Wireless Applications of a Refactored Prosthesis

"W.A.R.P."

Group 9 - Fall 2016

Daniel Mor	CpE
Niko Tubach	CpE
T. Brandon Ashley	CpE

limbitless SOLUTIONS

UNIVERSITY OF CENTRAL FLORIDA 1963 REACH FOR THE STARS

Motivation

- Research & Development
- Ease of Access
- Wireless Integration
- Additional Sensors
- Reduce Power Consumption
- Reduce Cost
- Toolkit for future Limbitless Engineers

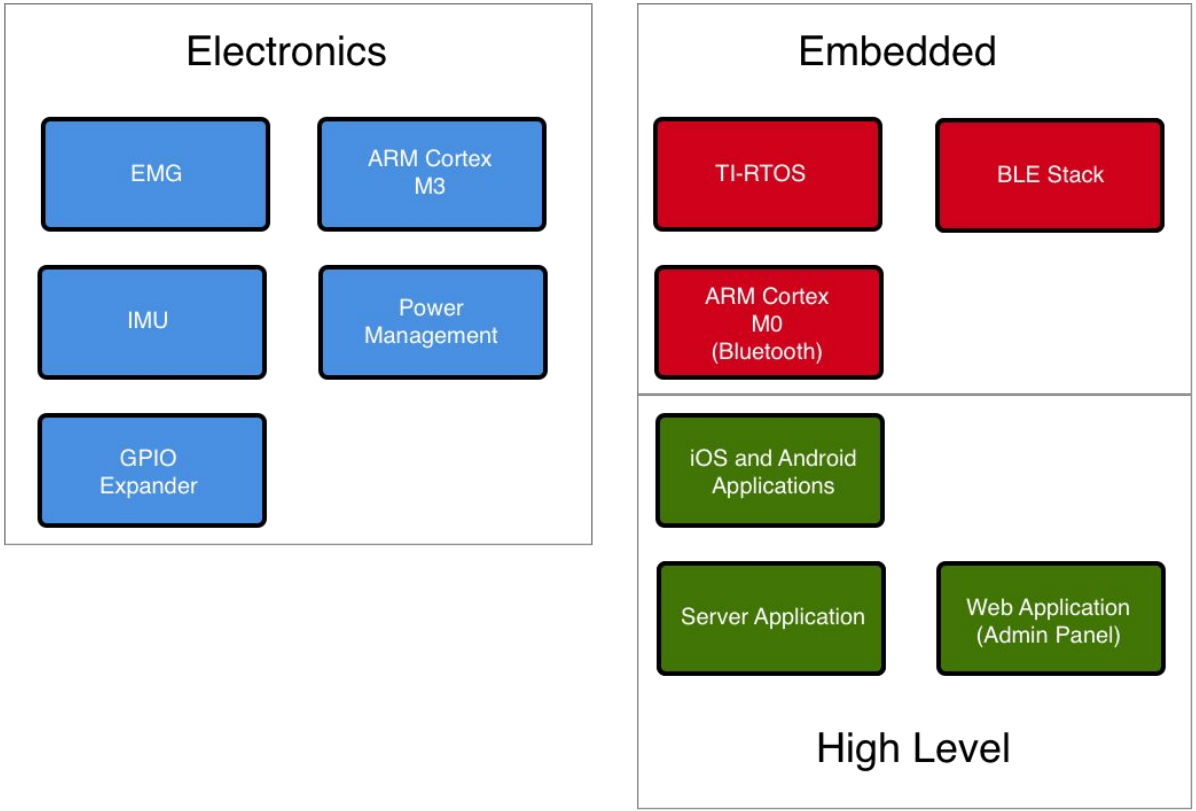
Goals and Objectives

- **Electronics**
  - Update regulator to reduce cost and increase efficiency
  - Update EMG Sensor
    - Lower operating voltage
    - Digitally controlled hardware threshold
    - Reduce Cost
  - Control 2 RGB LEDs
  - Control 2 Servos
  - Add IMU
  - Add external flash memory for wireless reprogramming

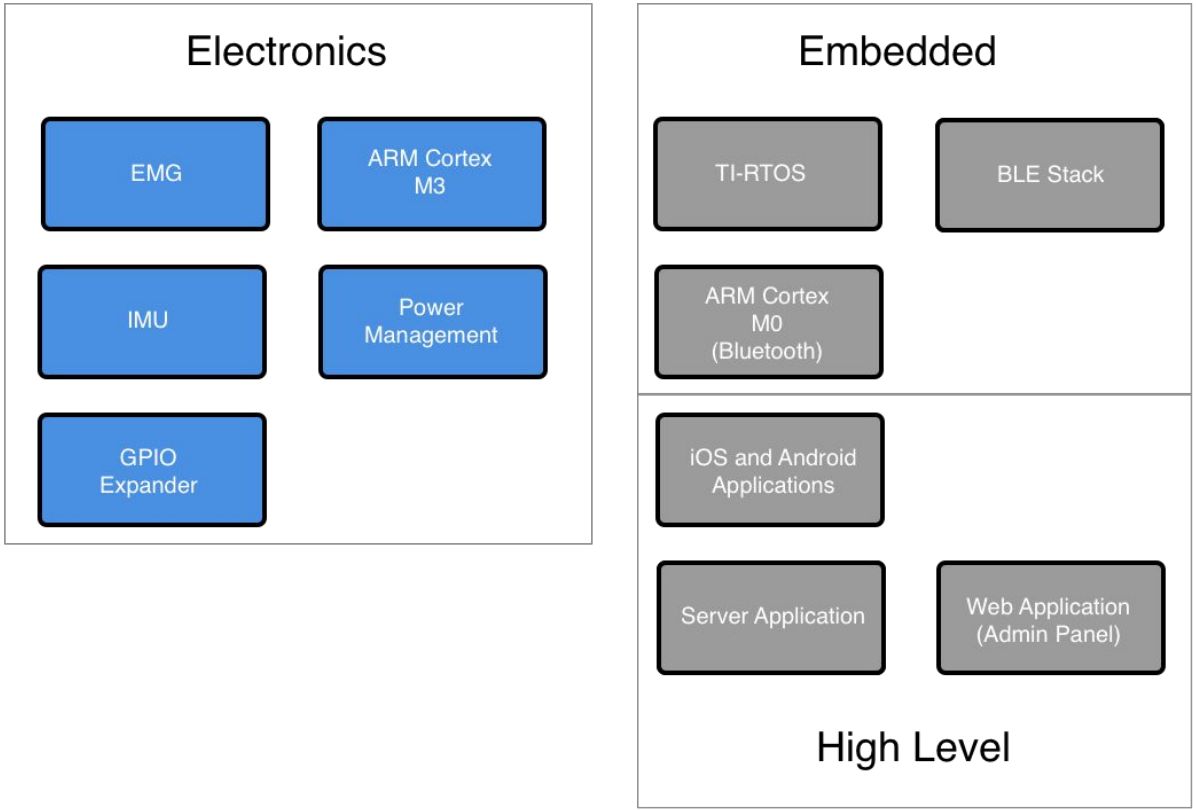
- **Embedded Software**
  - Utilize TI-Real Time Operating System for multithreaded processing
  - I^2C and SPI Interface
  - Utilize Bluetooth Low Energy Stack

- **Mobile & Server Development**
  - Transmit configuration data to and from PCB
  - Remote Data logging and diagnostics
  - Request assistance from Limbitless team in real time

Project-Scope Block Diagram



Electronics

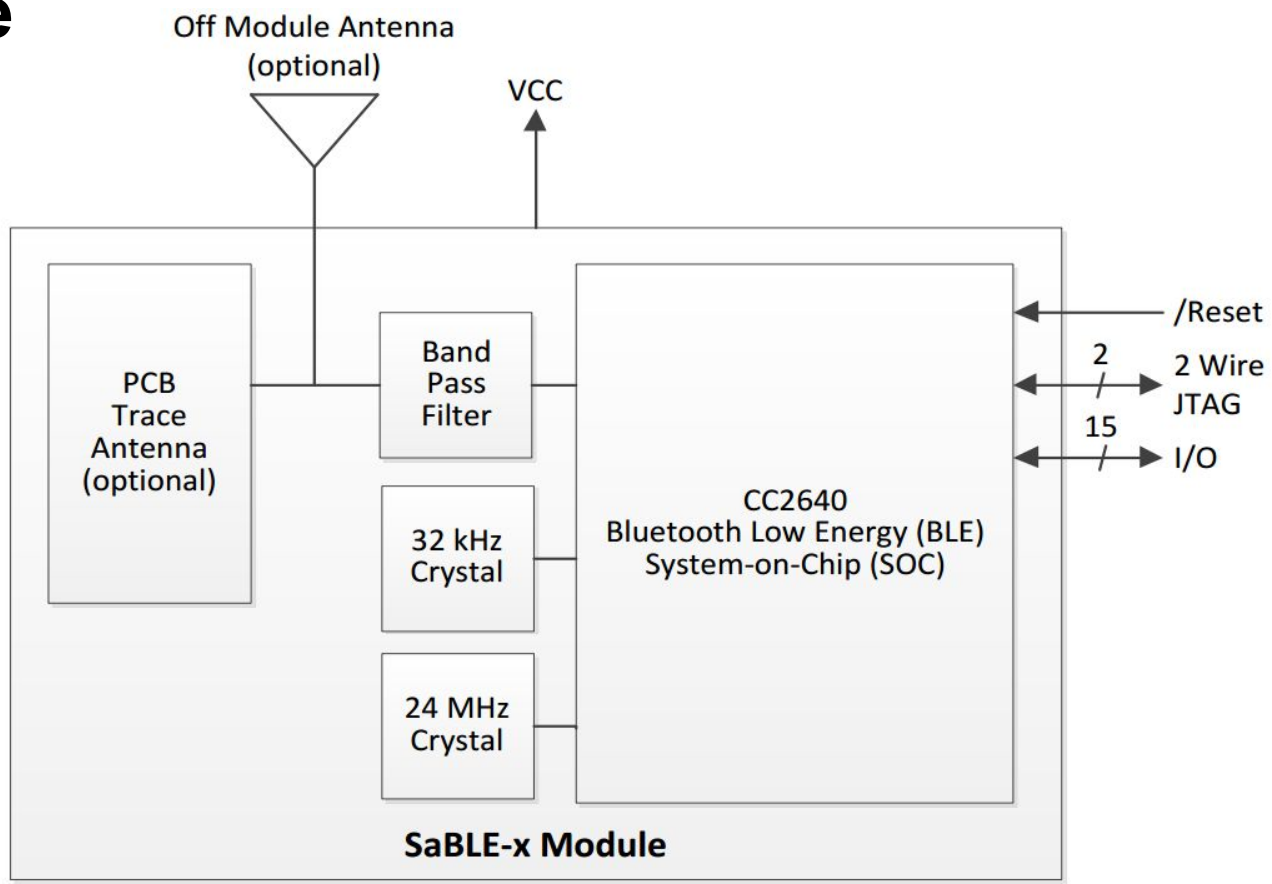


PCB Specifications

Description	Specification
Price	Under $100 for the final design
Input Voltage	6.5v - 8.5v (7.4v Nominal)
Operating Time	8 - 10 hours
Min Trace Width / Clearance / Via Size	8 mils / 8 mils / 13 mils
Layers	2 - 4
Dimensions (Max)	100mm x 100mm x 25mm (Approximately 4in x 4in x 1in)

