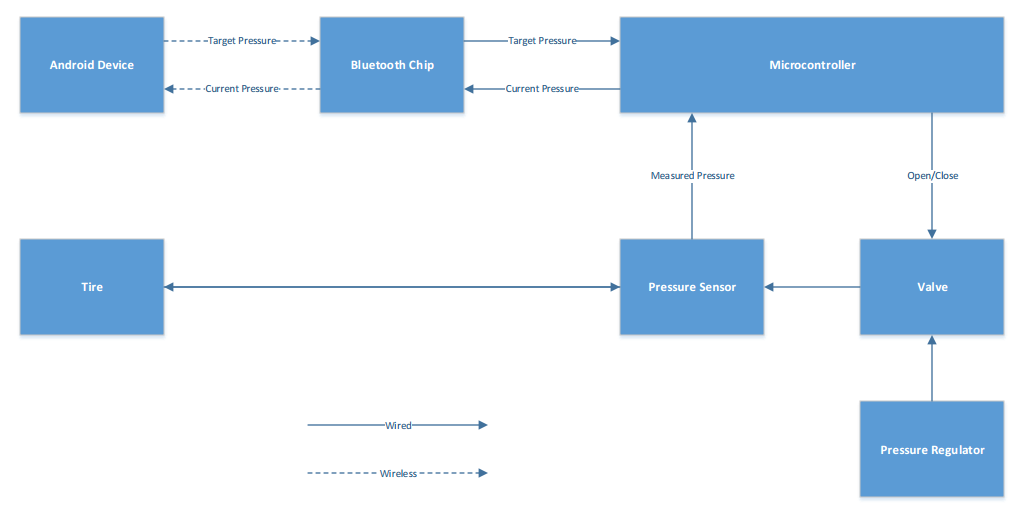
**3. Approach**

BPR is designed to dynamically monitor and adjust the air pressure of a bicycle tire based on user input. By changing the status of the tire pressure, the user will be able to more effectively manage different terrain.

**3.1. System Overview**

BPR consists of several subsystems in order to actively monitor tire-air pressure.

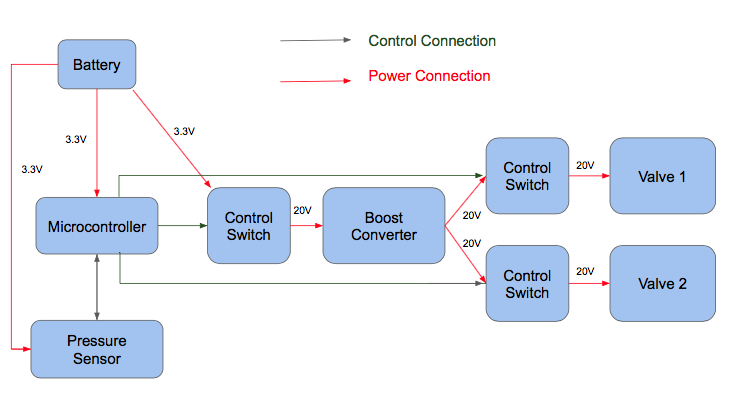
Below, Figure 3.1.1 shows an overview of the various subsystems that comprise a BPR unit. The device is intelligently controlled via an on-board microcontroller that interacts with the other components in order to further monitor pressure.



*Figure 3.1.1 Bike Pressure Regulator System Overview*

**3.2. Hardware**

The BPR contains a series of parts powered by a battery. A simple flowchart is shown in figure 3.1.1 to demonstrate the parts and their order in the design. The parts were chosen based on functionality and preferred results when compared with similar items. There are circuits created to power all of the functioning parts, and boost converters are used to supply the correct voltage where needed. The tables in this section will show the part comparisons in more detail.



