

Modular Smartwatch Documentation

Spring 2015

EECS 399/EECS 499

Brian Lichliter

Luke Ahn

Akhil Guliani

June 20, 2015

RFduino

Overview

The microprocessor we settled with this quarter was the RFduino. The RFduino is a very small SoC with a Bluetooth module on board. This allows for a small form device that can easily communicate with the phone. As we have stated in the previous report, the RFduino is a very reliable microprocessor due to its small size, low battery consumption¹, and user-friendly software IDE.

Software

To get started with the software navigate to the arduino website and download the latest IDE. You need the latest version (1.6.5 +) in order to program ARM microprocessor based SoC.

<http://www.arduino.cc/en/Main/Software>

For setting up the Arduino IDE for RFduino, please refer to the previous report as it shows detailed installation steps.

Purchases

To get started with development, we purchased a few components including:

- RFduino DIP + Programmer
 - <http://www.rfduino.com/product/rfd90101-rfduino-2pc-dev-kit/index.html>
- RFduino Battery Pack DIP
 - <http://www.rfduino.com/product/rfd22128-cr2032-coin-battery-shield-for-rfduino/index.html>
- Multiple RFduino SoCs
 - <http://www.rfduino.com/product/rfd22301-rfduino-ble-smt/index.html>

Other components are listed here: <http://www.rfduino.com/shop/index.html>

Getting Started

Instructions for getting started with the RFduino are listed here:

<https://github.com/RFduino/RFduino>

¹ RFduino with 1 wire thermometer - 0.02mA~0.04mA. RFduino with 1 wire analog thermometer : 0.04mA.

Project

Hardware

Sensors + Peripherals

We honed in on getting the system to work with a couple sensors this quarter to limit complexity when integrating the electronics with the housing built by the DTC teams (more on that in a later section). The sensors we picked are:

- Infrared Thermometer - MLX90614
 - <https://www.sparkfun.com/products/9570>
- Temperature Sensor - TMP36
 - <https://www.sparkfun.com/products/10988>
- LED - RGB Diffused Common Cathode
 - <https://www.sparkfun.com/products/9264>
- Flex Sensor 2.2"
 - <https://www.sparkfun.com/products/10264>

Cheat Sheet

Given to each DTC smartwatch team was a cheat sheet that described the key components we intended to include in the NU modular smartwatch v1.

<https://docs.google.com/document/d/1HmPRBFi7JoAw0H3BPowAba76oZdk8ZkcYvqi3K8GpV0/edit?usp=sharing>

