***EMBEDDED SYSTEMS ( 3-1 )***

“ARDUINO PROGRAMS” + PIC PROGRAMS

***QUESTIONS***

1. Write down the pin description of Arduino UNO. Difference between microcontroller and microprocessor

2. Write a Arduino source code to blink 4 LED’s.

3.Write a PIC16F877A code turn on the LED using Switch (odd time on & even time off).

4.Write down the PIC16F877A code for 8 LED’s (turn Led’s on in even and odd pattern)

5.Write a Arduino code interfacing for 1 switch & 1 led.

6.Differentiate the switch code in Arduino UNO & PIC16F877A controller.

7.Write an Arduino code to interface LCD for displaying the name.

8. Write an Arduino code to display the person count inside the room.

9.Write an Arduino to display the ADC value in LCD.

10. Write a PIC16F877A code to transmit the data in virtual terminal.

11.Write an Arduino code to turn on the load based on the temperature sensor value.

12.Write an Arduino code to control the load according to the data received from the HCO5 Bluetooth module.

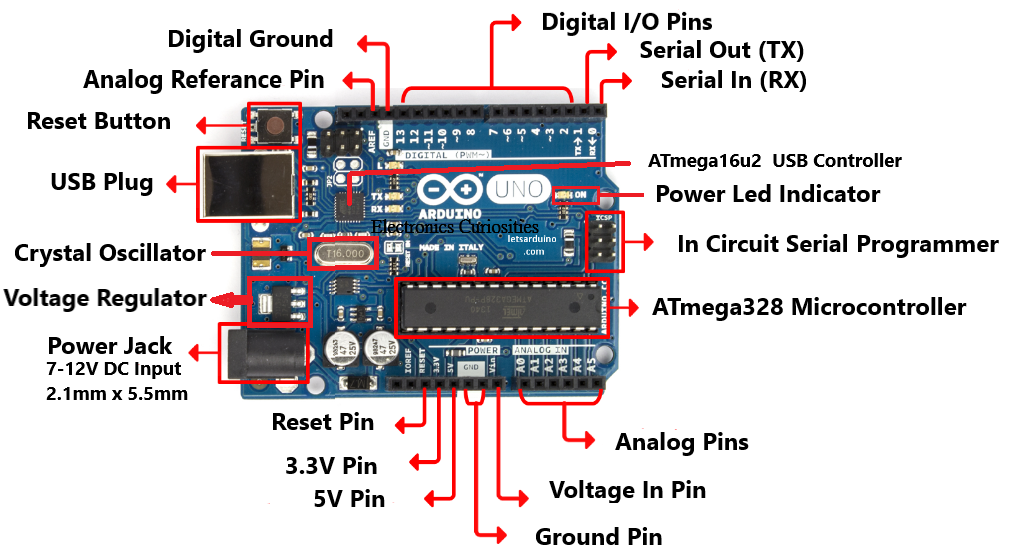
***ANSWERS***

1. A) Write down the pin description of Arduino UNO. B)Difference between microcontroller and microprocessor

Ans:

**Aim:**

**Arduino Pin Desciption**:



1. **Power USB**  
   Arduino board can be controlled by utilizing the USB cable from your PC/Laptop. You should simply interface the USB cable  to the USB port of arduino.

2) **Power Jack (7 - 12v DC) (2.1mm x 5.5mm)**  
 Arduino sheets can be powered straightforwardly from the DC power supply by associating it to the Barrel Jack

3)  **Voltage Regulator**  
The capacity of the voltage controller is to control the voltage given to the Arduino board and settle the DC voltages utilized by the Microcontroller and different components.