*Figure 3.2.1 Bike Pressure Regulator Battery/ Control Overview*

**3.2.1. Microcontroller**

*Table 3.2.1 Microcontroller Comparison*

|  |  |  |
| --- | --- | --- |
| **Microcontroller** | **Arduino Uno Rev3 [17]** | **MSP430G2553 [18] ✔** |
| **Low Power Mode** | N/A | 0.5µA ✔ |
| **Supply Voltage** | 1.8V - 5.5V ✔ | 1.8V - 3.6V |
| **Flash Memory** | 32kB ✔ | 16kB |
| **Cost** | $3.81 | $2.63 ✔ |

Table 3.2.1 demonstrates the capabilities of two frequently used microcontrollers, the MSP430 and the Arduino Uno Rev3. One major design constraint is that the device needs to provide power for a long enough duration to allow the user to complete their ride. Due to this constraint, an option such as the low power mode by presented by the MSP430 is very attractive, especially considering the low miniscule 0.5µA current draw in low power mode. Both microcontrollers operate at a low voltage that can be provided by the power system in the design. The Arduino Uno has a larger flash memory size than the MSP430, and this larger flash memory could be utilized to write more features for the device. The MSP430 is cheaper by about 45%, which could help make the device more affordable. The MSP430 will be used in the design primarily because of the ability to put the device in low power mode, thus providing the device with a longer battery lifespan.

**3.2.2. Bluetooth**

*Table 3.2.2. Bluetooth Comparison*

|  |  |  |
| --- | --- | --- |
| **Bluetooth System** | **CC2640 []✔** | **STLBC01 []** |
| **Operating Voltage** | 1.8 to 3.8 V ✔ | 1.9 to 3.6 V |
| **Low Energy Interface** | I2C, SPI, UART capable ✔ | SPI exclusive |
| **Current Draw (Transmit)** | 6.1 mA ✔ | 12.9 mA |
| **Cost** | $5.85 | $4.29 ✔ |

The Bluetooth system must be able to quickly communicate in low-energy mode. Table 3.2.2 demonstrates the capabilities of the most essential categories taken during consideration for the BPR Bluetooth module. Both Bluetooth choices require reasonable operating voltages for our design while also providing fast interface options. While the desire for fast communication between the microcontroller and the Bluetooth module favors SPI communication over I2C [2], the flexibility associated with the CC2640 allows for more design options. I2C allows the master to receive confirmation from slave devices, offers less interference between signal lines, and also allows for communication off of the PCB [3]. In order to provide more reliable feedback from a communication standpoint and compensate for geometric PCB design constraints, I2C is a necessary feature for the BPR design.

Due to the expected lifetime of the BPR product, resourceful current consumption is an additional consideration for Bluetooth systems. As demonstrated in Table 3.2.2, the CC2640 Bluetooth module requires almost half as much current while the Bluetooth radio is transmitting data. Since power supply will be a major constraint for the BPR design, the CC2640 is used for its resourcefulness.

**3.2.3. Pressure Sensor**

*Table 3.2.3 Pressure Sensor Comparison*

|  |  |  |  |
| --- | --- | --- | --- |
| **Pressure Sensor** | **Honeywell HSCSANN150PA2A3 [HS] ✔** | **NXP MPX5700GP-ND [NX]** | **TE M3021-000005-100PG [TE]** |
| **Max Operating Pressure** | 150 psi ✔ | 101.5psi | 100 psi |
| **Communication Mode** | I2C ✔ | Analog Voltage | Analog Voltage |
| **Current Draw** | 2.8 mA ✔ | 7.0 mA | 3.5 mA ✔ |
| **Accuracy** | 0.25 % ✔ | 2.50 % | 1.00 % |
| **Cost** | $46.17 | $15.21 ✔ | $53.66 |

Table 3.2.3 represents multiple pressure sensors and their associated properties that must be weighed to choose the pressure sensor. In order for the pressure sensor to function in the design, it needs to have a high operating pressure greater than 100 psi, which all listed options do. However, the Honeywell pressure sensor goes above this margin and provides more leeway with the pressure rating, thus assisting the device in achieving the safety constraints for the design. In order to access the data from the sensor, multiple modes of communication were analyzed, the prefered option being the I2C method utilized by the Honeywell sensor. All three options have low current consumption, which is necessary to provide a long, usable lifetime for the BPR. The pressure sensor in the design must be accurate in order for the device’s overall accuracy to be optimized. The Honeywell sensor is by far the most accurate with the TE sensor slightly less accurate and the NXP sensor much less accurate. However, the NXP sensor is much more affordable than both the Honeywell and TE sensors. Despite the high cost of the Honeywell sensor, it is used in the design because of its high operating pressure, I2C data communication mode, and measurement accuracy.

