Antonio Jimenez

Spaceship Interface

3/4/20-3/8/20

Lab

**Introduction:**

In this project we will be making a cool control panel that consist of a push button and LED’s. when you press the pushbutton, you can control the pattern of the lights. You will learn the about inputs and outputs, the programming of inputs and outputs. You will also get into how to use if statements in Arduino code.

**Materials:**

* 1- Pushbutton
* 2 – Red LED’s
* 1 – Green LED
* 3 – 220-ohm resistors
* 1 -10k ohm resistor
* Jumper wires

**Observations:**

All right so I have just made the spaceship interface and it was a very unique design I mean it uses LEDs but we made it unique by just putting a piece of Cardboard on it and making it look pretty and so far it looks nice.

So, in this project what was supposed to happen is that when you press the push button you create a pattern with the LED’s. what really happened was as described in the manual. There were a few changes I had to make however but they are trivial, for example the output pins had to change instead of using the pins described in the book I had to use pin 11, pin 7, and pin 4.

So, in this project we were supposed to learn inputs and outputs using a push button and LED’s. the way was constructed is practically simple all it consists of was a few jumper wires resistors LED's and the push button. The way the system worked was power exits from the Arduino then the Arduino checks whether the circuit is complete to where the push button is. Now if the circuit is not complete, the Arduino will tell itself “hey there is no circuit entering the Arduino therefore I turn on this specific LED”. now if we press the push button, the Arduino will tell itself “hey, there is current entering the pin therefore I execute this pattern”. Then the Arduino will loop through that over and over so long as it has power.

Alright so let's talk about the code. We have just been introduced to what the code looks like for the Arduino and so far, it makes sense. The Arduino code is divided into 3 sections, and this is common for almost every Arduino code, we have **initialization**, then **Setup,** then **Loop.** In the initialization we have define the switch state where it states to be zero. In the setup we have all the pins defined with output or input. And finally, in the loop we have a series of turning on these pins.

Due to this code, the behavior of the circuit displays that the green LED starts off being on then as I press the pushbutton the LED shifts to the two reds alternating as I hold down the push button. I modified the code so that it displays a more interesting effect. I called the code “ASK\_Project\_2\_V2” and what it does is when you push the button the bottom lights up for a while then the middle LED starts to blink. What I changes was I placed a for loop in the code. And that caused the middle light to blink.

