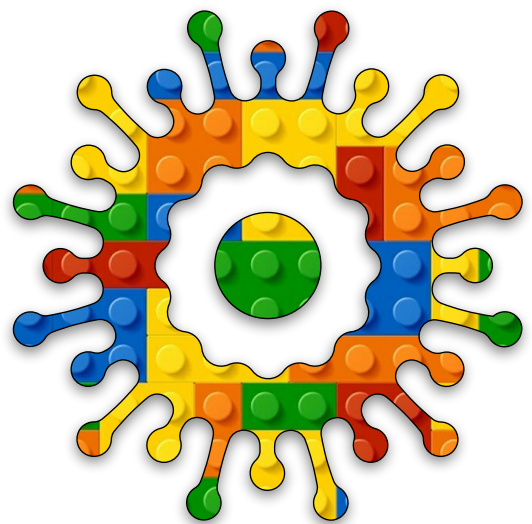


Internet of Things Module A Unit #6 arrays





Module A Unit #6 Random & Arrays

Sketch A6.1	a fast blink
Sketch A6.2	hard code
Sketch A6.3	an array
Sketch A6.4	a blinking array
Sketch A6.5	a new sketch
Sketch A6.6	not a random delay
Sketch A6.7	now a random delay
Sketch A6.8	random limits
Sketch A6.9	another new sketch
Sketch A6.10	an empty array
Sketch A6.11	filled random array
Sketch A6.12	using the random array



Introduction to random & arrays

Arrays are a very powerful tool for collecting, storing, and manipulating data. This is a simple introduction to their use and purpose.

The `random()` function gives you a random number. Although random for most purposes, it does have a pattern to it. There are ways to improve the randomness using `randomSeed()`.



Sketch A6.1 a fast blink

We will start with a basic sketch that you are familiar with. This will blink reasonably fast.

```
void setup()
{
  pinMode(2, OUTPUT);
}

void loop()
{
  blink(250);
}

void blink(int delayPeriod)
{
  digitalWrite(2, HIGH);
  delay(delayPeriod);
  digitalWrite(2, LOW);
  delay(delayPeriod);
}
```



Sketch A6.2 hard code

We will hard-code the **delayPeriod** for when the LED is off. Now it is on for **250** milliseconds and off for **100** milliseconds.

```
void setup()
{
  pinMode(2, OUTPUT);
}

void loop()
{
  blink(250);
}

void blink(int delayPeriod)
{
  digitalWrite(2, HIGH);
  delay(delayPeriod);
  digitalWrite(2, LOW);
  delay(100);
}
```



Sketch A6.3 an array

We are going to add an array for how long the LED is on for. The name of the array is **durations**. The data is stored in square brackets **[]**, separated by commas. We are going to have the LED on for three different periods of time.

```
int durations[] = {100, 100, 2000};
```

```
void setup()
```

```
{
```

```
  pinMode(2, OUTPUT);
```

```
}
```

```
void loop()
```

```
{
```

```
  blink(250);
```

```
}
```

```
void blink(int delayPeriod)
```

```
{
```

```
  digitalWrite(2, HIGH);
```

```
  delay(delayPeriod);
```

```
  digitalWrite(2, LOW);
```

```
  delay(100);
```

```
}
```



Notes

Nothing happens just yet; we have to cycle through the array.



Code Explanation

<code>durations[]</code>	An empty array called durations. The [] brackets denote an array, this is then filled with the elements in the {} braces.
--------------------------	---



Sketch A6.4 a blinking array

Now we need to call each element of the array in turn. The counting in arrays is different from the way we count. The counting starts with **0**, then **1**, then **2** rather than **1, 2, 3, 4, 5...** so the first element in the array is position **0** (not **1**).

Now we need to loop through each one in turn, and we can do this with a **for()** loop. Each **i** is an index. So...

index [0] is **100**,

index [1] is **100**, and

index [2] is **2000**.

This translates to...

durations[0] is **100**,

durations[1] is **100** and

durations[2] is **2000**, looping through one at a time till it gets to the last one and starts all over again.

Each value is passed onto the **delayPeriod** for the blink function. What you should see is two short blinks followed by two longer blinks and then two short blinks, etc.

```
int durations[] = {100, 100, 2000};

void setup()
{
  pinMode(2, OUTPUT);
}

void loop()
{
  for (int i = 0; i < 3; i++)
  {
    blink(durations[i]);
  }
}
```

```
}  
  
void blink(int delayPeriod)  
{  
    digitalWrite(2, HIGH);  
    delay(delayPeriod);  
    digitalWrite(2, LOW);  
    delay(100);  
}
```



Notes

You should get two short blinks followed by one long blink, and then repeat.