**3.2.4. Pressure Regulator**

*Table 3.2.4 Pressure Regulator Comparison*

|  |  |  |
| --- | --- | --- |
| **Pressure Regulator** | **UP100** **Regulator CO2 Charger Kit** **[AM]** **✔** | **TESCOM****44-2213-241 Pressure Reducing Regulator** **[TE2]** |
| **Inlet Pressure** | ~1000 psi | 3500 psi✔ |
| **Outlet Pressure Range** | 0-150 psi | 0-250 psi ✔ |
| **CO2 Cartridge Direct Connect** | Yes✔ | No |
| **Cost** | $24.55✔ | $199.00 |

Table 3.2.4 displays multiple pressure regulator/reducers. The pressure regulator is used to bring the air pressure down from the approximate 1000 psi in the CO2 cartridge to a value that will easily accommodate the pneumatic valve max pressure listed below. The Tescom pressure reducing regulator has a higher capability of handling a larger inlet pressure as well as maintaining a wider range of outlet pressures, but the total cost of this device would far exceed the design’s cost constraint with the other parts that are needed. The UP100 Regulator is the part chosen for the design of BPR. The UP100 is far more cost effective and is capable of handling exactly what the CO2 cartridge is outputting. One of the UP100 charger kit’s most prominent feature is its ability to connect directly to the CO2 cartridge which will cut down on total cost by eliminating the other adapters and hoses needed to connect the CO2 cartridge to the regulator.

**3.2.5. Pressure Valve**

*Table 3.2.5 Pressure Valve Comparison*

|  |  |  |  |
| --- | --- | --- | --- |
| **Pressure Valve** | **Nitra Solenoid AVP-31C1 []✔** | **Peter Paul**  **Series 40**  **Model 43 []** | **Gem**  **E Series**  **Solenoid Valve []** |
| **Max Inlet Pressure** | 130 psi | 150 psi | 175 psi ✔ |
| **Power Consumption** | 3.0 W (DC) | 5.0 W (DC) | 2.0 W (DC) ✔ |
| **Width**  **Height**  **Depth** | 22.0 mm  75.9 mm  62.4 mm | 39.6 mm  56.8 mm  36.9 mm | 18.54 mm ✔  40.89 mm ✔  18.54 mm ✔ |
| **Cost** | $18.50 ✔ | $50.00 | $208 |

Table 3.2.5 shows three pneumatic valves and the measurable constraints used to determine the best option for the BPR design. The BPR system will require a pneumatic valve to regulate pressure between the CO2 cartridge and the tire. Another valve will also be used to release air from the tire into the ambient air to reduce the tire pressure. The main factors used in comparison of the pressure valves were power usage, max pressure inlet, size, and cost. The Gem solenoid valve clearly is superior with most of these concerns, but due to the high cost, it was not chosen. The Peter Paul valve was another possible choice because of the lower cost than the Gem and the inlet pressure, but because of the high power consumption and larger size it was not chosen. The Nitra Solenoid valve was the best choice because of the extremely low cost. The major trade off with the Nitra Solenoid valve is much lower inlet pressure but because the design constraint require only 70 psi the valve will be able to operate in a safe manner far below this 130 psi value. Size was another trade off but the cost of this product was far less than the Gem valve and was within the design’s size constraint.

**3.2.6. Power Supply**

*Table 3.2.6 Power Supply Comparison*

|  |  |  |
| --- | --- | --- |
| **Battery** | **LiFePO4 []✔** | **Li-Ion []** |
| **Nominal Cell Voltage** | 3.3 V | 3.6 V |
| **Amp-Hours** | 3300 mAh ✔ | 3100 mAh |
| **Durability** | Lasts ~2000 charge/discharge cycles ✔ | Lasts ~1000 charge/discharge cycles |
| **Safety** | No explosion or leakage in shock or crush test ✔ | Possible explosion or leakage in both shock and crush tests |
| **Specific Energy** | 90-110 Wh/kg | 100-250 Wh/kg ✔ |
| **Cost** | $7.50 ✔ | $13.95 |

