

```

//
// Program Control Structure Examples with lots of Blinking Lights!
//
// First, define which pins to use
// NOTE: The output pins *must be* in ascending order by pin number!
//
const int Pin0 = 0; // LED output
const int Pin1 = 1; // LED output
const int Pin2 = 2; // LED output
const int Pin3 = 3; // LED output
const int PushButton = 12; // Control input
int count; // Define a "global" variable to count with

//
// By "defining" these symbols, we can change *a lot* of things in
// one easy step here!
//
#define NUM_LEDS 4 // The number of LEDs in use!
#define SMALL_DELAY 50 // A 50-millisecond delay
#define BIG_DELAY (SMALL_DELAY * 10) // Delay ten times as much!

//
// The setup() method runs once, when the sketch starts
// and initializes the digital pins as outputs or inputs.
//
void setup() {
int ii; // Variables called "ii", "jj", and "kk" are typically used to count!

for(ii = 0; ii < NUM_LEDS; ii++) // Do this for the # of LEDs in your display
{
pinMode(Pin0+ii, OUTPUT);
}
pinMode(PushButton, INPUT); // initialize the control input
}

//
// This method or function displays 4 bits on 4 LEDs
// NOTE: It also *assumes* that your pin numbers are in order
// starting with Pin0!
//
void LEDs_out(int my_nib)
{
int ii; // Variables called "ii", "jj", and "kk" are typically used to count!

for(ii = 0; ii < NUM_LEDS; ii++) // Do this for the # of LEDs in your display
{
if(my_nib & (1 << ii))
{
digitalWrite((Pin0+ii), HIGH); // set the LED on
}
else
{
digitalWrite((Pin0+ii), LOW); // set the LED off
}
}
} // END void LEDs_out(int my_nib)

```

```

//
// This method or function moves the lights in one direction
//
void strobe_up()
{
char shiftx;

    shiftx = 0b1000;    // Define the first LED to turn on

    LEDs_out(0b0000);    // Begin with NO LEDs on!
do
{
    delay(SMALL_DELAY);
    LEDs_out(shiftx);    // Turn on the LED defined by shiftx
    shiftx = (shiftx >> 1); // Change shiftx to the next LED
}while(shiftx & 0b1111); // Check to see if we're done

    delay(SMALL_DELAY);
    LEDs_out(0b0000);    // All LEDs off again...
} // END void strobe_up()

//
// This method or function moves the lights in the other direction
// in a different, simpler way!
//
void strobe_dn()
{
    LEDs_out(0b0000); // NO LEDs on!
    delay(SMALL_DELAY);
    LEDs_out(0b0001); // The first LED on!
    delay(SMALL_DELAY);
    LEDs_out(0b0010); // Now, the next LED...
    delay(SMALL_DELAY);
    LEDs_out(0b0100); // and so on...
    delay(SMALL_DELAY);
    LEDs_out(0b1000);
    delay(SMALL_DELAY);
    LEDs_out(0b0000); // All LEDs off again...
    delay(SMALL_DELAY);
} // END void strobe_dn()

//
// This method or function moves the lights in both directions
// using the methods we've already designed...
// Note: "Cylons" are the robots in Battlestar Galactica... :)
//
void cylon()
{
    strobe_up();
    strobe_dn();
} // END void cylon()

```

```

//
// This method or function blinks all the lights rapidly
//
void blink()
{
int ii;
  for(ii=0;ii<5;ii++)
  {
    LEDs_out(0b1111); // Turn on all the LEDs
    delay(SMALL_DELAY);
    LEDs_out(0b0000); // Turn off all the LEDs
    delay(SMALL_DELAY);
  }
} // END void blink()

//
// The loop() method runs over and over again,
// as long as the Teensy 3.0 has power
//
void loop()
{
int ii;
int limit = 5; // Specify how high to count - Try Different limits!

  LEDs_out(0b0000); // Turn off all LEDs
  delay(BIG_DELAY); // Delay for BIG_DELAY milliseconds

//
// The for() control structure is used for counting!
// In this example, "count" is the variable we count with.
// We start at zero and count up to display the "limit"
//
  for(count = 0; count < limit+1; count++)
  {
    LEDs_out(count); // Display current binary count on our LEDs
    delay(BIG_DELAY); // Delay for BIG_DELAY milliseconds
    if(0 == digitalRead(PushButton)) // Is the input low?
    {
      // If so, pause in our count until the button is released...
      while(0 == digitalRead(PushButton)); // then wait for input high
    }
  }
  delay(BIG_DELAY*3); // Leave our count displayed for a bit

  LEDs_out(0b0000); // Turn off all LEDs
  delay(BIG_DELAY); // Delay for BIG_DELAY milliseconds

```

```

//
// The switch() control structure helps us choose between a variety
// of actions! Here we will select the option that matches the count we set above...
//
switch(limit) // What is the present value of "limit?"
{
case 1: // Do this when limit is equal to 1

for(count = 0; count < 7; count++) // Try Different Counts!
{
    strobe_up();
}
break; // This takes us out of the switch() statement

case 2: // Do this when limit is equal to 2
for(count = 0; count < 14; count++) // Try Different Counts!
{
    strobe_dn();
}
break; // This takes us out of the switch() statement

case 3: // Do this when limit is equal to 3
for(count = 0; count < 3; count++) // Try Different Counts!
{
    cylon();
}
break; // This takes us out of the switch() statement

case 4: // Do this when limit is equal to 4
for(count = 0; count < 10; count++) // Try Different Counts!
{
    blink();
}
break; // This "breaks" us out of the switch() statement

case 5: // Do this when limit is equal to 5
default: // Do this ALSO when limit is equal to ANYTHING ELSE
for(count = 0; count < 7; count++) // Try Different Counts!
{
    // Now do a bunch of different things!
    blink();
    for(ii=0;ii<10;ii++)
    {
        cylon();
    }
    blink();
    strobe_up();
    blink();
    strobe_dn();
    blink();
}
break; // This "breaks" us out of the switch() statement

} // END switch(limit)

} // END void loop()

```