## Experiment-1A:Software Delay - Code

### 

#include "at89c51ed2.h"

void delay(unsigned int);

void main(void) {

while(1) {

P0 = 0x00;

delay(100);

P0 = 0x10;

delay(100);

P0 = 0x20;

delay(100);

P0 = 0x30;

delay(100);

}

}

void delay(unsigned int itime) {

unsigned int i, j;

for (i = 0; i < itime; i++) {

for (j = 0; j < 1275; j++);

}

}

## Experiment-1B: Hardware Delay - Code

#include "at89c51ed2.h"

void T0M1Delay();

void main(void) {

while(1) {

P0 = 0x00;

T0M1Delay();

P0 = 0x10;

T0M1Delay();

P0 = 0x20;

T0M1Delay();

P0 = 0x30;

T0M1Delay();

}

}

void T0M1Delay() {

unsigned char z;

for (z = 0; z < 5; z++) {

TMOD = 0x01; // configuring Timer 0 in mode 1

TH0 = 0x4B;

TL0 = 0xFD;

TR0 = 1; // start the timer

while (TF0 == 0); // monitor the timer overflow flag

TR0 = 0; // stop the timer

TF0 = 0; // clear timer overflow flag

}

}

## Experiment-2: Code

##### Square and Rectangular waveform

#include "at89c51ed2.h"

void delay(unsigned int);

void main(void) {

while(1) {

P0 = 0x00;

delay(200);

P0 = 0xff;

delay(200);

}

}

void delay(unsigned int itime) {

unsigned int i, j;

for (i = 0; i < itime; i++) {

for (j = 0; j < 1275; j++);

}

}

Square wave: equal delay (200, 200)

Rectangular wave: unequal delay (100, 200)

##### Triangular waveform

#include "at89c51ed2.h"

void main(void) {

unsigned char count;

while(1) {

for (count = 0; count != 0xff; count++)

P0 = count;

for (count = 0xff; count != 0; count--)

P0 = count;

}

}

##### Positive ramp

#include "at89c51ed2.h"

void main(void) {

unsigned char count;

while(1) {

for (count = 0; count != 0xff; count++){

P0 = count;

}

P0 = 0;

}

}

##### Negative ramp

#include "at89c51ed2.h"

void main(void) {

unsigned char count;

while(1) {

for (count = 0xff; count != 0; count–){

P0 = count;

}

P0 = 0xff;

