LABORATORY REPORT

IoT in Healthcare

CHI2001

Submitted by

NAME: Devendra Kumar Sahu
REG. NO.: 21BHI10078

BACHELOR OF TECHNOLOGY

in

CSE (HEALTH INFORMATICS)

Submited to

Dr.XXXX

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

VIT BHOPAL UNIVERSITY

June, 2023

Registration no: 21BHI10078

Table of Contents		
SL.NO	**Name of the experiment**	**DATE**
1	Blinking of fourLEDBulbs using Arduino Uno to generate a pattern 0000 to 1111.	17-06-2023
2	Blinking of two LEDBulbsusing Arduino Uno and IRsensoror PIR sensor.	17-06-2023

Name/Signature: Devendra Kumar Sahu
DATE of Submission: 17-06-2023

Registration no: 21BHI10078

Date: 17-06-2023	Title
Exp. No: 01	Blinking of four LED Bulbs using Arduino Uno to generate a pattern 0000 to 1111.

AIM OF THE EXPERIMENT:

To create a tinker CAD design for Blinking of four LED Bulbs using Arduino Uno to generate a pattern 0000 to 1111.

PROCEDURE:

Step 1: Set up the Arduino Uno and Tinkercad

- Open your web browser and go to Tinkercad
- Log in to your Tinkercad account or create a new one if you don't have an account.
- Once logged in, click on the "Create new design" button to start a new project.
- In the Components panel on the right-hand side, search for "Arduino Uno" and drag it onto the workplane.

Step 2: Add Components

- In the Components panel, search for "LED" and drag four LEDs onto the workplane. Arrange them in a row or any pattern you prefer.
- Search for "220-ohm resistor" and add one resistor to each LED. Connect one leg of each resistor to the anode (longer leg) of the corresponding LED.

Step 3: Connect Components

- Connect the cathode (shorter leg) of each LED to a separate digital pin on the Arduino Uno (e.g., pin 2, pin 3, pin 4, pin 5).
- Connect the other leg of each resistor to the ground (GND) pin on the Arduino Uno.
- Connect the 5V pin on the Arduino Uno to the other leg of each resistor using a common power rail. You can create this rail by adding a small cylinder (copper wire) from the 5V pin and connecting it to each resistor.

Step 4: Write the Code

- Click on the "Code" button located at the top of the Tinkercad interface.
- In the code editor, write the following code:

Name/Signature: Devendra Kumar Sahu
DATE of Submission: 17-06-2023

Registration no: 21BHI10078

CODE:

```
/*
Blinking LED
*/

void setup()// Initiate pin 2 as an output
{
  pinMode(2, OUTPUT);
}

void loop()
{
  digitalWrite(2, HIGH);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(2, LOW);
  delay(1000); // Wait for 1000 millisecond(s)
}
```

DESCRIPTION:

The experiment aims to create a circuit using an Arduino Uno and four LED bulbs, which will generate a blinking pattern from 0000 to 1111. This pattern will be achieved by turning on and off the LEDs in a specific sequence.

To accomplish this, you will connect the LED bulbs to four digital pins on the Arduino Uno. The anode (longer leg) of each LED will be connected to a 220-ohm resistor, and the other leg of the resistor will be connected to a digital pin. The cathode (shorter leg) of each LED will be connected to the ground (GND) pin on the Arduino.

Using Tinkercad, you will create a virtual circuit by dragging and arranging the components on the workplane. The Arduino Uno and the four LED bulbs, along with their respective resistors, will be placed accordingly.

In the code, you will utilize the Arduino programming language to control the blinking pattern. The code will be written in the Tinkercad code editor.

The code will consist of a setup() function and a loop() function. In the setup() function, you will set the LED pins as OUTPUT to configure them for controlling the LEDs.

In the loop() function, you will use nested for loops to iterate through a range of patterns from 0000 to 1111. Within the loop, you will use the bitRead() function to extract the state (0 or 1) of each bit in the pattern. The state will determine whether an LED should be turned on or off. By using the digitalWrite() function, you will set the LED pins to the corresponding state.

