

CONTROL FLOW: WHILE STATEMENTS

If we want to do something while a condition is true, then we can use a while statement. So let's write a program that turns on more and more LEDs while the button is pressed.

```
void loop() {  
    int i = 1;  
  
    while (EngduinoButton.isPressed()) {  
        EngduinoLEDS.setLED(i, RED);  
        i = i + 1;  
        delay(200);  
    };  
  
    delay(1000);  
    EngduinoLEDS.setAll(OFF);  
}
```

Like the `for` statement, we declare a variable, `i`, but we give it an initial value of 1 this time:

```
int i = 1;
```

Now comes the loop – with several statements grouped together inside the curly braces:

```
while (EngduinoButton.isPressed()) {  
    EngduinoLEDS.setLED(i, RED);  
    i = i + 1;  
    delay(200);  
};
```

What this means is:

- a) Check to see if the condition is true – in other words, if the button is pressed. If it is no longer pressed, go to step f).
- b) set LED number `i` to be RED
- c) add one to the value of `i`
- d) delay a little (0.2s)
- e) and go back to the step a)
- f) As soon as we let the button go (the condition is false) we carry on with the rest of the program. This waits for a second (`delay(1000);`) and then sets all the LEDs so that they are off again (`EngduinoLEDS.setAll(OFF);`).