Electronic Components Diagram

Flex Sensor

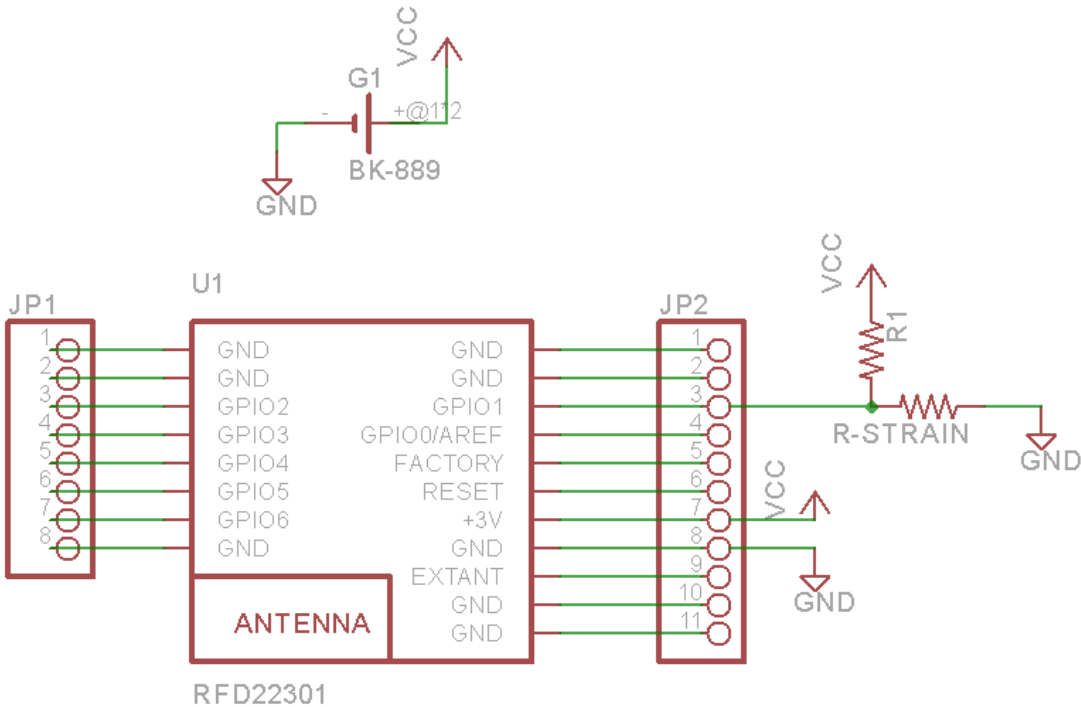


figure 1- a diagram of a flex sensor connected to the RFduino

In order to sense the change of the electrical current, we have built a voltage divider. One side of the voltage divider will be our flex/force sensor and the other side will be a resistor with the equal resistance value of the flex/force sensor. We have calculated the nominal resistance of the force sensor to be ~15k ohms. The GPIO pin 1 from RFduino will be set as an analog input and will receive voltage change from the node between the two resistors in our voltage divider circuit. The RFduino has 10 bit ADC ranging from 0 ~1023. The raw ADC value will be sent to the smartphone over the Bluetooth. The phone receives and plots the values.

MLX90614 IR temperature sensor

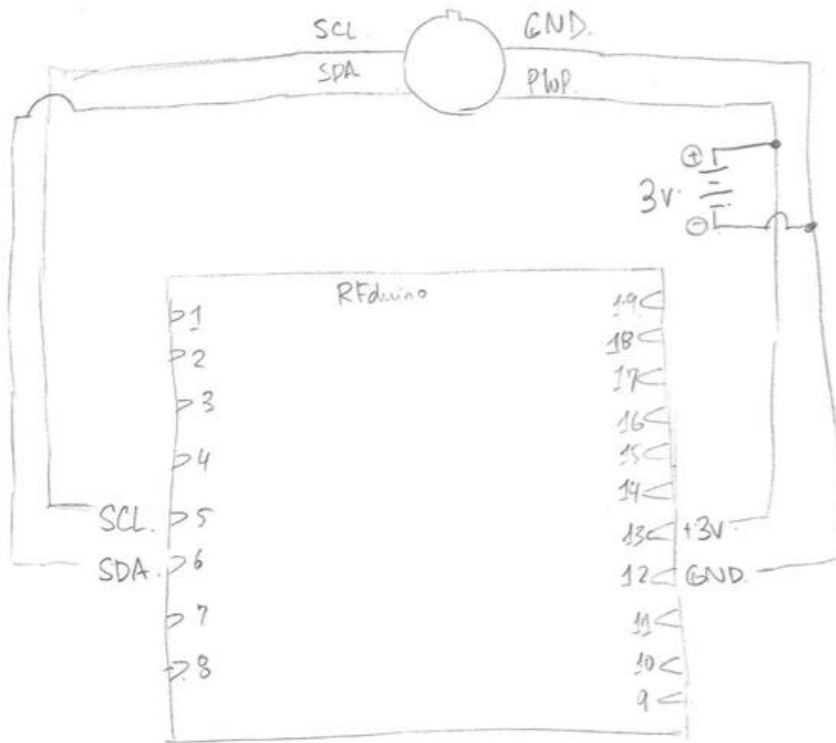


figure 2- a IR temperature sensor connected to the RFduino

The MLX90614 IR temperature sensor uses SMBus protocol (very similar to I2C). The RFduino has pin 5 dedicated for SCL and pin 6 for SDA. We have used standard Arduino I2C library along with MLX90614 library for the communication.

Software

Sensors

Each sensor we worked with had libraries already written for the arduino and easily integrated with the RFduino.