Table 3.2.6 shows two batteries suitable for recharging and use in the BPR. The battery has to be able to supply enough power to operate the circuitry and devices for at least one hour. The Lithium-Iron (LiFePO4) batteries are much safer, more environmentally friendly, and are more efficient when compared to Lithium Ion (Li-Ion) batteries [BATT]. LiFePO4 batteries provide better voltage to power the BPR as well as lasting longer. Also because of the nature of the project, the batteries will have to endure some shock during a ride, so it is important that safety not be an issue. Li-Ion batteries provide more watt-hours per kilogram, yet they are generally much bigger for the same weight. The LiFePO4 batteries are also cheaper to purchase, which help assist in meeting the design’s cost constraint. To measure the amp-hours supplied by the battery, the following calculations were applied:

*(Eq. 1)*

Where is the total current draw, *T* is the total number of hours, and *C* is the capacity of the battery in amp-hours.

*(Eq. 2)*

Where these *I* values are equal to the amount of current each component is drawing.

*(Eq. 3)*

Where are values coming out of the boost converter divided by the efficiency of the boost converter, which is 75%. Estimates show that the valves will be running a maximum of 15 minutes, so the amp-hour value can be divided by 4.

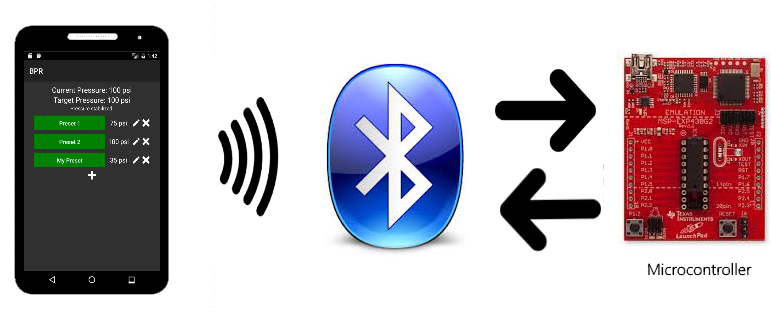
Given the value found in the valve currents, can be calculated.

Rearranging (*Eq. 1)*, the number of hours is calculated. The value of amp-hours for the battery was pulled from *table 3.2.6.*

Based off these calculations, the device will have enough battery to be utilized for 8.46 hours.

**3.3 Software**

The BPR will have three major software-heavy components: the microcontroller, the BlueTooth module and the Android phone application. The microcontroller will serve as the main mechanism for handling data and controlling the pressure components. The BlueTooth module will serve as a means to transmit/receive commands from the smartphone and the microcontroller. The Android phone application will act as an user-interface for system communication. Figure 3.3.1 illustrates each of these components and how they will be interconnected.

