BlepH – A reliable, power efficient and low cost Bluetooth pH sensor.

EE 149/249 Final Project Report

Jikang Chen, Siyuan “Jack” He, Yan Zhao
UC Berkeley

12/19/2014

1. Abstract

A reliable, robust, power efficient and low cost remote pH sensor was developed using Bluetooth Low Energy technology. A user is able to interface with the sensor using a customized iPhone App.
Video demo: http://youtu.be/7j5e0DGoxtg

2. Introduction

Existing pH sensors are mostly bulky and/or expensive. With the development of Internet of Things devices and increasing capability of data mining, there is an increasing demand for low cost wireless pH sensing device in industries like agriculture, food and health care. Moreover, existing pH sensors are usually designed for short time use with human intervention, but a bulk of these demand requires the sensor to run for long time with low or zero maintenance. This project aims to fill this demand by making a relatively low cost and power efficient wireless pH sensor that can be used towards such Internet of Things (IoT) applications.

3. Achievements

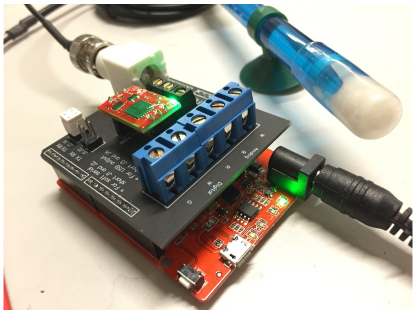


Figure 1: BLEpH Prototype

Over the course of two months, the team managed to build a prototype of the wireless pH sensor (Figure 1). It was implemented using Bluetooth Low Energy technology (with Nordic nRF51822 SoC on a RedBear nRF51822 board using mbed API) and a pH probe made of ion-selective glass electrode (American Marine pH Probe). Since the pH value from the probe cannot be measured directly using an ADC, an Atlas Scientific PH-EZO interface board was used to interface with the pH probe and transmit digital reading back through UART. The system design is shown in Figure 2. The team also built an iPhone app to interact with this pH sensor through Bluetooth.

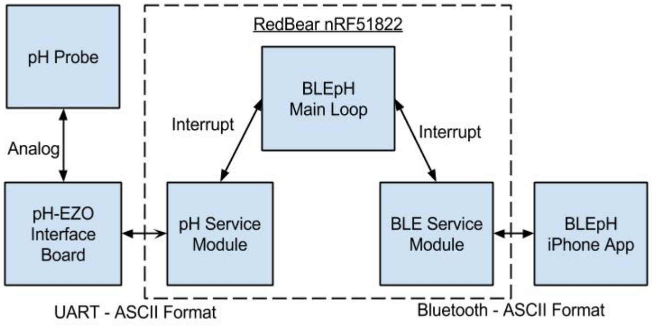


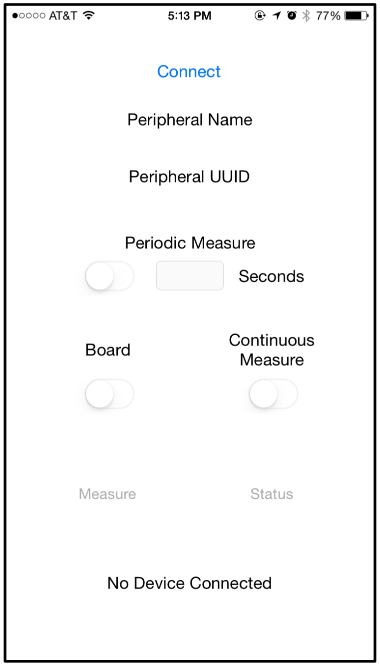
Figure 2: BLEpH System Functional Diagram

4. Software Functions

The final prototype involves several functions that can be accessed using an iPhone app:

1. Connect and disconnect from pH sensor
2. Turn on/off the pH interface board
3. Query the system status
4. Make a single pH measurement and display it on the phone.
5. Start/stop a continuous pH measurement and continuously display it on the phone
6. Start/stop a periodical pH measurement with a user defined period

User can access these functions using a graphical user interface as shown in Figure 3.



No Device Connected

Figure 3: BLEpH iPhone App UI

5. Modeling

As shown in Figure 2, the embedded system is identified to contain three main components: BLE

Service Module (ble_lib) - Yan), pH Service Module (ph_lib - Siyuan), and BLEpH Main Loop (Jikang).

