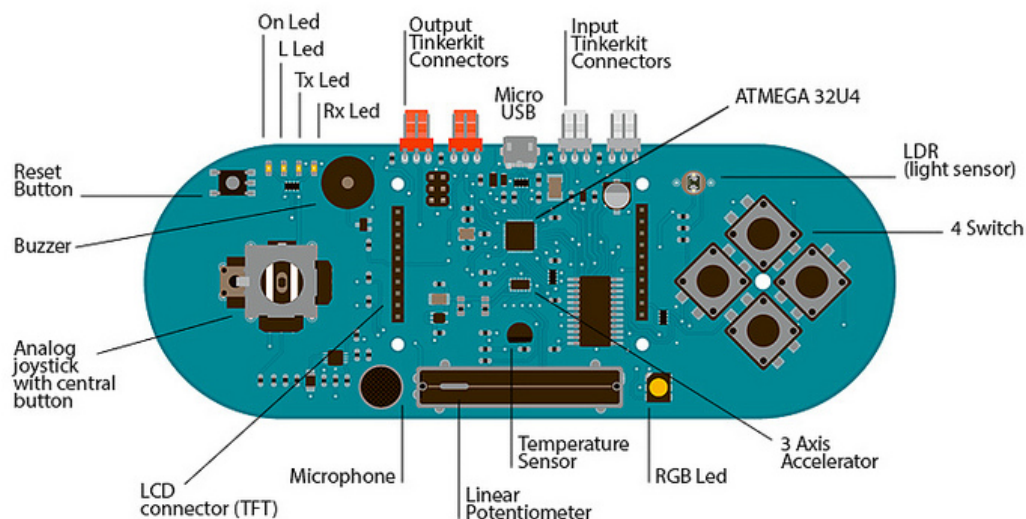


Activity

Now collect your designated Esplora board. Identify all the features that we discussed previously on the board.



Now we will write your first program.

A Simple Program

To compile the program (by that we mean take your *sketch* and turn it into something the Arduino can understand) the Arduino software goes through the file line-by-line. Let's take a look at a very simple program, that contains all the elements to compile and run, but it doesn't actually do anything.

```
/* This is a block comment that can span multiple lines. It's  
usually used to give a brief intro to the program and say what  
it does.  
*/
```

// This is a single line comment, anything after the "//" is ignored.

```
void setup() {
```

```
    // run once
```

```
}
```

```
void loop() {
```

```
    // run forever
```

```
}
```