4) **Crystal Oscillator ( 16 MHz)**  
  
The gem oscillator helps Arduino in managing time issues. The number imprinted over the Arduino precious stone is 16.000H9H. It discloses to us that the recurrence is 16,000,000 Hertz or 16 MHz

5) **Arduino Reset Pin**  
  
You can reset your Arduino board using this pin i.e., begin your program from the earliest starting point. You can reset the UNO board in two different ways. Initially, by utilizing the reset catch on the board. Second, you can interface an outside reset catch to the Arduino stick marked RESET

6) **Pins (3.3v, 5v, GND, Vin)**  
 **• Pin  3.3V − Supply 3.3  volt**  
 **• Pin  5V − Supply 5  volt**  
  
• Most of the parts utilized with Arduino board works fine with 3.3 volt and 5 volt.  
  
• **Pin  :  GND (Ground)**− There are a few GND sticks on the Arduino, any of which can be         utilized to ground your circuit.  
  
• **Pin  : Vin** − This stick additionally can be utilized to control the Arduino board from an   outside power source.  
  
7) **Analog Pins  (A0 – A5)**  
  
The Arduino UNO board has six Analog information pins A0 to A5. These pins can peruse the sign from an Analog sensor like the stickiness sensor or temperature sensor and convert it into a computerized worth that can be perused by the microcontroller.  
  
8)  **Microcontroller (ATmega328)**  
  
Each Arduino board has its own microcontroller (ATmega328). You can accept it as the cerebrum of your board. The principle IC on the Arduino is somewhat not quite the same as board to board. The microcontrollers are more often than not of the ATMEL Company. You should comprehend what IC your block has before stacking another program from the Arduino IDE. This data is accessible on the highest point of the IC.  
  
9) **ICSP Pins**  
  
For the most part, ICSP is an AVR, a small programming header for the Arduino comprising of MOSI, MISO, SCK, RESET, VCC, and GND. It is regularly alluded to as a SPI: Serial Peripheral Interface, which could be considered as an "extension"  for more use .  
  
10)  Power LED marker (indicator)  
  
This LED should illuminate when you plug your Arduino into a power source to show that your barricade is controlled effectively. In the event that this light does not turn on, at that point there is some kind of problem with the association.  
  
11) **TX and RX LEDs**  
  
TX (transmit) and RX (recieve). They show up in two places on the Arduino UNO board. Initially, at the computerized pins 0 and 1, to demonstrate the pins in charge of sequential correspondence. Second, the TX and RX drove . The TX drove flashes with various speed while sending the sequential information. The speed of blazing relies upon the baud rate utilized by the board. RX flashes during the getting procedure.  
  
12)  **Digital I/O Pins**  
  
The Arduino UNO board has 14 advanced I/O pins  (of which 6 give PWM (Pulse Width Modulation) yield. These pins can be arranged to function as information computerized pins to peruse rationale esteems (0 or 1) or as advanced yield pins to drive various modules like LEDs, transfers, and so forth. The pins marked "~" can be utilized to produce PWM.  
  
13) **AREF    (Analog Reference)**  
  
AREF represents Analog Reference. It is some of the time, used to set an outer reference voltage (somewhere in the range of 0 and 5 Volts) as far as possible for the Analog information pins.  
  
14)  **ATmega16u2 (USB Controller)**  
     Use to read the data from USB port to understand  the data to main Microcontroller.  
  
15)  **Reset Button**  
     Use to reset the ATmega328 Microcontroller by pressing the button

**Microprocessor**

* It can be understood as the heart of the computer system.
* It is a processor where the memory and I/O component are connected externally.
* The circuit is complex due to external connection.
* It can’t be used in compact system.
* It is not efficient.
* It has less number of registers.
* Most of the operations are based on memory.
* It has a zero status flag.
* It is generally used in personal computers.

**Microcontroller**

* It can be understood as the heart of the embedded system.
* It is a controlling device wherein the memory and I/O output component are present internally.
* It is present on chip memory.
* The memory and I/O components are available.
* The circuit is less complex.
* It doesn’t have a zero status flag.
* It can be used with a compact system.
* It is efficient.
* It is generally used in washing machines, and air conditioners.

2).Write a Arduino source code to blink 4 LED’s.

