



 **ENLIGHTENMENT**

Let's Get Beyond The Simple Stuff
And Head For The Nitty Gritty

- * You have learned some of the things that are available for your electronics pleasure.
- * That includes basic electronics and computers and software that are available for projects.
- * Other than making a LED flash, what is available and how do I choose.

* **INTRO**

- * We will start by looking at some basic computer concepts.
- * These concepts will not just help you with the project you choose but is the basis of all computers and how they operate.

* **UNDERSTANDING**

- * We talked about the Arduino and what is required to make the LED flash.
- * Basic electronics - understanding some of the components of the hardware and what Ohms law is.
- * Understanding these gives a good foundation toward the understanding of Microprocessors.

 **REVIEW**

- * Program

- * Operating System

- * Application (app)

- * Data

- * The above definitions help to understand the difference between Arduino operation and an OS like Windows)

*** SOFTWARE DEFINED**

- * A program (just a group of instructions) can do only one thing at a time such as flash an LED.
- * An Operating System is not a program per se but allows for applications to be run.
- * An application is a program that runs under an operating system
- * Examples (verbal Q and A)

* SOFTWARE DIFERENCES

- * A single program such as flashing the LED is fast. It runs as fast as the machine's CPU.
- * In general the program only has one thing in mind to do and therefore is NOT susceptible to outside influences, like hacking.
- * Small in size both physically and for memory.

* PROGRAM ADVANTAGES

- * Operating systems make (in theory) programming applications easier.
- * Treats all applications the same way by interfacing to low level hardware.
- * If an error occurs in the application, it can be reported to the user.
- * User interface (GUI) can be the same for many different programs. Ease of operation.
- * Good control of resources.

* OS ADVANTAGES

- * Programs can run forever. (infinite loop)
- * Resources all have to be programmed. (no high level calling of hardware like an OS)
- * Will only run on the specific CPU involved. (Compiled code is only good on one specific type of uP)
- * Not forgiving. (errors that are not picked up by the compiler can be hard to find)

* THE DARK SIDES

- * OS can be very complex.
- * Can inhibit operation even with a correct program.
- * Susceptible to outside influences (hacking etc.).
- * Slows down application execution.
- * Requires updating as time goes on.

*** THE DARK SIDES cont.**

- * That depends on the project
 - * If the project is simple and does only one thing then a Arduino or equiv. will work fine.
 - * If your project requires many resources and data, then an operating system is best.
 - * If you need security or speed then an OS may not provide what you need.
- * Cost a factor? Then Arduino has a small advantage over the Raspberry Pi.

*** WHICH TO USE?**

- * From a hardware stand point, the Pi has got most of what you need. USB, HDMI, audio out, later models have WiFi and SD memory.
- * Both have plenty of I/Os but the Pi appears to be more delicate than the Arduino.
- * The Arduino has analog input and the Pi doesn't.
- * Both have a wealth of hardware add-ons.

* ARDUINO VS. PI

- * Software wise - both have lots of programs that can be downloaded and are fun to use.
- * Arduino programs are compiled and need some experience to change etc.
- * Pi programs (applications) can be downloaded and run. They are limited to how the application was programmed.
- * Generally applications cannot be changed but compiled programs can (if you have the source code).

* ARDUINO VS. PI cont.

- * Experimentation is always appropriate. Trying these or other uPs as they are called can be fun, educational and useful.
- * If you have a specific project in mind, then research is the first step. See what the internet says. Ask people in the IOT group and the Robo group and see if they have any experience with the project you are considering.

*** SO WHERE FROM HERE**

- * Try downloading software onto your home PC and see how much you understand of the project.
- * Ask for help from the two groups previously mentioned.
- * Watch Youtube videos that explain how things work and the steps needed to complete.
- * Make a hobby of the project and if you purchase something that won't work for you, in general it's very low cost and that's the cost of education....

*** SO WHERE FROM HERE**
continued

- * Understand the following - look them up as needed:
 - * Program, Application, Windows, Linux
 - * Compiler, interpreter, Arduino, Basic
 - * CPU, I/O, uP, USB
 - * Serial communication
- * There are plenty of books out there just waiting to be read. Even ones that can be read on your tablet.

* **glossary**