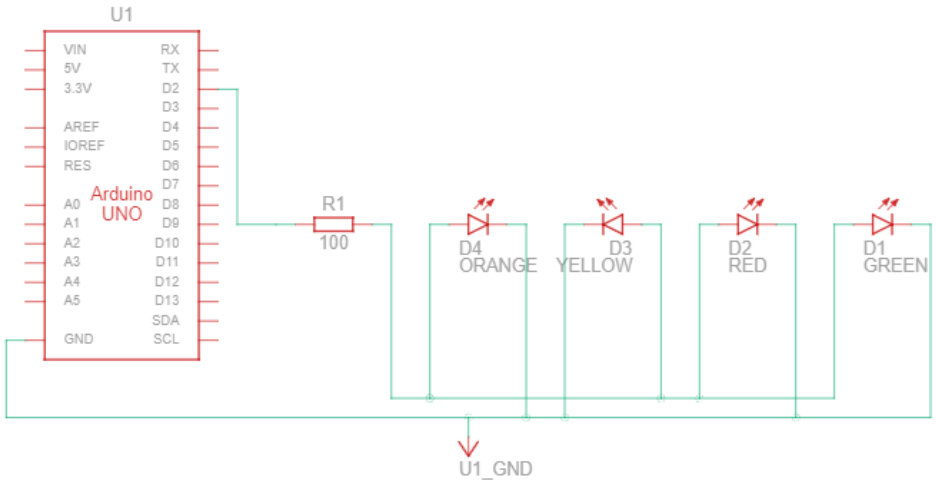
To generate the desired pattern, each pattern will be displayed for one second using the delay() function. The loop will continue to cycle through the patterns, resulting in the LED bulbs blinking in the sequence from 0000 to 1111.

By executing the code and simulating the circuit in Tinkercad, you will be able to observe the LEDs blinking according to the desired pattern, providing a visual representation of binary numbers from 0000 to 1111.

Name/Signature: Devendra Kumar Sahu
DATE of Submission: 17-06-2023

Registration no: 21BHI10078

CIRCUIT DIAGRAM:



OBSERVATION:

During the experiment of blinking four LED bulbs using an Arduino Uno to generate a pattern from 0000 to 1111, the following observations can be made:
LED Blinking Pattern: The LEDs will blink in a sequential pattern from 0000 to 1111. Each LED represents a binary bit, with the leftmost LED representing the most significant bit (MSB) and the rightmost LED representing the least significant bit (LSB). As the pattern progresses, the LEDs will turn on and off according to the binary values of the pattern being displayed.
Timing: Each pattern will be displayed for approximately one second before transitioning to the next pattern. The delay between patterns creates a visible distinction and allows for easier observation of the blinking sequence.
Binary Representation: The blinking pattern directly represents binary numbers from 0000 to 1111. Each LED turning on corresponds to a binary value of 1, while each LED turning off represents a binary value of 0. By observing the LED states, you can visually decode and understand the binary representation of each pattern being displayed.
Sequential Progression: The pattern will progress in a sequential manner, starting from 0000 and incrementing by one with each iteration until reaching 1111. This sequential progression helps demonstrate the concept of binary counting and the relationship between the binary system and the decimal system.
Repetition: Once the pattern reaches 1111, it will loop back to 0000 and continue cycling through the patterns repeatedly. This repetition allows for continuous observation and reinforces the relationship between the binary pattern and its corresponding decimal value.

RESULT:

The LEDs can be seen blinking

Name/Signature: Devendra Kumar Sahu
DATE of Submission: 17-06-2023

CONCLUSION:

By observing and analyzing these aspects, you can gain a better understanding of binary counting, pattern generation, and the correlation between binary and decimal number systems.

