ELECOMP CAPSTONE
VIRTUAL SUMMIT
FRIDAY, MAY 7, 2021

THE UNIVERSITY OF RHODE ISLAND

THINK BIG WE DO℠

SPBBTR

SmartPlug Bluetooth Button Transmitter and Receiver

Team Members: Andrew Crawford (ELE), David Hwang (CPE), Kevin Kwan (ELE)

Taco Comfort Solutions™ A Taco Group Company

Technical Directors: Phil Manning, Nick Costello, Mike Smith

Project Motivation

In certain parts of the world where there are water shortages like California, USA, there are mandatory requirements for on demand water heating. The average residence can waste up to 12,000 gallons of water per year waiting for running water to heat up to the desired hot temperature. Taco Comfort Solutions currently has a product called the SmartPlug® that can upgrade any recirculation pump to “Smart” operation. The project is to upgrade the current SmartPlug® so that it can be wirelessly activated via bluetooth with the push of a button when hot water is required. Manual activation of hot water recirculation increases the efficiency of water conservation even further by not having to rely on water usage patterns to assume when users might need hot water.

Key Accomplishments

Decide on a bluetooth microprocessor for the button and SmartPlug: The best option for a bluetooth processor is the nRF52840 by Nordic Semiconductor. It has the specifications that this project requires when it comes to range and frequency. It is bluetooth low energy which is perfect for the desired usage. This chip is used by a variety of vendors that implement it in their own products for ease of prototyping.

Acquired appropriate bluetooth modules: Adafruit Industries developed their own module called the nRF52840 feather express which implements the Nordic nRF52840 chip. Thanks to the MDBT42Q-U512KV2 chip by Raytac Corporation which contains the Nordic chip, Adafruit has been able to design a board that has a micro-USB for programming and other several components that make the board easy to test code. Adafruit also provides example codes relating to the bluetooth functionality of the nRF52840. The team has decided to choose this module for prototyping the code.

BLE Implementation: We made use of the Arduino IDE for the majority of the BLE implementation. Referencing the Arduino and the Adafruit library examples in order to pair the two modules. Usage of the callbacks were very helpful in coding both the Central and Peripheral modules; as the connection and disconnection tasks were performed in this way.

Central module: This module was coding to perform the task that the Button would perform. This module works to detect the advertised signal from the Peripheral module and pair to it. Acting only when a debounced button push is received from the user. After the module would begin to search for a specified signal to pair to. Once the button has been paired with the Peripheral module, acting as the SmartPlug, it would instruct the Peripheral module to begin their processes. See **Fig. 1**.

Peripheral module: The partner module of the Central module, acting as the Smartplug. This module is constantly advertising it’s UUID in order for the two modules to pair with one another. Setting up characteristics for the button, this allows for the use of a characteristic write to be performed. Making it possible for the paired modules to communicate with one another. See **Fig. 2**.

Testing the modules: Testing the modules together, took any attempts with much trial and error. Pairing the two modules together is a quite quick step in the testing. The team was able to make a simple connection between the two modules. Issues arose when we were looking to send a specified debounce signal through our designated characteristics. We attempted to perform a characteristic write from the Central to the Peripheral. We were able to resolve this issue when we discovered that permissions were required for the write to be performed. See **Fig. 3**.

Designed the PCB for the button: Since the modules worked the way the project intended, it is clear that the nRF52840 was the right processor for the job. When it comes to designing a standalone PCB for the button, the MDBT42Q-U512KV2 will be used and the other components will be implemented. The other components required for the standalone PCB to work include capacitors for decoupling, reset switch, LEDs for troubleshooting, a micro-USB for bootloading and programming, and a voltage relay for power and filtering. All components are surface mounted. See **Fig. 4**.

Anticipated Best Outcome

A new version of the current SmartPlug® in its prototype stage of development that includes a redesigned PCB within the current enclosure and updated firmware to interact with a remote button through bluetooth to control the hot water recirculation pump. The hardware and firmware for the button separate from the SmartPlug® is developed. The whole system will include other functions such as a learning algorithm and programmable timer for automatic recirculation on top of manual activation.

Project Outcome

The outcome of this project includes a prototype of the SmartPlug equipped with an Adafruit nRF52840 bluetooth module and a prototype of the SmartButton with a custom PCB. The team has met the anticipated best outcome that was set.