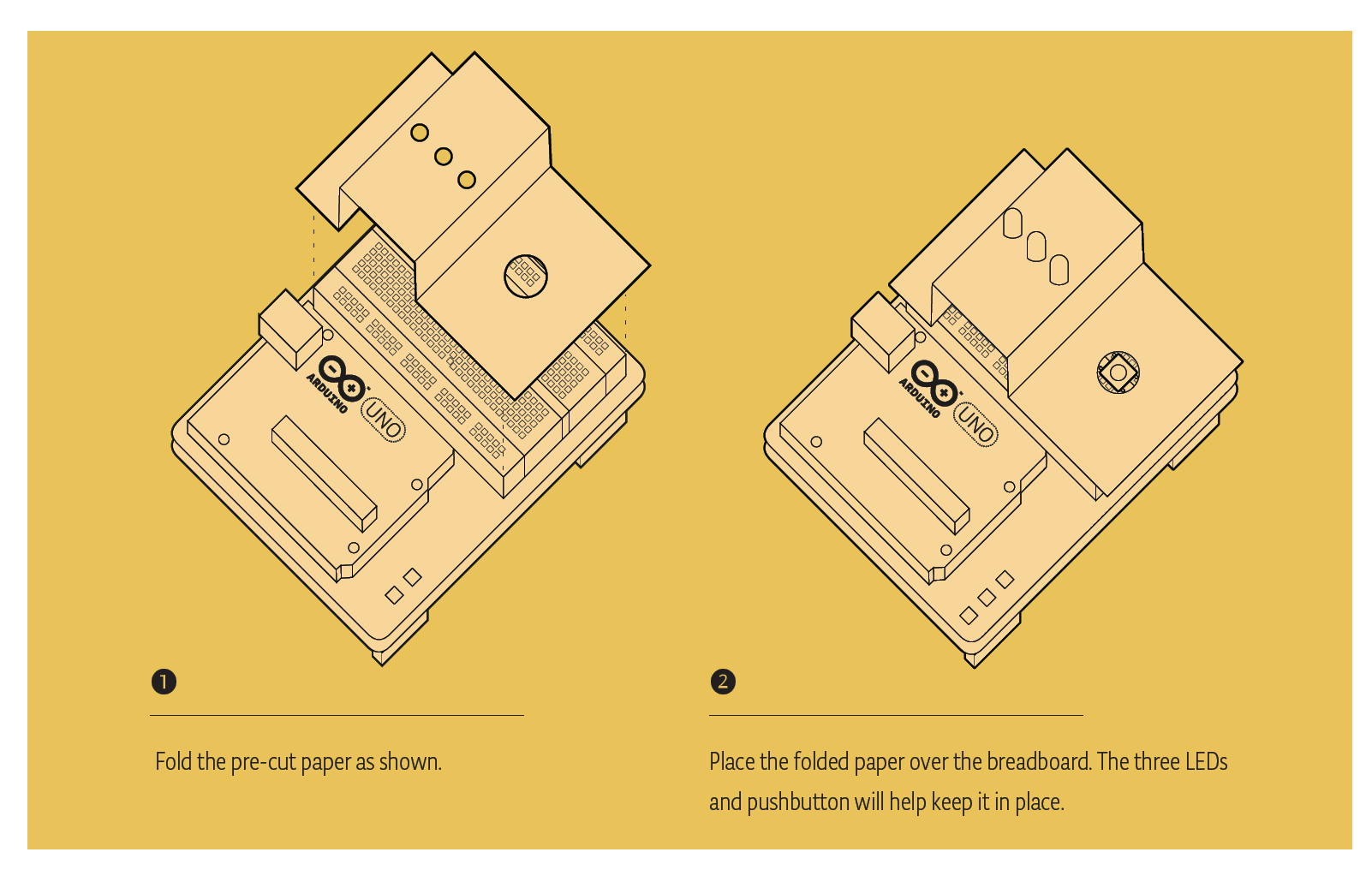
**Notes:**

In this project we learn about inputs and outputs. The Arduino pins can read two states, on or off. These inputs are called digital (or binary for on or off). The code refers to on and off as High and Low respectively. We turn on the output pin by using the digitalWrite () command. When measuring the output voltage of the pin you get five volts.

The digital pins can act as inputs or outputs. You can configure them in your code. When you configure a pin to be an output it will give off voltage. If you make your pin an input, then it can read voltage as it goes in.

One note when building your circuit, make sure you insert a 10k ohm resistor from the ground to the switch pin so that when the switch is open the pin can read LOW when there is no voltage coming, this is what you call a pull down resistor.

**Note:**  you can even cover the circuit in cardboard and show what each light means, kind of like a control panel.



Now let us look at the code:

Arduino code always has 3 major sections, initialization, setup, and loop. The initialization section is where we define variables so that you can keep track of everything. Be sure that the names are descriptive. One variable would be “switchState” which stores the state of a switch.

To create a variable, you must specify what kind of variable it is. In this case the switch state variable is an integer “int” meaning that the value must be a whole number. the variable must end with a semi colon “;”.

Now we move into the setup section, the setup section only runs once. In the setup, this is where you can configure the pins to be inputs or outputs. By using the “PINMODE” function you can specify which pin and whether that pin is an input or an output.

The loop runs continuously, the loop is where the voltage is checked in the inputs and when the LED’s turn on and off. To check the voltage you have to use **digitalRead()** that checks the chosen pin for the voltage then stores it in a variable we already declared, but first it needs an argument meaning that it needs to be given some information, such as the pin number. The **switchState** variable will get either High (1) or Low (0). We then use if statements (**if ()**) to compare two things and determine if its true or false.

…

In general, a conditional statement is saying “if something is true then do this if not then do this instead.”. also note that a double equal sign means equal to.

is a command that sends 5 volts or 0 volts from the pin specified. The only arguments needs is the pin number and the state HIGH or LOW. After you have closed the switch the Arduino reads that the switch state is no longer low then executes the else statement. In the else statement you can see that there are functions, they are there because we want the program to slow down the transition of alternating the LEDs or else we would just get light s alternating so fast that all we see are two dim LED’s. the only has one argument and that is the time in milliseconds.

**Questions:**

* How could you get the red LED to be blinking when your program starts?
* How could you make a larger more complex interface with your LED and switches?

**Additional notes:**when you start creating your interface, think about what people’s expectations are when using it. When they press the button, would they want immediate feedback, or will they want a delay in feedback? Put yourself in their shoes and see if your design functionality expectations go in sync with other users.

**Conclusion:**

In this project, you created your first Arduino program to control the behavior of some LED’s based on a switch. You’ve used variables and the if else functionality to use them to read inputs and give certain outputs.