- Infrared Thermometer - MLX90614
 - <https://learn.adafruit.com/using-melexis-mlx90614-non-contact-sensors/wiring-and-test>
- Temperature Sensor - TMP36
 - <https://learn.adafruit.com/tmp36-temperature-sensor/using-a-temp-sensor>
- LED
 - <https://learn.adafruit.com/adafruit-arduino-lesson-3-rgb-leds/overview>

Bluetooth communication

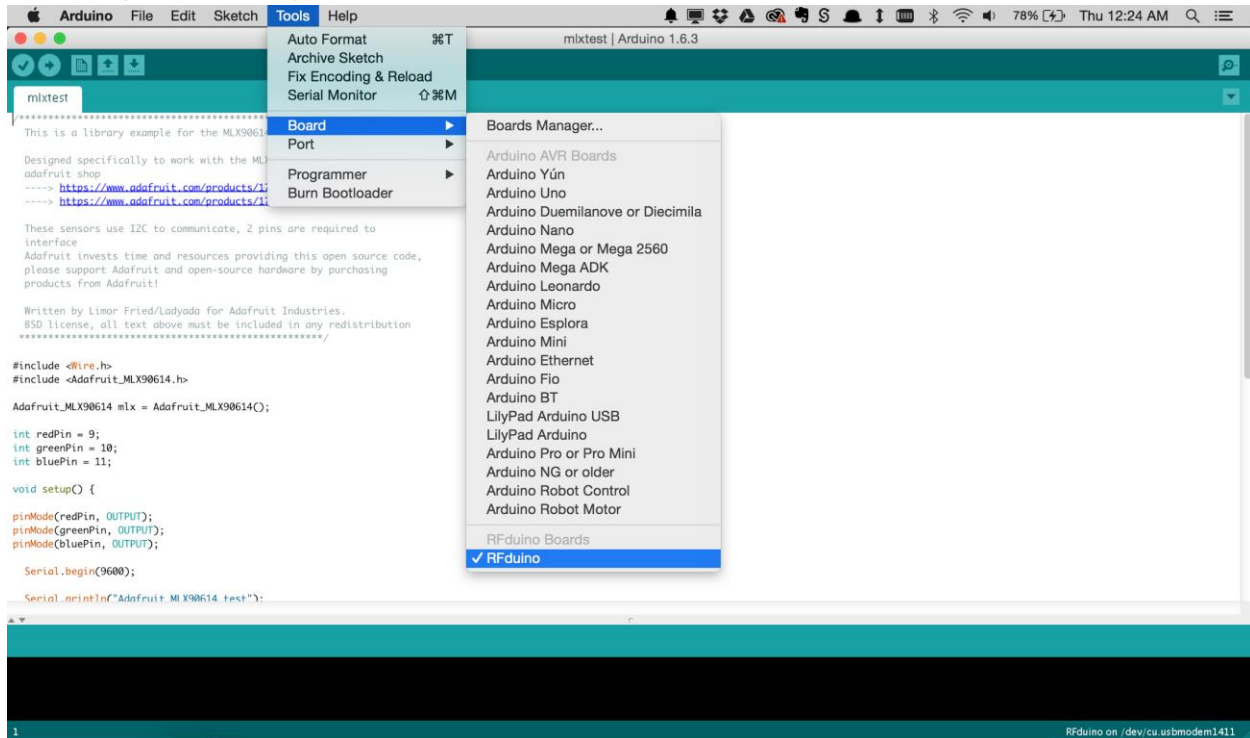
RFduino has a bluetooth reference guide online for once you begin programming along with

many example programs that come with the library.