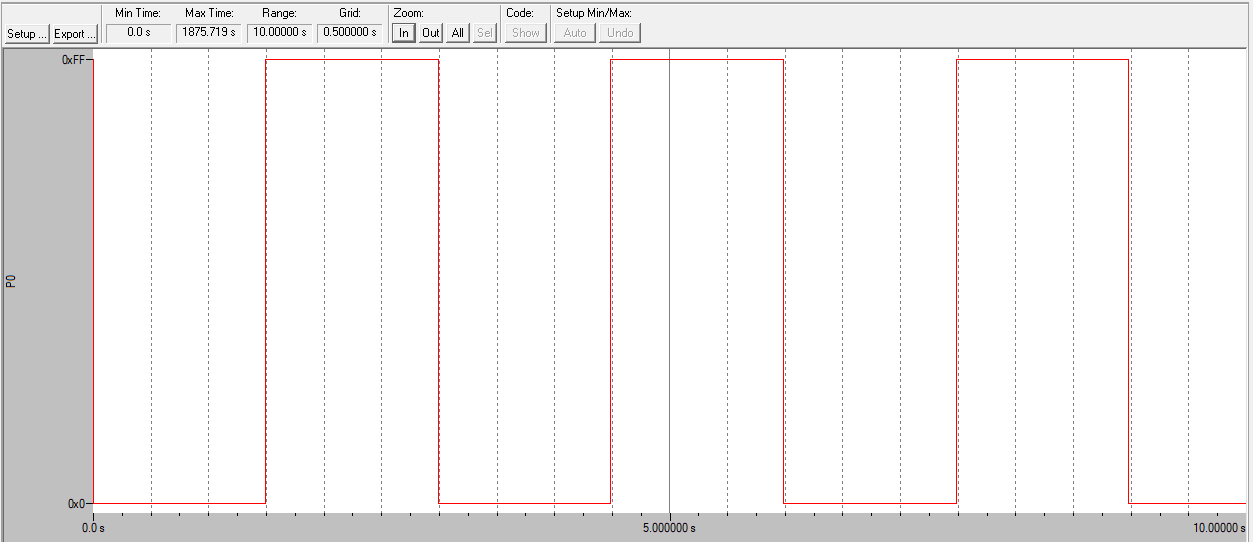
}

}

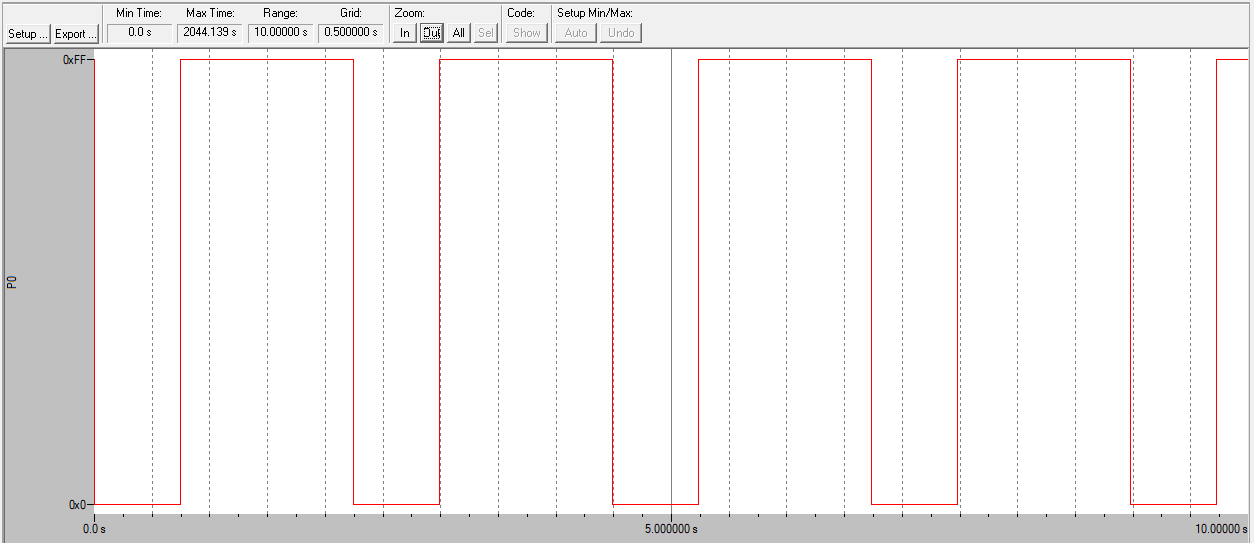
## Experiment-2: Outputs

##### 

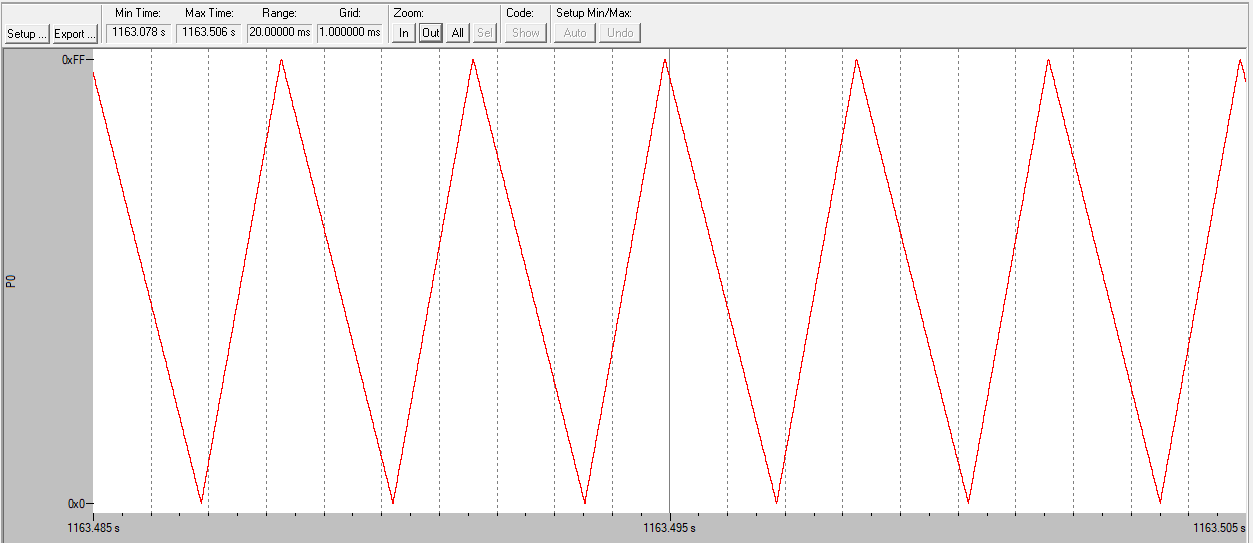
##### Square wave (TON = TOFF)