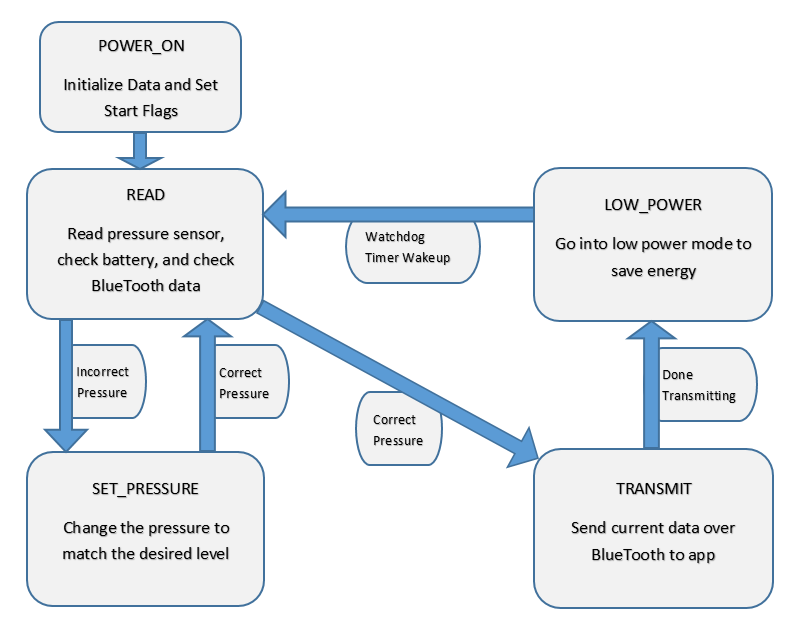


*Figure 3.3.1 - System Interaction*

**3.3.1. Implementation Details**

Data communication will be handled through software programmed interfaces. The mobile application will allow user communication (through a GUI interface) to the microcontroller via Bluetooth 4.0. The software datapath will be a full-duplex system, allowing for user and system data exchanges. The BPR software uses C-syntax languages. The microcontroller and the Bluetooth module are written in C, while the mobile application is written in C#. The C programming language is implemented due to the gcc compiler associated with the integrated development environment (IDE) for the Texas Instrument microcontroller and BlueTooth module. The C programming language provides runtime advantages compared to higher programming languages, as well as an easier method to access registers. The Android application is written in Visual C# due to familiarity. The approach for using the previously mentioned languages provides an efficient and powerful way to ensure proper communication between devices.

**3.3.2. Command Layer**



*Figure 3.3.2 - States of BPR software*

The command layer of BPR is responsible for obtaining inputs, controlling the tire pressure, and transmitting data to the user. The software will need to handle inputs from user commands via BlueTooth, measurements of the battery status, and pressure sensor values via I2C. Figure 3.3.2 demonstrates the state machine that will be utilized in the software of BPR. Other than when the user needs to change the pressure, the device will spend most of its time in the LOW\_POWER mode in order to conserve energy. To escape the LOW\_POWER mode, a watchdog timer will be enabled to wake the device up and go back to the READ state in order to get new data.

**3.3.3 Data Handling**

In order for the BPR to interact with the phone and to properly handle internal data transfer, standards have to be created to ensure accurate, reliable transmission of data. To obtain a usable pressure in psi from the pressure sensor that can be used by the microcontroller and sent to the user on the phone application, the software must utilize an equation to calculate the actual pressure value.

(*Eq. 4)*

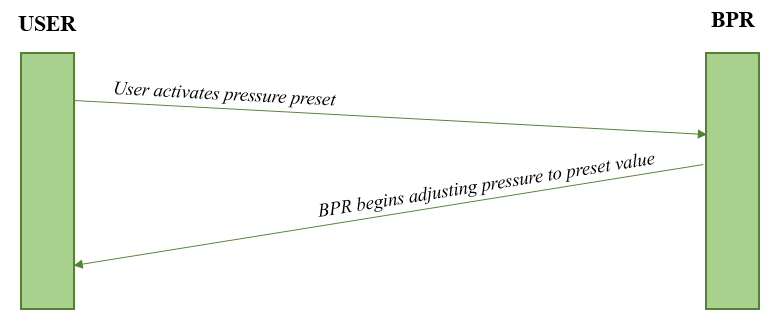
*Eq. 4* shows how the pressure will be calculated from the reading of the pressure sensor. In *Eq. 4*, *Pressure* is the pressure level in psi that the is calculated from the equation, *Output* is the hex value that the pressure sensor sends over I2C to the microcontroller, and *Outputmax*, *Outputmin*, *Pressuremax*, and *Pressuremin*are constant-hex-values that are determined by the Honeywell pressure sensor.

**3.3.4. Mobile Application**

The Android phone application will be used to send data to the microcontroller and relay information back to the user. The application will continually receive data from the microcontroller in a 4-byte format that will contain the current measured tire pressure as well as any pertinent error flags. Data received by the microcontroller will be displayed to the user on the main menu of the application. The application will also transmit a 4-byte value to the microcontroller containing a user-selected target pressure in addition to any outside commands (such as force-idle or power-down). The app will allow the user to add up to ten custom target pressures.