The BLE Service module is a library built with the mbed BLE library. It behaves as a BLE slave and manages communications with external BLE master device. It runs as a state machine as shown in Figure 4.

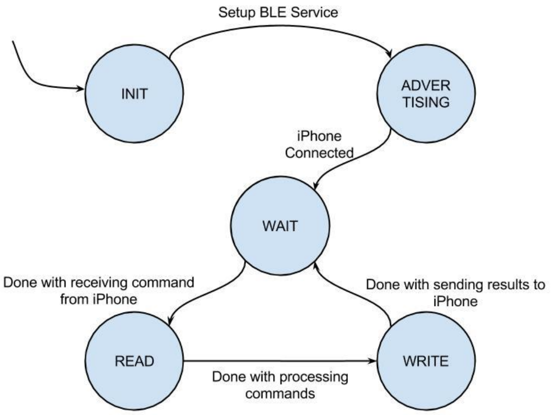


Figure 4: BLE Service Module State Machine Model

The PH Sensing module is a library that interfaces with the PH-EZO measurement circuit. It manages UART connection with PH-EZO and provide methods for administrative and measurement functions. Two modes of measurement are available:

- **Single**: A synchronous mode - the measurement routine blocks the caller until a value is returned.
- **Continuous**: An asynchronous mode - the measurement routine returns immediately and UART interrupt is used to call a callback function when a new pH value is ready.

The operation of this module can be modeled using the state machine as shown in Figure 5.

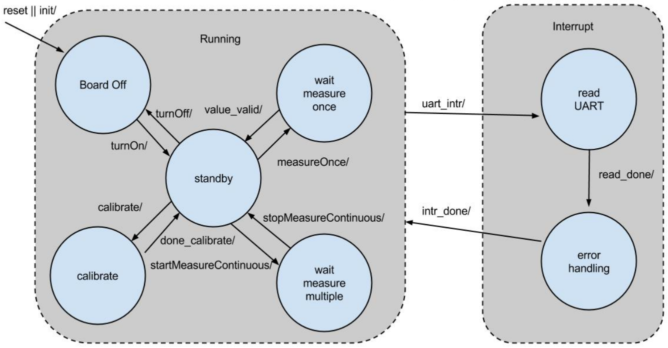


Figure 5: PH Service Module State Machine Model

The main application loop uses both the PH Service module and the BLE Service module to interface with the iPhone app.

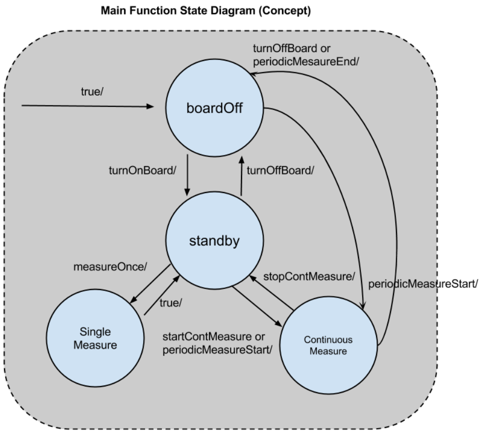


Figure 6: Main Loop State Machine Model

6. Design Considerations

Concurrency

Since the system requires several modules to run simultaneously and involves shared variables, concurrency plays a big role in design considerations. Since there is no operating system running on the embedded platform, concurrency is mostly realized through hardware interrupts. While system is idle, it is in a deep sleep mode and will only be interrupted by a BLE connection request or packet arrival. During continuous measurement, the system need to listen to BLE packet as the client may request to stop continuous measurement, query the status, etc. In this case, the main loop sleeps awaiting BLE interrupt, while an UART interrupt will call a callback function that sends pH reading through BLE channel.

Critical sections are created through disabling single or all interrupts, depending on whether the interrupt handler will access the protected data structure.

Reliable Real Time Behavior

Since user can support a self-defined period for periodical measurements, it is necessary for our system to ensure this period is accurately implemented. The team achieved this with two hardware timers. The *periodicMeasureStart* timer interrupt calls *periodiclyMeasuring* to start a continuous measurement at the user defined period. The *periodicMesaureEnd* timer interrupt calls periodiclyMeasuringStop to stop that continuous measurement and send the mean pH reading to BLE master using the last 5 pH readings.