**Aim:**

**Program**::

void setup() {

pinMode(1,OUTPUT);

pinMode(2,OUTPUT);

pinMode(3,OUTPUT);

pinMode(4,OUTPUT);

}

void loop() {

digitalWrite(1,1);

digitalWrite(2,1);

digitalWrite(3,1);

digitalWrite(4,1);

delay(500);

digitalWrite(1,0);

digitalWrite(2,0);

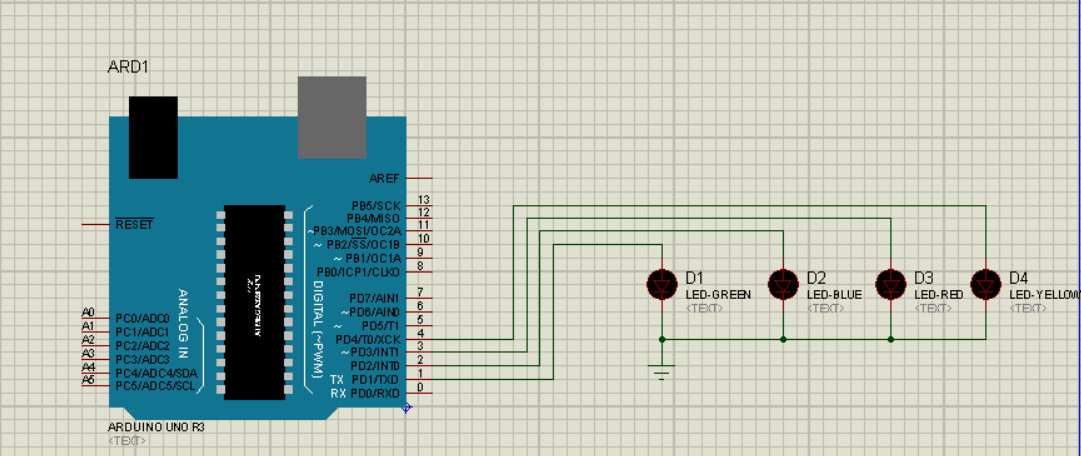
digitalWrite(3,0);

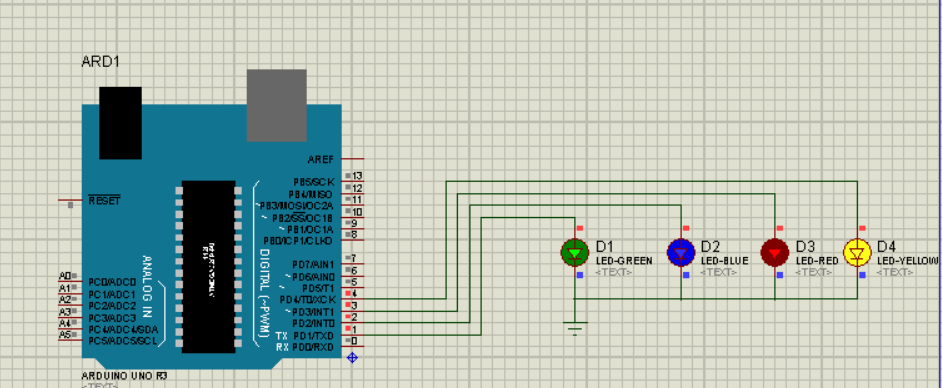
digitalWrite(4,0);

