Lab Report: Project 2 - Spaceship Interface

Lucas Porter

Spring 2025 TESY 3300

Trevor P. Robinson, PhD

Lab 2 – Spaceship Interface

1/28/2025

Abstract

The objective of this lab was to design and implement a spaceship interface using an Arduino Uno. The circuit consisted of a pushbutton, a green LED, and two red LEDs. The goal was to demonstrate digital input and output control by having the green LED remain on until the button was pressed, at which point the green LED would turn off and the red LEDs would blink alternately. The lab reinforced concepts of digital logic, pin configuration, and structured programming using conditional statements and functions.

Materials

• Arduino Uno Board

• Breadboard

• One Green LED

• Two Red LEDs

• One Pushbutton

• One 10kΩ Resistor

• Three 220Ω Resistors

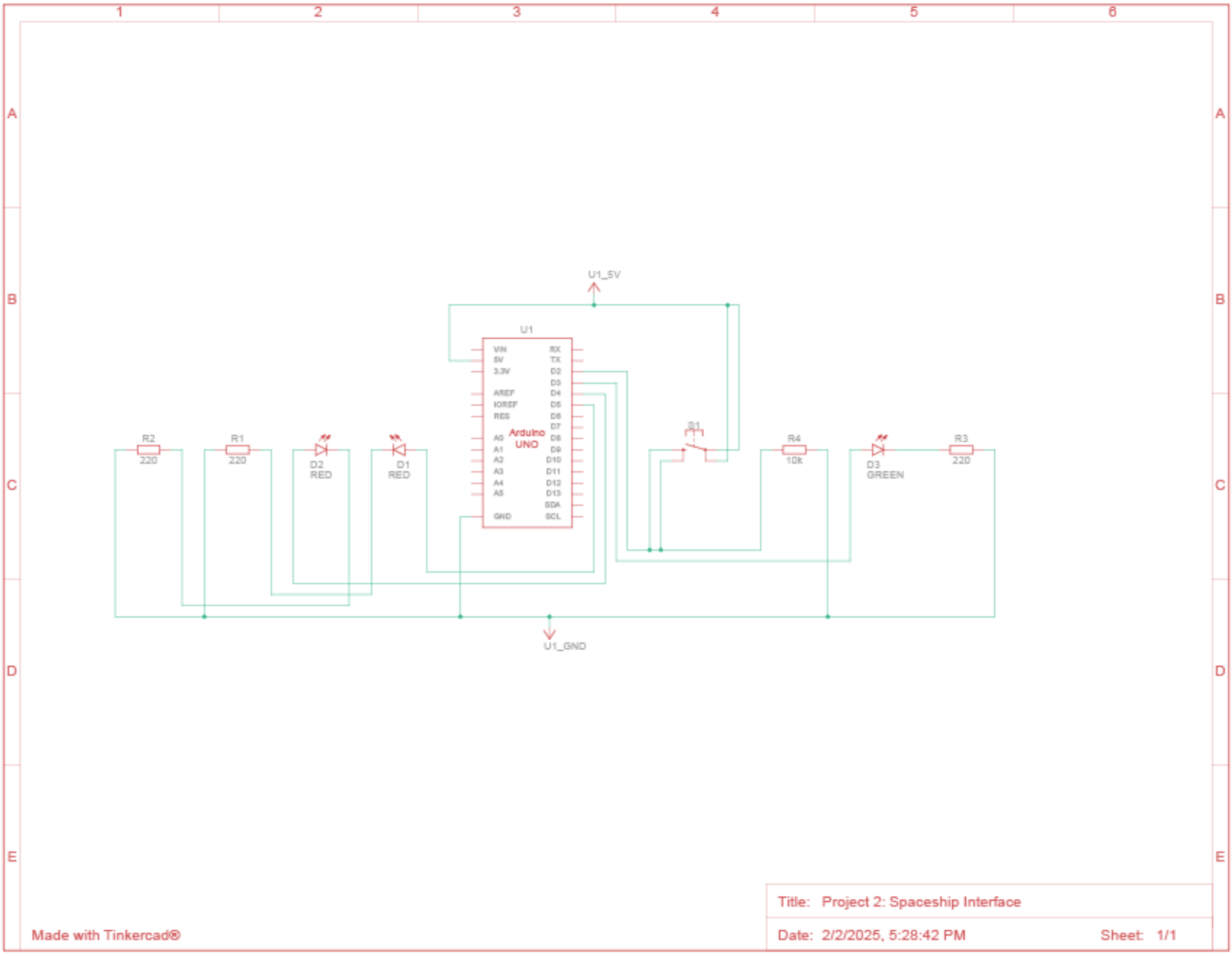
• Jumper Wires

• USB Cable

• Computer with Arduino IDE

Procedure

Circuit Diagram



Steps

• Connected the Arduino's 5V and GND pins to the power and ground rails of the breadboard.

• Placed the green LED's anode on pin 3 and connected its cathode to ground through a 220Ω resistor.

• Placed the two red LEDs' anodes on pins 4 and 5 and connected their cathodes to ground through 220Ω resistors.

• Connected one side of the pushbutton to 5V and the other side to digital pin 2 on the Arduino.

• Added a 10kΩ pull-down resistor between ground and the pushbutton's pin connected to the Arduino.

• Uploaded the provided Arduino sketch and tested the circuit.

Code

```
const int redLed1=5;
const int redLed2=4;
const int greenLed=3;

const int switchPin=2;
void setup() {
  // put your setup code here, to run once:
pinMode(redLed1, OUTPUT);
pinMode(redLed2, OUTPUT);
pinMode(greenLed, OUTPUT);

pinMode(switchPin, INPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
int switchState;

switchState=digitalRead(switchPin);

if(switchState == HIGH){
  digitalWrite(greenLed, LOW);
  digitalWrite(redLed1, HIGH);
  digitalWrite(redLed2, LOW);
  delay(250);
  digitalWrite(redLed1, LOW);
  digitalWrite(redLed2, HIGH);
  delay(250);
} else{
  digitalWrite(redLed1, LOW);
  digitalWrite(redLed2, LOW);
  digitalWrite(greenLed, HIGH);
}


}

```

Output

```
Sketch uses 1160 bytes (3%) of program storage space. Maximum is 32256 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables. Maximum is 2048 bytes.
```