SaBLE-x Module

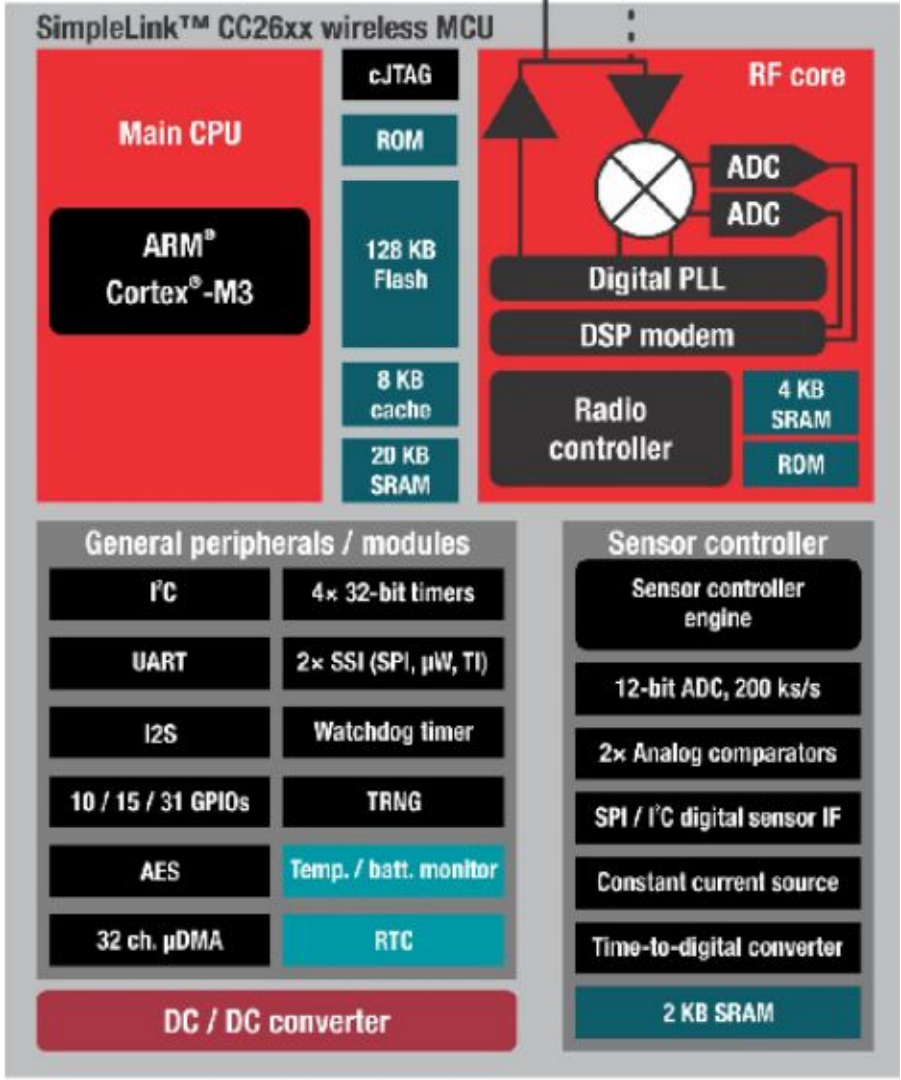
- Integrated CC2640 with integrated passive components
- Includes FCC approved PCB Trace Antenna
- Dimensions <11.63 x 17.86>



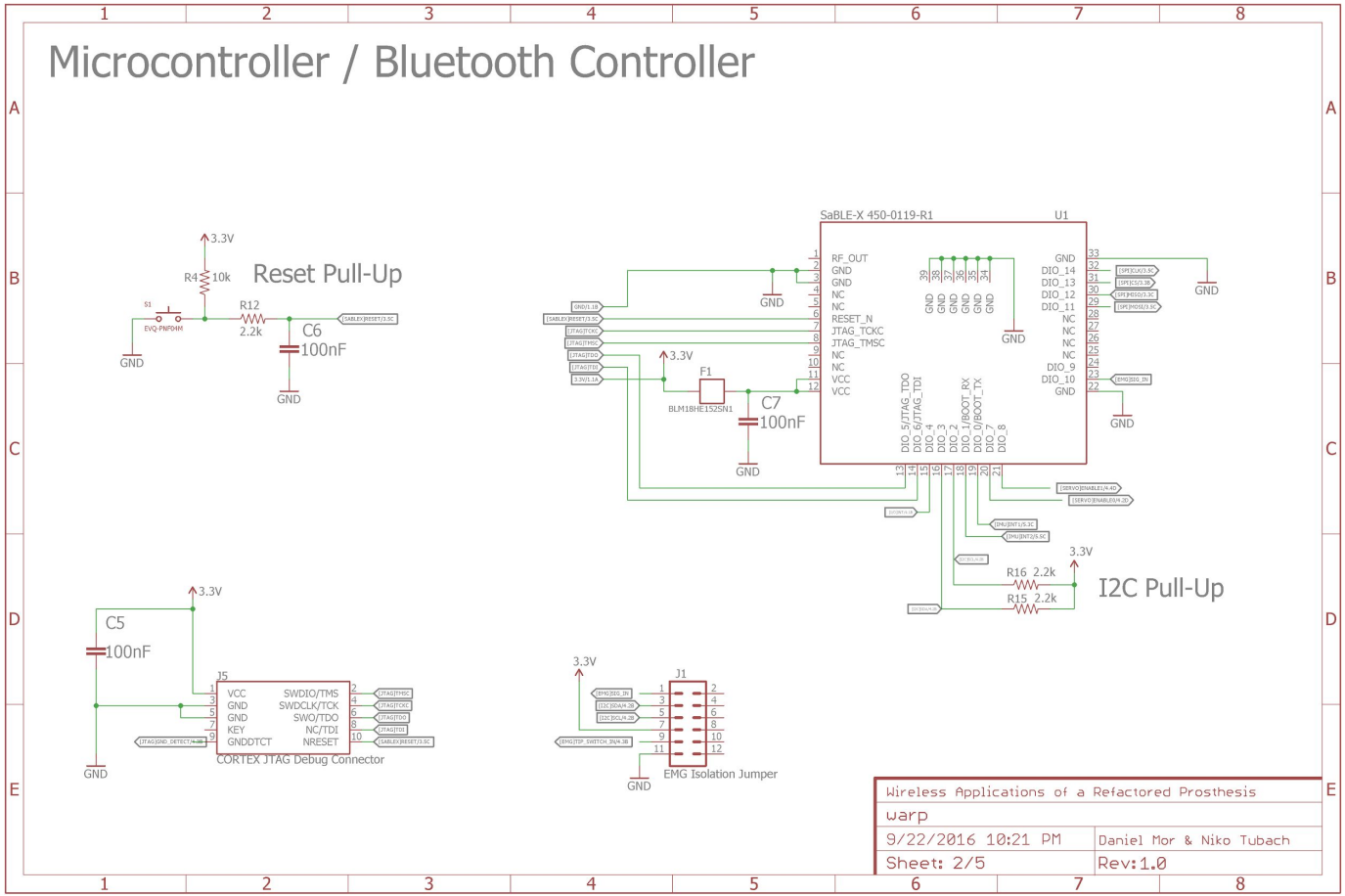
CC26xx

Main Features:

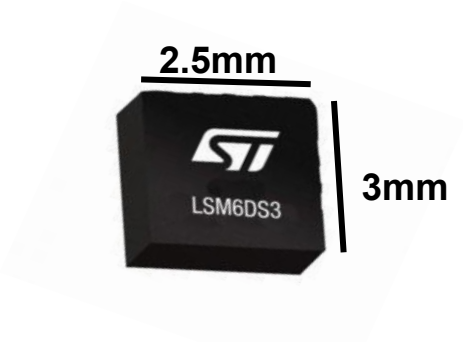
- ARM Cortex-M3 processor (System Core)
- 128 KB of Main Flash Memory
- 28 KB of SRAM (8KB cache)
- Compatible with all common transfer protocols
- ARM Cortex-M0 processor (Radio Core)
- 15 GPIO pins
- 12 Bit ADC
- Dedicated Sensor Controller