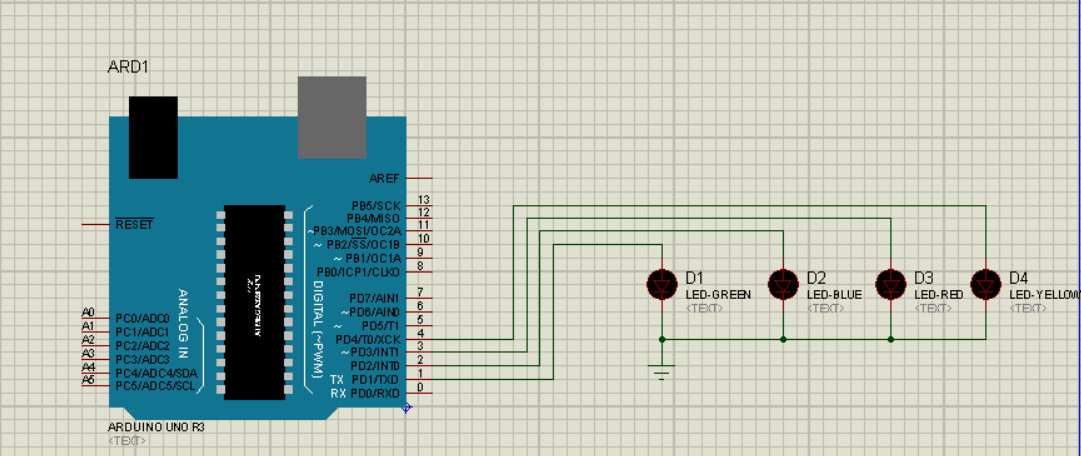
delay(500);

}

**OUTPUT**:







3)Write a PIC16F877A code turn on the LED using Switch (odd time on & even time off).

**Aim:**

**Program**::

#include <1 SWITCH 1 LED.h>

#bit sw=0X06.1

#bit tsw=0x86.1

#bit led=0x07.0

#bit tled=0x87.0

void main()

{

sw=0;

tsw=1;

led=0;

tled=0;

while(TRUE)

{

if(sw==1)

{

led=1;

}

else

{

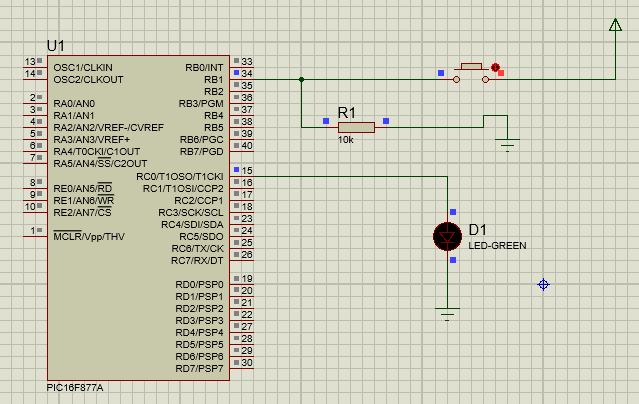
led=0;

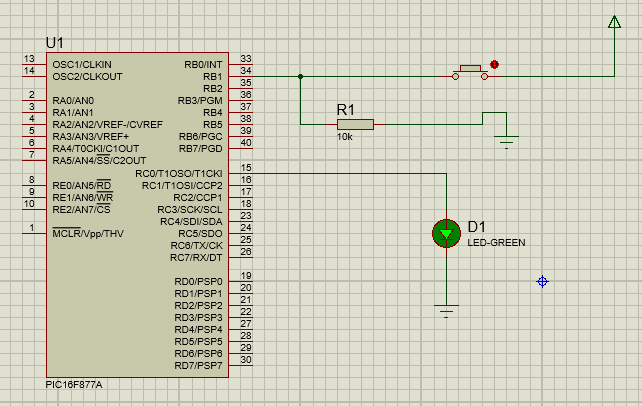
}

}

}

**SIMULATION OUTPUT:**





4).Write down the PIC16F877A code for 8 LED’s (turn Led’s on in even and odd pattern)

**Aim:**

**Program**::

#include <8 LED even odd.h>

#bit LED1=0X06.0

#bit TLED1=0x86.0

#bit LED2=0X06.1

#bit TLED2=0X86.1

void main()

{

LED1=0;

TLED1=0;

LED2=0;

TLED2=0;

while(TRUE)

{

LED1=1;

delay\_ms(1000);