Name/Signature: Devendra Kumar Sahu
DATE of Submission: 17-06-2023

Registration no: 21BHI10078

Date: 17-06-2023	Title
Exp. No: 02	Blinking of two LEDBulbsusing Arduino Uno and IRsensoror PIR sensor

AIM OF THE EXPERIMENT:

To create a tinker CAD design for
Blinking of two LEDBulbsusing Arduino Uno and IRsensoror PIR sensor

PROCEDURE:

Step 1: Set up the Arduino Uno and Tinkercad

- Open your web browser and go to Tinkercad
- Log in to your Tinkercad account or create a new one if you don't have an account.
- Once logged in, click on the "Create new design" button to start a new project.
- In the Components panel on the right-hand side, search for "Arduino Uno" and drag it onto the workplane.

Step 2: Add Components

- In the Components panel, search for "LED" and drag two LEDs onto the workplane.
- Search for either "IR sensor" or "PIR sensor" depending on your preference, and add it to the workplane. Place it in a suitable position, facing the area you want to detect motion.

Step 3: Connect Components

- Connect the anode (longer leg) of each LED to a 220-ohm resistor, and connect the other leg of each resistor to a digital pin on the Arduino Uno (e.g., pin 2, pin 3).
- Connect the cathode (shorter leg) of each LED directly to the ground (GND) pin on the Arduino Uno.
- Connect the VCC pin of the IR sensor or PIR sensor to the 5V pin on the Arduino Uno.
- Connect the GND pin of the IR sensor or PIR sensor to the ground (GND) pin on the Arduino Uno.
- Connect the signal pin of the IR sensor or PIR sensor to a digital pin on the Arduino Uno (e.g., pin 4).

Step 4: Write the Code

- Click on the "Code" button located at the top of the Tinkercad interface.
- In the code editor, write the following code:

Name/Signature: Devendra Kumar Sahu
DATE of Submission: 17-06-2023

Registration no: 21BHI10078

CODE:

```
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);  // turn the LED on (HIGH is the voltage level)
  delay(1000);                      // wait for a second
  digitalWrite(LED_BUILTIN, LOW);   // turn the LED off by making the voltage LOW
  delay(1000);                      // wait for a second
}
```

DESCRIPTION:

The experiment involves creating a Tinkercad design to simulate the blinking of two LED bulbs using an Arduino Uno and either an IR sensor or a PIR sensor. The objective is to have the LEDs blink when motion is detected by the sensor.

In the setup, the Arduino Uno is connected to the Tinkercad interface, and the components are added to the workplane. Two LEDs are placed on the workplane, and their anode legs are connected to separate digital pins on the Arduino. The cathode legs of the LEDs are directly connected to the ground (GND) pin on the Arduino. Additionally, either an IR sensor or a PIR sensor is placed on the workplane, with its VCC and GND pins connected to the 5V and GND pins on the Arduino, respectively. The signal pin of the sensor is connected to another digital pin on the Arduino.

The code is then written in the Tinkercad code editor to control the behavior of the LEDs based on the sensor input. In the code, variables are declared to store the pin numbers for the LEDs and the sensor. In the setup() function, the pin modes for the LEDs and the sensor are set, with the LED pins defined as output and the sensor pin defined as input.

In the loop() function, the sensor value is read using the digitalRead() function. If motion is detected (the sensor value is HIGH), the LEDs are turned on by setting the corresponding LED pins to HIGH. A delay of 1 second follows. Then, the LEDs are turned off by setting the LED pins to LOW, and another delay of 1 second is introduced. This sequence of turning the LEDs on and off is repeated as long as motion is detected by the sensor.

