Arduino Programming Fundamentals

Key Terminology

Variable – a name that represents a value. Think of it like algebra, where ‘a’ could equal any number, but ‘a’ can also equal words too. Variables can also be changed as the program runs.

Key Commands

I/O (Input, Output)

- `pinMode(PinNumber, Mode)` – sets a pin to be either an INPUT or an OUTPUT.
  e.g. `pinMode(2, OUTPUT);`
- `digitalRead(PinNumber)` – read the value of the specified PinNumber. This vaule is usually saved to a variable or used as part of another command.
  e.g. `state = digitalRead(3);`
- `digitalWrite(PinNumber, STATE)` – set the digital pin PinNumber to the state HIGH (on) or LOW (off).
  e.g. `digitalWrite(2, HIGH);`

Serial (Debugging)

- `Serial.begin(baud)` – Begins a Serial transmission over the USB connection to the computer. If using, always use in the setup() section. The typical baud rate is 9600. Must be used for other Serial commands to work.
  e.g. `Serial.begin(9600);`
- `Serial.println(string)` – Prints whatever string (text surrounded by “”) to the serial monitor in the Arduino IDE. Also makes each new command use a new line to make it easier to follow.
  e.g. `Serial.println(“Hello World!”);`

Other

- `delay(time)` – Make the program wait for the specified amount of milliseconds (1000 = 1 sec) before proceeding onto the next line.
  e.g. `delay(1000);`

File Layout

After creating a new file, you will notice that there are two sections, labelled `setup()` and `loop()`. The only things that do NOT go in one of these sections are GLOBAL variables, i.e. variables that need to be accessible by all parts of the code. All other code, at least for these basic

programs, MUST go in one of these sections. This includes CHANGING the values of those global variables.

setup()

The `setup()` section contains all the code that only runs ONCE, such as setting pins to the right modes and enabling Serial.

loop()

The `loop()` section, as the name implies, contains all code that is meant to run FOREVER. This will be where the bulk of the code should be, such as reading inputs and changing outputs.

if() Statements

if() statements are a form of logic, which allows a program to change it's behaviour based on if something else is or isn't true. They only run once, then continue through the rest of the code. For example, if you want to turn the LED on when it is pressed, you could write:

```
if (button == HIGH) {
     digitalWrite(ledRed, HIGH);
}
```

Firstly, you may notice that there are two equals (=) signs. In the Arduino IDE, a single equals is for SETTING a value (e.g. Pin = 2), while double equals is for checking if the values are the same as part of logic.

You will also notice that there are a set of {} brackets, which is how the program knows where the if() statement starts and ends.

else if and else statements

As an extension of if() statements, else if() can be used for alternative sets of code which can run if the statement provided is true. These run in the order they are found in the code, and the first one that is found to be true will be the ONLY one to run. If you want multiple statements which can all run if they are all true, then use multiple if() statements instead.

```
if (button == HIGH) {
     digitalWrite(ledRed, HIGH);
}
else if (button == LOW) {
     digitalWrite(ledRed, LOW);
}
```

else() statements are always the last statement in a if() statement set, as it is the catch-all for if nothing else is true. For instance, with the example from earlier, if we wanted to turn the light off when the button is NOT pressed:

```
if (button == HIGH) {
     digitalWrite(ledRed, HIGH);
}
else {
     digitalWrite(ledRed, LOW);
}
```

while() loops

Unlike if() statements, while() loops run through whatever code is contained within them until the statement is no longer true. Any code AFTER the while() loop will not be run until the statement is no longer true.

```
while (statement) {
     code
}
```

For instance, this could be used to make the LEDs flash in a pattern while the button is held, but once it is let go it will turn all the LEDs off.