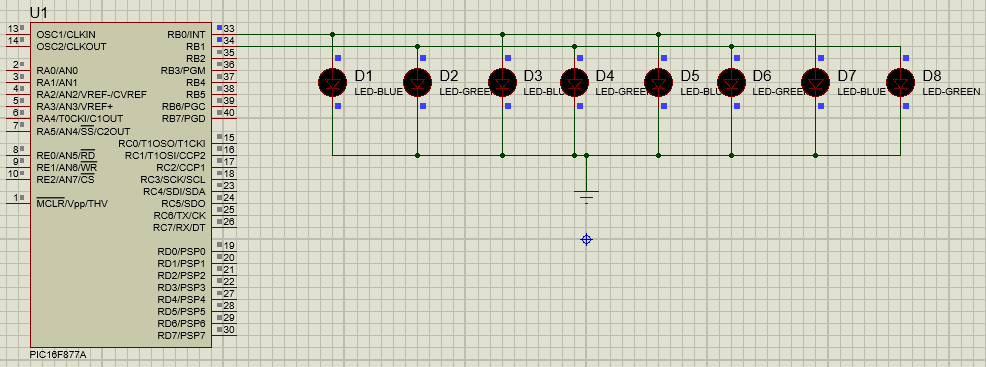
LED2=1;

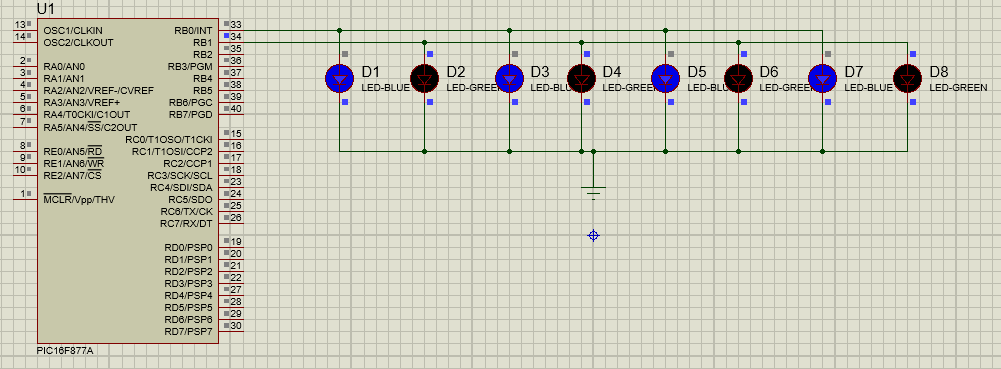
delay\_ms(1000);

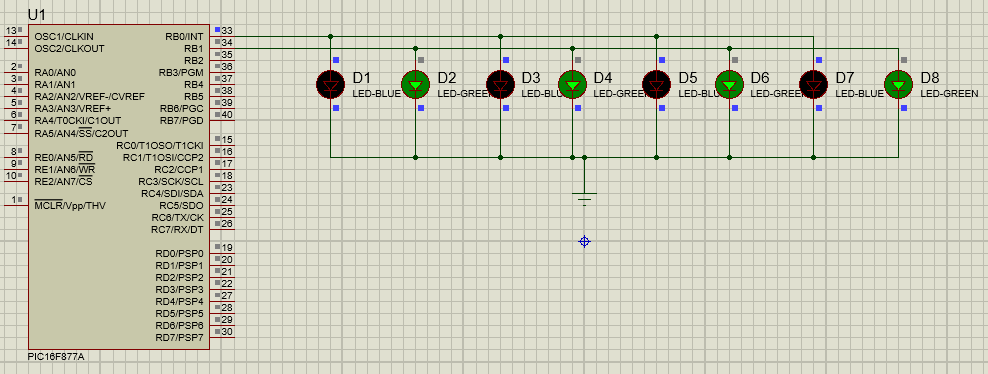
}

}

**SIMULATION OUTPUT:**

****

****



5).Write a Arduino code interfacing for 1 switch & 1 led.