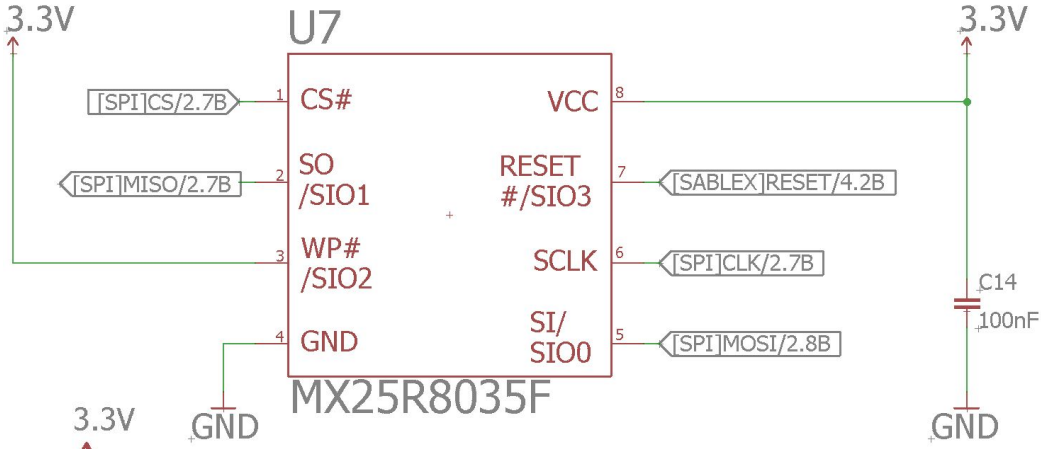


PCB Schematics

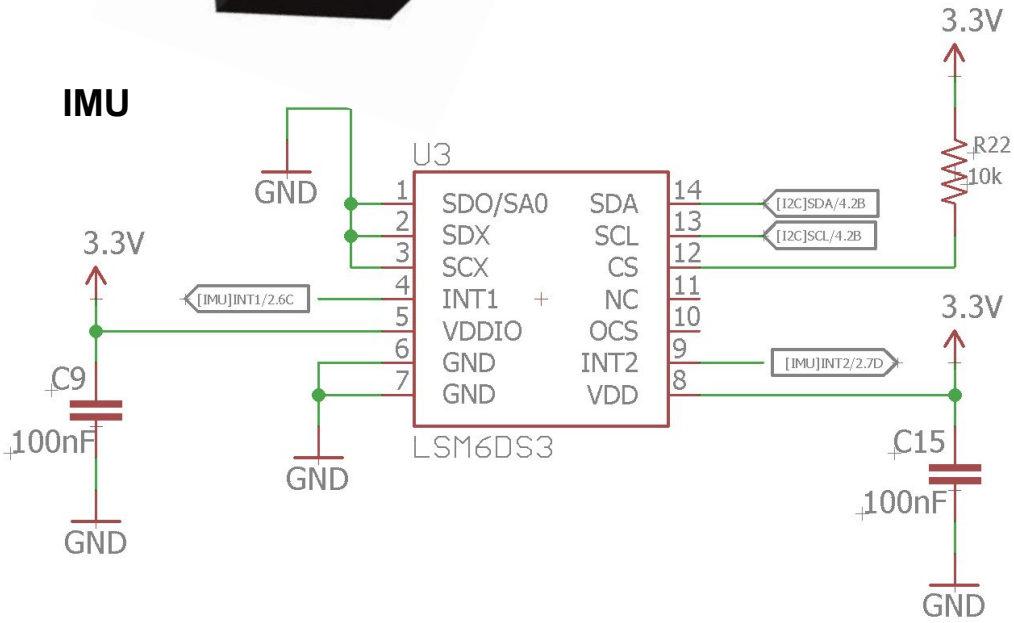


PCB Schematics

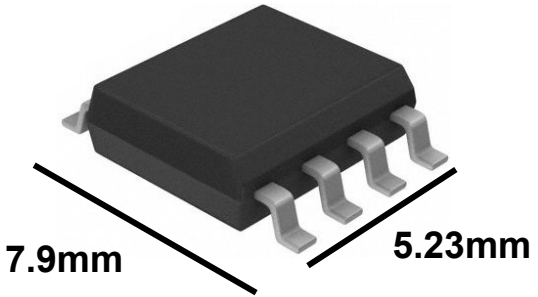




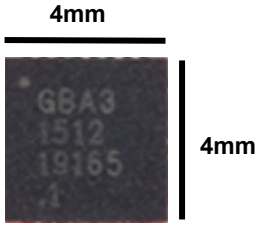
IMU

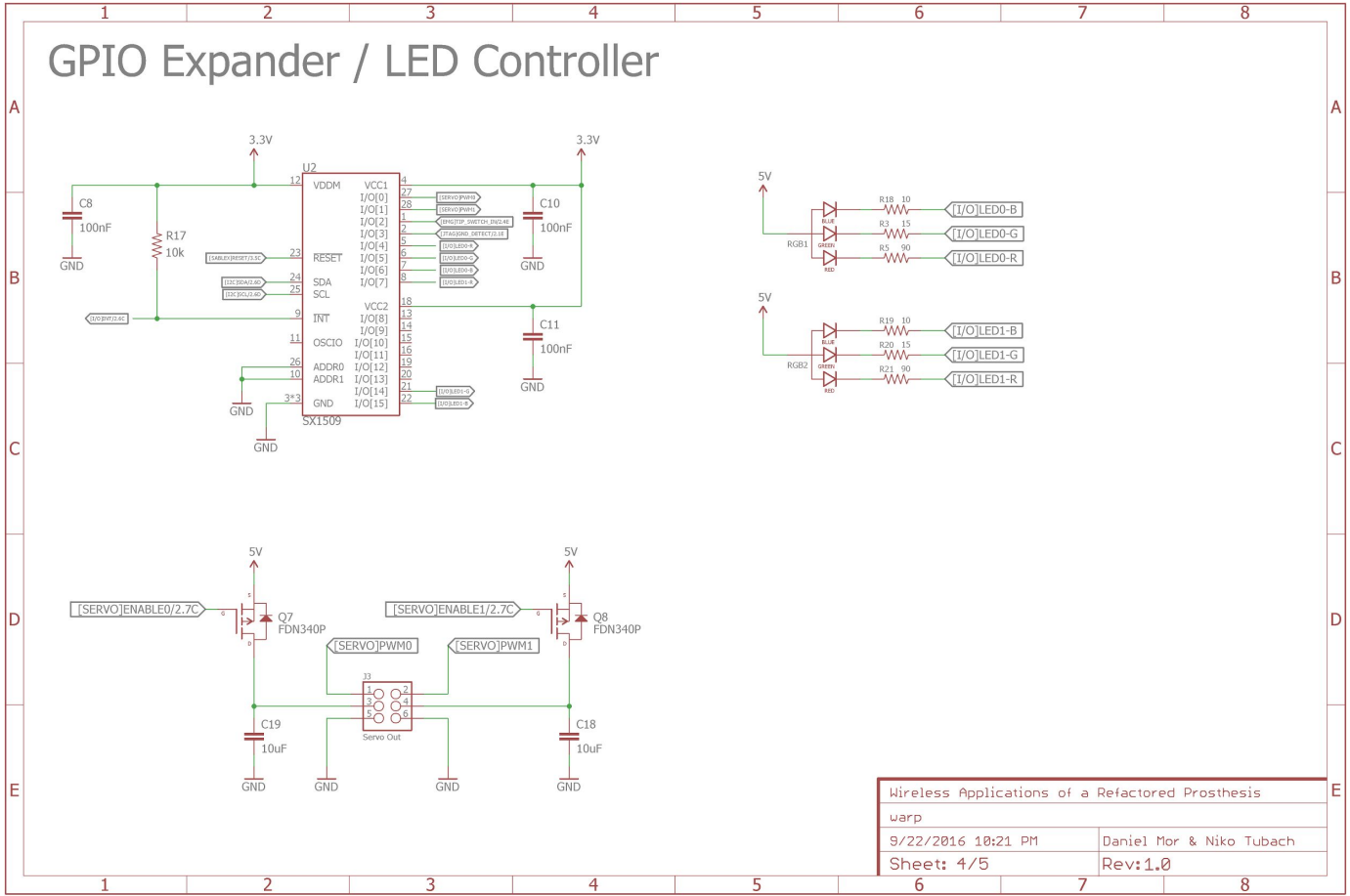


External Flash Memory

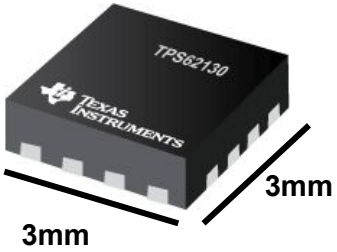


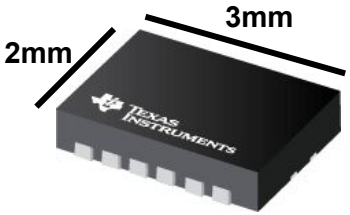
PCB Schematics

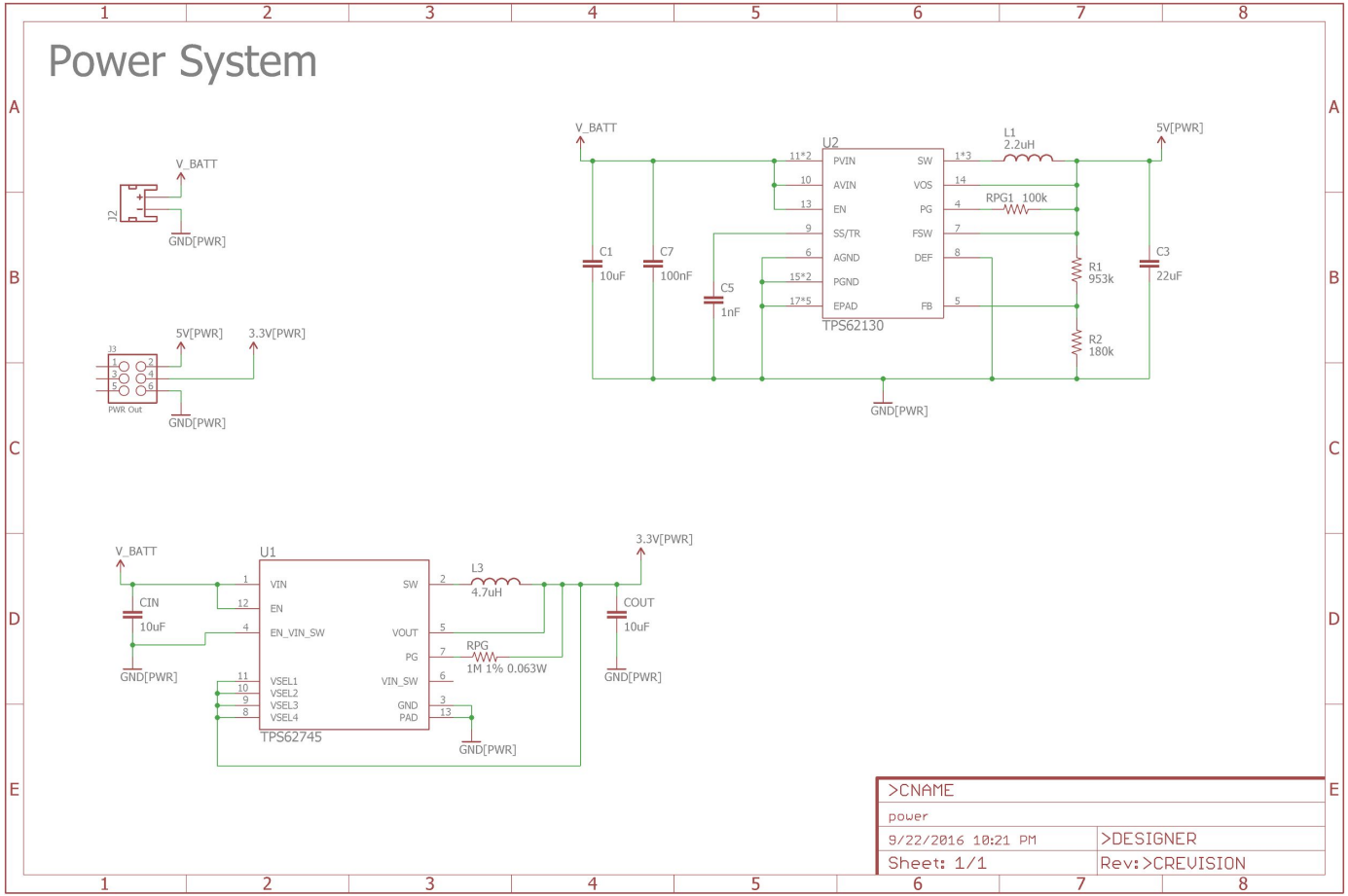




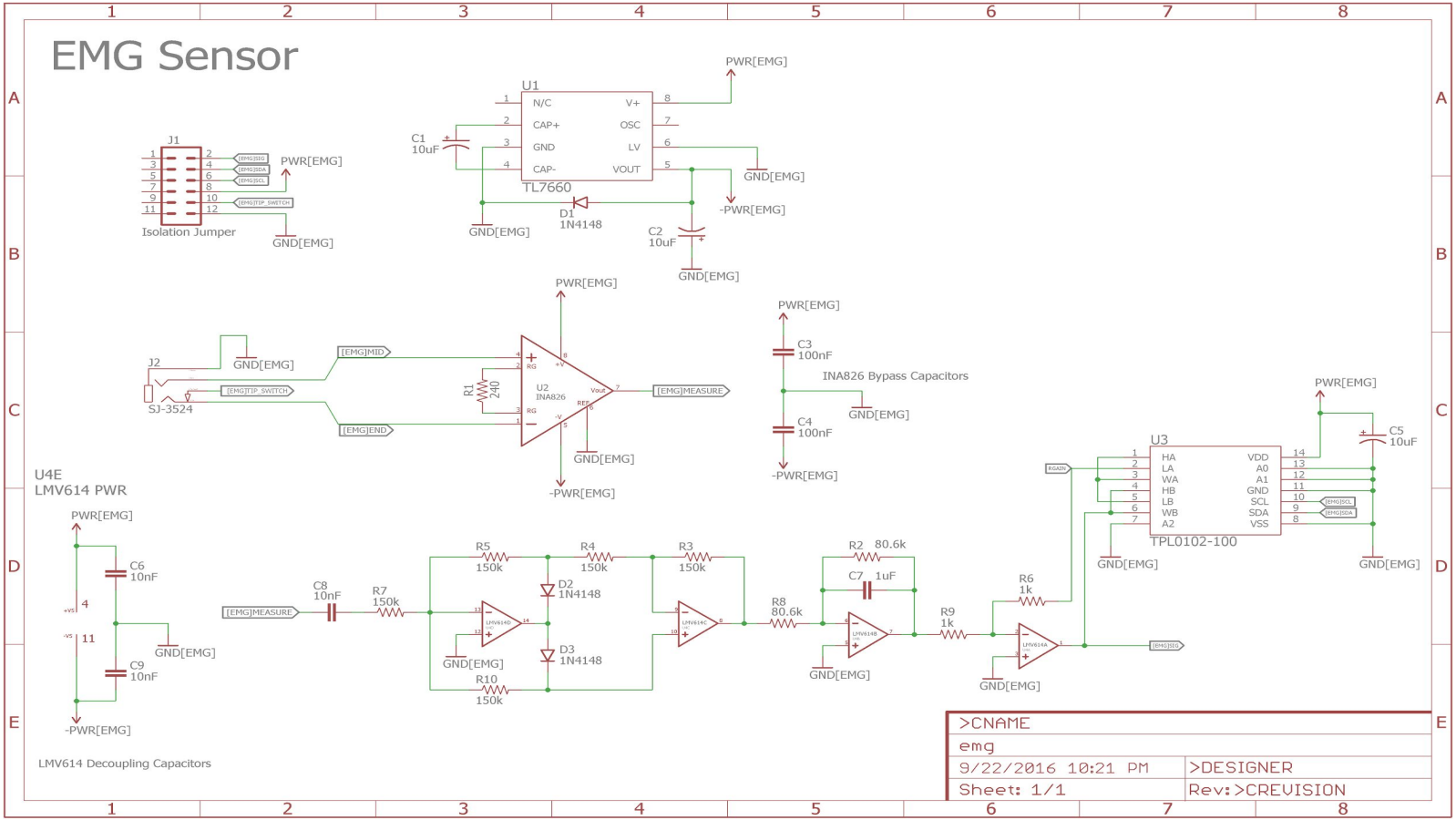
