Arduino Programming :

Switch-Case

Objectives

- Understand the Switch-Case structure
- Understand applications of Switch-Case

Branching / Looping methods

So far we have discussed three ways to branch or loop: if, while and for.

if – if/else: Used to "branch" based on a variable value or input

while: Used to stay in a loop until a condition changes

for: Used to loop a specific number of times

It is important to determine which structure to use in your program.

Switch / Case

Another powerful method of branching is the switch / case structure.

The first part – *switch* – determines what variable will be evaluated:

```
switch (x)
{

}
```

Switch case

The second part – *case* – determines what code runs based on the value of the variable

```
switch (x)
{
case 1:
  digitalWrite(LED1,HIGH);
  break;
case 2:
  digitalWrite(LED2,HIGH);
  break;
case 3:
  digitalWrite(LED3,HIGH);
  break;
}
```

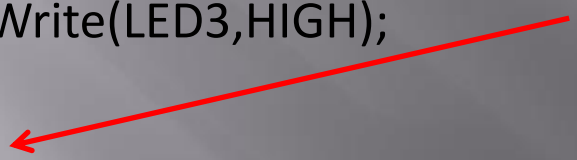
Does this code if x is equal to 1

If this code runs, break out

Default

If none of the values of the variable match a number, then default can be included:

```
switch (x)
{
…
case 3:
   digitalWrite(LED3,HIGH);
   break;
default:
   digitalWrite(LED1,LOW);
   digitalWrite(LED2,LOW);
   digitalWrite(LED3,LOW);
}
```



Run this code if x does not equal 1 to 3

Why is switch case so great?

It eliminates a large number of *if – else* loops

It allows for a condition (default) if none of the values are true

It can be used to run through parts of a program sequentially (more on this later…)