Haven's Light is Our Guide

Rajshahi University of Engineering and Technology

Department of Computer Science & Engineering

Lab Report

Course Title

Computer Interfacing and Embedded System Sessional

Course No.

CSE – 3106

Submitted by

Name: Mysha Ahmed
Roll: 2003166
Series: 20
Department of CSE, RUET

Submitted to

Md. Nasif Osman Khansur
Lecturer
Department of CSE, RUET

Experiment 1

Name of the experiment: Blinking LED

Background Information:

Light-emitting diodes (LEDs) are electronic components that emit light when electric current passes through them. The color of the light emitted depends on the materials used in the LED. In this experiment, we have explored the concept of blinking LEDs using a Blue Pill microcontroller and with help of programming.

Objective:

The objective of this experiment was to observe the effect of a program on the blinking behavior of an LED.

Procedure:

1. **Connecting the circuit:**

    - Connect the positive leg (longer leg) of LED to a 220-ohm resistor and connect to 3V3 pin.
    - Connect the negative leg (shorter leg) of the LED to pin 13 (PC13) of the Blue Pill board through jumper wire.

2. **Upload the program:**

    - Open the Arduino IDE software on computer.
    - Copy and pasted the following code into the Arduino IDE software:

Code:

```
void setup() {
    pinMode (PC13, OUTPUT); // Set pin 13 (PC13) as output for LED
}
void loop() {
    digitalWrite(PC13, HIGH); // Turn on LED connected to pin 13
    delay(1000);          // Wait for 1 second (1000 milliseconds)

    digitalWrite(PC13, LOW);  // Turn off LED connected to pin 13
    delay(1000);          // Wait for 1 second (1000 milliseconds)
}
```

3. **Run the program:**

   - Connect the Arduino board to computer using a ST Link cable.
   - Select the appropriate board type “Generic STM32F1 Series” and serial port from the Arduino IDE menus.
   - Click the "Upload" button to upload the program to the Arduino board.
   - Observe the behavior of the LED connected to pin 13(PC13). It should blink on and off with a one-second interval.

Discussion:

The program controls the blinking behavior of the LED by setting the pin PC13 connected to the LED as an output and then using the digitalWrite function to turn the pin on and off at regular intervals. The delay function introduces a pause between turning the LED on and off, which creates the blinking effect.

In this experiment, the blinking interval is set to one second. By modifying the value in the delay function (Ex: delay (100) for faster blink or delay (10000) for slower blink), we can change the blinking speed of the LED.

Experiment 2

Name of the experiment: Turning on an LED

Background Information:

Light-emitting diodes (LEDs) are electronic components that emit light when electric current passes through them. The color of the light emitted depends on the materials used in the LED. In this experiment, we have explored the concept of turning on LEDs using a Blue Pill microcontroller and with help of programming.

Procedure:

1. Connecting the circuit:

- Connect the positive leg (longer leg) of LED to a 220-ohm resistor and connect to 3V3 pin.
- Connect the negative leg (shorter leg) of the LED to pin PA15 of the Blue Pill board through jumper wire.

2 .Upload the program:

- Open the Arduino IDE software on computer.
- Copy and pasted the following code into the Arduino IDE software:

Code:

```
void setup()
    {
    pinMode (PA15, OUTPUT); // Set the LED pin as an output
    digitalWrite(PA15, HIGH);  // Set the LED pin high
    }
```

```
void loop()
{
// We are not doing anything in the loop!
}
```

3. Run the program:

- Connect the Arduino board to computer using a ST Link cable.
- Select the appropriate board type “Generic STM32F1 Series” and serial port from the Arduino IDE menus.
- Click the "Upload" button to upload the program to the Arduino board.
- Observe the behavior of the LED connected to pin PA15 the LED will stay on indefinitely.

Discussion:

The code provided controls the LED using an Arduino board. The pinMode function sets the pin connected to the LED as an output pin. The digitalWrite function sets the voltage on the output pin to high, which turns on the LED. The loop function is empty in this code, so the LED will stay on indefinitely.