PCB Schematics





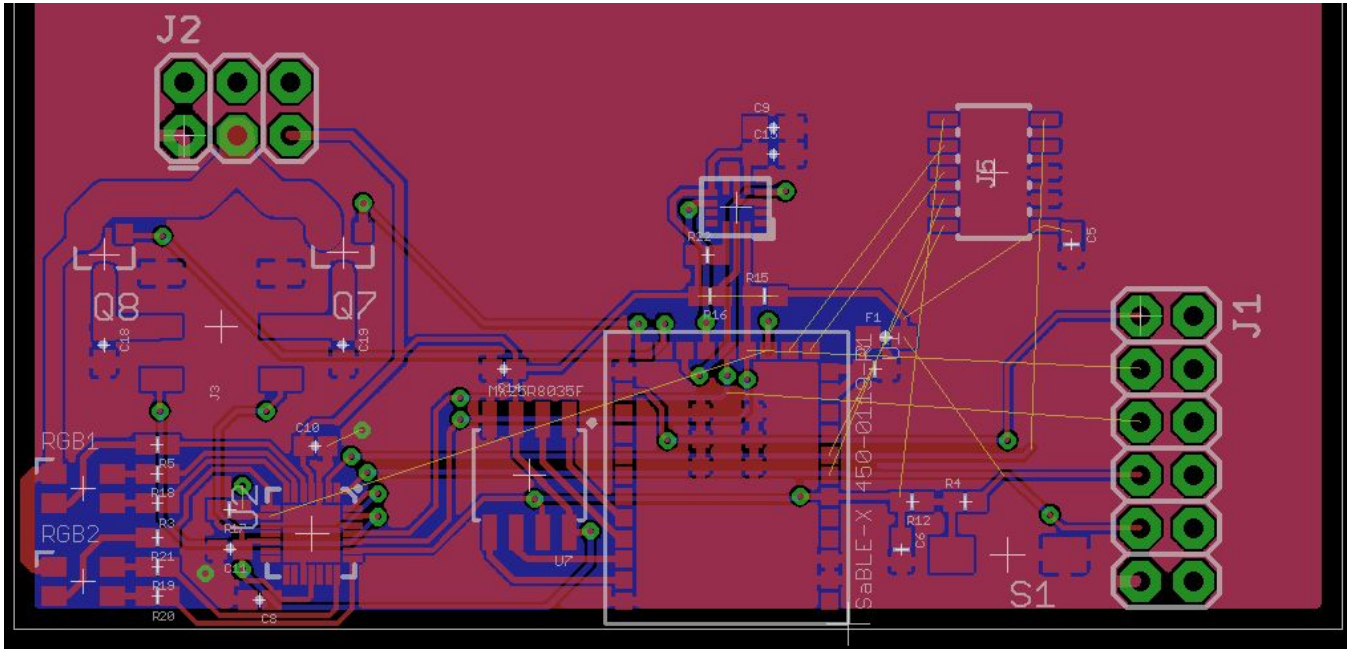


PCB Schematics



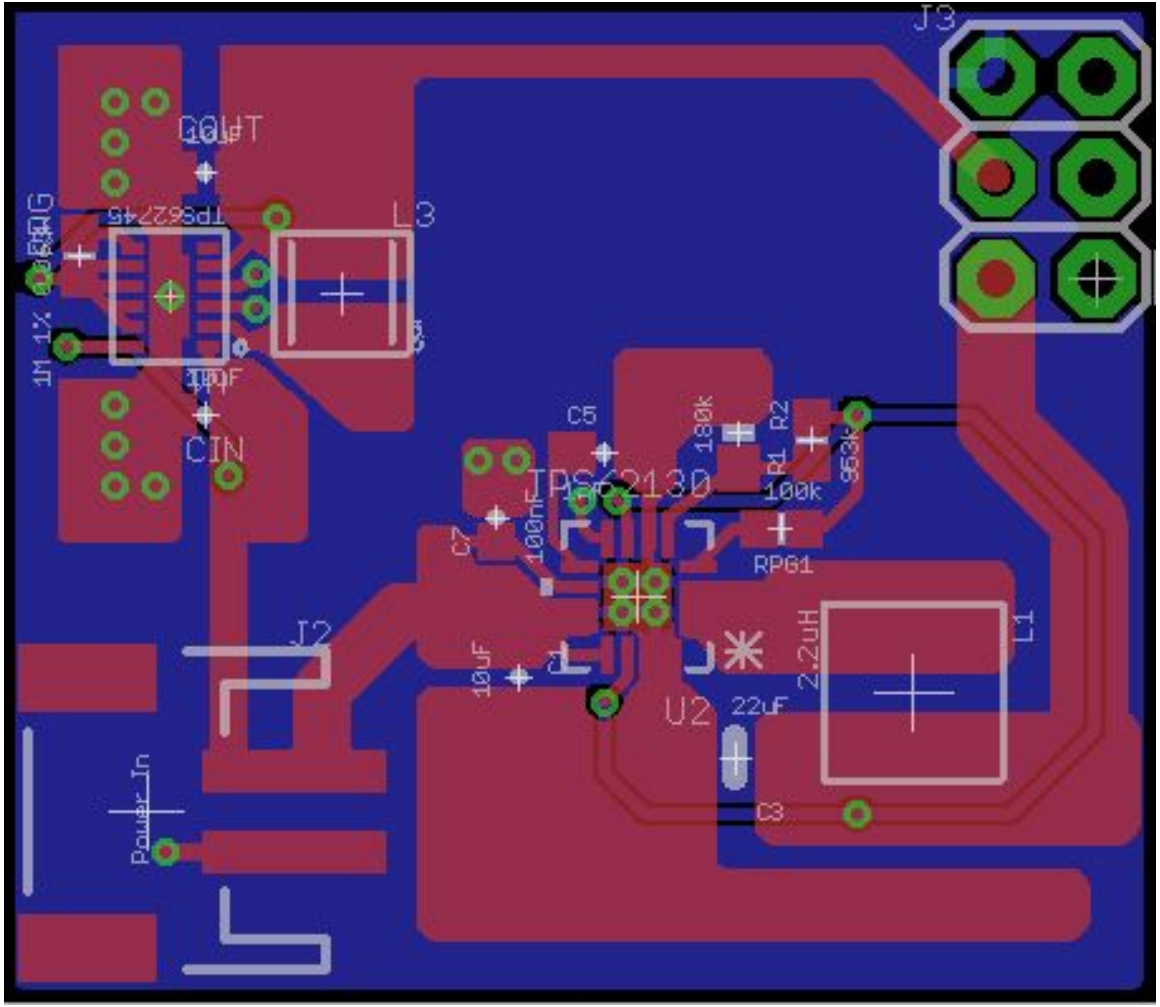
PCB Layout

Digital Logic Board



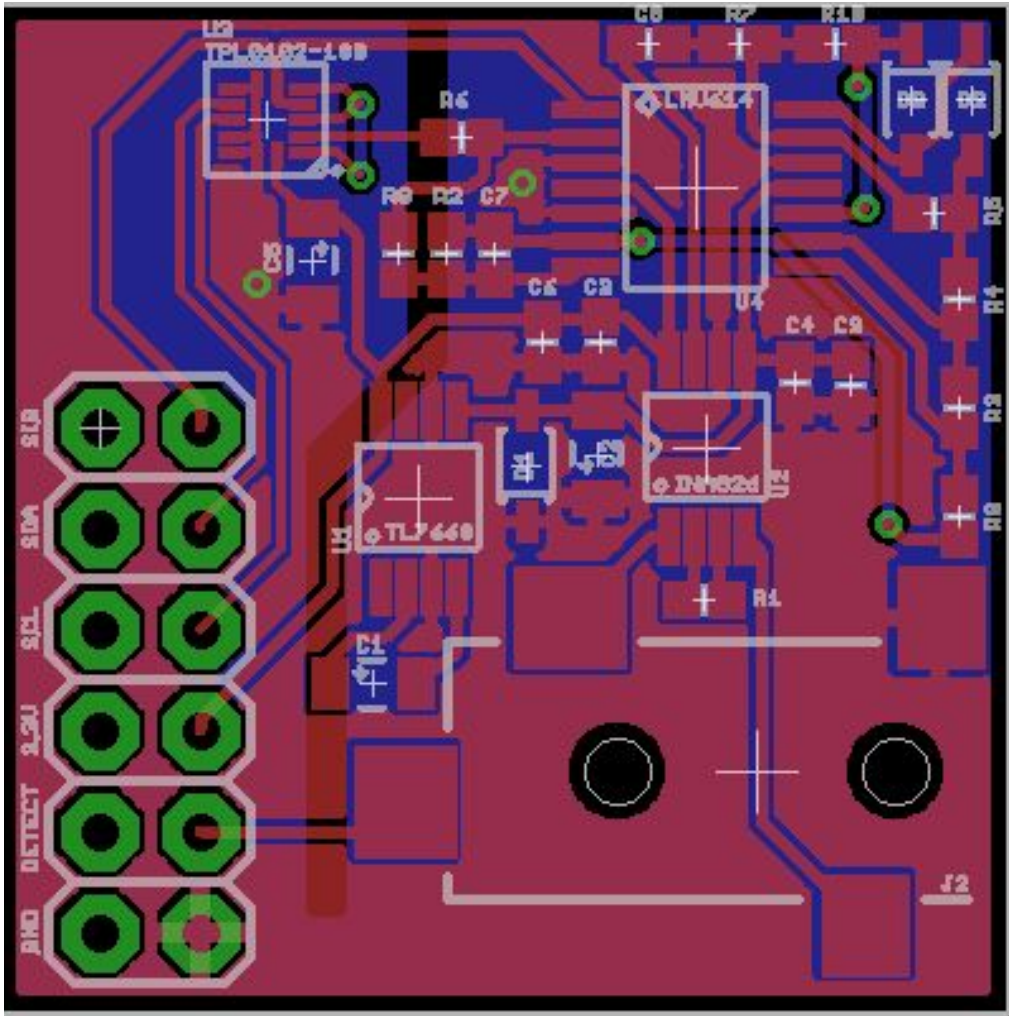
PCB Layout

Power board

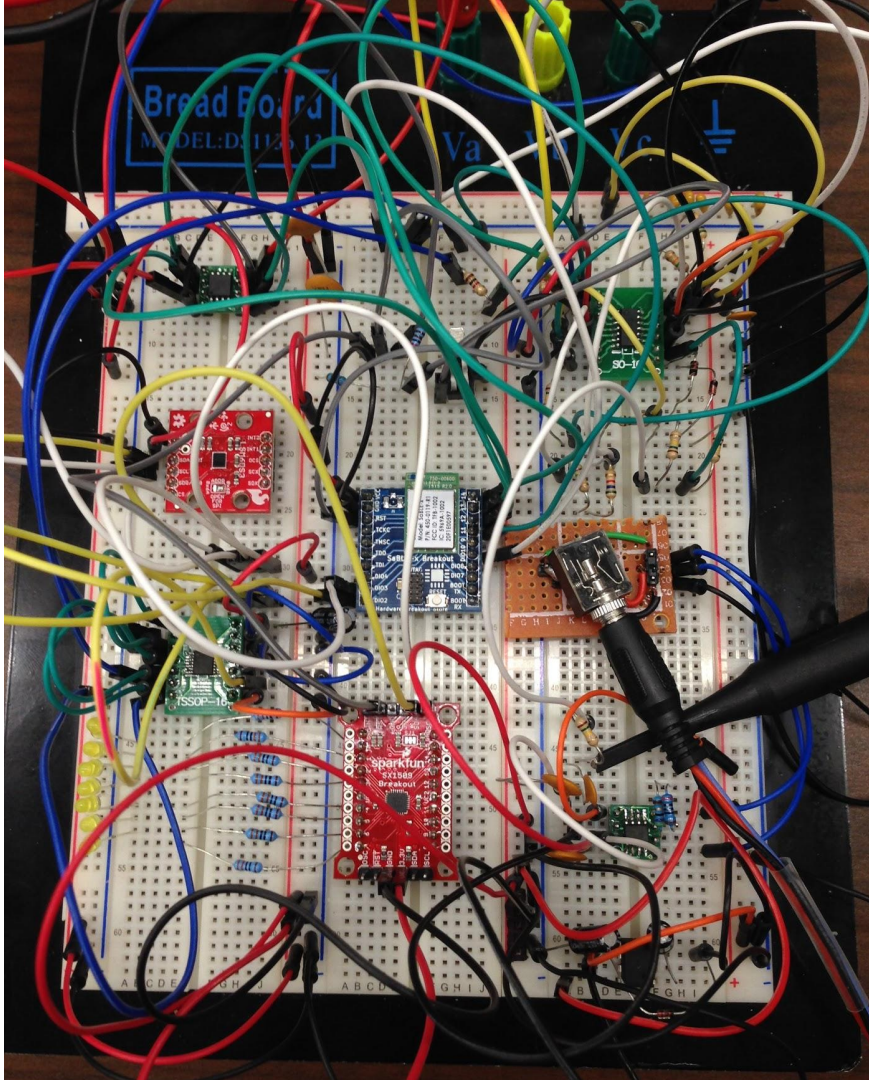


PCB Layout

EMG Board

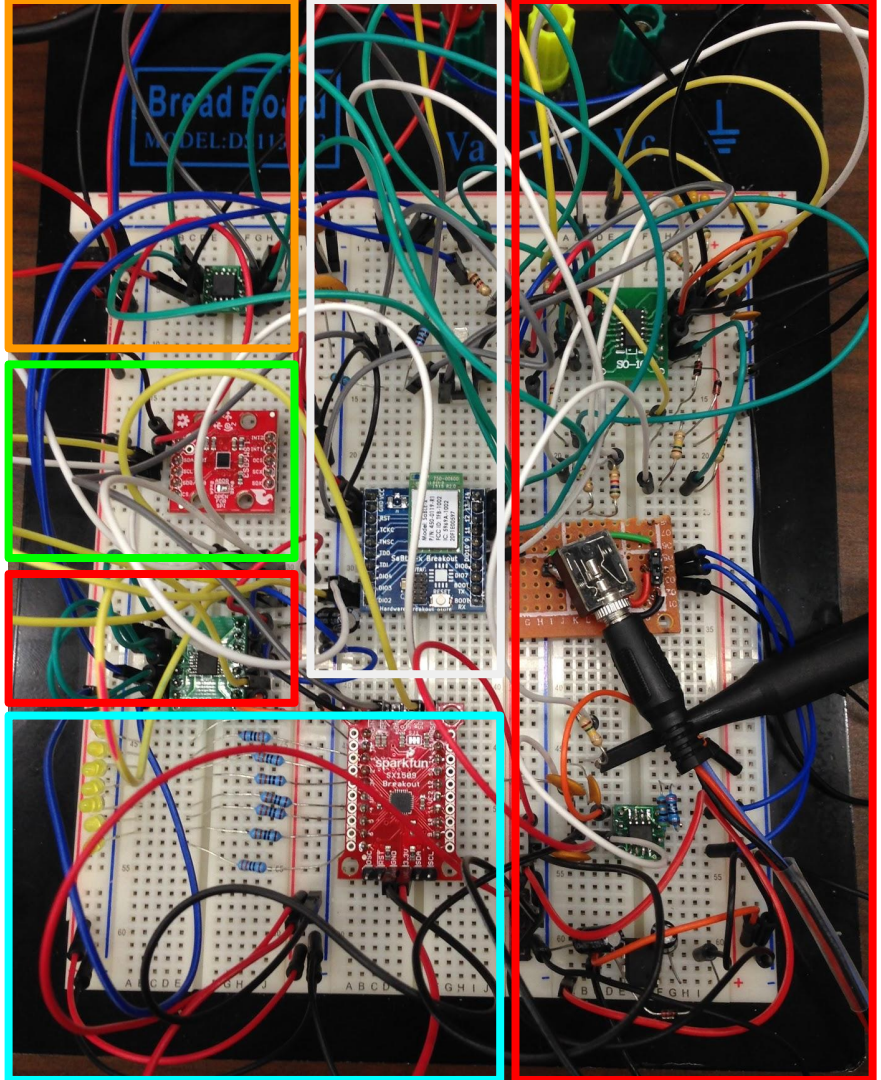


Prototyping



Prototyping

- SaBLE-x
- EMG Sensor
- 1 MB External Flash Memory
- Accelerometer / Gyroscope (IMU)
- GPIO Expander

