdf?HQS=TI-null-null-digikeymode-df-pf-null-wwe&ts=1600912252796. [Accessed: 23-Oct-2020].

"P2110B Datasheet," Powercast, 2016. [Online]. Available: https://www.powercastco.com/wp-content/uploads/2016/12/P2110B-Datasheet-Rev-3.pdf. [Accessed: 23-Oct-2020].

6.3 Appendices

Any additional information that would be helpful to the evaluation of your design document.

If you have any large graphs, tables, or similar data that does not directly pertain to the problem but helps support it, include it here. This would also be a good area to include hardware/software manuals used. May include CAD files, circuit schematics, layout etc,. PCB testing issues etc., Software bugs etc.