The reason for using a continuous measurement during periodical measurement is to achieve better accuracy. The two timers have a constant phase difference larger than the maximum time required for taking 5 pH readings. This maximum time was determined in the **Analysis** section.

Power Saving

By letting the main program loop sleep for most of the time and using interrupts for concurrent event, the team makes the ARM Cortex-M0 core on the nRF51822 use as little as 8 μW average.

By using the BLE technology, the team make the Bluetooth connection consumes very little energy as well.

The PH-EZO circuit consumes about 75mW at standby and even 5mW at sleep. To completely turn it off when not in use, a BJT transistor was added to solve this awkward situation as shown in Figure 7. A GPIO pin on the nRF51822 was then used to control this BJT transistor.

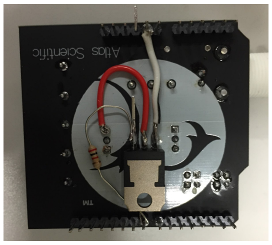


Figure 7: BJT Power Switch

7. Analysis

To validate the power consumption characteristics of the system, a NI PXI-4072 DMM was used to sample current consumptions of the system from a 9VDC power supply at 1K samples/s with LABVIEW. The team obtained a power profile for the system when it is running at different modes: pH measurement mode, BLE communication mode, and sleep mode.

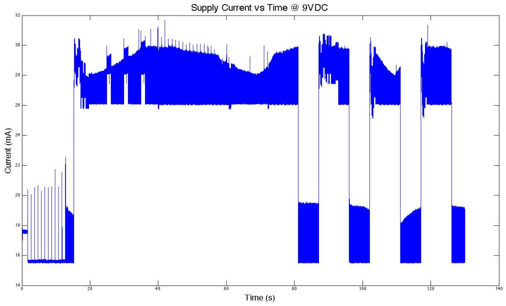


Figure 8: Overall Power Profile

As shown in Figure 8, the power profile not only shows the energy consumption of the system at different modes, which is a significant design goal for our project, it also reveals the system's real-time behaviors to a millisecond precision because of the 1K/s sampling frequency.

BLE Module Behavior & Power Consumption

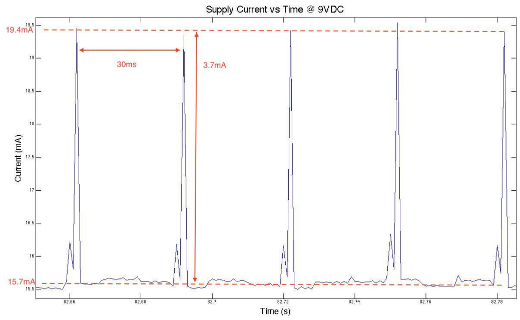


Figure 9: BLE Power Profile

As shown in Figure 9, the BLE module waits up and communicate with the iPhone every 30ms with a roughly 6% duty cycle. BLE consumes roughly 3.7mA during wait-up and whenever it is idle, it goes to sleep mode to conserve power. After integrating over an hour, the charge that BLE consumes is 0.247mAh @ 9VDC, an average 2.22mW power consumption.

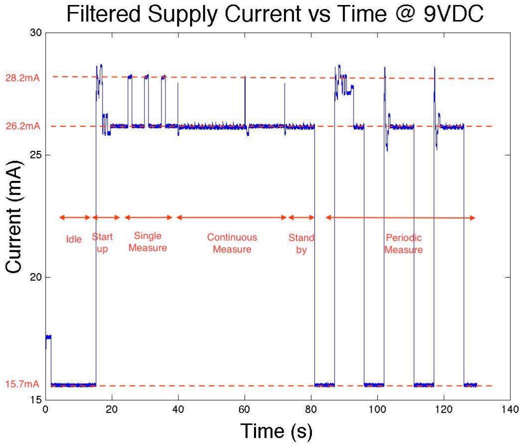


Figure 10: pH Sensing Circuit Power Profile

PH Module Behavior & Power Consumption

After applying a 15ms median filter on Figure 8, which eliminates the BLE current peaks, the power consumption for the PH-EZO board is shown in Figure 10. As labeled in the graph, the filtered current graph precisely reveals the real-time behavior of the PH-EZO board. For example, after turning on the pH sensing circuit at t = 15s, the power consumption raises and saturates at 26.2mA. Following the startup is a sequence of single measurements that are shown in the graph by the three consecutive pulses. A continuous measurement is then followed for another 20 seconds until the pH circuit is shut off.