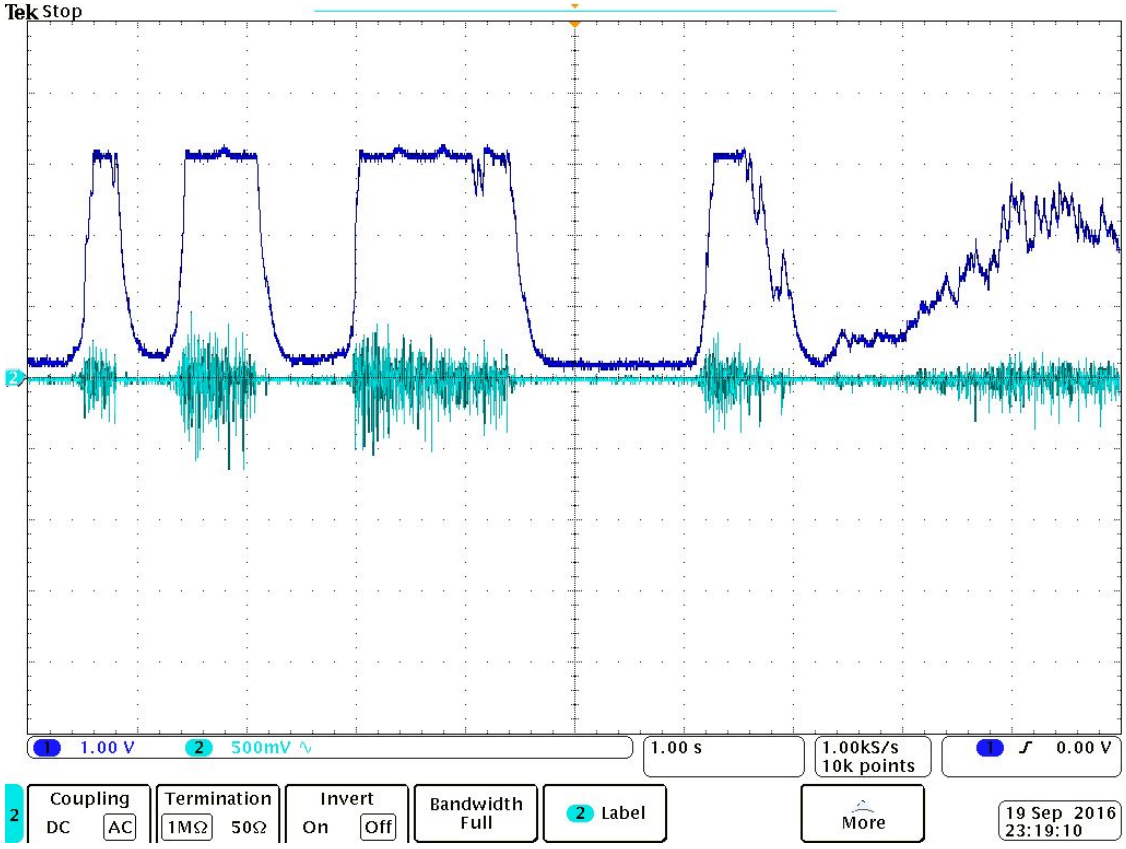


EMG Breadboard Output

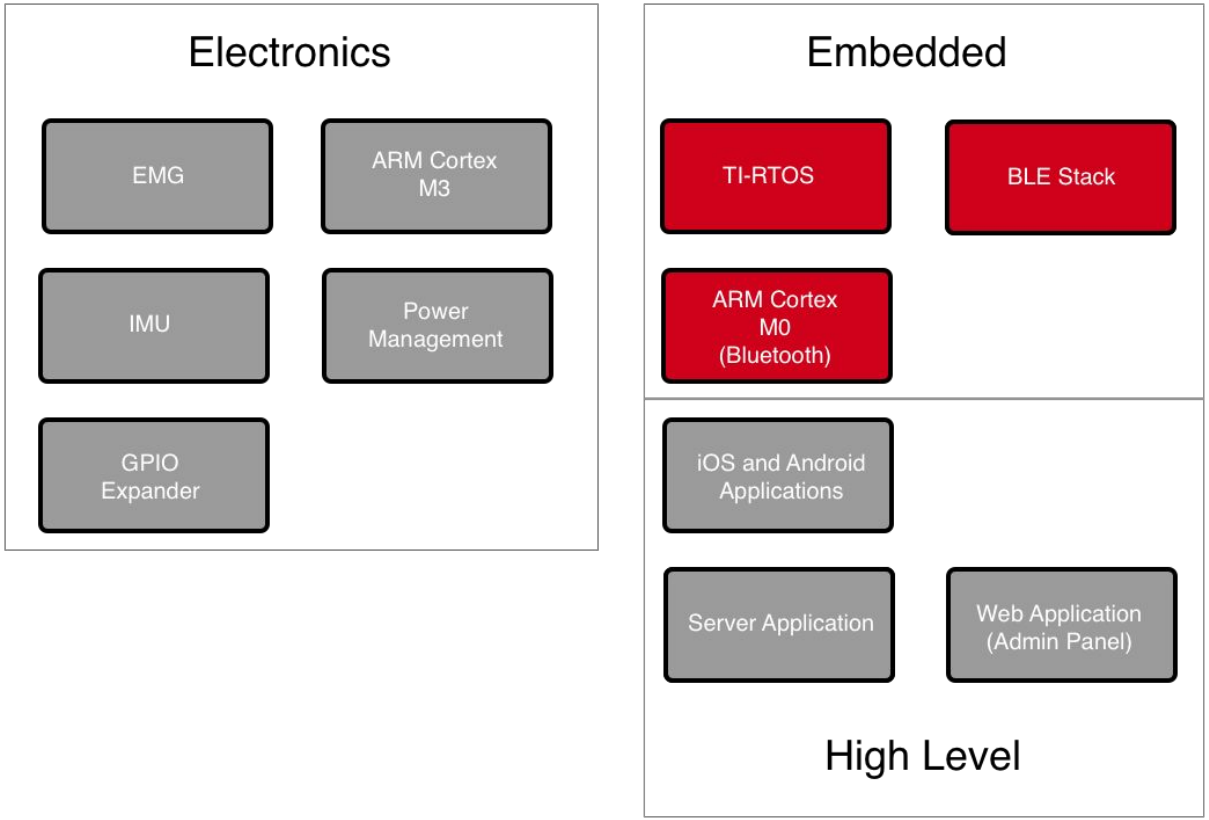
Processed Signal (1v Scale)

Raw EMG Signal (500mV Scale)

Average of +/- 50mV Ripple



Embedded

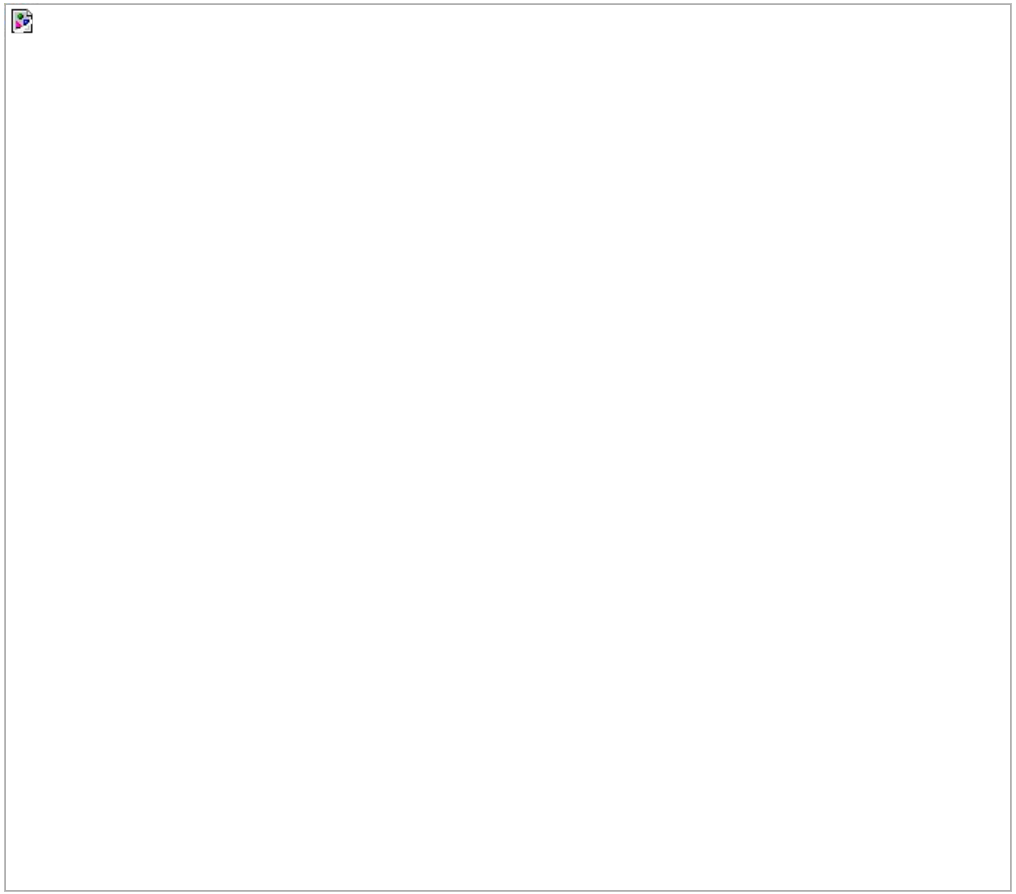

Electronics
EMG
ARM Cortex M3
IMU
Power Management
GPIO Expander
Embedded
TI-RTOS
BLE Stack
ARM Cortex M0 (Bluetooth)
iOS and Android Applications
Server Application
Web Application (Admin Panel)
High Level

Embedded Software

- **TI-RTOS**
  - Interface with external peripherals through I^2C and SPI
  - Utilize ADC to digitize EMG Sensor Output
  - Schedule tasks and allocate system resources
  - Design multi-threaded application software
    - Thread Synchronization (semaphores, monitor, queues, mailbox)
- **Bluetooth Low Energy (BLE) Stack**
  - Manage BLE Pairing / Profiles / Services
  - Transfer data between low level RTOS and mobile apps
  - Provide read/write functionality to external application
- **Boot Image Manager (BIM)**
  - Over the Air Download image management for wireless reflashing