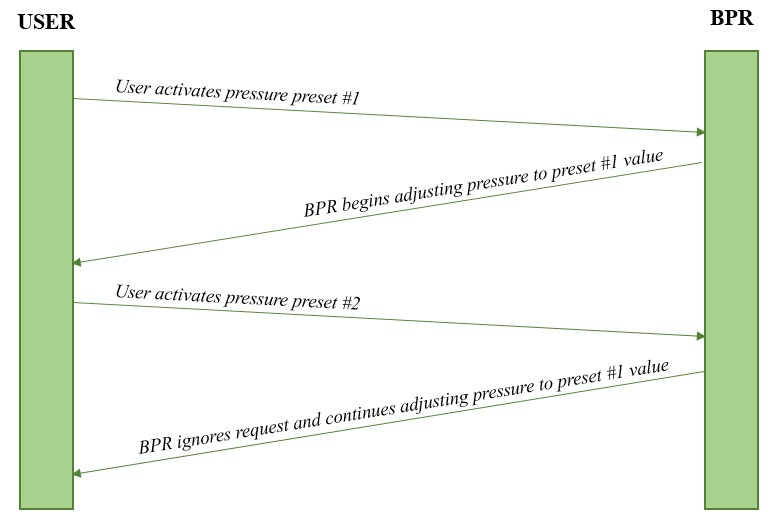
**3.3.5. Use Cases**

There are several different interactions that may occur from using the BPR. *Figure 3.3.1* illustrates an ideal case with a “Sunny Day” user interaction.



*Figure 3.3.5a – Sunny Day User Interaction*

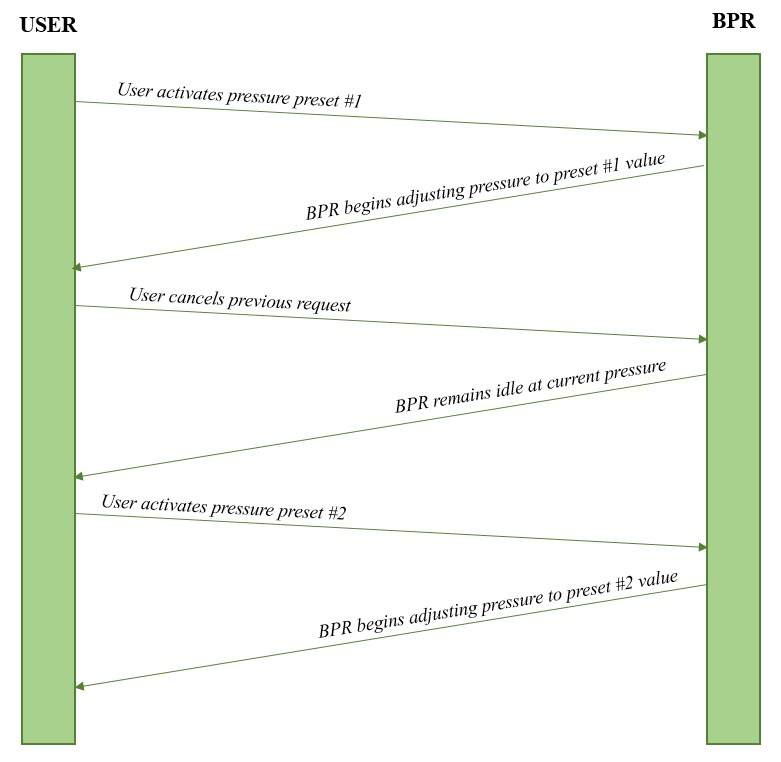
The BPR should be able to adjust the pressure to a given target pressure, supplied by the user. This interaction should be very simple and require very little effort by the user. The “Rainy Day” interaction, shown in *Figure 3.3.2*, depicts how the BPR will handle a more conflicting series of instructions from the user.



*Figure 3.3.5b – Rainy Day User Interaction #1*

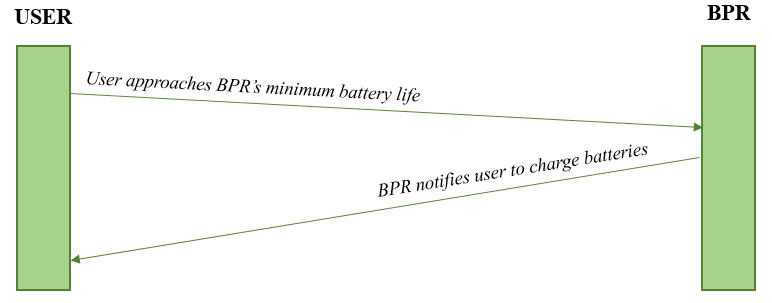
In this case, the user attempts to activate a new pressure preset before the previous request has been satisfied. BPR will solve this conflict by ignoring any new preset requests until the current preset has been met.

There is also an alternative solution to this issue. The BPR application will have a ‘Cancel’ button on the main menu which will allow the user to cancel a previous request before it has been met. This will allow the user to select a different preset without having to wait. Figure *Figure 3.3.3* details how this interaction will proceed.



