Arrays for Data Lists

There are many ways to store data within any computer system. You can create a variable. You can sometimes write data to a file (for long term storage). But one common way to temporarily hold a large amount of data is in an Array. An array is basically a list of variables, stored in a specific order. \*A lot can be said about the physical memory locations of an array, but that is outside of the scope of this lesson/ activity.

# Example Sketch 1

Below is an example array implemented in a program. You can see that the array is implemented in the ‘void setup()’ just so you can see it work once. \*an important fact to know about arrays is that they are zero-referenced. That means the first location is position 0 (Zero) and the positions are numbered up from there.

**void setup() {** // the setup function runs once

**Serial.begin(9600);**

**int favoriteNumbers[] = {1, 6, 14, 32, 64};**

**Serial.println(favoriteNumbers[0]);** //print text

**Serial.println(favoriteNumbers[1]);** //print text

**Serial.println(favoriteNumbers[2]);** //print text

**Serial.println(favoriteNumbers[3]);** //print text

**Serial.println(favoriteNumbers[4]);** //print text

**Serial.println("Reassigning ‘64’");** //print text

**favoriteNumbers[4] = 11;** //reassigning array location 4

**Serial.println(favoriteNumbers[4]);** //print text

**}**

**void loop() {** //ignore the 'void loop()' for this one...

**delay(1000);** //ignore this...

**}** //ignore this...

Explain the Serial Monitor behavior of the above sketch with written sentences:

Describe the output of this program on a serial monitor. (what text and time intervals?)

Answer here

How is this output achieved by the program? (Explain the logic of the program)

Answer Here

# 

# Example Sketch 2

Below is an example array implemented in a program. You can see that the array is implemented in the ‘void loop()’ so this code will run forever, while powered.

**int ledPin = 13;**

**void setup() {** // the setup function runs once

**pinMode(ledPin, OUTPUT);**

**}**

//morse code for the letters “A.I.”

**int times[] = {250,80,750,80,250,80,250,2000};**

**void loop() {** // will run forever, while there is power

**digitalWrite(ledPin, HIGH);** //5V output on pin 13

**delay(times[0]);** //wait some time

**digitalWrite(ledPin, LOW);** //0V output on pin 13

**delay(times[1]);** //wait some time

**digitalWrite(ledPin, HIGH);** //5V output on pin 13

**delay(times[2]);** //wait some time

**digitalWrite(ledPin, LOW);** //0V output on pin 13

**delay(times[3]);** //wait some time

**digitalWrite(ledPin, HIGH);** //5V output on pin 13

**delay(times[4]);** //wait some time

**digitalWrite(ledPin, LOW);** //0V output on pin 13

**delay(times[5]);** //wait some time

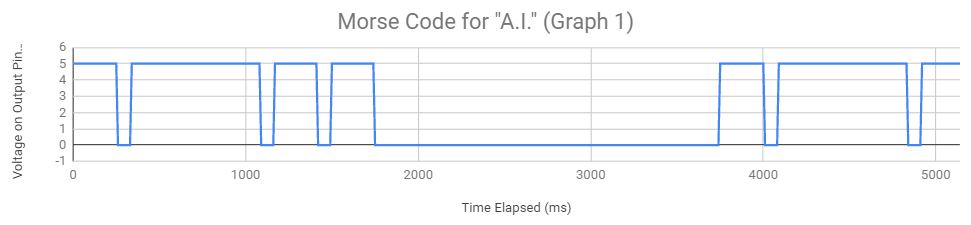
**digitalWrite(ledPin, HIGH);** //5V output on pin 13

**delay(times[6]);** //wait some time

**digitalWrite(ledPin, LOW);** //0V output on pin 13

**delay(times[7]);** //wait some time

**}**

The example sketch above would output the voltage shown in the graph below: 

Understanding how this voltage relates to the program is an important part of understanding microcontrollers. Please explain in at least one (6 sentence) paragraph how and why the voltage output shown above comes from this program.

Answer here (will probably overflow to another page)

# Arduino’s Own Explanations

[Array Reference Page](https://www.arduino.cc/reference/en/language/variables/data-types/array/) << a written explanation of arrays, and how they work on Arduino

[Array Example Sketch](https://www.arduino.cc/en/Tutorial/KnightRider) << an example implementation of an array, used in a sketch

# Create your own Array Sketches

*You don't know if you understand it, until you can create it from nothing...*

Using what you have seen above, in the example sketches and from Arduino’s reference materials. Create your own sketch that implements an array in some way. You have plenty of creative license in this goal, but you must make a sketch that successfully uses an array. A screenshot of your sketch and output is half of your response to these. A written explanation of the logic is also required to prove that you understand what you are doing with this work. Primary tasks are required for all students. Secondary tasks are required to get a top grade.

* Primary tasks:
  + successfully implement an array in the ‘void setup()’ function

Screenshot & Explanation here

* + successfully implement an array in the ‘void loop()’ function

Screenshot & Explanation here

* Moderate understanding secondary tasks:
  + Change the values in an existing array

Screenshot & Explanation here

* + Use a while loop with an array, so that the while loop sometimes runs, and sometimes does not, based on the array values

Screenshot & Explanation here

* Advanced understanding secondary tasks:
  + Store [analog input](https://www.arduino.cc/en/Tutorial/AnalogInput) values in an array of 10 numbers and “serial.println” their average value each time the program updates a value in the array. \*This is a way to “smooth” the inputs on a robot controller\*

Screenshot & Explanation here

* + Use a [2-D array](https://forum.arduino.cc/index.php?topic=275584.0), which is an array made up of other arrays

Screenshot & Explanation here

# 