Discussion

1. **What states can a digital pin on the Arduino read?**

   A digital pin on the Arduino can read two states: HIGH (5V) and LOW (0V).

2. **In Arduino programming, what is the purpose of the `void setup()` function?**

   The `void setup()` function is executed once when the Arduino is powered on or reset. It initializes settings such as pin modes and serial communication.

3. **In Arduino programming, what is the purpose of the `void loop()` function?**

   The `void loop()` function runs continuously after `setup()` completes. It contains the main logic of the program and allows the Arduino to execute repeated actions indefinitely.

4. **What is the importance of the `==` operator?**

   The `==` operator is used for comparison in conditional statements. It checks if two values are equal and returns true if they are.

5. **What does the `digitalRead()` command do in Arduino programming?**

   The `digitalRead()` function reads the state of a specified digital pin and returns HIGH (1) or LOW (0).

6. **What does the `delay()` command do?**

   The `delay()` function pauses the execution of the program for a specified number of milliseconds.

7. **What is the ideal resistance and acceptable tolerance levels of a five-band resistor with the following color code: orange, green, red, black, gold?**

   The color code translates to a resistance of 3.5kΩ (3500Ω) with a tolerance of ±5%.

Troubleshooting

1. Issue: Missing semicolons in the code.

   Solution: Used the verify feature in the Arduino IDE before uploading to catch syntax errors.

2. Issue: Video and book instructions differed in expected output.

   Solution: Adjusted the variable in the if statement to match the book's expected output (solid green LED, blinking red LEDs upon button press).

Conclusion

This lab reinforced key programming and circuit design concepts using an Arduino. By integrating digital input and output, I successfully implemented a functioning spaceship interface where LEDs changed states based on button input. Troubleshooting minor issues like syntax errors and expected output mismatches helped solidify my understanding of Arduino coding logic. The project demonstrated the importance of structured programming, real-time debugging, and attention to design specifications, all of which are skills that can be utilized for more complex electronic projects.