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.

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. With this hysteresis loop, there now exists an opportunity where the device is not powered on at a time when it is critical for the device to be on, such as during a synchronization event. A method of ensuring that the device is active at specific times is thus 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.

For the 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.

For the 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.

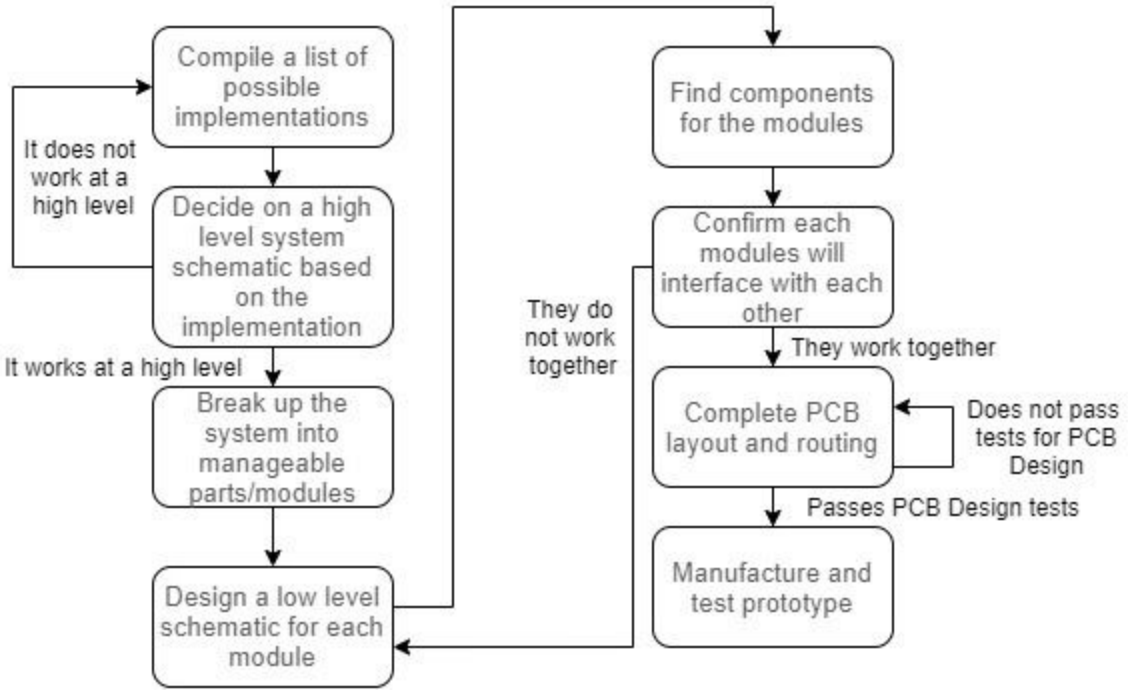
The RF Harvester will not be something the user will be able to interface with or access. Any attempt to change the operation of the harvester may result in unexpected behaviour and possibly failure of the whole device.

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 flowchart below.



2.2 Risks And Risk Management/Mitigation

The biggest risk associated with this project involves the system used to power the wake-up module. 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. Although the probability that the efficiency will be high enough is not definite, using a dedicated system definitely increases the chances.

Another risk is the wake-up module itself, which is essentially a receiver and filter. 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.

