1. Abstract
2. Introduction

This lab is to introduce the Arduino UNO microprocessor board based upon the Atmel Atmega 328 P microcomputer and the associated Arduino development environment. In the first lab, it introduces the basic use of LED, Button, serial monitor, lcd and i2c (Inter Integrated Circuit). Besides it shows how to work with c language, the Arduino, and its associated language in the Arduino development environment.

1. Discussion of the lab
   1. Program 1-4
      1. **Brief Design Specification**

The requirements are to modify the Blink program so that LED is on and off for specific seconds and repeats. For the Blink program has implemented the function of blinking, these programs are completed by changing the delay time and right order of digitalWrite().

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Item | Name | Description | Type | Range |
| Inputs | / | / | / | / |
| Output  (Program 1-2) | Pin 13 | LED on and off | / | HIGH,  LOW |
| Output  (Program 3-4) | Pin 8 | LED brick on and off | / | HIGH,  LOW |
| Side effects | / | / | / | / |

Pseudo English description

Setup pin of LED

void loop() {

LED on(seconds);

LED Off(seconds);

}

LED on(seconds){

Write HIGH to LED;

delay for required seconds;

}

LED off(seconds){

Write LOW to LED;

delay for required seconds;

}

* + 1. **Hardware Implementation**

**Software Implementation**

These four programs do almost the same job with the only peripheral LED. There is no external control in these designs. Specific cycles are built to meet the requirements. In program 1 and 2, instructions are directly written in loop function. But in Program 3 and 4, a function is written to turn on the LED or the LED brick, in which HIGH signal is written to the pin and the program will delay for seconds input. The LED can be turned off by changing HIGH into LOW.

* 1. Program 5

Brief Design Specification

it is required to connect the Button Brick and the LED Brick to the microcontroller, and use Button to control LED.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Item | Name | Description | Type | Range |
| 4 | Inputs | Pin 9 | Press the button to send a signal | int | High/ Low |
| 6 | Outputs | Pin 8 | LED on or off | int | High/ Low |
|  | Side effects | / | / | / | / |

Pseudo English description

set the pin of LED and Button

void setup()

{

set the pinMode of LED Output

set the pinMode of Button Input

}

void loop()

{

if Button is pressed

LED on

else //Button is not pressed

LED off

}

Hardware Implementation

Software Implementation

* 1. Program 7

Brief Design Specification

it is required to use the Serial to print the result of Homework Problem 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Item | Name | Description | Type | Range |
| 4 | Inputs | / | / | / | / |
| 6 | Outputs | Serial | Send Message | / | / |
|  | Side effects | result | The result of Homework problem 1 | float | / |

Pseudo English description

define a float varible “result”

void setup()

{  
 set the baud rate of Serial 9600

calculate the result of Homework problem 1

}

void loop()

{  
 print text in the serial monitor : "The price of the trip is $’result’ "

}

No Error

Hardware Implementation

Software Implementation

* 1. Program 8

Brief Design Specification

It is required to print the name of team on the LCD screen

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Item | Name | Description | Type | Range |
| 4 | Inputs | / | / | / | / |
| 6 | Outputs | Pin{2,3,4,5,6,7,8} | lcd display | Receive type char/ int | / |
|  | Side effects | / | / | / | / |

Pseudo English description

define the lcd

void setup()

{

initialize the lcd

print the first two team name on the lcd

set cursor at the second line

print the last name

}

void loop()

{

set the lcd not display, delay 0.5s, set the lcd display, delay 0.5s

}

Hardware Implementation

Software Implementation

* 1. Program 9

In this program, the requests are using the “cursor” function on LCD to let the display move by one space, left or right, by signals from bricks. Actually, the request is to design a game like a tug-of-war, and whether to win or lose the game depends on the frequency of the buttons pressed. Usually, the Arduino’s library does not support the encased function that distinguishes the positive edge of pressing the button, but our team design a new algorithm to distinguish it. Because if the function does not exist, when the two buttons keep pressed, the game will be a tie forever. The function we designed make it a real game.