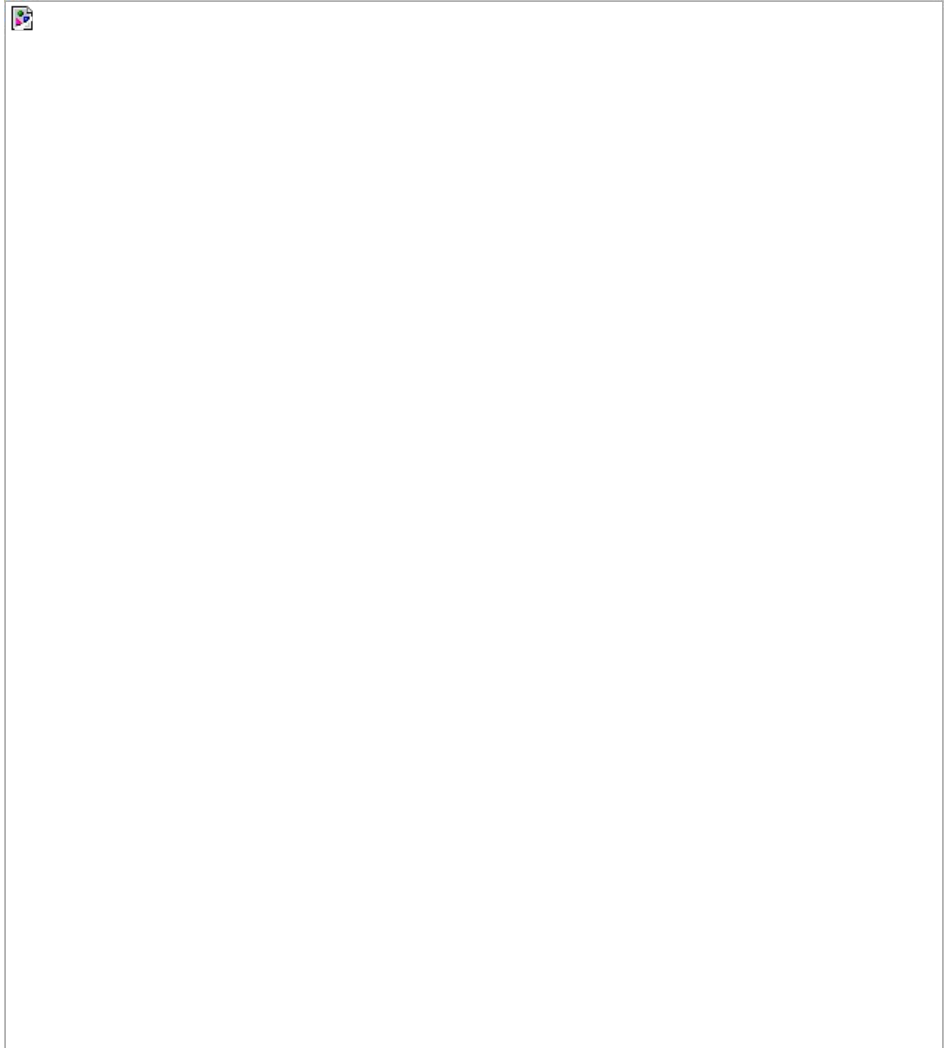
Top Level Embedded Diagram

- Hardware interrupt fires
- Kernel processes interrupt
- TI-RTOS evaluates interrupt and sends to BLE Manager
- ICall function transfers data to BLE process
- BLE Stack wraps and passes data to RF Core for wireless transmission
- Return data processed in reverse

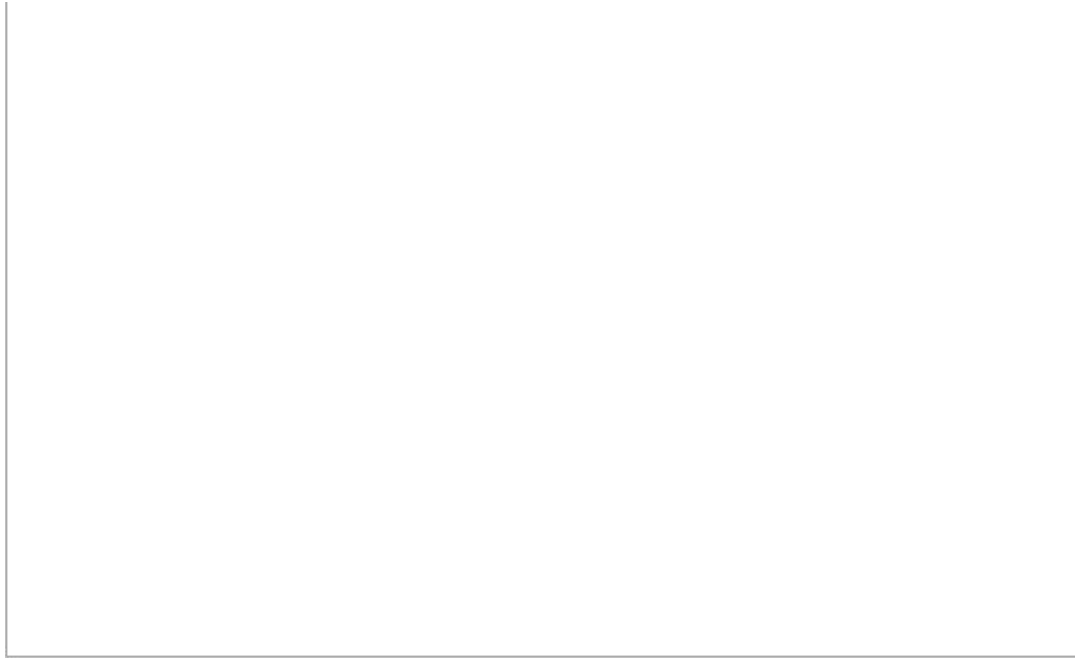


TI-RTOS Architecture
System Initialization

- Boot Image Manager (for Over the Air Download handling)
- TI-RTOS Reset Calls (on init)
- Main function calls (utilizes Driver configs)
- All Tasks are started and interrupts are enabled



TI-RTOS Architecture
Main Loop

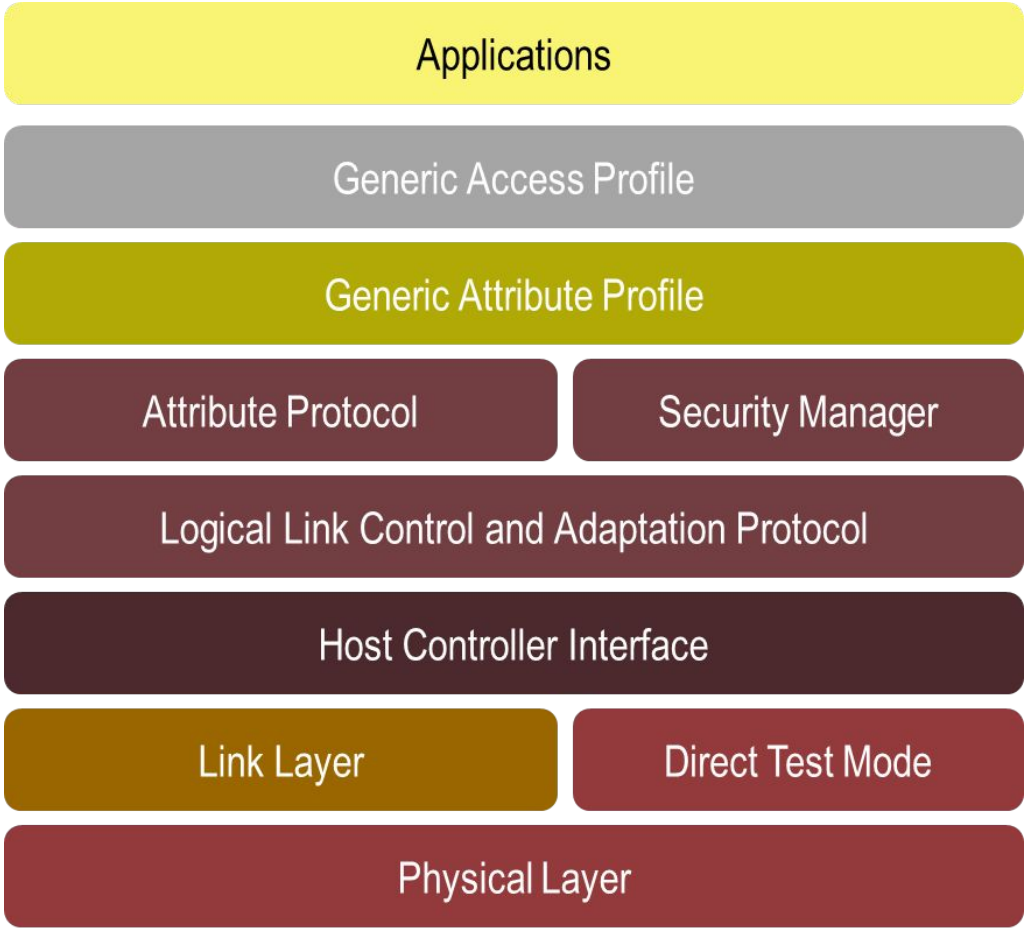


- SYS/BIOS Scheduler for multi-threaded event handling
- Main functions run from Tasks; called by SWIs and HWIs
- Power down mode can be planned or accidental

BLE Architecture

Key Features:

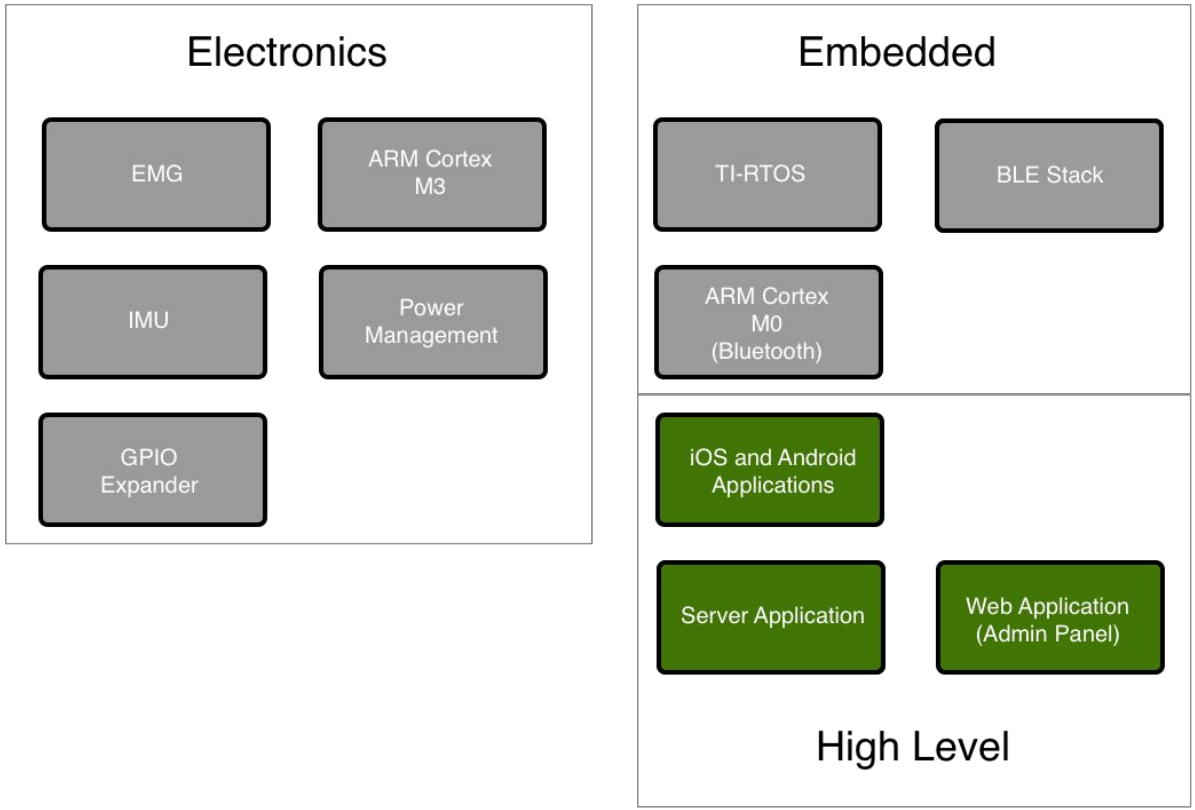
- Utilize wireless data
- Defines the general topology of the BLE network stack
- Describes in detail how attributes (data) are transferred once devices have a dedicated connection
- Allows for reads and/or writes to certain attributes exposed in a non-complex, low-power manner
- Internal interpretation of data to/from HCI
- Manages main Controller types and generic host information
- Transports Bluetooth packets between devices on the piconet (connection)
- The actual device hardware



Simplified Generation of BLE Profile

1. Have data needed to be sent through BLE connection
2. Decide how you want that data packaged (Boolean, uint8, etc.)
3. After making a general profile, add a service with a characteristic to fit your data (BDS a plus)
4. Implement this service in code as a library called by your main function
5. Add any relevant handling of your data (pre/post processing)
6. Turn on your host device and pair with BLE profile

High Level Software



Mobile Application

- Remote firmware updates
- User selected gestures
- LED color changer
- Data logging, pushed to server
- Diagnostics
- Real time communication with Limbitless Solution
- Social Networking

iOS vs Android

iOS

- Is a phone? true

Android

- Is a phone? true

iOS and Android (Love is Love)

Developing for both platforms

- Facebook’s React Native
- Functional and Declarative UI
- State management with Redux
- Side Effect handling with Redux-Saga
- ~80% Code reuse

