Sub-Gigahertz Arduino Shield + Remote

Team 16

Members: Alexander Beck, Christopher Baldwin, Faruk Toy

TA: Anthony Schroeder

Professor: Arne Fliflet

1 Introduction

1.1 Objective

Electronic devices become more and more common in our everyday lives. Some of us are hobbyists and want to build their own projects. No matter who makes the electronic equipment, they need to be controlled. Although switches are the main source of control in today’s world, they require physical contact. It is more convenient for users to be able to operate multiple devices remotely. We, as a team, are motivated to provide solutions to this problem.

We recognized that Arduinos are easy-to-use tools that can be utilized by many people, including non-experts. In recent years, Arduino shields were introduced to the market to create new capabilities using Arduino devices and platforms. They can be used to solve the problem we are interested in. The goal of this project is to create a low-cost, low-power, long-range remote controller Arduino shield. This will allow people to include remote control capability in projects or add remote control to their existing devices.

1.2 Background

People increasingly want to interact with things remotely. Whether it be led lights, home automation, or our vehicles. What we found is a gap in the market for cheap, low power, and long range wireless communication. What our project hopes to solve is allowing people to remotely control any of their Arduino projects. The advantages to the Sub-Gigahertz protocol is that it is lower power and longer range than other protocols like WiFi, Bluetooth, ZigBee, etc. while also allowing for many devices to be connected at once. Users can easily create their own network of IoT devices without having the security risk of connecting them to their home network. All of these devices can be controlled from one central unit and all at a low cost and easy to use platform.

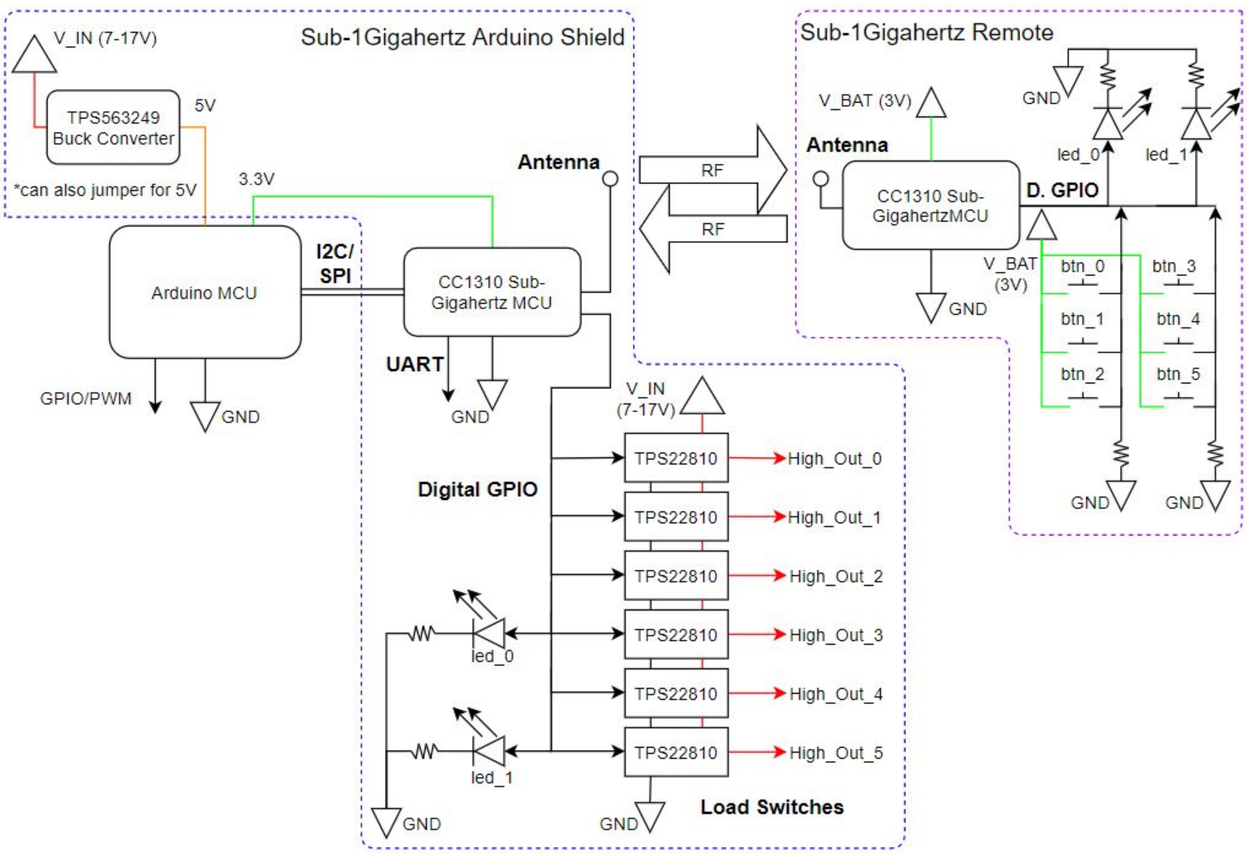
Our project will be to build a shield with a Sub-Gigahertz MCU and develop a library which offers the Arduino user an easy to use experience. We plan on connecting the shield with a serial protocol to maximize the number of IO pins for the user and allow for other shields to be stacked. We also hope to add load switches to the shield to allow for added voltage and current output capability. The shields can be configured to connect to other shields or to our remotes. The battery powered remote will have it's own MCU with RF and GPIO to handle accepting button presses and being able to give visual and auditory feedback to the user.

1.4 High Level Requirements

- The shield should regulate DC input voltage to 5V and supply 450-600 mA for both the Arduino and the CC1310 microcontroller.
- The remote control must be able to send wireless signals with frequencies less than 1 gigahertz to the arduino shield.
- Having received a signal from the remote, the arduino must execute its pre-programed response.

