LogiSteps nRF52 App Design

Overview

This document explains the basic design of the nRF52 app, and the various ‘modules’ that compartmentalize the work done to achieve the apps goal of staying low power, reading step data, and transmitting it to a mobile device over Bluetooth Low Energy.

The sequence diagram for the nRF52 embedded app is shown on the right. It shows the high-level logic flow of the application. It begins with Main querying the Energy module to ensure there is enough energy stored to begin. Once there is enough power Main initializes the modules, and the BLE module begins advertising. Once a device has connected Main begins the Trigger module to wait for an event to be registered. From there three things can happen. In normal function an event occurs, the Energy module is queried to ascertain the current power levels of the system. Based on the power levels the saadc samples at a specified frequency until the event ends. Then the BLE transmits the collected data, and the system goes back into a waiting state. The second option is that the connected device disconnects, in which case the system reverts back to the beginning of operation. The third option is that the system is running out of power, in which case main attempts to gracefully shut everything down.

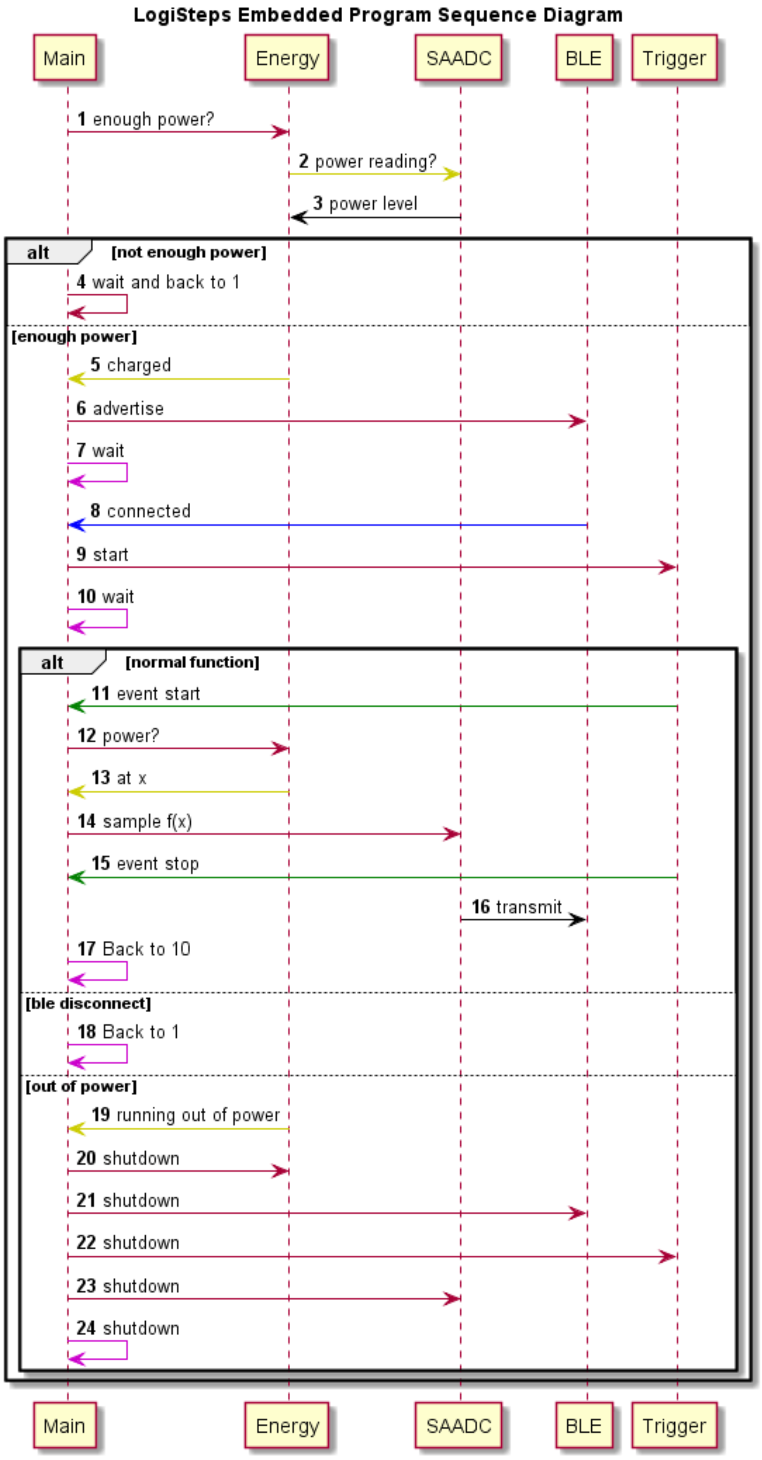

Figure 1 Sequence Diagram of the nRF52 app

Modules

Main

The purpose of Main in our nRF52 app is to handle the initialization of all other modules and ensure proper logic flow of the program. Main also begins low power management and puts the device into sleep mode when it is not working so as to save power.