Experiment 3

Name of the experiment: LED with Changing Blink Rate

Background Information:

Light-emitting diodes (LEDs) are electronic components that emit light when electric current passes through them. The color of the light emitted depends on the materials used in the LED. In this experiment, we have explored the concept of blinking LEDs using a Blue Pill microcontroller and with help of programming.

Objective:

The objective of this experiment was to observe the effect of a program on the Changing Blink Rate of an LED.

Procedure:

1. Connecting the circuit:

- Connect the positive leg (longer leg) of LED to a 220-ohm resistor and connect to 3V3 pin.
- Connect the negative leg (shorter leg) of the LED to pin PA0 of the Blue Pill board through jumper wire.

2. Upload the program:

- Open the Arduino IDE software on computer.
- Copy and pasted the following code into the Arduino IDE software:

Code:

```
void setup()
        {
                pinMode (PA15, OUTPUT); // Set the LED pin as an output
        }
void loop()
{
        for (int i=100; i<=1000; i=i+100)
        {
                digitalWrite(PA15, HIGH);
                delay(i);
                digitalWrite(PA15, LOW);
                delay(i);
        }
}
```

3. Run the program:

- Connect the Arduino board to computer using a ST Link cable.
- Select the appropriate board type “Generic STM32F1 Series” and serial port from the Arduino IDE menus.
- Click the "Upload" button to upload the program to the Arduino board.
- Observe the behavior of the LED connected to pin PA15. It should blink on and off with a one-second interval.

Discussion:

In the code provided, the for loop iterates from 100 to 1000 in increments of 100. Inside the loop, the digitalWrite function turns on the LED, then the program pauses for the value of the variable i. Then the digitalWrite function turns off the LED, and the program again pauses for the value of the variable i. Because the value of i increases each time through the loop, the amount of time the LED stays on and off increases as well. This creates the blinking effect, with the blink rate slowing down as the experiment progresses.

Experiment 4

Name of the experiment: Fading LED

Background Information:

Light-emitting diodes (LEDs) are electronic components that emit light when an electric current passes through them. The brightness of an LED is related to the amount of current flowing through it. Pulse width modulation (PWM) is a technique that can be used to control the brightness of an LED by rapidly turning it on and off. The duty cycle of the PWM signal determines the average brightness of the LED. A higher duty cycle (more on time than off time) results in a brighter LED.

Objective:

The objective of this experiment was to observe the effect of a program on the fading behavior of an LED.

Procedure:

1. **Connecting the circuit:**

   - Connect the positive leg (longer leg) of LED to a 220-ohm resistor and connect to 3V3 pin.
   - Connect the negative leg (shorter leg) of the LED to pin PB0 of the Blue Pill board through jumper wire.

2. **Upload the program:**

   - Open the Arduino IDE software on computer.
   - Copy and pasted the following code into the Arduino IDE software:

Code:

```
{
pinMode (PB0, OUTPUT); //Set the LED pin as an output
}
void loop()
{
for (int i=0; i<256; i++)
{
analogWrite(PB0, i);
delay(10);
}
for (int i=255; i>=0; i--)
{
analogWrite(PB0, i);
delay(10);
}}
```

3. **Run the program:**

   - Connect the Arduino board to computer using a ST Link cable.
   - Select the appropriate board type “Generic STM32F1 Series” and serial port from the Arduino IDE menus.
   - Click the "Upload" button to upload the program to the Arduino board.
   - Observe the behavior of the LED connected to pin 13(PC13). It should blink on and off with a one-second interval.

Discussion:

The provided code controls the LED using an Arduino board. The pinMode function sets the pin connected to the LED (PB0 in this case) as an output pin. The analogWrite function is used for PWM. In the code provided, the analogWrite function is used to vary the voltage output from 0 to 255. The for loop iterates from 0 to 255, which means the voltage output will increase gradually over each iteration of the loop. This will cause the LED to slowly brighten. After the loop reaches 255, the loop decrements down to 0, causing the LED to slowly dim. The delay function introduces a pause in the program between each iteration of the loop. Hence, the code successfully creates a fading LED effect.