Figures

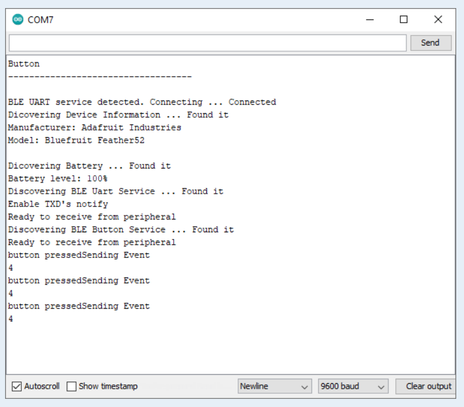

Figure 1: Central Module Serial Monitor: The serial monitor in arduino gives the user information about what the module is doing in the IDE. The top portion of the code indicates the module has connected to the peripheral module. The last 6 lines indicate the button on this module has been pressed 3 times.

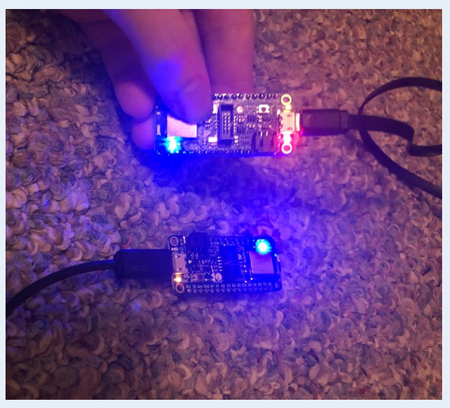

Figure 3: Central Module (TOP) and Peripheral Module (Bottom): When the blue LED is blinking on both modules, this indicates that neither of them are connected to the other. When it is a steady blue light, this means that they are connected. The red LED was used in troubleshooting for the button press in the central module. When the button is pressed the red LED will remain on for about 5 seconds to indicate the debounce function which prevents the user from pressing the button too many times in an instance.

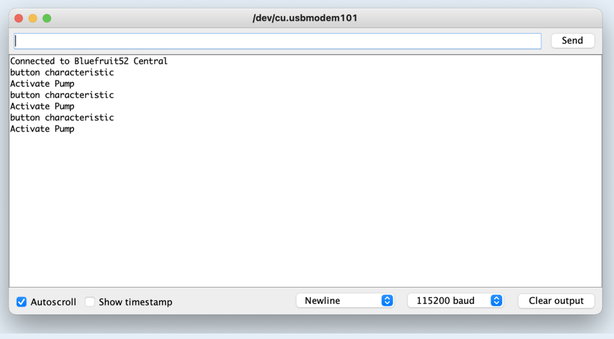

Figure 2: Peripheral Module Serial Monitor: The serial monitor for the peripheral indicates that it has received a signal from the central module that the button has been pressed. This will then activate the pump function which sends a signal to the pin that performs this function. This module was in sync with the module in Fig 1 which is why it has received 3 button presses.

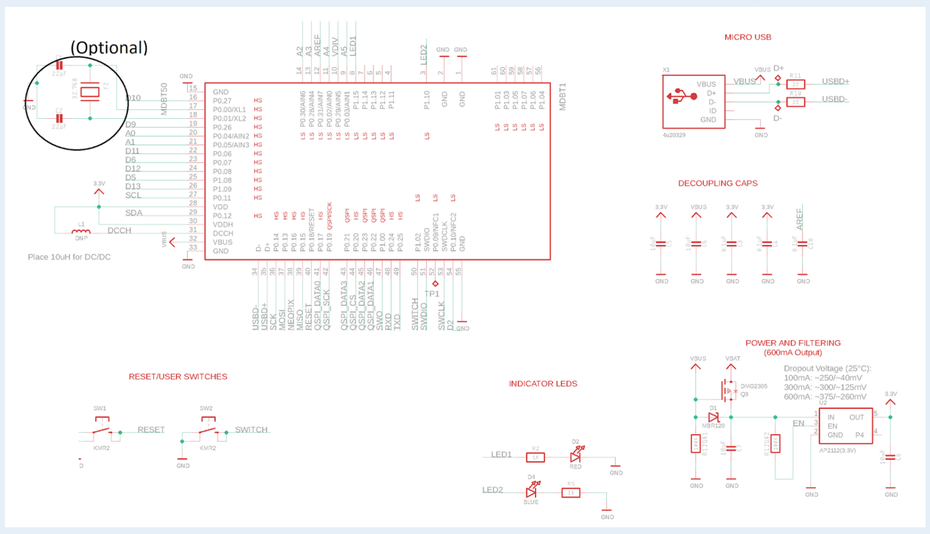

Figure 1: Central Module Serial Monitor: The serial monitor in arduino gives the user information about what the module is doing in the IDE. The top portion of the code indicates the module has connected to the peripheral module. The last 6 lines indicate the button on this module has been pressed 3 times.

ELECOMP Capstone Program Director: Professor Harish R D Sunak

email: sunak@ele.uri.edu telephone: 401-874-5859 ELECOMP Website: uri.edu/elecomp-capstone