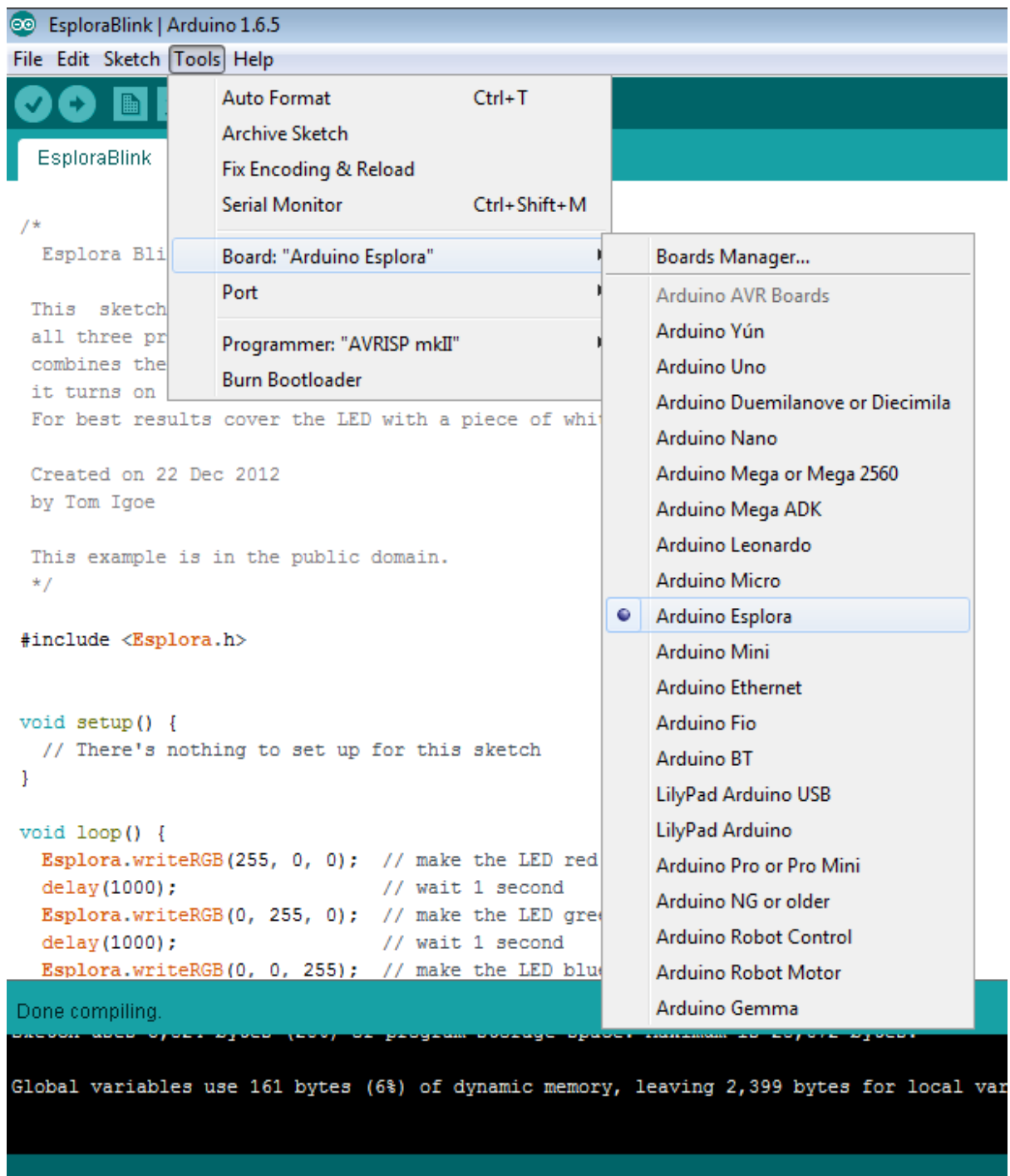
It runs all the commands at the beginning, and goes through the file sequentially until it finds the setup() function. The setup runs exactly once, but it must be included, even if it is empty like the example above. It keeps going through the file until it gets to the loop() function. While the Arduino is turned on, it will go through the loop and repeat it when it gets to the end.

Since there is only a comment (you can tell by the //) in both the setup() and loop() functions, this program does absolutely nothing. But it will compile and run.

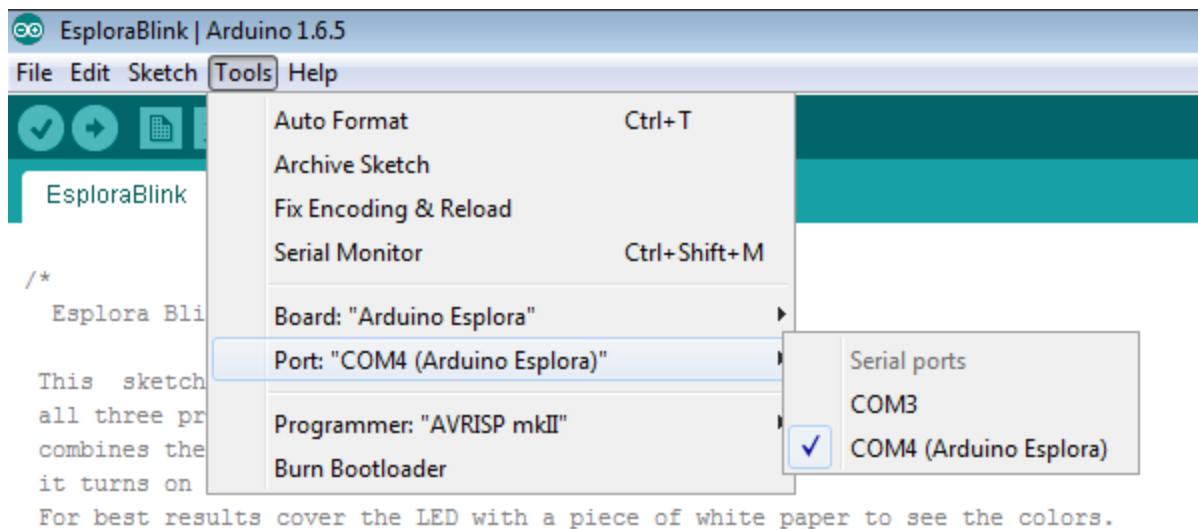
Your First Program!