Experiment 5

Name of the experiment: Simple LED control with a button

Background Information:

Buttons and LEDs are common electrical components used in various circuits. Buttons are used as electrical switches to control the flow of current in a circuit. LEDs are electronic components that emit light when an electric current passes through them. In this experiment, we will explore how a button can be used to control the state (on or off) of an LED.

Objective:

The objective of this experiment is to design and build a circuit that controls the on/off state of an LED using a pushbutton.

Procedure:

1. Connecting the circuit:

The circuit is constructed on the breadboard as follows:

- Connect the positive leg of the LED (longer leg) to pin PB0 of the Arduino board.
- Connect a resistor (220 ohm) to the negative leg (shorter leg) of the LED.
- Connect the other end of the resistor to pin PC15 of the Arduino board.
- Connect a jumper wire from the ground pin (GND) of the Arduino board to the negative terminal of the breadboard.
- Connect a jumper wire from the 5V pin of the Arduino board to the positive terminal of the breadboard.
- Connect a jumper wire from one leg of the pushbutton to pin PC15 of the Arduino board.

- Connect a jumper wire from the other leg of the pushbutton to the ground pin (GND) of the Arduino board.

2. **Upload the program:**

- Open the Arduino IDE software on computer.
- Copy and pasted the following code into the Arduino IDE software:

Code:

```
const int LED=PB0;  // The LED is connected to pin 9
const int BUTTON=PC15; // The Button is connected to pin 2
void setup()
{
        pinMode (LED, OUTPUT);  // Set the LED pin as an output
        pinMode (BUTTON, INPUT);  // Set button as input (not required)
}
void loop()
{
        if (digitalRead(BUTTON) == LOW)
        {
                digitalWrite(LED, LOW);
        }
        else
        {
                digitalWrite(LED, HIGH);
        }
}
```

3. Run the program:

- Connect the Arduino board to computer using a ST Link cable.
- Select the appropriate board type “Generic STM32F1 Series” and serial port from the Arduino IDE menus.
- Click the "Upload" button to upload the program to the Arduino board.
- Observe the behavior of the LED connected to pin 13(PC13). It should blink on and off with a one-second interval.

The code uploaded to the Arduino board controls the circuit. It defines two constant variables:

- LED: This variable is assigned the value PB0, which corresponds to pin PB0 on the Arduino board that the LED is connected to.
- BUTTON: This variable is assigned the value PC15, which corresponds to pin PC15 on the Arduino board that the button is connected to.

The setup function prepares the input and output pins for the circuit:

- pinMode(LED, OUTPUT): This line sets pin PB0, which is connected to the LED, as an output pin. The Arduino can only control the voltage on output pins.
- pinMode(BUTTON, INPUT): This line sets pin PC15, which is connected to the button, as an input pin. The Arduino can read the voltage level on input pins.

The loop function continuously monitors the button and controls the LED accordingly:

- digitalRead(BUTTON) == LOW: This condition checks if the voltage level on the button pin (PC15) is low. When the button is not pressed, the pin is connected to ground (GND) which is considered a low voltage level.
- If the condition is true (button is pressed), the digitalWrite(LED, LOW) line sets the voltage on the LED pin (PB0) to low, turning off the LED.
- else: If the button is not pressed (not a low voltage level on the button pin), the digitalWrite(LED, HIGH) line sets the voltage on the LED pin (PB0) to high, turning on the LED.

Discussion:

The code successfully controls the LED using the button. When the button is pressed (low voltage level detected on the button pin), the LED turns off. When the button is not pressed (high voltage level detected on the button pin), the LED turns on. This demonstrates how a simple circuit with a button and an LED can be used to create an interactive electronic device.