Microcontoller + Wireless Options – some of Jon's favorites

(Last modified: 28 Jan 2019)

The list below summarizes some commercially available options that are likely viable for your capstone project. There are many, many options out there; new modules pop up every few months. This list was selected based on my experience working with many of these over the past 3-4 years. Another strong factor in making this list was the data rates required for your projects. Everyone has a fairly low data transmission rate – maybe a few bytes sent every second (compared to high speed which is ~10 kB/s). Without further ado:

Microcontroller Options

1. Teensy. Pretty much an Arduino on steroids—faster computation, more peripheral ports, better ADC, on-board microSD storage, smaller physical footprint, etc. Programs interface is the same as Arduino IDE, using a small library add-on called Teensyduino. No built-in wireless capability; requires additional wireless module, if needed. Note there are several different models – Teensy LC, Teensy 3.2 and Teensy 3.6. Links:

- https://www.pjrc.com/teensy/ (Paul Stoffregen who developed the Teensy was one of the core contributors to the Arduino libraries; his website doesn't look awesome, but I assure you his product is)
- Teensyduino getting started
- https://www.adafruit.com/product/2756
- Teensy 3.6

2. Adafruit Feather Series: Adafruit has developed their own line of boards. These are brand new and look rock-soild. Adafruit products typically are. They are attractive because they are USB native (can plug into a computer, but also have wireless built in). There are many models in the Feather series, such as:

- https://www.adafruit.com/product/2829 (Straight from the horses mouth: *This is the Adafruit Feather 32u4 Bluefruit - our take on an 'all-in-one' Arduino-compatible + Bluetooth Low Energy with built in USB and battery charging. Its an Adafruit Feather 32u4 with a BTLE module, ready to rock!*)
- https://www.adafruit.com/product/2995 (MCU +BLE, high speed MO processor)
- https://www.adafruit.com/product/2796 (USB native, can add wireless module. Has a native microSD card for storage which can be great for testing and verification)

3. Light Blue Bean. All-in-one Microcontroller (MUC) integrated blue tooth low energy (BLE) module. Also features on on-board accelerometer and temperature read-out, which could be handy in some applications. Fairly easy to work with and program in my experience – though it is much more Mac friendly, as has been difficult to use with Windows because Windows 7 and prior lacked a built-in API for BLE (but that is starting to change). This is a wireless only module with over-the-air (OTA) programming; no wired configuration is possible.

- https://punchthrough.com/bean (scroll to bottom to see the Bean)
- https://www.adafruit.com/product/2732

4. ESP8266. All-in-one MCU + WiFi module, the darling of the maker world the past couple of years. Relatively easy to program, though historically documentation was not greatSparkfun and Adafruit both make nice breakout boards. Small, cheap, mostly just works. Note that WiFi makes this chip power hungry, so really only worth a look if you need WiFI high data rates.

- https://www.adafruit.com/product/2471
- https://www.adafruit.com/product/2821
- https://www.sparkfun.com/products/13804
- http://espressif.com/en/support/explore/get-started/esp8266/getting-started-guide

5. Particle Photon/Boron. Photon was one of the original kickstarter projects in MCU + WiFi that grew up, not blew up. The hardware and support ecosystem have really matured over the past couple of years. This module is meant to be programmed OTA, so it has the same pro/con as with the BlueBean (see above). I haven't used this module recently, but many people report is performance is rock solid. Documentation excellent too. All that said, this is a power hungry WiFI module that will drain a battery relatively quickly. The Boron offers LTE cellular network connectivity

- https://docs.particle.io/guide/getting-started/start/photon/
- https://www.sparkfun.com/products/13774
- Particle Photon module
- Particle Boron module

6. Arduino boards: The original classic. There are many variants of Arduino boards: Uno, Duemilanove, Micro, etc. Choose one that is small, has ready wireless add-on capability, such as the ones below. See the entire selection guide here: https://www.sparkfun.com/standard_arduino_comparison_guide

- https://www.sparkfun.com/products/12587 (can add-on bluetooth serial module easily)

- https://www.sparkfun.com/products/11520 (kind of expensive, but has XBee ready socket)

Wireless Modules Comparison

Bluetooth vs. BLE vs. WiFi vs. Xbee – which to choose? The basic tradeoff is data rate and transmission range vs. power consumption, as summarized in Table 1 below. You should also know at the outset that Bluetooth has 2 subtypes – "classic" and "low energy" (BLE). The BLE modules are likely much more appropriate for projects with low-power requirements to extend battery life. You should also consider ease of use/programming: My experience, in general, Bluetooth and BLE are the easiest, followed by WiFi and Xbee.