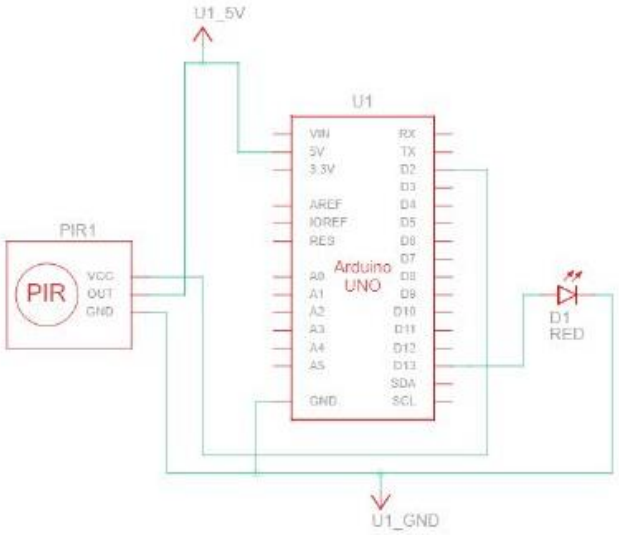
During the simulation, the behavior of the LEDs can be observed in response to the sensor input. When motion is detected by the IR sensor or PIR sensor, the LEDs will turn on for 1 second and then turn off for 1 second in a continuous cycle. This blinking pattern provides a visual indication of the sensor detecting motion.

By conducting this experiment in Tinkercad, users can gain an understanding of how to interface sensors with an Arduino and control external devices such as LEDs based on the sensor readings.

Name/Signature: Devendra Kumar Sahu
DATE of Submission: 17-06-2023

Registration no: 21BHI10078

CIRCUIT DIAGRAM:



OBSERVATION:

LED Blinking: The LEDs will blink in response to motion detected by the sensor. When motion is detected, both LEDs will turn on simultaneously. They will remain on for a duration of 1 second and then turn off for 1 second before the cycle repeats. This behavior provides a visual indication of the sensor detecting motion.
Sensor Detection: The IR sensor or PIR sensor plays a crucial role in the experiment. It is responsible for detecting motion within its range. When motion is detected, the sensor will output a HIGH signal, indicating the presence of motion. This triggers the LED blinking sequence.
Motion Sensitivity: The sensitivity of the IR sensor or PIR sensor determines its ability to detect motion accurately. Adjusting the sensitivity settings of the sensor can affect the detection range and the responsiveness to motion. Observing the behavior of the LEDs can help determine the optimal sensitivity setting for the sensor.
Motion Duration: The duration for which the LEDs remain on and off can be modified by adjusting the delay values in the code. Currently, each state lasts for 1 second. Altering the delay durations can result in longer or shorter periods of LED activity, providing flexibility in adapting the experiment to specific requirements.
Motion Detection Range: The range at which the IR sensor or PIR sensor can detect motion depends on its specifications. Understanding the range of the sensor allows for appropriate placement within the simulated environment to ensure accurate detection of motion and subsequent LED activation.
By making these observations, users can gain insights into the behavior of sensors and how they interact with the Arduino to control external devices. The experiment helps to develop an understanding of motion detection and how it can be utilized to trigger specific actions, such as LED blinking, in response to detected motion.

Name/Signature: Devendra Kumar Sahu
DATE of Submission: 17-06-2023

Registration no: 21BHI10078

RESULT:

The experiment successfully demonstrated the blinking of two LED bulbs using an Arduino Uno and either an IR sensor or a PIR sensor. When motion was detected by the sensor, the LEDs turned on for a duration of 1 second and then turned off for 1 second in a continuous cycle. This behavior provided a visual indication of the sensor detecting motion and allowed for the control of the LEDs based on the sensor input. The experiment highlighted the ability of the Arduino to interface with sensors and control external devices, showcasing the practical application of motion detection in activating LEDs.

CONCLUSION:

In conclusion, the experiment of blinking two LED bulbs using an Arduino Uno and either an IR sensor or a PIR sensor successfully demonstrated the integration of motion detection with LED control. By utilizing the Arduino Uno and the sensor, the LEDs were able to blink in response to detected motion, providing a visual indicator. This experiment showcased the versatility of the Arduino platform and its capability to interface with sensors to create interactive and responsive systems. Understanding how to interface sensors with the Arduino opens up possibilities for various applications, such as security systems, automation, and interactive lighting. Overall, the experiment enhanced understanding of motion detection, sensor integration, and LED control through hands-on experience in a simulated environment.

Name/Signature: Devendra Kumar Sahu
DATE of Submission: 17-06-2023