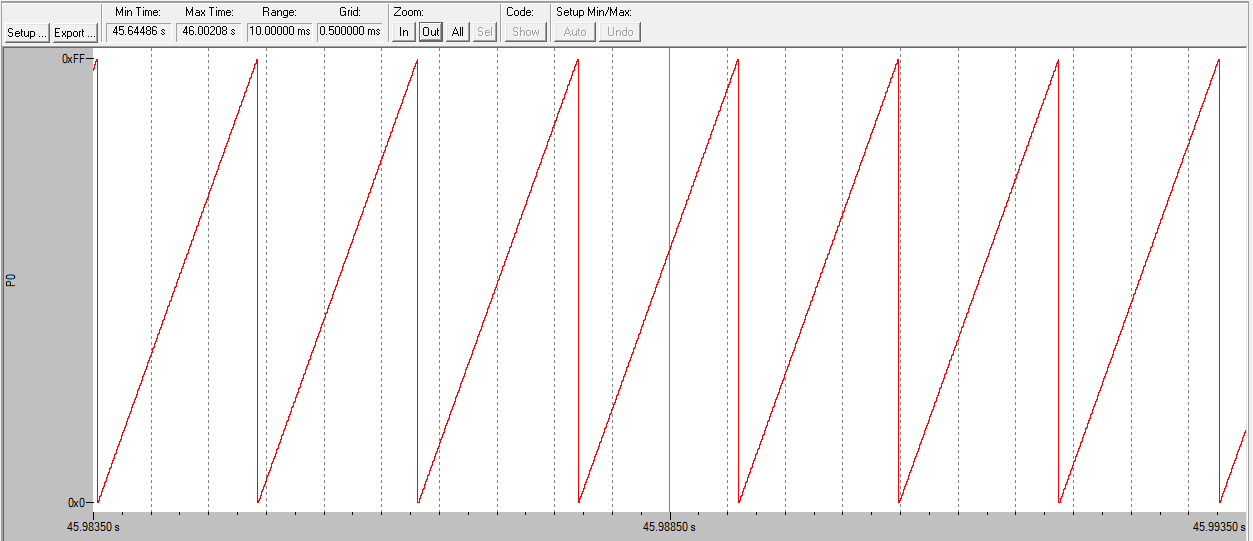
##### Rectangular wave (TON ≠ TOFF)



