**Arduino**

An Arduino is a small microcontroller based kit that is used to build and develop digital and interactive objects primarily to sense and alter the physical world. The first Arduino board was built in 2005 and aimed to be a low cost solution for hobbyists and professionals to experiment with creating devices that interact with or alter the real world using sensors and actuators.

The boards use either various forms of 8-bit Atmel AVR microcontrollers or 32-bit Atmel ARM processors at their core. They provide a set of digital and analogue I/O pins that can be used to connect to various controllers, sensors, or extension parts. Some models include USB so that they can load programs from various personal computers.

They are programmed using Arduino’s very own IDE which includes support for C, C++ and Java programming languages. Programs written in this IDE can be uploaded and stored onto the Arduino’s on-chip flash memory. This simplifies development by allowing the use of any general purpose machine instead of an external system to program.

All of the software systems and hardware used by Arduino is open source, and as such the circuitry can be replicated or have its capabilities expanded by an experienced circuit designer. Its software libraries can be expanded via the use of various C++ libraries.

Arduino boards have been used by people all over the globe to create gadgets and systems, and are simple enough that both professionals and beginners to access its features. The capabilities of the system are limited only but the users experience with the system and imagination. They have been used to create things as complex as bipedal robots, world clocks, remote control lawnmowers and even an entire gardening ecosystem.

The Arduino supports and has access to almost every type of sensor available including thermostats and accelerometers, as well as a variety of motors including servos. As long as the power supply is strong enough, the project’s complexity is not limited by any hardware.

The Arduino is much more lightweight in terms of hardware when compared to the Raspberry Pi, and the nature of our project does not require the raw power a Raspberry Pi provides. As such we decided that an Arduino board would be used exclusively over the Raspberry Pi.

|  |  |
| --- | --- |
| Microcontroller | [ATmega2560](http://www.atmel.com/Images/Atmel-2549-8-bit-AVR-Microcontroller-ATmega640-1280-1281-2560-2561_datasheet.pdf) |
| Operating Voltage | 5V |
| Input Voltage (recommended) | 7-12V |
| Input Voltage (limit) | 6-20V |
| Digital I/O Pins | 54 (of which 15 provide PWM output) |
| Analog Input Pins | 16 |
| DC Current per I/O Pin | 20 mA |
| DC Current for 3.3V Pin | 50 mA |
| Flash Memory | 256 KB of which 8 KB used by bootloader |
| SRAM | 8 KB |
| EEPROM | 4 KB |
| Clock Speed | 16 MHz |
| Length | 101.52 mm |
| Width | 53.3 mm |
| Weight | 37 g |

Raspberry Pi

The Raspberry Pi is a small low budget computing device that allows users of all experience levels to delve into the world of computing. It is essentially just a miniature version of any other personal computer, and has the same capabilities such as plugging it into a monitor, TV or other USB controlled devices.

It can also do almost anything a PC can do in regards to software as well. A user can load an operating system like Linux onto it then use it like a normal PC, allowing the browsing of the web, and even playing games so long as the hardware is good enough.

It was developed with the intention of providing a low cost solution for educational institutes to teach basic computer science.

The Raspberry Pi is mainly limited by its hardware in terms of comparisons to a regular PC. The processor runs at around 600-1000 MHz, which is equivalent to the Intel Pentium processors, and the GPU is comparable to that of the original Xbox. Given the size, these statistics are very impressive.

Since most of the hardware, such as the memory, is built into the board, it cannot be replaced or added to. This limitation is due to the foundation’s goal of providing the device at as low a cost as possible while also maintaining a decent set of hardware capabilities.

The power supply used in the Raspberry Pi is independent of the system, and thus it depends on the application that dictates how much power will be needed. Around 2.5A is recommended by the foundation if you want to make sure you can use it to maximum efficiency.

We initially conceptualized using the Raspberry Pi in order to communicate with the robot arm via a phone interface using Bluetooth or Wi-Fi. We discovered after our initial research, however, that Arduino boards have components available to

Robot Armunio Considerations Dealing with the Above

Multiple Moving Parts

The robotic arm is going to have multiple servo motors controlling each movable part of the arm. There are two main things to take into consideration when thinking about this part of the specification.

1. Control Scheme. Since the robot is more complicated than just an X and Y axis, special consideration needs to be given to the way the user will control the robot. An interface that clearly labels the part and how it is controlled is crucial. Some ways this could be done are through a standard keyboard input scheme, accompanied by a graphic that shows which key controls which part.

We also thought of the idea that we could use a mobile interface and control it either via

Wi-Fi or Bluetooth, both of which are easily accessible by a Raspberry Pi or an Arduino. The interface would include a basic graphic of the robotic arm, with selectable joints that could then be controlled either by dragging or tapping arrow keys.

1. Complicated wiring. Having up to five servo motors as well as a Bluetooth or Wi-Fi adaptors may become complicated in the hardware aspects of the project. Luckily this whole worry is alleviated by the fact that Arduino has servos specifically developed for use by the Arduino.

This means that at most we will need to extend some of the servo wires in order to reach the desired part of the arm.

Using a Phone as a Controller

The robotic arm will have some sort of interface built into a mobile phone. This means that whatever hardware we use will need to have some way to communicate with the phone, via either Bluetooth or Wi-Fi.

1. Bluetooth – Bluetooth is a technology used to send data over short distances. Bluetooth uses a master and slave model, with the master device being able to send data to the slave. The roles can be switched if both devices agree, meaning that both linked devices can act as a receiver or transmitter. The limitation of this is that only one of the devices can send data to the other at the same time meaning it would be a less effective option if the specification required a lot of back and forth between the devices. There is also very little user configuration options as it is a simple symmetric connection between two devices
2. Wi-Fi – Wi-Fi is an alternative solution to Bluetooth and was designed as a replacement for high speed cabling in larger networks such as work offices. Wi-Fi is different to Bluetooth in that all traffic is centred on an access point from which all traffic is routed. Wi-Fi is also user configurable meaning it can be set up for several different purposes and supports much higher speeds than Bluetooth.

One of the main factors we took into consideration when making the decision between the two was cost factor. Wi-Fi transmitters and receivers are both quite expensive. We looked at what we needed – a (probably) simple message sent from the phone controller to the receiver on the Arduino board, so we made the decision that Bluetooth would be better for our needs. Using Wi-Fi would be too expensive and heavyweight for what we need.