Let's go through all the steps now to load your first program onto the Arduino Esplora board.

First, we need to tell the Arduino software that you are using the Arduino Esplora board. Otherwise it won't know what type of board you have plugged in. To do this we go to **Tools > Board > Arduino Esplora**, like in the screenshot below.



Next, we need to tell the Arduino software what Port you are communicating to the board on. To do this go to **Tools > Port > COMx (Arduino Esplora)**, where x is a number. Probably 4 or 5, it depends on your computer. On a mac and linux, this will be a little different. If you don't have an option that says "(Arduino Esplora)", please check that you have installed the drivers on windows, and that the board is plugged into your computer.



Type the following into the Arduino software:

```
/* This is a modified version of Esplora blink. This program cycles through the
   three primary colours - red, green and blue.
*/

#include <Esplora.h>

void setup() {

  // There's nothing to set up for this sketch

}

void loop() {

  Esplora.writeRGB(255, 0, 0);    // make the LED red (red, green, blue)

  delay(1000);                    // wait 1 second, the units are in milliseconds

  Esplora.writeRGB(0, 255, 0);    // make the LED green

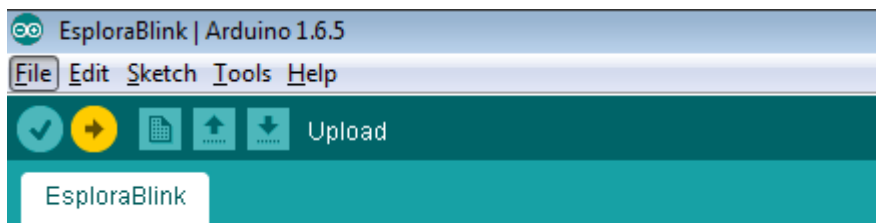
  delay(1000);                    // wait 1 second

  Esplora.writeRGB(0, 0, 255);    // make the LED blue

  delay(1000);                    // wait 1 second

}
```

Next you click the upload button

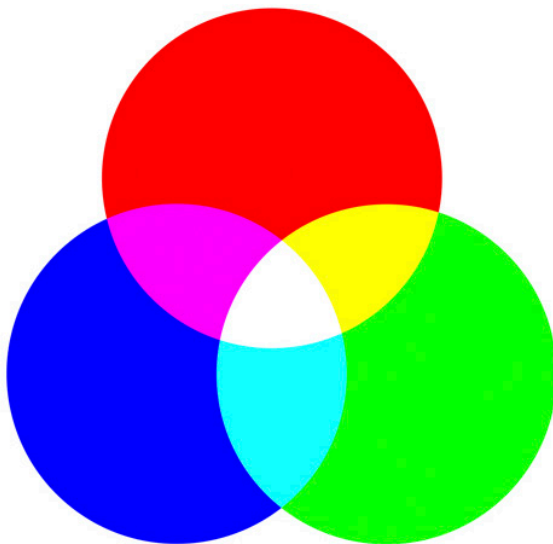


If that worked correctly, it usually takes about 10 seconds, but your LED should be flashing between RED, GREEN and BLUE colours.

Activity

Colors

Magenta is a color that can be created by combining the RED and BLUE LEDs. We turn them both on at once and the colors will ADD to make Magenta. The maximum value of each LED is 255, where there are 255 different brightness levels for each color, just like in Photoshop or any program used on a computer to represent color.



What single line of code will make the Esplora RGB LED turn Magenta at full brightness?

Look at the program above to find the right line - you will just have to change the numbers. Each line must end with a semi-colon!

Change your program to blink in turn, Magenta, Yellow and Turquoise.