Sketch A6.5 a new sketch

! Starting with our very basic blink sketch.

```
void setup()
{
  pinMode(2, OUTPUT);
}

void loop()
{
  digitalWrite(2, HIGH);
  delay(1000);
  digitalWrite(2, LOW);
  delay(1000);
}
```



Sketch A6.6 not a random delay

Introduce a variable we will call **delayRandom** (made-up word) and give it an initial value of **1000**. It is always a good idea to give a variable an initial value.

```
int delayRandom = 1000;

void setup()
{
  pinMode(2, OUTPUT);
}

void loop()
{
  digitalWrite(2, HIGH);
  delay(delayRandom);
  digitalWrite(2, LOW);
  delay(delayRandom);
}
```



Notes

Nothing will happen yet. There is no randomness yet, we have simply called it variable name.



Sketch A6.7 now a random delay

Now we change the **1000** milliseconds to a random number between **0** and **1000**. It will change every time it loops through.

```
int delayRandom = 1000;

void setup()
{
  pinMode(2, OUTPUT);
}

void loop()
{
  delayRandom = random(1000);
  digitalWrite(2, HIGH);
  delay(delayRandom);
  digitalWrite(2, LOW);
  delay(delayRandom);
}
```



Notes

The LED will blink random lengths and pause for random lengths up to **1000** milliseconds (**1** second).



Code Explanation

`random(1000);`

Gives you a random number between 0 and 1000



Sketch A6.8 random limits

We can create upper and lower limits, so now it has a random number between **500** and **2000** milliseconds.

```
int delayRandom = 1000;

void setup()
{
  pinMode(2, OUTPUT);
}

void loop()
{
  delayRandom = random(500, 2000);
  digitalWrite(2, HIGH);
  delay(delayRandom);
  digitalWrite(2, LOW);
  delay(delayRandom);
}
```



Code Explanation

<code>random(500, 2000);</code>	Returns a random number between 500 and 2000
---------------------------------	--



Sketch A6.9 another new sketch

! Starting with our basic sketch, with the delay set to **1000**.

```
void setup()
{
  pinMode(2, OUTPUT);
}

void loop()
{
  digitalWrite(2, HIGH);
  delay(1000);
  digitalWrite(2, LOW);
  delay(1000);
}
```



Sketch A6.10 an empty array

We add an array of ten elements. This is a blank or empty array, effectively ten zeros.

```
int durations[10];

void setup()
{
  pinMode(2, OUTPUT);
}

void loop()
{
  digitalWrite(2, HIGH);
  delay(1000);
  digitalWrite(2, LOW);
  delay(1000);
}
```



Notes

Nothing new will happen.



Code Explanation

```
int durations[10];
```

This array is effectively empty but it has been given the dimensions of 10 elements.



Sketch A6.11 filled random array

In `setup()`, we fill each element with a random number between `0` and `1000`. We do this by looping through ten times, each time selecting a random number.

```
int durations[10];

void setup()
{
  pinMode(2, OUTPUT);
  for (int i = 0; i < 10; i++)
  {
    durations[i] = random(1000);
  }
}

void loop()
{
  digitalWrite(2, HIGH);
  delay(1000);
  digitalWrite(2, LOW);
  delay(1000);
}
```



Notes

Nothing to see just yet. This is storing the random elements in the array, it still isn't going to give a random blink, not yet...



Code Explanation

```
durations[i] = random(1000);
```

As it cycles through the loop, `i` starts at zero and increments by 1 each loop. So for each position in the array it puts a random number from 0 to 1000 into that array. The `i` value is the index reference for the array, counting starts at zero not 1.



Sketch A6.12 using the random array

Now we need to read from each element in a similar loop inside the `loop()` function. You should see the same pattern of blinks every ten blinks.

```
int durations[10];

void setup()
{
  pinMode(2, OUTPUT);
  for (int i = 0; i < 10; i++)
  {
    durations[i] = random(1000);
  }
}

void loop()
{
  for (int i = 0; i < 10; i++)
  {
    digitalWrite(2, HIGH);
    delay(durations[i]);
    digitalWrite(2, LOW);
    delay(durations[i]);
  }
}
```

Code Explanation

```
delay(durations[i]);
```

This is the reverse of the previous operation. It goes through the array one element at a time starting at index 0 through to index 9, it reads its value and that is the length of the delay.