2 Design

2.1 Block Diagram



2.2 Functional Overview

This project has 2 subsystems: Sub-1Gigahertz Arduino Shield and Sub-1Gigahertz Remote.

Sub-1Gigahertz Arduino shield will receive input from a 7-17 V DC input. We originally design the project with a vehicle application in mind although it can be used in many Arduino applications. As a result, we set our input voltage range as 7-17 V which includes vehicle battery range. This subsystem will have a Buck converter to regulate voltage down to 5 V with 5% range to power Arduino. To ensure the system can power both Arduino and Buck converter, maximum current drawn from the converter would be at most 600 mA with a safety margin of 1.5x-2x the maximum rated current.

The shield will have a CC1310 microcontroller to enable remote control communications. The CC1310 microcontroller will be powered by Arduino's 3.3 V supply output for external devices. The microcontroller will establish a communication bus with Arduino through GPIO pins. Provided Arduino libraries will analyze the input from bus pins and configure the outputs of other GPIO pins for load switches. Therefore, load switches will be turned ON or OFF according to signal from remote. Finally, load switches will connect to external electrical devices that are desired to be controlled remotely. Those devices will be powered from the external sources, not by the shield.

The second subsystem will be the Sub-1Gigahertz remote system. This subsystem will be powered by a coin battery which supplies 3V to the system. The button on the remote corresponds to different controls on the shield load switches. Based on the button pressed, CC1310 microcontroller will generate RF signals to be processed in the shield.

2.3 Block Requirements

2.3.1 Buck Converter/PMIC

Arduino and shield will be powered by a 12 V vehicle battery. However, Arduino needs 5V input and CC1310 needs 3.3 V input. This block will serve as voltage regulator.
Requirements:

1. Provide 5 V ±5% from a 7-17V DC source in the range of 0-600 mA.
2. The system should remain below the maximum operating temperature of all the devices at room temperature (20 °C - 30 °C).
3. Efficiency of the DC-DC conversion from 7-17 V to 5 V must be 80% or higher at 600 mA.

2.3.2 Sub-1Gigahertz Arduino Shield

To enable remote control operations, CC1310 microcontroller will interact with the Arduino through GPIO pins. Also, Arduino's 3.3V output will supply power to the CC1310 microcontroller.

Requirements:

1. CC1310 controller will be powered from the 3.3 V output of Arduino
2. The shield will be able to send and receive instructions and data with the Arduino over the I2C and/or SPI bus protocols.
3. The shield will trigger RF/GPIO signals from commands received over I2C and/or SPI bus protocols. The shield will also buffer commands sent by the remote and/or other shields.
4. The load switches will output the supply voltage to the VOUT pin when EN is high and GND when EN is low. This the output will reach 90% of the supply voltage within 500 us of EN reaching 50% of its target voltage.

2.3.3 Sub-1Gigahertz Remote

The shield receives RF signals representing the commands from the user.

Requirements:

1. The remote will be powered by a coin style battery without the need for a DC-DC converter.
2. The remote will be able to trigger a GPIO output given a command over RF.
3. The remote will send a command over RF when a button is pressed.

2.4 Risk Analysis

We do have some risk especially with board assembly. Parts like the CC1310 and some of the associated passives use extremely small pitch components. This can lead to bridged solder pads and ruining PBCs and components. We will need to be especially careful when assembling boards and will likely need multiple operating boards in order to reduce the risk of failing the project demo.

There is risk associated with the PCB layout, specifically the RF antenna design as well as the higher voltage traces. We are looking at best practices by isolating parts of the board for HV so that we reduce the risk of shorting traces.

3 Ethics and Safety

Since the arduino can be used in outdoor projects, there is a possibility it will get rained on. The water can damage the arduino and may cause short circuits. We plan on using an enclosure to try and reduce the moisture around the PCB traces and to keep dust off the PCB. This will help mitigate the risk of shorts.

We plan to use a battery to power our remote control. This is a potential safety risk since batteries do not offer short circuit protection and can corrode. We will take additional steps to reduce the risk of short circuits by isolating traces as well as possibly implementing an eFuse. eFuses can detect over-current events and cut power to the system. They also allow for the system to re-try so that events like ESD cause the system to be reset by removing power all together. To reduce the risk of battery corrosion exposure, the battery will be contained within the case of the remote. The case should only be opened when the battery needs to be replaced. This will greatly reduce the time the user touches the battery, therefore reducing the risk of potential battery corrosion exposure.

Since the arduino shield accepts wireless signals, someone may try to replicate those signals to access the arduino shield without the owner's permission. What we plan to do is investigate the use of encrypted data messages such that the packet structure will be obfuscated and an adversary would not be able to simply listen to an exchange and decode the message structure.

4 References

[1] Texas Instruments, *CC1310 SimpleLink™ Ultra-Low-Power Sub-1 GHz Wireless MCU*, 2018. [Online]. Available: https://www.ti.com/lit/ds/swrs181d/swrs181d.pdf?ts=1600391985951

[2] Texas Instruments, *TPS563249 17-V, 3-A, Constant 1.4-MHz Synchronous Step-Down Voltage Regulator*, 2018. [Online]. Available: https://www.ti.com/lit/ds/symlink/tps563249.pdf?ts=1600392228838

[3] Texas Instruments, *TPS22810, 2.7-18-V, 79-mΩ On-Resistance Load Switch With Thermal Protection*, 2018. [Online]. Available: https://www.ti.com/lit/ds/symlink/tps22810.pdf?ts=1600392338461

[4] Octopart, *Arduino Uno*, n.d.. [Online]. Available: https://datasheet.octopart.com/A000066-Arduino-datasheet-38879526.pdf