I know what I want to say but I have writer’s block. I don’t intend this to have dot points in its final form; I’m just a shitty writer. I don’t expect to require this to be finished until Raymond returns.

Arduino vs Pi

**Overview**

**Main**

/\* This is intended to be a persuasive argument as to why we are choosing whatever we choose. Sorry for presupposing pi, we can change it if we change our minds. \*/

/\* Mike edits \*/

The guiding principle of the OARKit project, is to lower the barrier of entry to the experimentation of robotics. Mainly, to give access to robotics to people with minimal programming experience. Currently our aim is to try and make the system do as much as possible so the user doesn’t have to.

With this in mind, the hardware we use will affect what is possible. The current choices are the popular hobbyist platforms Ardiuno and the Raspberry Pi, these are the forerunners as they’re both popular and have a community with goals similar to OARKit. Namely, providing cheap, easy to use platforms to enable easy access to people with low specialised knowledge.

## Arduino

The Arduino is an electronics platform based on the Atmel system on a chip processors, it’s largely self-contained, needing minimal supporting circuitry to be operational. The Arduino offers a number of different base systems with various performance and features. Various modules, called shields, are available that offer further features such as WiFi, ethernet, radio communications, GPS, etc. These modules can be stacked, and the Arduino development environment offer a set of libraries so that software can be easily developed.

## Raspberry Pi

The Raspberry Pi is a credit-card sized computer, it is comparable to a low-powered desktop computer and has provision for attaching a monitor and a keyboard. The goal behind the Raspberry Pi is to lower the barrier of entry for experimentation into computing.

/\* It feels clumsy ending this description here, but it that’s what it is. \*/

## Comparison

The Raspberry Pi is more powerful. The Arduino platform runs an Atmel 8-bit microcontroller, while the Raspberry Pi uses a 32-bit microprocessor. The Arduino’s microcontroller runs from about 8MHz to 16MHz, while the Raspberry Pi can be clocked from 700Mhz to 1GHz. Available RAM on both systems also follows the same trend, 2-8k for the Arduino, up to 1GB for the Raspberry PI.

There are exceptions to this trend for Arduino. Boards like the Arduino Due, and Arduino Yún have higher clock rates (84MHz and 400 MHz respectively) , but they have far lower RAM sizes than the Raspberry Pi. Arduino also have the Tre system, that is like the Pi, and can run Linux, but it is not yet available.

Comparing the price of both is difficult, because there are so many Arduino systems, and none approach the performance of the Pi. The Arduino Uno Rev 3 costs approximately $30, compared to $40 of the latest Raspberry Pi 2, the Raspberry Pi still has a far higher performance to price ratio.

Why would anyone pick an Arduino over the Pi then?

* Less power
* Simplier
* Anything else?
* We are an isolated sub-project of OARKit but we should not forget their goals. We need to take into account cost, replaceability and re-usability. The arduino UNO Rev3 costs approximately $30 compared to about $40 of the Pi. This saving is nearly insignificant when it is viewed alongside the fact that a single Dynamixel AX-12 servo costs ~$50 but a saving is still a saving. The real comparison occurs when the ratio of cost to performance is considered for both the Pi and the Arduino. The Pi wins this contest by a sizeable margin /\* probably, I don’t actually know \*/. Obviously, in embedded applications all of that performance may not be needed but we believe that we can properly exploit this performance to provide higher level interfaces to the hardware than is possible on the arduino.
* Arduino has many versions, pi very few. May be inconsistencies across versions. Might have to look into the standards specified for the arduinos.
* Pi has ROS
  + ROS is \*the\* movement in open source robotics right now
  + Easiest environment for our code to be maintained/extended
  + If our code uses ROS then it is an organic process for a student to begin using other ROS features for additional functionality
  + I’m pretty sure that our code will be irrelevant in a few years if we don’t use ROS.
* How they interface with components
  + Presumably arduino is slightly better when it comes to this
  + Depends on our scope. If only dynamixel servos then we may be fine.
* Built-in hardware
  + Arduino micro pro requires that RF24 wireless communications adapter whereas a PI has appropriate communications solutions for that built-in.  
    /\* Note: I’m sure that the RF24 is a radio transmitter module, this is not the same as wireless (802.11) communications. \*/

**Significant Tradeoffs**

Tradeoff low overhead on Arduino for more languages/libraries on Pi.

Tradeoff better understanding of hardware on Arduino for simpler to program software on Pi.

**Pi**

**Advantages**

**Disadvantages**

* It may be hard for us to hide away the behemoth overhead of ROS and still have the student understand what is really happening.
* The student is taken further away from the hardware and I feel like Raymond’s overall goal is more hardware-oriented.

**Arduino**

**Advantages**

* Been around longer. More likely that people still have old ones lying around. That old arduino may be significantly different to the one specified in the kit.

**Disadvantages**