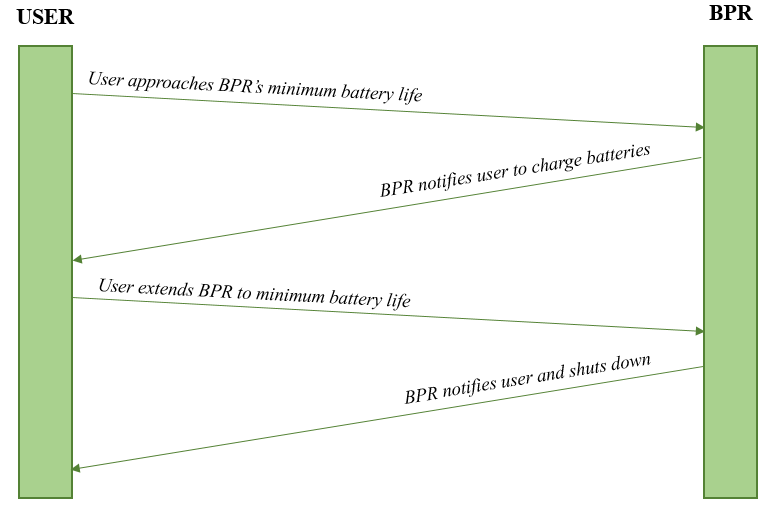
*Figure 3.3.5c – Rainy Day User Interaction #2*

Battery life is also an area of concern in the BPR’s design. The solution for this is explained in the following Sunny and Rainy day cases.



*Figure 3.3.5c – Sunny Day Interaction #2*

Ideally there should be no issues, assuming the user changes the batteries once they are nearing depletion. The issue arises when the user ignores BPR’s notification and continues to use BPR past the recommended battery percentage. *Figure 3.3.5d* shows how BPR will handle this case.



*Figure 3.3.5c – Rainy Day User Interaction #3*

As detailed in *Figure 3.3.5d*, BPR will manually shut itself down once the battery reaches its minimum percentage. Since the valves must be opened to either pressurize or release the tire, the bike’s tire pressure should remain relatively idle once in shutdown mode and safe for the user to continue to ride.

**References:**

[17] <http://www.atmel.com/images/atmel-8271-8-bit-avr-microcontroller-atmega48a-48pa-88a-88pa-168a-168pa-328-328p_datasheet_summary.pdf>

[18] <http://www.ti.com/lit/ds/symlink/msp430g2553.pdf>

[1] <http://www.ti.com/lit/ds/symlink/cc2640.pdf>

<http://www.st.com/content/ccc/resource/technical/document/datasheet/1f/59/bb/bc/72/89/46/65/DM00051480.pdf/files/DM00051480.pdf/jcr:content/translations/en.DM00051480.pdf>

<https://upverter.com/upn/b0a4ae1a959041b9/> (all together)

[2] <http://www.byteparadigm.com/applications/introduction-to-i2c-and-spi-protocols/>

[3] <https://www.lifewire.com/selecting-between-i2c-and-spi-819003>

[HS] <http://www.digikey.com/product-detail/en/honeywell-sensing-and-productivity-solutions/HSCSANN150PA2A5/480-5523-ND/2270189>

[NX ] <http://www.nxp.com/assets/documents/data/en/data-sheets/MPX5700.pdf>

[TE] <http://www.te.com/commerce/DocumentDelivery/DDEController?Action=srchrtrv&DocNm=MSP300&DocType=Data+Sheet&DocLang=English>

[TE2]

<http://www.amronintl.com/tescom-pressure-reducing-regulator-1-4-inch-44-2213-241.html>

[AM]

<https://www.amazon.com/UP100-Regulator-Charger-Disconnect-Kegerator/dp/B01EHVVU24/ref=as_li_ss_tl?srs=12586382011&ie=UTF8&qid=1484757870&sr=8-22&linkCode=ll1&tag=hombrefin-20&linkId=cdb0ade06c9db2db0ff83e63c9ff2f09>

[BATT]

<http://www.brighthubengineering.com/power-generation-distribution/123906-comparison-of-lithium-ion-to-lithium-iron-battery/>