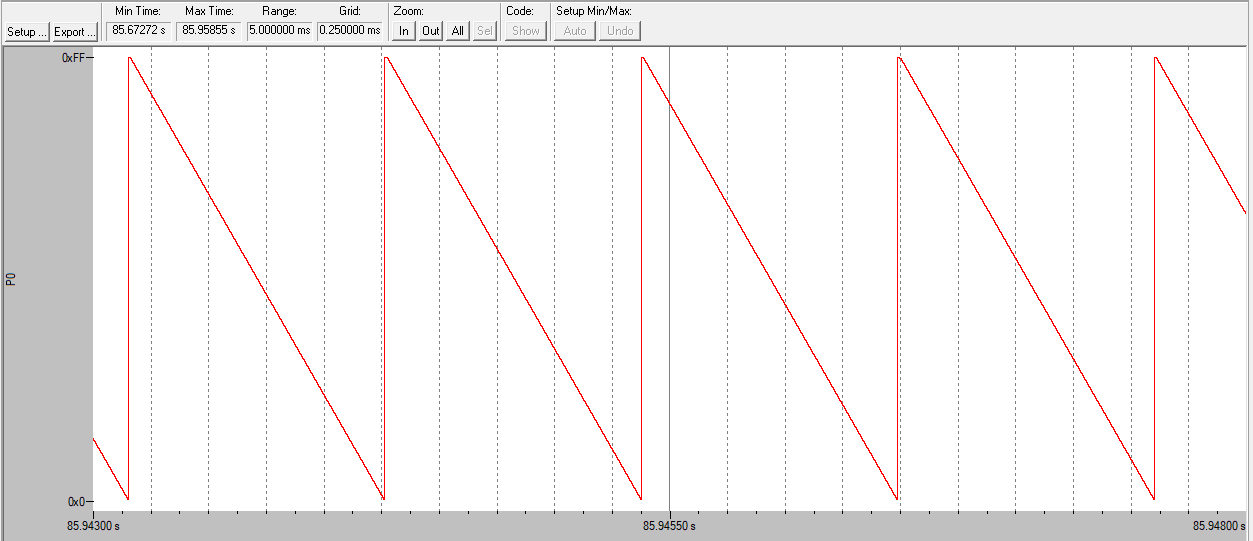
##### Triangular wave



##### Positive ramp



##### Negative ramp



## Experiment-3: Code

NOTE: Add LCD\_ROUTINE header to the source group

#### Case-1 (output on first row)

##### Code

#include "at89c51ed2.h"

#include <intrins.h>

// LCD function prototype

void lcd\_init(void);

void lcd\_comm(void);

void lcd\_data(void);

unsigned char xdata arr[16] = {"HELLO WORLD"};

unsigned char temp1 = 0x00;

unsigned char temp2;

unsigned int i = 0;

void main(void) {

AUXR = 0x10;

lcd\_init();

temp1 = 0x80;

lcd\_comm();

for (i = 0; i < 12; i++) {

temp2 = arr[i];

lcd\_data();

}

}

##### Output



#### 

#### Case-2 (output on rows 1&2)

##### Code

#include "at89c51ed2.h"

#include<intrins.h>

// LCD function prototype

void lcd\_init(void);

void lcd\_comm(void);

void lcd\_data(void);

unsigned char xdata arr[16] = {"Line One"};

unsigned char xdata arr1[16] = {"Line Two"};

unsigned char temp1 = 0x00;

unsigned char temp2;

unsigned int i = 0;

void main(void) {

AUXR = 0x10;

lcd\_init();

temp1 = 0x80;

lcd\_comm();

for (i = 0; i < 8; i++) {

temp2 = arr[i];

lcd\_data();

}

temp1 = 0xC0;

lcd\_comm();

for (i = 0; i < 8; i++) {

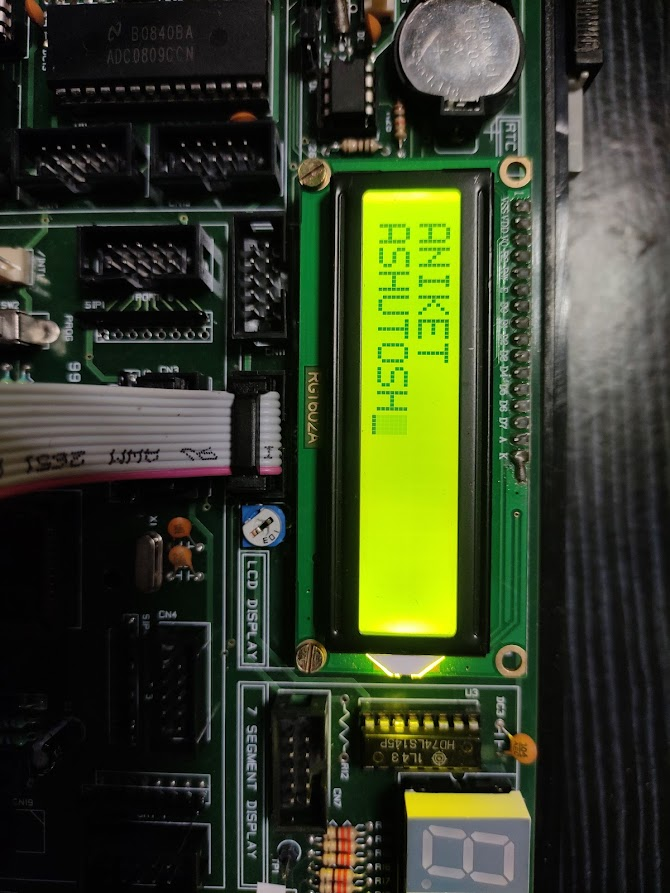
temp2 = arr1[i];

