Bluetooth 4.0 Low Energy
+
CoreBluetooth

Alicia M. F. Key

amfkey@gmail.com

akey7.com/bluetooth-low-energy

github.com/akey7/bluetooth-low-energy

Hardware: **Demo**

- For demonstration purposes
- Does nothing useful
- Features
  - Motor that spins
  - Light sensor
  - LED that turns on and off

Intro: **Possible Interfaces**

- AirPlay
- iPhone Access Protocol
- Lightning
- Audio jack
- Bluetooth
- Apple MFI Program

https://developer.apple.com/programs/mfi/

Intro: **MFi Program**

- Benefits
  - Hardware and documentation
  - Use of logos
  - Technical support from Apple
- Requirements
  - Requires legal entity
  - Credit check
  - NDA

https://developer.apple.com/programs/mfi/

Intro: **MFi, Bluetooth LE, WWDC**

- But *MFi isn’t required for Bluetooth LE*
- And it is what Apple pushes at WWDC
  - WWDC 2013 Session 700
  - WWDC 2012 Session 705
  - WWDC 2013 Session 703

http://mfi.apple.com/faqs#4-2

Bluetooth: **Overview**

- Originally a wireless replacement for RS-232
- Evolved through versions up to 4.0
- Managed by Bluetooth SIG
- Operates in Industrial, Scientific, and Medical band between 2.400 GHz and 2.480 GHz
- Profiles for major use cases

http://en.wikipedia.org/wiki/Bluetooth

Bluetooth: **BLE vs. Classic** (1 of 2)

- I will abbreviate as Bluetooth 4.0 Low Energy as *BLE*
- BLE similar range, lower power consumption than Classic
- BLE and Classic share same 2.400 GHz antenna
- Different modulation schemes.

http://en.wikipedia.org/wiki/Bluetooth_low_energy
http://www.bluetooth.com/Pages/Smart-Logos.aspx
https://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=229737

Bluetooth: **BLE vs. Classic** (2 of 2)

- BLE only = single mode
- BLE + Bluetooth Classic = dual mode
- Bluetooth SMART = single mode
- Bluetooth SMART Ready = dual mode
- Rest of talk about BLE only

http://en.wikipedia.org/wiki/Bluetooth_low_energy
http://www.bluetooth.com/Pages/Smart-Logos.aspx
https://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=229737

BLE: **Use Cases**

- Health care
- Sports & Fitness
- Proximity Detection
- Time
- Home automation
- Security

http://en.wikipedia.org/wiki/Bluetooth_low_energy
http://www.bluetooth.com/Pages/Smart-Logos.aspx
WWDC 2012 703, WWDC 2013 700

BLE: **Firmware / Software**

- Little device called *Peripheral*
- Main device called *Central*
- Peripherals:
  - 1 or more *services*
  - Services have 1 or more *characterisitcs*
- Discover and use peripherals, services, characteristics from *CoreBluetooth*