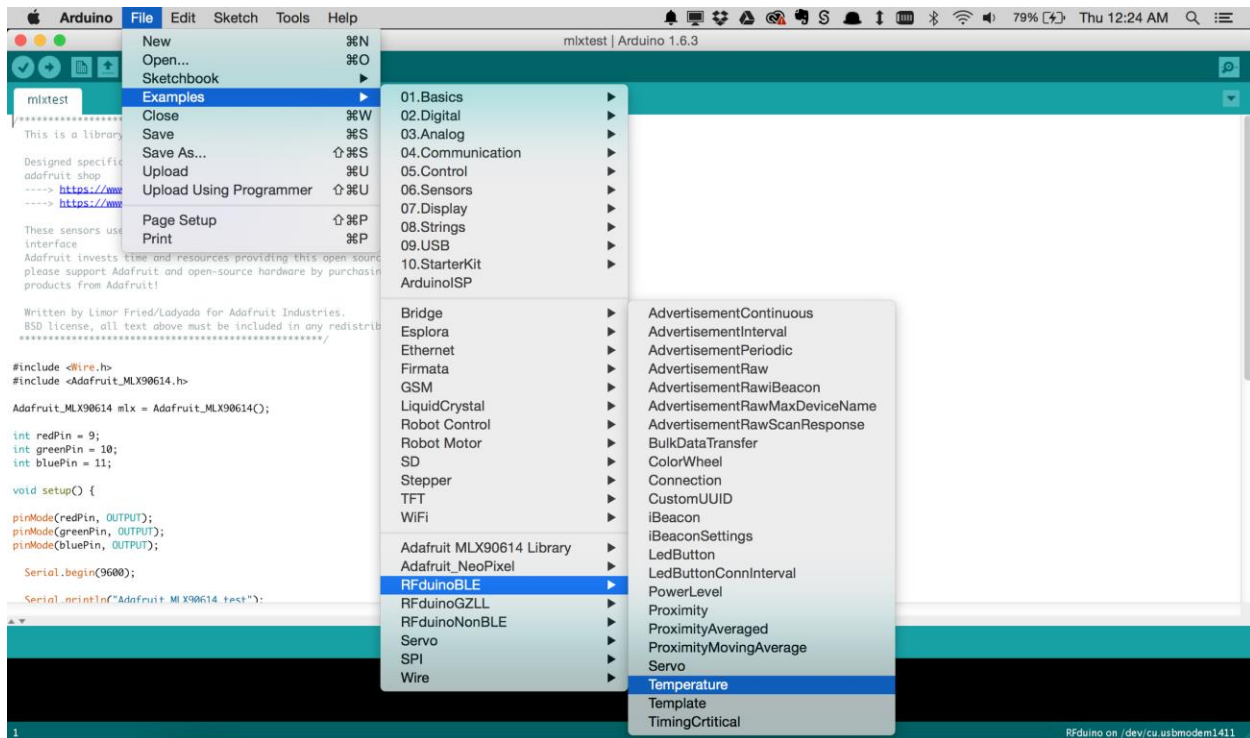
BLE reference

http://www.rfduino.com/wp-content/uploads/2014/03/rfduino.ble_programming.reference.pdf

How to open an example



Step 1: Select RFduino from the tools > board > RFduino



Step 2: Select an example from file > examples > RFduinoBLE > {{example}}

DTC -- Casing

Two DTC teams were tasked with creating a housing using CAD and 3d printing for the sensors we outlined early in the project. All the assists from their projects will be included in the google drive/github repo.

backsjnsdlkcmnas

Next Steps

- Begin to solidify the exact sensors to move forward with and understand the specifications and requirements surrounding those sensors
 - GSR is an obvious choice
 - <http://www.seeedstudio.com/depot/Grove-GSR-p-1614.html>
 - http://www.seeedstudio.com/wiki/Grove_-_GSR_Sensor
- Understand the power consumption of the RFduino
- Develop a way to determine the power consumption of arbitrary systems of sensors and the RFduino
- Understand and integrate the lithium ion charging circuit
- Begin work on output from Android to RFduino
 - Understand tricolor LED power consumption
 - Is it visible enough

- Write the code for both the android platform and the RFduino platform
- Research whether there is a smarter more efficient device for the needs of the research study
- Reach out to students to gauge interest in joining the smartwatch team
- Look into the idea of creating a protoboard for the RFduino smartwatch
 - Look into manufacturers both traditional and of flex PCBs
 - Look into flex protoboard
 - <http://www.adafruit.com/products/1518>
- Look into SMD protoboard
 - <https://www.sparkfun.com/products/retired/8708>
- Look into low profile headers to create a smaller version of the RFduino dev board