lcd\_data();

}

}

##### Output



## 

## Experiment-4

NOTE: This experiment is only for demonstration purposes.

##### Code

/\* This program displays the temperature sensor output of the ADC0809 IC on LCD

/\* display \*/ //#include "D:\Keil\C51\INC\Atmel\reg51f.h"

#include<at89c51ed2.h>

#include<stdio.h>

// LCD FUNCTION PROTOTYPE

void lcd\_init(void);

void lcd\_comm(void);

void lcd\_data(void);

void delay(int);

sbit EOC = P0^4;

sbit START\_ALE = P0^7;

unsigned char xdata arr1[12]={"ADC Value = "};

unsigned char xdata arr2[12]={"TEMP 'C = "};

unsigned char temp\_msg[]={" "};

unsigned char temp1=0x00;

unsigned char temp2;

unsigned char buf[8],a;

unsigned char i,temp\_hi,temp\_low,adc\_val;

unsigned int vtemp1;

float analog\_val,temp\_float;

#define VREF 5000

#define FULLSCALE 0xff

void main ()

{

START\_ALE = 0;

lcd\_init();

temp1 = 0x80; // Display the data from first position of first line

lcd\_comm(); // Command Writing

for(i=0;i<12;i++)

{

temp2 = arr1[i];

lcd\_data(); // Data Writing

}

temp1 = 0xC0; // Display the data from first position of first line

lcd\_comm(); // Command Writing

for(i=0;i<12;i++)

{

temp2 = arr2[i];

lcd\_data(); // Data Writing

}

P1 = 0xff; // Configure P1 as input to read the ADC o/p

delay(200);

while(1)

{

P0 &= 0xFA; // Select the temparrature sensor as input channel

// EOC = 0;

START\_ALE=1;

delay(5);

START\_ALE = 0;

do

{

vtemp1=P0;

vtemp1=vtemp1 & 0x10;

} while(vtemp1 == 0x10); // POLL EOC LINE HI TO LOW

do

{

vtemp1=P0;

vtemp1=vtemp1 & 0x10;

} while(vtemp1 == 0x00); // LOW TO HIGH

// EOC = 1; // SET

adc\_val = P1;// display adc result on the data field

analog\_val = ((float)adc\_val \* (float)VREF)/(float)FULLSCALE;

temp\_float = (float)(analog\_val - 2731.4)/10;

temp\_float = (int)(temp\_float\*10);

i = 100; // used to extract digit of temp\_float

for(a=0;a<4;a++)

{

buf[a] = temp\_float / i;

temp\_float -= buf[a] \* i;

buf[a] += '0';

i /= 10;

}

buf[3] = buf[2];

buf[2] = buf[1];

buf[2] = '.';

buf[4] = '\0';

temp1 = 0xCC;

lcd\_comm();

for(i=0;(buf[i]!='\0');i++)

{

temp2 = buf[i];

lcd\_data();

}

temp\_hi=adc\_val>>4;

temp\_hi=temp\_hi & 0x0f;

temp\_low=adc\_val & 0x0f;

if(temp\_hi>9) // Convert the received ADC o/p into ASCII code

temp\_hi = temp\_hi + 0x37;

else

temp\_hi = temp\_hi + '0';

if(temp\_low>9)

temp\_low = temp\_low + 0x37;

else

temp\_low = temp\_low + '0';

delay(100);

temp\_msg[1] = temp\_hi ;

temp\_msg[2] = temp\_low ;

temp1 = 0x8C; // To display the ADC o/p

lcd\_comm(); // Command Writing

temp2 = temp\_hi;

lcd\_data();