The flow of control:

|  |
| --- |
| int Button1 = 9; |
| int Button2 = 10; |

Test plan

Firstly, the button pressing is designed to tested, because it is vital to the program. The test contains whether one press gives one space move on the LCD. Secondly, the initialization should be tested to determine that whether the game will re

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Item | Name | Description | Type | Range |
| Inputs | Pin 9 | Press the Button1 to send a signal | int | High/ Low |
| Inputs | Pin 10 | Press the Button2 to send a signal | int | High/ Low |
| Output | Pin{2,3,4,5,6,7,8} | LCD Display | / | / |
| Side effects | / | / | / | / |

Pseudo English description

Setup for LCD, Button1, Button2

void loop() {

If (Read the positive of Button1)

{

Move a space right on LCD;

}

If (Read the positive of Button2)

{

Move a space left on LCD;

}

If(display over right edge of the LCD)

{

Print(“Right win!);

Initialization();

}

If(display over left edge of the LCD)

{

Print(“Left win!);

Initialization();

}

}

Initialization()

{

Put the ‘X’ in the middle of LCD;

}

In the program, we use “cursor” function to move a space on the LCD display. To distinguish the positive edge of the button pressing, we designed a simple structure. We set a flag to conserve the previous value of the button which means when the program find the button pressed, it will firstly inspect the flag, and if it shows the previous value is LOW, the press counts, otherwise, it is an invalid press.

* 1. Program 10:

**Brief Design Specification**

For this program, it is required to work with Bus and send a command form the master to the slave telling it to blink the LED ON or OFF.

Master

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Item | Name | Description | Type | Range |
| Inputs | Pin 8 | Button | / | HIGH,  LOW |
| Outputs | Serial.print | Print messages | / | “LED is ON”,  “LED is OFF” |
|  | Wire.write | Write commands | / | “LED is ON”+1,  “LED is OFF”+0 |
| Side effects | / | / | / | / |

Slave

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Item | Name | Description | Type | Range |
| Inputs | Wire.read | Read commands | / | “LED is ON”+1,  “LED is OFF”+0 |
| Outputs | Serial | Print messages | / | “LED is ON”+1,  “LED is OFF”+1 |
|  | Pin 8 | LED brick on and off | / | HIGH,  LOW |
| Side effects | / | / | / | / |

Pseudo English description

Master:

void loop(){

if (button press){

Send command to turn on the LED;

Print through serial;

}

else{

Send command to turn off the LED;

Print through serial;

}

Slave:

void receiveEvent(how ){

detect the last bytes;

if (it is true)

turn on LED;

else

turn off LED;

}

**Hardware Implementation**

Cut two wires and connect one of them between pins A4 and the second between pins A5 on each of the two boards. Then cut another three wires. Connect one end of the three-pin cables to

the LED brick while connecting the pins as follows: Digital Pin 8, ground and +5 volts on the board.

**Software Implementation**

The graph has shown the main structure. The slave board is controlled by the last byte of the command. If it is 1, then LED will be turned on while the LED will be turned off if it is 0.

1. Test plan:

Program 1-4

Use the USB as power supply, test if LED bricks work as the specification.

Program 5

Firstly, when button is pressed, test whether the LED keeps bright. Secondly, when the press released, test if the LED quenches immediately. Finally, test when the button is pressed and released quickly, whether the LED will blink.

Program 7-8

Test whether  the contents are printed correctly.

Program 9

Firstly, the button pressing is designed to tested, because it is vital to the program. The test contains whether one press gives one space move on the LCD. Secondly, the initialization should be tested to determine that whether the game will reset.

Program 10-11

Test whether the button can control transmission of the I2C block.

1. Analysis

After tests, all programs work correctly and the results proved to be substantiated. The programs meet all the requests of the specification.

1. Summary and Conclusion:

It is the first lab of EE299 which requires the students to design programs on an Arduino UNO board. This lab introduces the Arduino board step by step and implement what they have learned in class into real functions as well. Ten programs are required here including many basic modules and peripherals of microprocessors, such as the LED, button, LCD, Bus and other things. Although it is not very tough to do them right, it must be difficult to do them well. Indeed doing the project well should be the only goal for engineers. To achieve this success, the group must design a good structure before coding, improve the algorithm, and test its robustness and effectiveness.