State Management

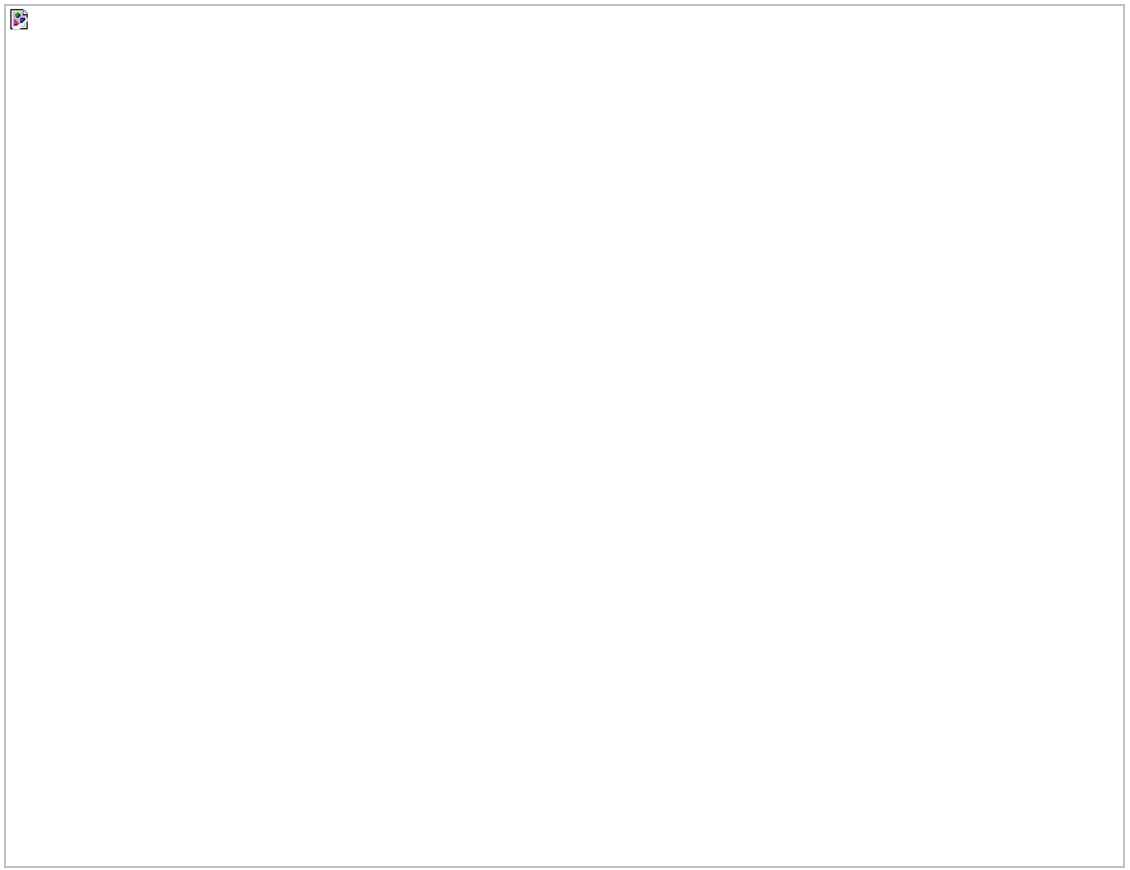
```
15 const initialState = fromJS({
16   isScanning: false,
17   connectedDevice: false,
18   availableDevices: [],
19 });
20
21 function deviceScreenReducer(state = initialState, action) {
22   switch (action.type) {
23     case START_SCANNING:
24       return state.set('isScanning', true);
25     case STOP_SCANNING:
26       return state.set('isScanning', false);
27     case SET_CONNECTED_DEVICE:
28       return state.set('connectedDevice', action.payload);
29     case ADD_DEVICE:
30       return state.update(
31         'availableDevices',
32         List(),
33         list => list.push(action.payload)
34       );
35     default:
36       return state;
37   }
38 }
```

Sagas - What the Fork?

```
function createBleChannel() {
  return eventChannel(emitter => {
    const events = NativeAppEventEmitter.addListener(
      'BleManagerDiscoverPeripheral',
      (data) => {
        emitter(data);
      }
    );

    BleManager.scan([], 5, false)
      .catch((err) => {
        console.log('*** ble error ***', err);
        emitter(END);
      });

    return () => {
      BleManager.stopScan();
      events.remove();
    }
  });
}

export function* closeChannelLater(channel) {
  // Debounce the same amount of time as BLE scan
  yield call(delay, 5000);
  channel.close();
  yield put(stopScanning());
}
```

```
export function* startScanning() {
  const bleChannel = yield call(createBleChannel);
  yield fork(closeChannelLater, bleChannel);

  while(true) {
    const payload = yield take(bleChannel);
    yield put(addDevice(payload));
  }
}

export function* watchScanRequest() {
  while(true) {
    yield take(START_SCANNING);
    yield fork(startScanning);
  }
}
```