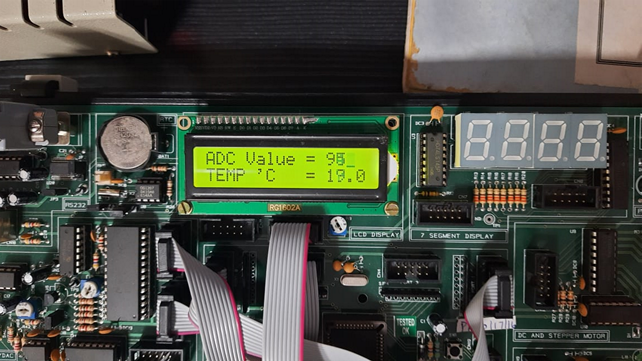
temp2 = temp\_low;

lcd\_data();

} // end of while(1)

}

##### Output



## 

## Experiment-5

##### Code

const int buttonPin1 = 13;

int buttonState1 = LOW;

const int ledPin1 = A5;

const int buttonPin2 = 12;

int buttonState2 = LOW;

const int ledPin2 = A4;

const int buttonPin3 = 11;

int buttonState3 = LOW;

const int ledPin3 = A3;

const int buttonPin4 = 10;

int buttonState4 = LOW;

const int ledPin4 = A2;

void setup() {

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

pinMode(buttonPin1, INPUT);

pinMode(ledPin1, OUTPUT);

pinMode(buttonPin2, INPUT);

pinMode(ledPin2, OUTPUT);

pinMode(buttonPin3, INPUT);

pinMode(ledPin3, OUTPUT);

pinMode(buttonPin4, INPUT);

pinMode(ledPin4, OUTPUT);

Serial.begin(9600);

}

void loop() {

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

buttonState1 = digitalRead(buttonPin1);

buttonState2 = digitalRead(buttonPin2);

buttonState3 = digitalRead(buttonPin3);

buttonState4 = digitalRead(buttonPin4);

if (buttonState1 == HIGH) {

digitalWrite(ledPin1, LOW);

}

else {

digitalWrite(ledPin1, HIGH);

}

Serial.println(buttonState1);

if (buttonState2 == HIGH) {

digitalWrite(ledPin2, LOW);

}

else {

digitalWrite(ledPin2, HIGH);

}

Serial.println(buttonState2);

if (buttonState3 == HIGH) {

digitalWrite(ledPin3, LOW);

}

else {

digitalWrite(ledPin3, HIGH);

}

Serial.println(buttonState3);

if (buttonState4 == HIGH) {

delay(1000);

digitalWrite(ledPin4, LOW);

}

else {

digitalWrite(ledPin4, HIGH);

}

Serial.println(buttonState4);

}

##### 

##### Output

## Experiment-6

##### Code

#include <dht.h>

#define DHT11\_PIN 4

dht DHT;

void setup() {

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

Serial.begin(9600);

}

void loop() {

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

int chk = DHT.read11(DHT11\_PIN);

Serial.print("Temperature: ");

Serial.println(DHT.temperature);

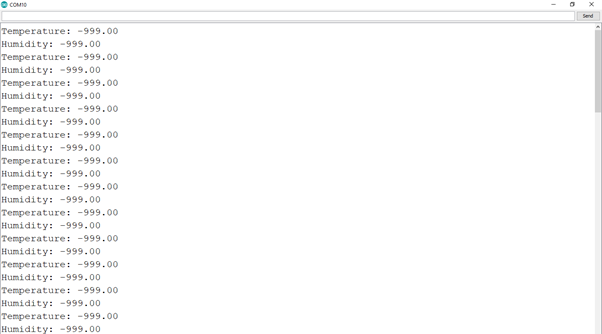
Serial.print("Humidity: ");

Serial.println(DHT.humidity);

delay(200);

}

##### Output



## Experiment-7

##### Code

int light\_pin = 5;

void setup() {

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

pinMode(light\_pin, INPUT);

Serial.begin(9600);

}

void loop() {

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

int light\_data = digitalRead(light\_pin);

if (light\_data)

Serial.println("Light not detected!");