As mentioned earlier, single measurement was implemented with busy-waiting, whereas continuous measurement was implemented with hardware interrupts. While busy-waiting reduces the latency of single-measurement, it potentially uses more energy than continuous measurement does. As shown in Figure 10, the current pulses during single measurement and the constant current during continuous measurement prove that hardware interrupts successfully conserves a large amount of energy.

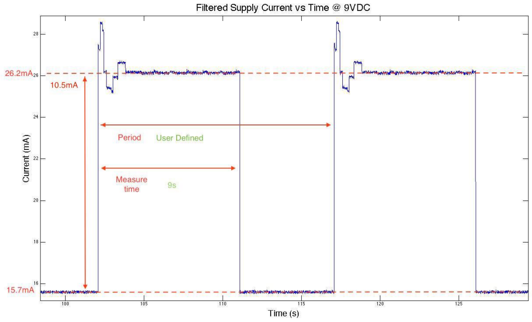


Figure 11: pH Sensing Circuit Power Profile in periodical measurement

Figure 11 shows a more detailed view for periodical measurement. The pH circuit will be periodically turned on with a user-defined period by one timer and shut down by another timer, as described in the previous section. The time it takes to measure within each period is predefined to 9s, a minimum duration that allows the pH sensing circuit to generate enough reliable data for an average calculation.

Other Findings

There is a constant 15.7mA current offset that is superimposed on the power profile for all components. This offset is believed to be consumed by the on-board MK20 companion chip and other accessories that are needed for USB communication between the RedBear nRF51822 board and computer. Since this offset will not exist in any final product that is used in the field, it can be removed in the battery life estimation for the prototype system. As shown in Figure 12, the estimated battery life for the system can easily achieve two years with a 7 hours period, which is sufficient for our purposes.

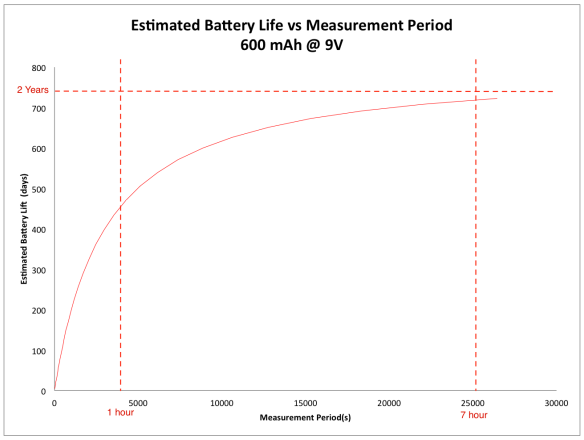


Figure 12: Estimated Battery life

8. Material Cost

Item	Cost
RedBear Lab nRF51822 Board	$39.90
RedBear Lab nRF51822 Board	$39.90
American Marine pH Probe	$39.99
PH-EZO Circuit	$34.00
Arduino Rapid Development Shield	$24.00
Battery & Connector	$4.00
Total	**$141.89**

9. Future Plan

The team has the following items on agenda to improve this prototype:

- Design and validate customized an integrated circuit such that any unnecessary components are excluded
- Improve software features such as a better iPhone app and BLE standby procedure
- Build a distributed system for large scale data acquisition
- Enables the design of other BLE systems

10. Conclusion

It is totally possible to make such a reliable, power efficient and low cost Bluetooth pH sensor and it will perform better with more engineering effort.

References

[1]. Project Github (private, ask author for access): https://github.com/hsysuper/bleph
[2]. Nordic nRF51822 Specifications: https://www.nordicsemi.com/eng/Products/Bluetooth-Smart-Bluetooth-low-energy/nRF51822
[3]. RedBear nRF51822 Datasheet: http://redbearlab.com/redbearlab-nrf51822/
[4]. Freescale MK20DX128VFM5 Datasheet: http://cache.freescale.com/files/32bit/doc/data_sheet/K20P32M50SF0.pdf