Activity

Compiling

Type the following into the Arduino software:

```
/* This program flashes the LED at a given speed determined by the value set
   inside the delay function. You will need to uncomment one line at the top of the
   program to get this program to compile.
*/

// #include <Esplora.h>

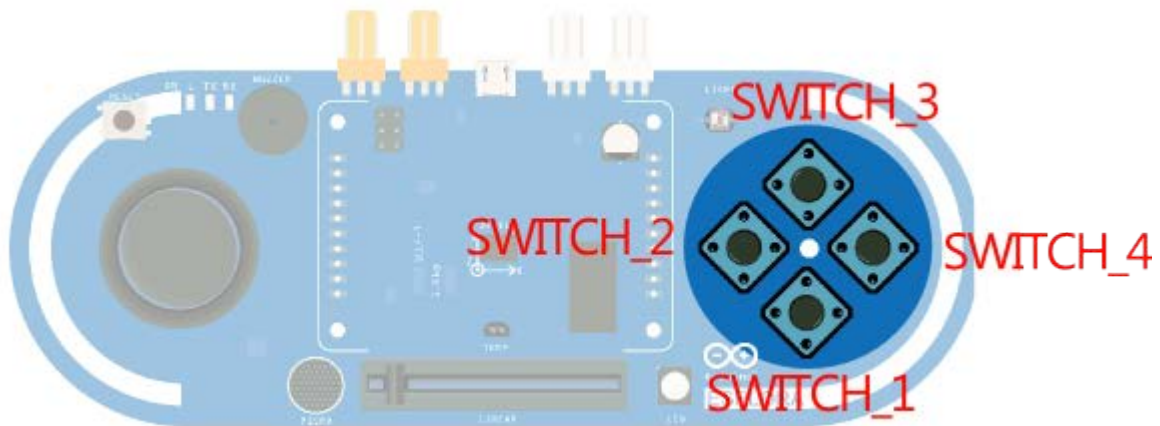
void setup() {
  // Nothing to setup in this program
}

void loop() {
  Esplora.writeRGB(0, 0, 255); // Turn on blue LED
  delay(1000);                // Wait one second (units are in milliseconds)
  Esplora.writeRGB(0, 0, 0);  // Turn off blue LED
  delay(1000);                // Wait one second
}
```

If you click on the Verify button, you will notice that the above program says 'Error compiling'. Inside the Error messages it says error: 'Esplora' was not declared in this scope.

Why do you think we get this error? Can you make this program work?

Using the Buttons



In these exercises we'll look at how you can use the buttons to control the LEDs.

```
/* This program turns on the red led after SWITCH_1  
is pressed.
```

```
*/
```

```
#include <Esplora.h>
```

```
void setup() {
```

```
    // Nothing to setup
```

```
}
```

```
void loop() {
```

```
    if (Esplora.readButton(SWITCH_1) == LOW) {
```

```
        Esplora.writeRed(255);
```

```
    }
```

```
}
```

After the code is uploaded, hit switch one, which is the bottom push button. You'll see the RED led turn on. So let's take a look at the code and see what happening.

On the first line of the loop() function we see a new statement that we haven't seen before, the **if** statement. This checks that if a condition is true, and *if it is*, the following section gets executed.

So in the example above, we check **if** Esplora.readButton(SWITCH_1) *is the same as* (==) LOW (or pressed down). Since when the button is pressed down, this condition is true, the RED LED turns on.

But How do we turn the LED off....?

In the previous example, we turned the red LED on by pressing the button, but the LED never turned off. Once the button was pressed, the LED stayed on. This is because we never gave a condition in our program of when the LED could turn off. *The Arduino will only ever do exactly what you tell it to do.*

In the program below, we add a statement that lets us turn the LED off as well. Load the program onto your Arduino Esplora, have a read through the program and see if you can figure out how it works.

```
#include <Esplora.h>

void setup() {
  // Nothing to setup
}

void loop() { // Check if switch 1 is pressed.
  if (Esplora.readButton(SWITCH_1) == LOW) {
    Esplora.writeRed(255);    // Turn red led on
  }
  else {
    Esplora.writeRed(0);      // Turn red led off
  } }
}
```


In this program we add a new statement, the **else** statement. You can only use the else statement when paired with an **if** statement. Only one of them will get executed. If the *if* condition is true, that section will get executed, otherwise the *else* section will get executed.

Using Multiple Switches

What happens if we want to use multiple switches to do different things? Then we'll need multiple *if* statements in order to check each condition. Remember, computers in general only ever do what you tell them to do.