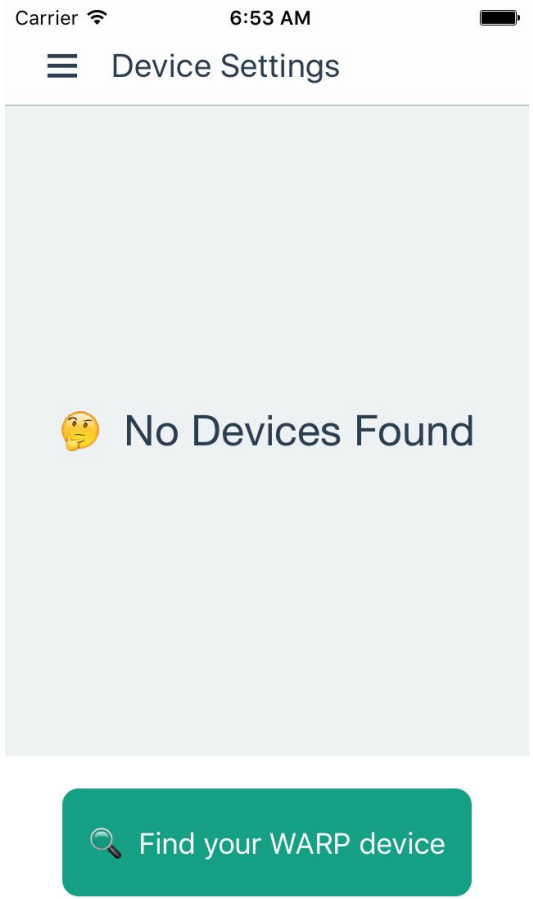
Layered Software Architecture

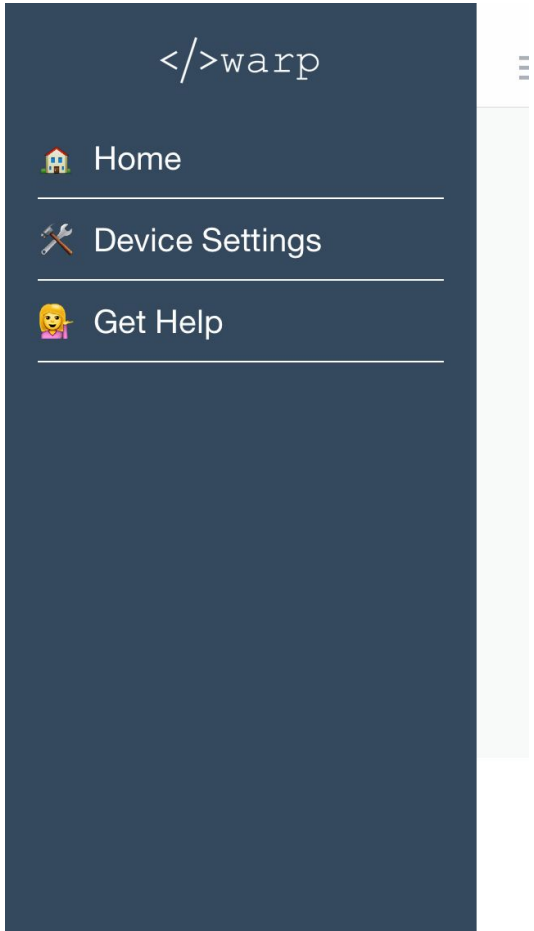


State Relation Flowchart

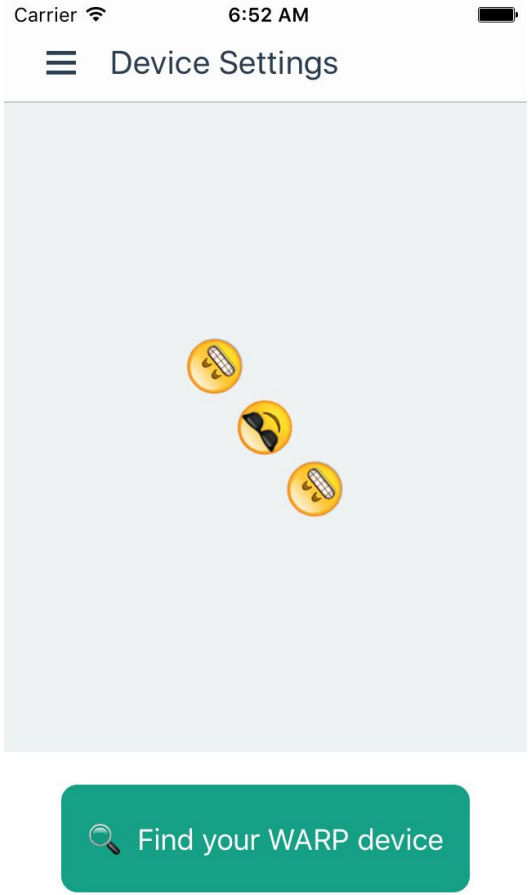
State Relation Flowchart

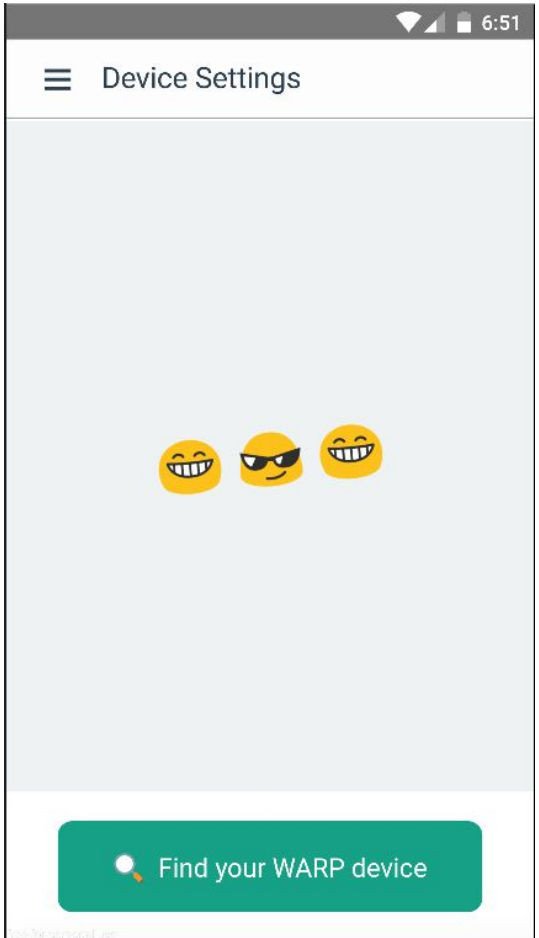
GUI pics


Carrier
6:53 AM
Device Settings
No Devices Found
Find your WARP device

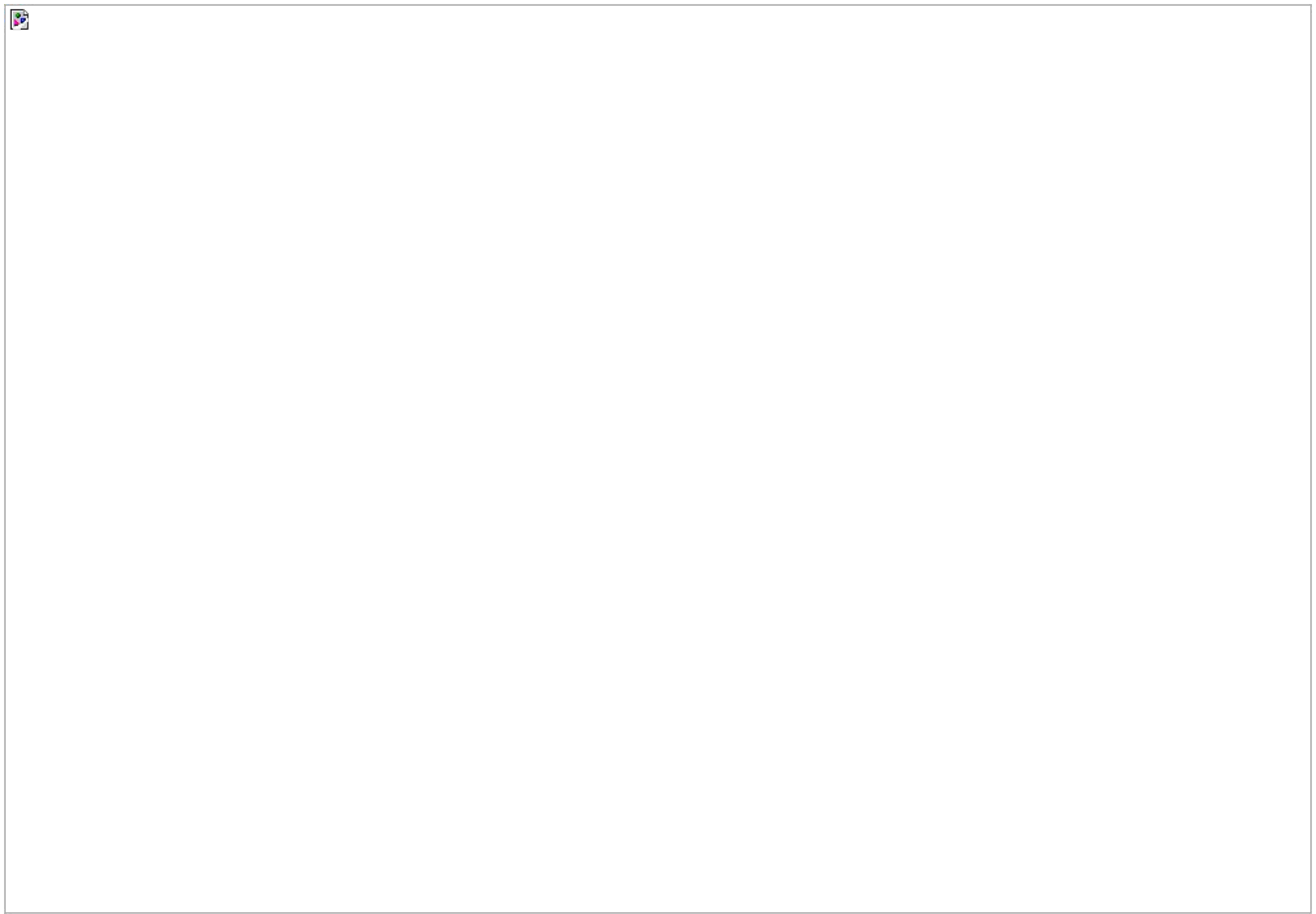

warp
Home
Device Settings
Get Help

GUI pics


Carrier
6:52 AM
Device Settings
Find your WARP device


6:51
Device Settings
Find your WARP device

Entity Relationship Diagram



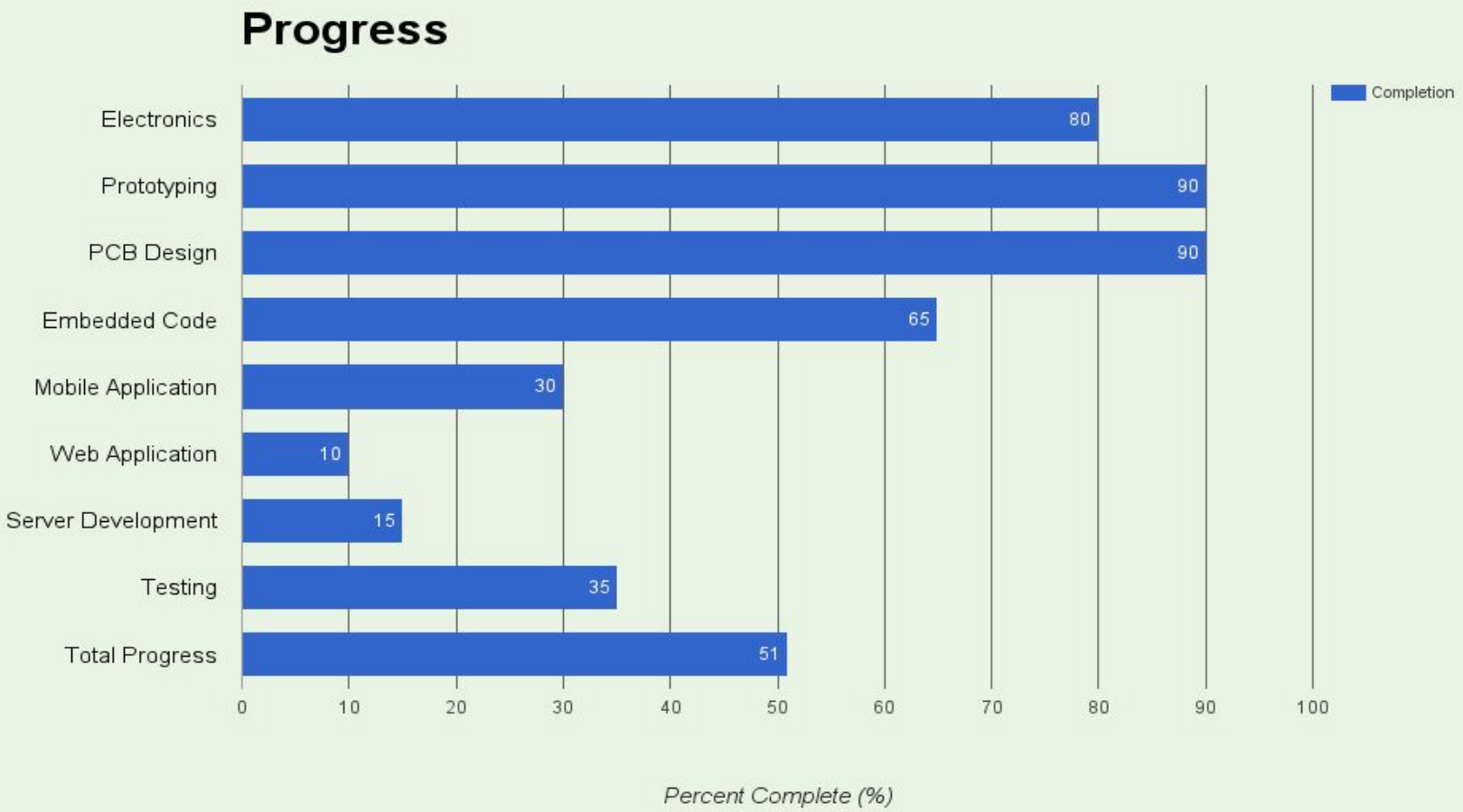
Administrative Content

Work Distribution

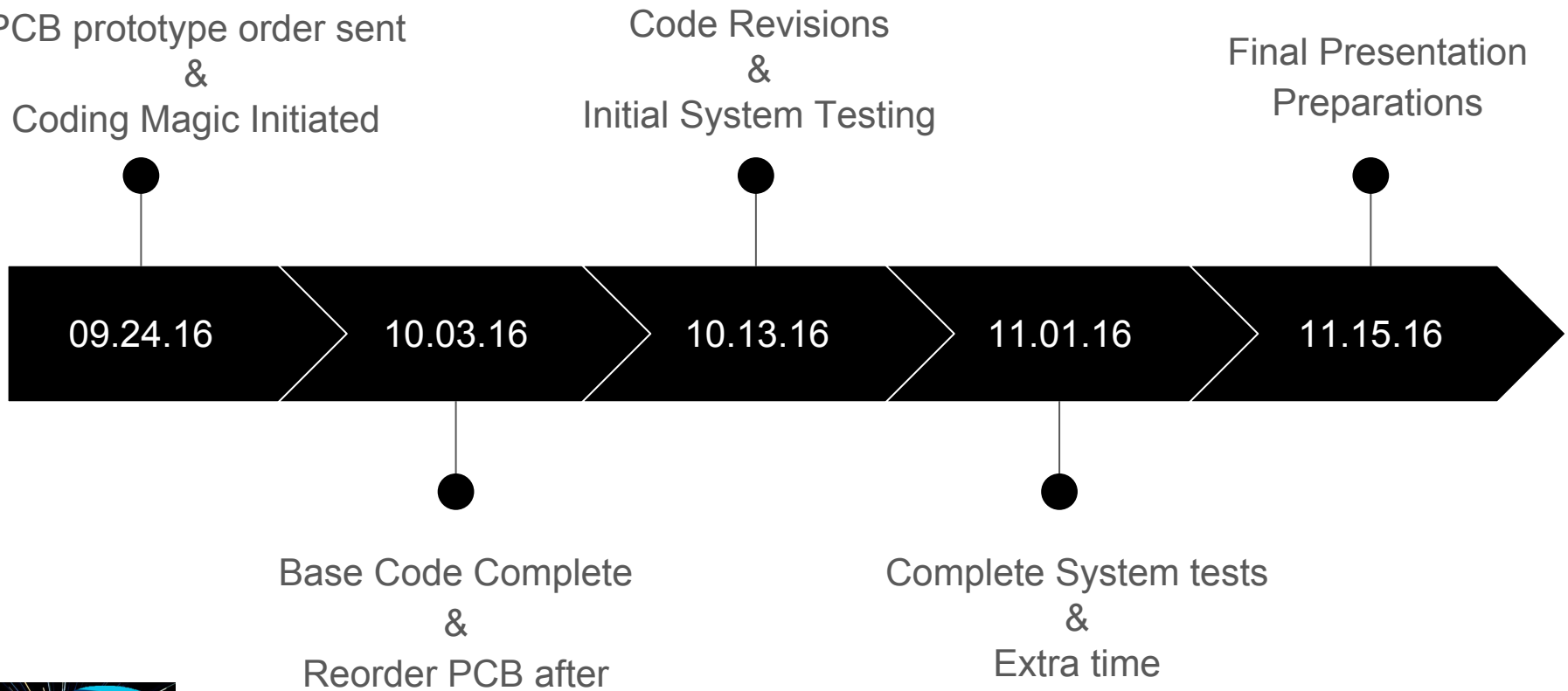
	Electronics	*Embedded Software (TI-RTOS)*	*Embedded Software (BLE-Stack)*	*Mobile Software Development*	*Server Development*
Daniel Mor	**Lead**	**Co-Lead**	**2nd**	**2nd**	
Niko Tubach	**2nd**	**Co-Lead**	**Lead**	**2nd**	
Brandon Ashley			**2nd**	**Lead**	**Lead**

Budget

Part Name	Manufacturer	Part Number	Quantity	Unit Cost	Total Cost
SaBLE-x (Trace Antenna)	LSR	450-0119	1	$16.52	$16.52
Push-Button	Panasonic	EVQ-PNF04M	1	$0.72	$0.72
1MB Flash Memory	Macronix	MX25R8035FM1IL0	1	$0.63	$0.63
Accelerometer / Gyroscope	ST	LSM6DS3	1	$3.93	$3.93
GPIO Expander	Semtech Corp	SX1509BIULTRT	1	$2.60	$2.60
RGB LED	Broadcom Limited	ASMB-TTB0-0A3A2	2	$1.31	$2.62
Voltage Inverter	TI	TL7660CDGKR	1	$1.43	$1.43
Instrumentation Op-Amp	TI	INA826AIDGKR	1	$3.01	$3.01
Quad Op-Amp	TI	LMV614MTX/NOPB	1	$0.92	$0.92
Digital Potentiometer	TI	TPL0102-100RUCR	1	$1.76	$1.76
3.5mm Jack	CUI Inc.	SJ-3524-SMT-TR	1	$1.37	$1.37
High Power Voltage Regulator	TI	TPS62130RGTR	1	$2.93	$2.93
Logic Level Voltage Regulator	TI	TPS62745DSSR	1	$2.6	$2.60
Male JST Connector	JST	BM02B-GHS-TBT	1	$0.43	$0.43
PCB Creation and Part Placement					$40
	Non-Discrete Total Cost				**$81.47**
	Total Cost				**$103.36**


Progress
Completion
Electronics
Prototyping
PCB Design
Embedded Code
Mobile Application
Web Application
Server Development
Testing
Total Progress
80
90
90
65
30
10
15
35
51
0
10
20
30
40
50
60
70
80
90
100
Percent Complete (%)

Future Dates



Challenges

Electronics

- **Offering improved capabilities at a reduced size and lower price**
- **High speed PCB layout**
- **Mix of sensitive analog and digital components in close proximity**

Embedded Software

- **Steep Learning Curve for TI-RTOS & BLE Stack**
- **High level Software Development Concepts**
- **Size Considerations for BLE Profile using OAD**

High-Level Software

- **Creating a UI that is cross-platform friendly**
- **Making native calls to the Bluetooth Module without memory leaks**
- **DevOps**

Q & A