WWDC 2012 705

BLE: **Hardware for Demo**

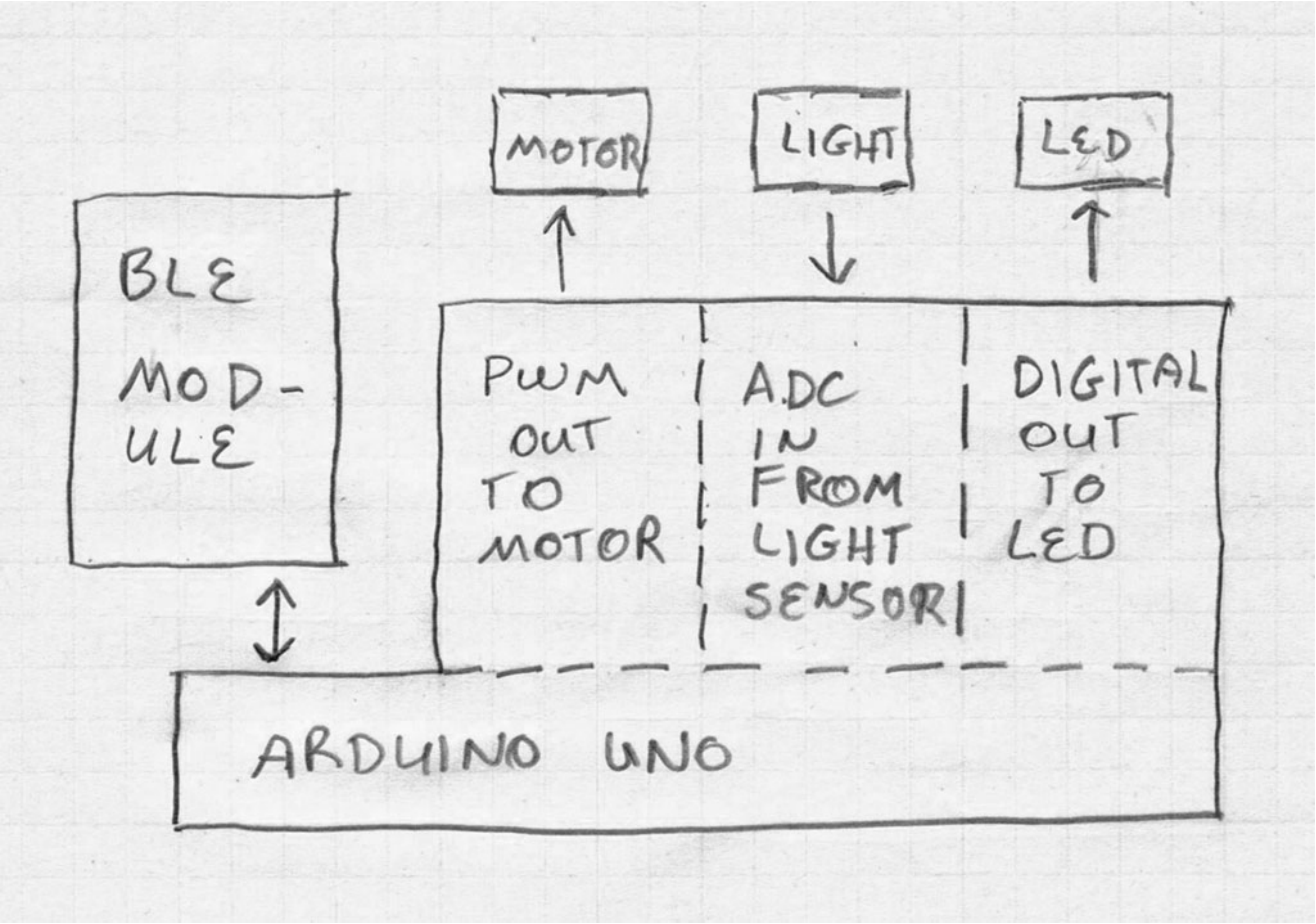
- Demo for those that want to make their own devices.
- Does nothing useful
- Not meant to set new EE standards.
- Uses Arduino but not strictly an Arduino project.
- I call it the *Gadget*

http://akey7.com/bluetooth-low-energy

Hardware: **Block Diagram** (1 of 2)

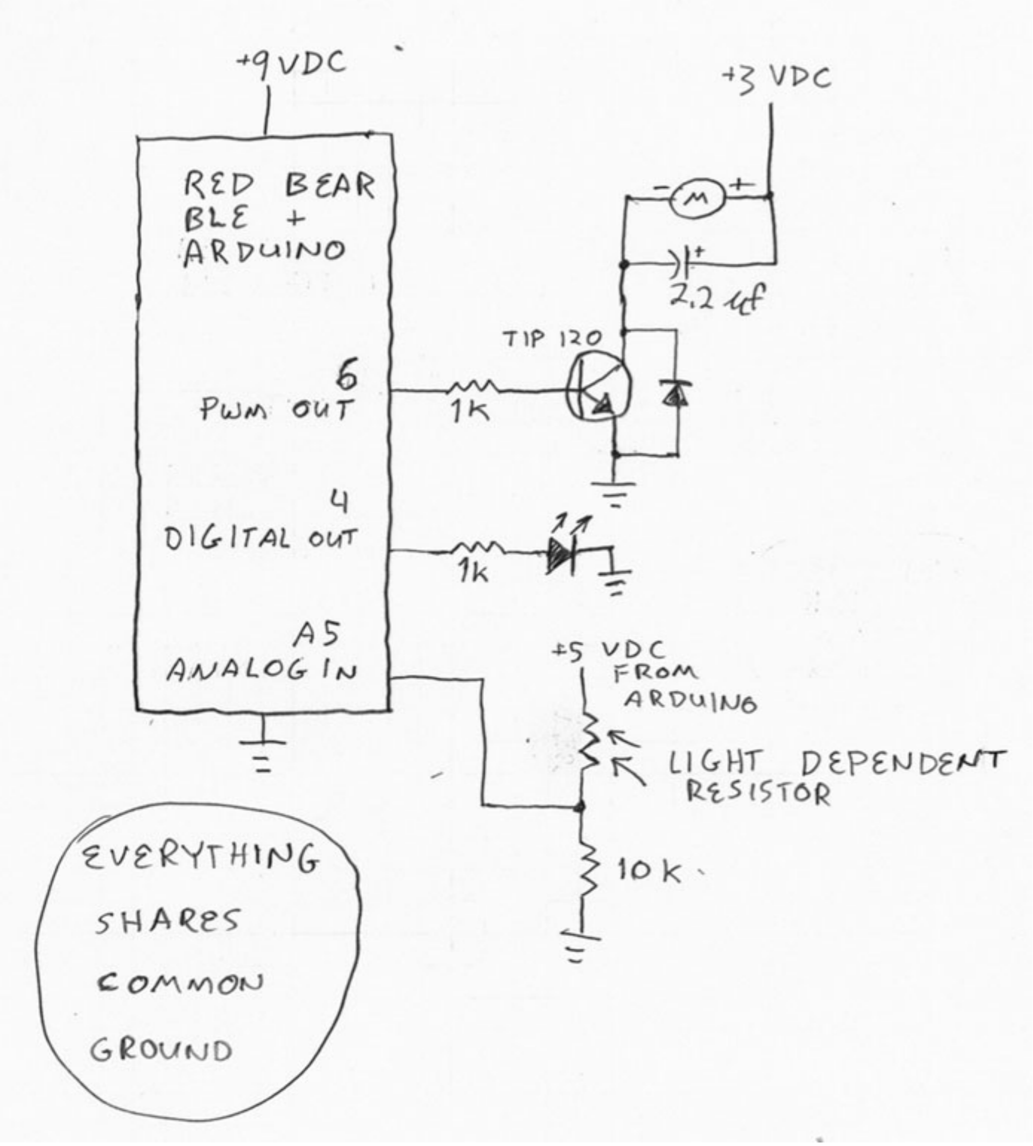
- Arduino Uno
- Red Bear Labs BLE
- “Services”:
  - Motor
  - LED to toggle
  - Light sensor