Below is an example of two *if statements* to check SWITCH_1 and SWITCH_2, to turn both the red and green LEDs on and off.

```
#include <Esplora.h>

void setup() {
    // Nothing to setup
}

void loop() {
    // Check if switch 1 is pressed.
    if (Esplora.readButton(SWITCH_1) == LOW) {
        Esplora.writeRed(255);    // Turn red led on
    }
    else {
        Esplora.writeRed(0);      // Turn red led off
    }

    // Check if switch 2 is pressed.
    if (Esplora.readButton(SWITCH_2) == LOW) {
        Esplora.writeGreen(255); // Turn green led on
```

```
}  
  
else {  
    Esplora.writeGreen(0);    // Turn green led off  
}  
}
```

Upload the program above onto your Arduino Esplora and give it a go!

Activity

All Four Switches

Now modify the above code so all four switches produce the following outputs:

- Switch one - Red
- Switch two - Green
- Switch three - Blue
- Switch four - Magenta (magenta is red and blue)

Test it out on the Arduino Esplora, make sure you add else statements to turn the LEDs off!

Conditional statements

We've seen conditional if statements used before in our Esplora sketches.

For example,

1. `if (Esplora.readButton(SWITCH_1) == LOW) { // if switch_1's value is equal to low,`
2. `Esplora.writeRed(255); // turn red led on to max brightness (255)`

```
3.  }  
4.  else {                      // otherwise,  
5.    Esplora.writeRed(0);    // turn red led off  
6.  }
```

It's time we learn how to write some of these statements ourselves.

A conditional statement simply tests whether a statement is **true** or **false**.

If the answer to the test is **true**, it will run the lines of code within the curly braces immediately after the statement.

If the answer to the test is **false**, it will **skip** all the lines of code within the curly braces, and continue on later.

So what tests can we apply?

Since we mainly deal with numbers in programming, there are usually two types of tests we come across:

- comparison tests and,
- logical tests

Comparison tests come in the following forms:

- == Equal to
- != Not Equal to
- > Greater Than
- < Less Than
- >= Greater Than or Equal To
- <= Less Than or Equal To

Activity

Conditional Problems

State whether these conditional statements are true or false:

1. $3 > 4$
2. $-5 < 1$
3.

```
int a=5;
int b=10;
b>a
```
4.

```
int a=4;
int b=4;
a>=b;
```
5.

```
int a=-1;
int b=0;
a++; // same as a = a+1
a!=b;
```
6.

```
int a=4;
int b=0;
b=a/a; // backslash is a divide operator
b==1;
```

There are three main **logical** tests (NOT, AND and OR):

NOT:

- Which returns the opposite value for **TRUE** or **FALSE**.
- We use the operator **!** to represent the NOT function. For example:
 - **!TRUE** returns **FALSE**
 - **!FALSE** returns **TRUE**

AND:

- Which returns a TRUE value if ALL values tested are true.
- We use the operator **&&** to represent the AND function. For example,
 - **TRUE && TRUE** returns **TRUE**
 - **TRUE && FALSE** returns **FALSE**
 - **FALSE && FALSE** returns **FALSE**

- **TRUE && TRUE && FALSE** returns **FALSE**

OR:

- returns a TRUE value if ANY values are true.
- We use the operator `||` (above the enter key on a US layout keyboard) to represent the OR function. For example,
 - **TRUE || TRUE** returns **TRUE**
 - **TRUE || FALSE** returns **TRUE**
 - **FALSE || FALSE** returns **FALSE**
 - **TRUE || TRUE || FALSE** returns **TRUE**

Activity

Logical Problems

State whether these logical statements are true or false:

1. **TRUE && TRUE**
2. **FALSE && TRUE**
3. **FALSE || !TRUE**
4. `int a=0;`
`int b=1;`
`(a!=b) && (b>a)`
5. `int a = -2;`
`(a<0) || (a>0)`

6. What colour will the light be with the following code?

```
int a = 0;
int b = 1;
a=a+1;
if (a==b){
  Esplora.writeRed(255);
}
else {
  Esplora.writeGreen(255);
}
```

7. What colour will the light be with the following code?

```
boolean proceed = TRUE;
if (!proceed){
    Esplora.writeRed(255);
}
else {
    Esplora.writeGreen(255);
}
```