Name	Bluetooth Classic	Bluetooth 4.0 Low Energy (BLE)	ZigBee	WiFi
IEEE Standard	802.15.1	802.15.1	802.15.4	802.11 (a, b, g, n)
Frequency (GHz)	2.4	2.4	0.868, 0.915, 2.4	2.4 and 5
Maximum raw bit rate (Mbps)	1-3	1	0.250	11 (b), 54 (g), 600 (n)
Typical data throughput (Mbps)	0.7-2.1	0.27	0.2	7 (b), 25 (g), 150 (n)
Maximum (Outdoor) Range (Meters)	10 (class 2), 100 (class 1)	50	10-100	100-250
Relative Power Consumption	Medium	Very low	Very low	High
Example Battery Life	Days	Months to years	Months to years	Hours
Network Size	7	Undefined	64,000+	255

Table 1. Summary comparison of wireless protocols. From: https://learn.sparkfun.com/tutorials/bluetooth-basics/all

Further elaborations on differences between these can be found on this very intelligent answer on stackexachange. Further elaborations on the difference between Bluetooth Classic vs. BLE can be found here; Comparison of BLE vs. Zigbee are summarized here; BLE vs. WiFi summary comparison here. Main properties/trade-offs of each wireless communication are summarized below. Suggested modules are included on the following page.

Bluetooth**:**
Medium data rate and medium power consumption. IEEE standard has different device profiles to enable interoperability between devices. 'Pairing' type network however, so not useful for sensor networks, but good for controlling devices using a laptop or phone which usually have Bluetooth on board. Communicates with Arduino via serial RX and TX pins. link. Getting started with classic Bluetooth is really easy. For example, follow this hook-up guide from sparkfun. Note that you can replace the Sparkfun BlueSMiRF with just about any other standard Bluetooth module. Take particular note of the pass-through sketch to see how you can communicate with your Bluetooth module.

Bluetooth Low Energy:
Lower data rate and very low power. These RF chips can run for years on an AA battery and so are used for things like heart rate sensors. The data rate is quite low so only good for transferring a small amount of data such as sensor readings. Most devices are setup as peripherals and can only connect to one 'central' device, so no good for sensor networks. Bluetooth low energy is supported by the latest mobile phones and laptops however. Useful to enable control of a device using a phone or laptop, for example a LED lamp.

XBee:
Low data rate (250kbit/s) and low power consumption. Used to make mesh-type sensor networks; each XBee device can communication with each other, and *through* each other via the mesh to devices that are out of range. Used for networks of devices, particular home automation and sensors. link.

WiFi:
High data rate (54Mbit/s +) but also high power consumption. Used when you need to connect directly to the internet, such as an internet-of-things device, and have an external power source.

Nordic Semiconductor nRF24L01 / RFM12B / RFM22B(SI4432)
Popular proprietary transceivers. They can connect via SPI and have a lot of fancy features like low power modes, multiple channels, channel hopping, frequency calibration, CRC, retransmit and so on. The nRF24L01 from Nordic Semiconductor link operates in the 2.4GHz band, others use the ISM band 433/470/868/915Mhz. These are todays state of the art highly integrated and low cost hardware for reliable telemetry and consumer applications. The range is between 10m and 150m. The SI4432/RFM22B has the range of about 1.5Km.

Potential/Suggested Modules for Integrating into Electronics Project

BLE modules

- https://www.adafruit.com/product/2479 (easy to use, just plugs into Tx/Rx on Arduino or Teensy)
- https://www.adafruit.com/product/1697 (more power efficient but a bit trickier to implement)

Bluetooth classic modules

RN-42: (classic, rock solid, use this in my research lab, a little bit on the expensive side and a little bit bulky, about the size of a stick of gum. Other potential downside is 50 mA max power consumption is not so power efficient)

https://www.sparkfun.com/products/12576

HC-06: the old-school maker classic. It cheap, it generally works, just a bit finicky and not as robust as RN-42.

https://www.amazon.com/Pass-Through-Communication-Compatible-Atomic-Market/dp/B00TNOO438

XBee modules:

- https://www.sparkfun.com/products/11215 paired with https://www.sparkfun.com/products/11812
- https://www.adafruit.com/product/247 paired with https://www.adafruit.com/product/247