Lab Work -04

Course Name: Microprocessor and Microcontroller Lab

Course: CSE 316

Submitted by:

Name:Hashibur Rahman Redoy

ID: 18192103276

Inteke: 41 Section: 03

Program: BSc in CSE (BUBT)

Submitted To:

Jahirul Islam Babar

Lecturer: Dept. of CSE

BUBT

Submission Date: 2-03-2023

1st Code:

```
int led1=6;

void setup() {

  // put your setup code here, to run once:

//pinMode(pin,Mode);

pinMode(led1,OUTPUT);

}

void loop() {

  // put your main code here, to run repeatedly:

digitalWrite(led1,HIGH);

delay(1000);

digitalWrite(led1,LOW);

delay(1000);

}
```

Introduction

The given code is written in the Arduino Integrated Development Environment (IDE) and is intended to control the blinking of an LED connected to Pin 6 of the Arduino board. This code is a basic example of how to use digital input/output pins in Arduino.

Methodology

The code has two main functions: setup() and loop().

In the setup() function, the pinMode() function is used to set the LED1 pin as an output. The pinMode() function sets the mode of the given pin as either INPUT or OUTPUT. In this case, the LED1 pin is set as OUTPUT to drive the LED.

In the loop() function, the digitalWrite() function is used to set the value of the LED1 pin to HIGH, which turns on the LED. The delay() function is used to pause the program for 1000 milliseconds (1 second). Then, the digitalWrite() function is used to set the value of the LED1 pin to LOW, which turns off the LED. Again, the delay() function is used to pause the program for 1000 milliseconds (1 second). This process repeats continuously until the Arduino board is turned off.

Results

The result of running this code is that the LED connected to Pin 6 of the Arduino board will blink on and off repeatedly with a 1 second on-time and a 1 second off-time.

Conclusion

In conclusion, the given code is a simple example of using digital output pins in Arduino. The code uses the pinMode() function to set the LED1 pin as an output and then uses the digitalWrite() function to control the LED connected to it. By repeatedly setting the value of the LED1 pin to HIGH and LOW with a delay of 1 second, the LED will blink on and off continuously.

2nd code:

```
const int a = 3;
const int b = 4;
const int c = 5;
const int d = 6;
const int e = 7;
const int f = 8;
const int g = 9;
const int h = 10;
void setup() {
pinMode(a,OUTPUT);
pinMode(b,OUTPUT);
pinMode(c,OUTPUT);
pinMode(d,OUTPUT);
pinMode(e,OUTPUT);
pinMode(f,OUTPUT);
pinMode(g,OUTPUT);
pinMode(h,OUTPUT);
}
void one()
{
digitalWrite(a,LOW);
digitalWrite(b,HIGH);
digitalWrite(c,HIGH);
digitalWrite(d,LOW);
digitalWrite(e,LOW);
digitalWrite(f,LOW);
digitalWrite(g,LOW);
digitalWrite(h,LOW);
}
void two()
{
digitalWrite(a,HIGH);
digitalWrite(b,HIGH);
digitalWrite(c,LOW);
digitalWrite(d,HIGH);
digitalWrite(e,HIGH);
digitalWrite(f,LOW);
digitalWrite(g,HIGH);
digitalWrite(h,LOW);
}
void three()
{
digitalWrite(a,HIGH);
digitalWrite(b,HIGH);
digitalWrite(c,HIGH);
digitalWrite(d,HIGH);
digitalWrite(e,LOW);
digitalWrite(f,LOW);
digitalWrite(g,HIGH);
digitalWrite(h,LOW);
}
void four()
{
digitalWrite(a,LOW);
digitalWrite(b,HIGH);
digitalWrite(c,HIGH);
digitalWrite(d,LOW);
digitalWrite(e,LOW);
digitalWrite(f,HIGH);
digitalWrite(g,HIGH);
digitalWrite(h,LOW);
}
void five()
{
digitalWrite(a,HIGH);
digitalWrite(b,LOW);
digitalWrite(c,HIGH);
digitalWrite(d,HIGH);
digitalWrite(e,LOW);
digitalWrite(f,HIGH);
digitalWrite(g,HIGH);
digitalWrite(h,LOW);
}
void six()
{
digitalWrite(a,HIGH);
digitalWrite(b,LOW);
digitalWrite(c,HIGH);
digitalWrite(d,HIGH);
digitalWrite(e,HIGH);
digitalWrite(f,HIGH);
digitalWrite(g,HIGH);
digitalWrite(h,LOW);
}
void seven()
{
digitalWrite(a,HIGH);
digitalWrite(b,HIGH);
digitalWrite(c,HIGH);
digitalWrite(d,LOW);
digitalWrite(e,LOW);
digitalWrite(f,LOW);
digitalWrite(g,LOW);
digitalWrite(h,LOW);
}
void eight()
{
digitalWrite(a,HIGH);
digitalWrite(b,HIGH);
digitalWrite(c,HIGH);
digitalWrite(d,HIGH);
digitalWrite(e,HIGH);
digitalWrite(f,HIGH);
digitalWrite(g,HIGH);
digitalWrite(h,LOW);
}
void nine()
{
digitalWrite(a,HIGH);
digitalWrite(b,HIGH);
digitalWrite(c,HIGH);
digitalWrite(d,HIGH);
digitalWrite(e,LOW);
digitalWrite(f,HIGH);
digitalWrite(g,HIGH);
digitalWrite(h,LOW);
}
void zero()
{
digitalWrite(a,HIGH);
digitalWrite(b,HIGH);
digitalWrite(c,HIGH);
digitalWrite(d,HIGH);
digitalWrite(e,HIGH);
digitalWrite(f,HIGH);
digitalWrite(g,LOW);
digitalWrite(h,LOW);
}
void dot()
{
digitalWrite(a,LOW);
digitalWrite(b,LOW);
digitalWrite(c,LOW);
digitalWrite(d,LOW);
digitalWrite(e,LOW);
digitalWrite(f,LOW);
digitalWrite(g,LOW);
digitalWrite(h,HIGH);
```

```
}
void loop() {
  zero();
  delay(1000);
  one();
  delay(1000);
  two();
  delay(1000);
  three();
  delay(1000);
  four();
  delay(1000);
  five();
  delay(1000);
  six();
  delay(1000);
  seven();
  delay(1000);
  eight();
  delay(1000);
  nine();
  delay(1000);
  dot();
  delay(1000);
}
```

Introduction

The provided code is a simple Arduino program that displays digits (0-9) and a decimal point on a seven-segment display. It uses the digitalWrite() function to control the individual segments of the display.

Methodology

The program starts with the definition of constant integer variables a, b, c, d, e, f, g, and h, each representing a pin on the Arduino board. These variables are used to define the pin numbers for the corresponding segments on the seven-segment display.

The setup() function is used to set all of the pins as output pins using the pinMode() function.

The program then defines ten functions - one() through nine() and zero() - each of which uses the digitalWrite() function to turn on or off the appropriate pins to display the corresponding digit on the seven-segment display. There is also a dot() function that lights up the decimal point segment.

In the loop() function, the program calls each of the functions in turn, displaying each digit for one second before moving on to the next.

Results

The program successfully displays each digit (0-9) and the decimal point on the seven-segment display in sequence, with each digit being displayed for one second.

Conclusion

The provided code demonstrates a simple example of how to use an Arduino to control a seven-segment display to display digits and a decimal point. The code can be easily modified to display different patterns or sequences on the display.