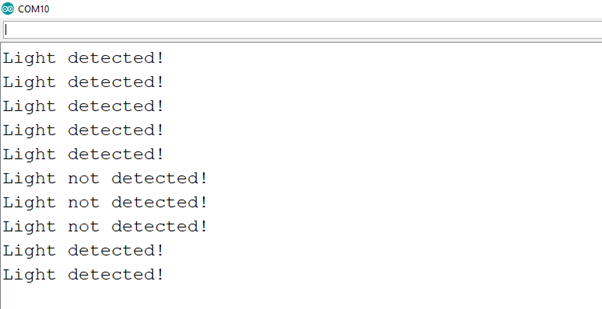
else

Serial.println("Light detected!");

delay(1000);

}

##### Output



## Experiment-8

##### Code

int relay\_pin = 8;

void setup() {

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

pinMode(relay\_pin, OUTPUT);

Serial.begin(9600);

digitalWrite(relay\_pin, HIGH);

}

void loop() {

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

digitalWrite(relay\_pin, LOW);

Serial.println("Relay is OFF");

delay(1000);

digitalWrite(relay\_pin, HIGH);

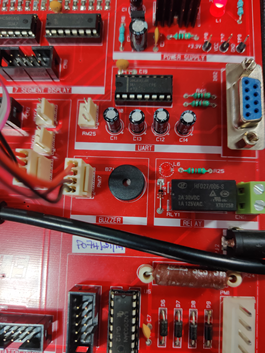
Serial.println("Relay is ON");

delay(1000);

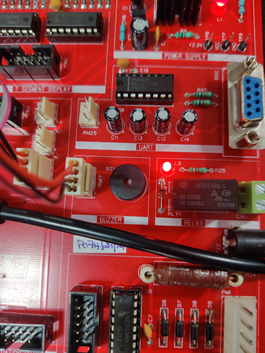
}

##### Output

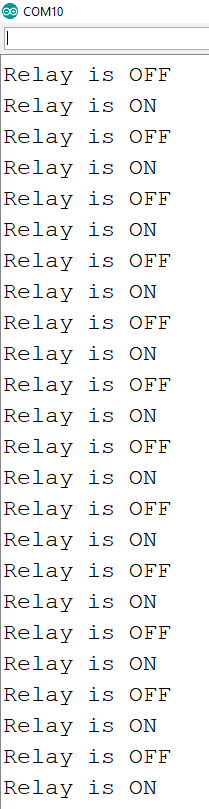
Relay OFF, LED OFF



Relay ON, LED ON



Serial Monitor



## Experiment-9

##### Code

import RPi.GPIO as GPIO

import time

import math

import requests

import json

import Adafruit\_DHT as dht

import urllib3

import Adafruit\_GPIO.SPI as SPI

import Adafruit\_MCP3008

myAPI = '8N16NCBNW7HCBJFP'

baseURL = 'https://api.thingspeak.com/update?api\_key=8N16NCBNW7HCBJFP&field1=0'

SPI\_PORT = 0

SPI\_DEVICE = 0

mcp = Adafruit\_MCP3008.MCP3008(spi=SPI.SpiDev(SPI\_PORT, SPI\_DEVICE))

GPIO.setWarnings(False)

GPIO.setMode(GPIO.BCM)

GPIO.setup(13, GPIO.IN)

GPIO.setup(19, GPIO.OUT)

GPIO.setup(26, GPIO.IN)

sensor = dht.DHT11

dht11\_pin = 4

temp = 0.0

humidity = 0.0

while True:

try:

print("Scanning for sensors data....")

time.sleep(2)

print("---------------------")

humidity, temp = dht.read\_retry(sensor, dht11\_pin)

if math.isnan(temp) == False and math.isnan(humidity) == False:

print("-----------------")

print("Temperature = %.02fC" % (temp))

print("-----------------")

print("Temperature = %.02f%%" % (humidity))

print("-----------------")

conn = urllib3.urlopen(baseURL + "&field = %s&field2 = %s" % (temp, humidity))

print(conn)

print(temp + " " + humidity)

conn.close()

print("")

print("Data sent to cloud")

print("-----------------")

print("")

print("")

time.sleep(2)

except:

print('exception')

break

## 

## Experiment-10

##### Code