Hardware: **Block Diagram** (2 of 2)



Hardware: **Schematic**

- Voltage divider (outputs 0-1024)
- PWM motor driver (accepts 0-256)
- LED (on or off)
- *Issue*: Common ground voltage drop?



show assembly images ->

Firmware / Software: **Protocol**

- 3-byte or 1-byte I/O, X = ignored, V = variable

action	direction	bytes
LED	iOS -> Gadget	0x01, 0x00, X = off 0x01, 0x01, X = on
Motor (0-255)	iOS -> Gadget	0x02,V, X V = PWM duty cycle, speed
Light sensor (0-1023)	Gadget -> iOS	0x0B, V, V V = bytes of analog value
Reset transceiver	iOS -> Gadget	0x01, X, X *On different characteristic*

Firmware: **Show & Tell**

- Firmware:
  - Watchdog
  - setup()
  - loop()
- *http://github.com/akey7/bluetooth-low-energy*
- Show code

Software: **This Implementation**

- GadgetController
- GadgetControllerDelegate
- ViewController
- NIB
- Red Bear Lab quirks
  - Services
  - Characteristics
  - Same UUID (UUIDs explained later)

Show code ->

Software: **CoreBluetooth (1 of 4)**

- iOS Side, the Central
  - CBCentralManager
  - CBCentralManagerDelegate
- Gadget side, the Peripheral
  - CBPeripheral
  - CBPeripheralDelegate
  - CBService
  - CBCharacteristic
  - CBUUID (NSUUID in iOS 7)
  - Important value, but not a class: RSSI
- Why iOS 6 for this implemetation?

Asynchronous!

Software: **CoreBluetooth (2 of 4)**

- Discover *peripherals*
- *CBCentralManager, CBCentralManagerDelegate*
  - Manager: *scanForPeripheralsWithServices*
  - Delegate: *didDiscoverPeripheral*
  - Manager: *stopScan*
  - Manager: *connectPeripheral*
  - Delegate: *didConnectPeripheral*

Show code ->

Software: **CoreBluetooth (3 of 4)**

- Discover *services*
- *CBPeripheral, CBPeripheralDelegate*
  - Peripheral: *discoverServices*
  - Delegate: *didDiscoverServices*
- Discover *characterisitics*
- *CBCharacteristic*
  - Peripheral: *disocverCharacterisitics:forService*
  - Delegate: *didDisocverCharacteristics*

Show code ->

Software: **CoreBluetooth (4 of 4)**

- Read from and write to *characterisitics*
- *CBPeripheral, CBPeripheralDelegate, CBCharacteristic*
- Polling (ask the peripheral)
  - Peripheral: *readValueForCharacteristic*
  - Delegate: *didUpdateValueForCharacteristic*
- Notifications (wait for peripheral to send)
  - Peripheral: *setNotifyValue:forCharacterisitic*
  - Delegate: *didUpdateValueForCharacteristic*

Show code ->

Software: **API, other operations**

- Other operations not covered
  - Reconnection (as specific CoreBluetooth operation)
  - Some error handling because my hardware always works ;) (explained in API docs)
- Changes from iOS 6 to iOS 7
  - Peripheral retrieval and reconnection (didn't cover in this talk)
  - NSUUID, not CFUUIDRef

WWDC 2013 Session 703

Resources

- WWDC **2013** Session 700
  - Start at 21:15, go to 40:20 for BLE
- WWDC **2012** 705
  - Best intro to API, including things I didn't cover. And it has a heart rate monitor!
- WWDC **2013** 703
  - Review and goes over iOS 7 changes
- Wikipedia
- BlueTooth SIG Specification Version 4.0
  - *https://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=229737*

Thanks!

Alicia M. F. Key

amfkey@gmail.com

akey7.com/bluetooth-low-energy

github.com/akey7/bluetooth-low-energy