University of California, Santa Cruz

Independent Study

Course Code

Implementing Cyber Physical Systems on top of Bluetooth Smart

Student Name:
Ryan Rodriguez

Student ID:
1336732

December 17, 2015

Contents

1 Motivation

The research focus for my individual study during the Fall of '15 was the RFduino platform and how it can be used to implement cyber-physical systems. In order to allow students to engage with control problems involving a diverse set of sensors and actuators, we seek to tap into the momentum of open source movements like Arduino to build a sensible framework for experimenting with cyber-physical systems.

The success of the Arduino platform is owed in no small part to its ease of use, stemming from a robust hardware abstraction layer (HAL). While traditional embedded systems design relies on thorough understanding of proprietary architectures and sifting through stacks of manuals for hardware registers, taking advantage of Arduino compatible chipsets is often as simple as importing a library. The implication for researchers is that we can spend less time building boiler-plate hardware and software to develop embedded applications.

This ease of use, of course, has traditionally been at the cost of performance and customizability. A recent trend amongst microcontroller manufacturers is to integrate Arduino support into their own products by including bootloaders for flashing code from the Arduino IDE, and integrating the Arduino HAL into their APIs. This has given the Arduino community a broader range of chips to choose from, with an increasingly diverse set of capabilities. Notably, we've seen the rise of RF system-on-chips (SOCs) which incorporate radio hardware into microsystems.

The RFduino is nothing more than a Nordic Bluetooth Low Energy (BLE) SOC equipped with an Arduino compatible bootloader, and HAL. The device is roughly the size of a quarter, and uses energy quite conservatively making it well-suited for battery powered applications.

2 Background

3 Hardware

The RFduino, founded as a Kickstarter project, is a small, low-power, bluetooth enabled microcontroller. Under the hood, the device feature a Nordic NRF-51822, which uses an Arm Cortex-M0 as its processing core. The device features standard serial protocols like SPI, I2C, and UART, along with GPIO functions and a 10-bit ADC. Most notably, the chip features a fully operational bluetooth low-energy stack capable of wireless communication with bluetooth hosts, and other BLE devices. The chip is clocked at 16MHz, which is conservative, but preferable for low-power. The chip features

3.1 Hardware Abstraction

4 Software

4.1 Hardware Constraints on Software

5 Toward Increasing Hardware Abstraction

6