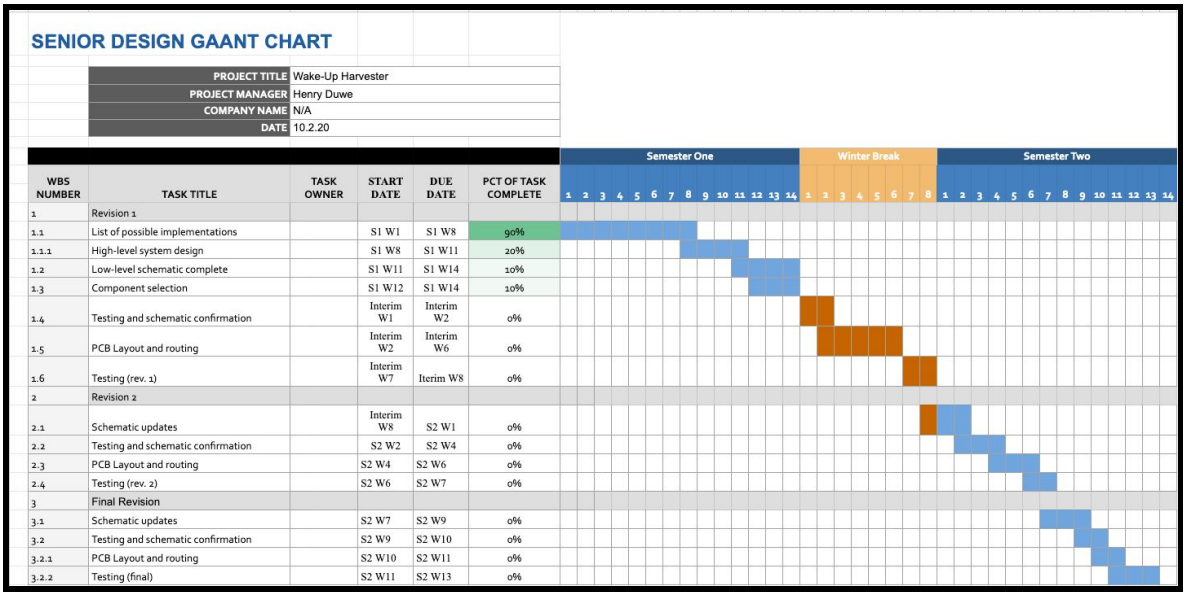
Milestone 6: Testing of revision one is complete and changes necessary to ensure the proper operation of revision two have been identified.

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



SENIOR DESIGN GAANT CHART

PROJECT TITLE	Wake-Up Harvester
PROJECT MANAGER	Henry Duwe
COMPANY NAME	N/A
DATE	10.2.20

WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUE DATE	PCT OF TASK 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%

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

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

Include relevant background/literature review for the project

– If similar products exist in the market, describe what has already been done

– If you are following previous work, cite that and discuss the **advantages/shortcomings**

– Note that while you are not expected to "compete" with other existing products / research groups, you should be able to differentiate your project from what is available

Detail any similar products or research done on this topic previously. Please cite your sources and include them in your references. All figures must be captioned and referenced in your text.

3.2 Design Thinking

Detail any design thinking driven design "define" aspects that shape your design. Enumerate some of the other design choices that came up in your design thinking "ideate" phase.

3.3 Proposed Design

Include any/all possible methods of approach to solving the problem:

- Discuss what you have done so far – what have you tried/implemented/tested?
- Some discussion of how this design satisfies the **functional and non-functional requirements** of the project.
- If any **standards** are relevant to your project (e.g. IEEE standards, NIST standards) discuss the applicability of those standards here
- This design description should be in **sufficient detail** that another team of engineers can look through it and implement it.

3.4 Technology Considerations

Highlight the strengths, weakness, and trade-offs made in technology available.

Discuss possible solutions and design alternatives

3.5 Design Analysis

- Did your proposed design from 3.3 work? Why or why not?
- What are your observations, thoughts, and ideas to modify or iterate over the design?

3.6 Development Process

Discuss what development process you are following with a rationale for it – Waterfall, TDD, Agile. Note that this is not necessarily only for software projects. Development processes are applicable for all design projects.

3.7 Design Plan

Describe a design plan with respect to use-cases within the context of requirements, modules in your design (dependency/concurrency of modules through a module diagram, interfaces, architectural overview), module constraints tied to requirements.

4 Testing

Testing is an **extremely** important component of most projects, whether it involves a circuit, a process, or software.

1. Define the needed types of tests (unit testing for modules, integrity testing for interfaces, user-study or acceptance testing for functional and non-functional requirements).
2. Define/identify the individual items/units and interfaces to be tested.
3. Define, design, and develop the actual test cases.
4. Determine the anticipated test results for each test case
5. Perform the actual tests.
6. Evaluate the actual test results.
7. Make the necessary changes to the product being tested
8. Perform any necessary retesting
9. Document the entire testing process and its results

Include Functional and Non-Functional Testing, Modeling and Simulations, challenges you have determined.

4.1 Unit Testing

– Discuss any hardware/software units being tested in isolation

4.2 Interface Testing

– Discuss how the composition of two or more units (interfaces) are to be tested. Enumerate all the relevant interfaces in your design.

4.3 Acceptance Testing

How will you demonstrate that the design requirements, both functional and non-functional are

being met? How would you involve your client in the acceptance testing?

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).

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.