Energy

The purpose of the Energy module is to monitor the system's energy supply. With that in mind it has two jobs. It answers queries about the current supply when asked by main, and alerts main when power drops too low.

SAADC

The saadc module handles all of the systems saadc operations, which entail the measurement of all of the systems sensors, and the system's specific energy level. It provides the data so that it can be sent to BLE module to allow it to transmit the data to the device the system is connected to.

BLE

The BLE module handles all BLE operations required for the nRF52 app. It advertises the system so the user can connect, it receives the necessary data from the user's phone to continue operation, and it transmits the sensor data to the user's device for use. Specifically, the BLE handles one Bluetooth service which holds two Bluetooth characteristics. For a device to work with the app, on connection it must transmit the current time to the time characteristic. Then whenever there is data to send the data characteristic notifies the device with the sensor data, and the time characteristic notifies the device with the systems time pairing for the data.

Trigger

The trigger module simply handles the event that the system uses to begin gathering data from the sensors. After it is initialized by main it will wait for a signal to register an interrupt which will alert main to begin reading sensor data.

Interfaces

Sensors

In order for the nRF52 app to interface with the sensor hardware, the input power provided by the energy harvesting circuit is expected to be in a voltage range between 1.8V and 3.6V as the microcontroller specifications require. The sensor inputs, and other inputs from the energy harvesting circuit, are expected to be in a voltage range between zero and the input power as the saadc specification requires.

With those voltage ranges decided the nRF52 app will work with the sensors by watching a p-good signal from the energy harvesting circuit at all times to ensure the power supply is sufficient. It will also read the voltage from one of the energy harvesting capacitors to determine the actual energy levels for use in the application. The last way it will interface is with the sensors, which depending on the energy levels, it will sample using the saadc read the sensor levels.

Mobile Device

For the nRF52 app to interface with a user's mobile device there are a few things of importance. First is that the nRF52 uses one custom BLE service. This service has two characteristics, one of which is a time characteristic and one is a sensor data characteristic.

When the device connects it must send the current time over the time characteristic so the app knows the current time so it can pair it with sensor data. From that point on whenever there is sensor data to transmit, the data characteristic will be updated with a 16-bit signed integer detailing the sensor data. At the same time the time characteristic will update with the time paired with that sensor data.

Bill of materials

There were many different pieces of hardware equipment that were required to program the microcontroller for our project. Luckily, they were all easily accessible and only two different vendors were needed to buy what we needed.

The two main components needed for our project were the nRF52832 Sparkfun breakout board and the nRF52832 hardware development kit. The hardware development kit is available through various electronics vendors, but we purchased two of them from Digikey for $39 each with $7 shipping for a total of around $90 with tax included. The Sparkfun breakout board was purchased through the Sparkfun website and we bought 3 of them for $20 each plus $7 shipping. In addition to the breakout board itself, an FTDI basic module was needed to allow for USB connection to program and power the breakout board. We purchased three of these for $15 each, making the total for the breakout boards around $112 and an overall total of around $202 for the microcontroller hardware.

In addition to the two development boards, various other electrical components were needed. A micro USB cord was needed to power the hardware development kit, and an additional micro USB cord (or a mini USB cord depending on which FTDI module was used) was needed to power and program the breakout board. The breakout board also required pins to be soldered onto the chip to allow for connections to be made with the GPIO pins. The breakout board would then need to be placed on a breadboard to allow for connections to be made. Finally, various wires are needed to make connections, and specifically, M/F wires are needed to connect the breakout board to the hardware development kit.

Luckily, our project team already had all of these electrical components from earlier classes or from home and they didn't need to be purchased. However, if a person didn't have any of these components, they could all be purchased on the Sparkfun website along with the breakout board and FTDI basic module. To buy all of the various components needed to program the microcontroller for this project, a person would have to pay, at a maximum, around $120.