**Aim:**

**Program**::

void setup() {

pinMode(1,INPUT);

pinMode(2,OUTPUT);

}

void loop() {

int a=digitalRead(1);

if(a==1)

{

digitalWrite(2,1);

delay(100);

}

else

{

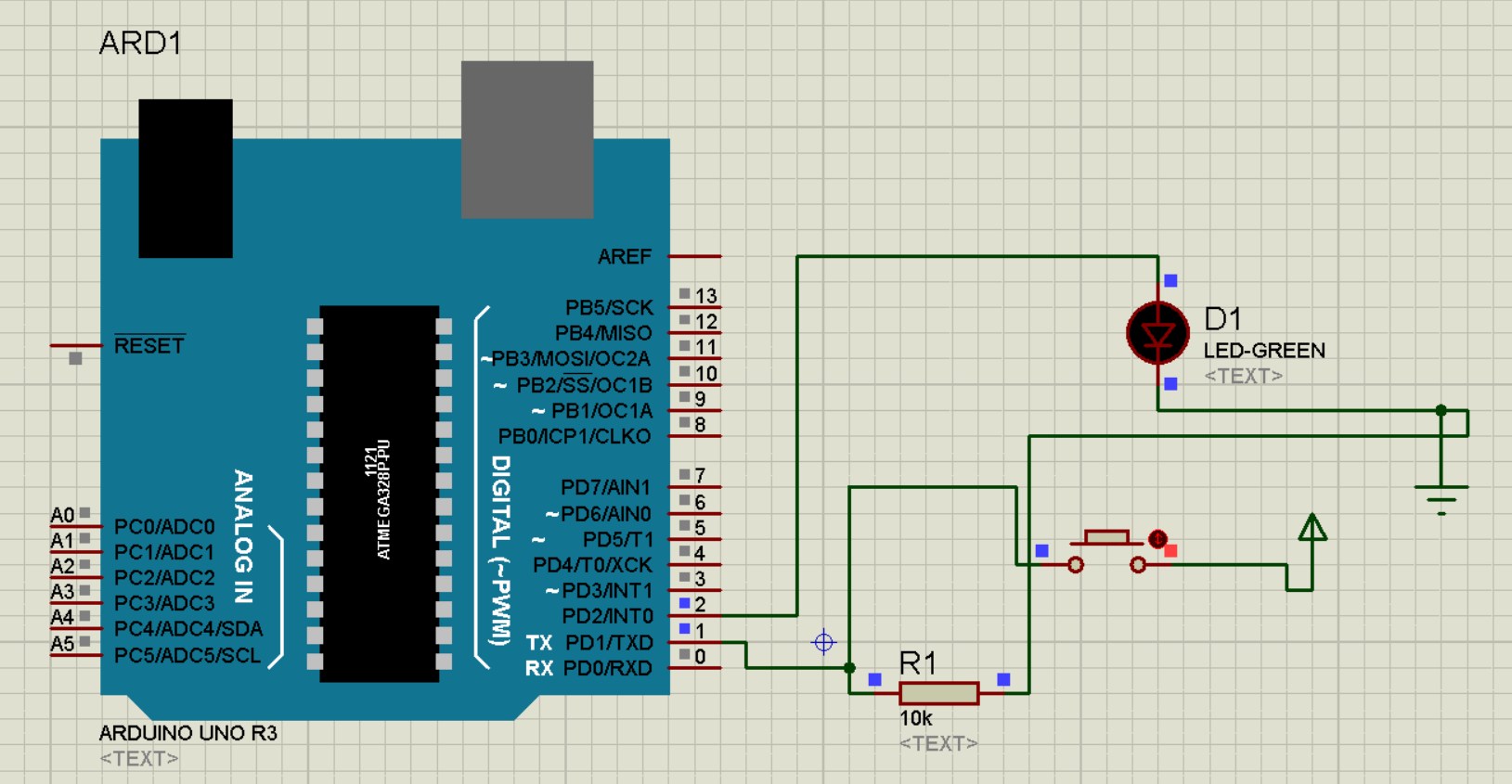
digitalWrite(2,0);

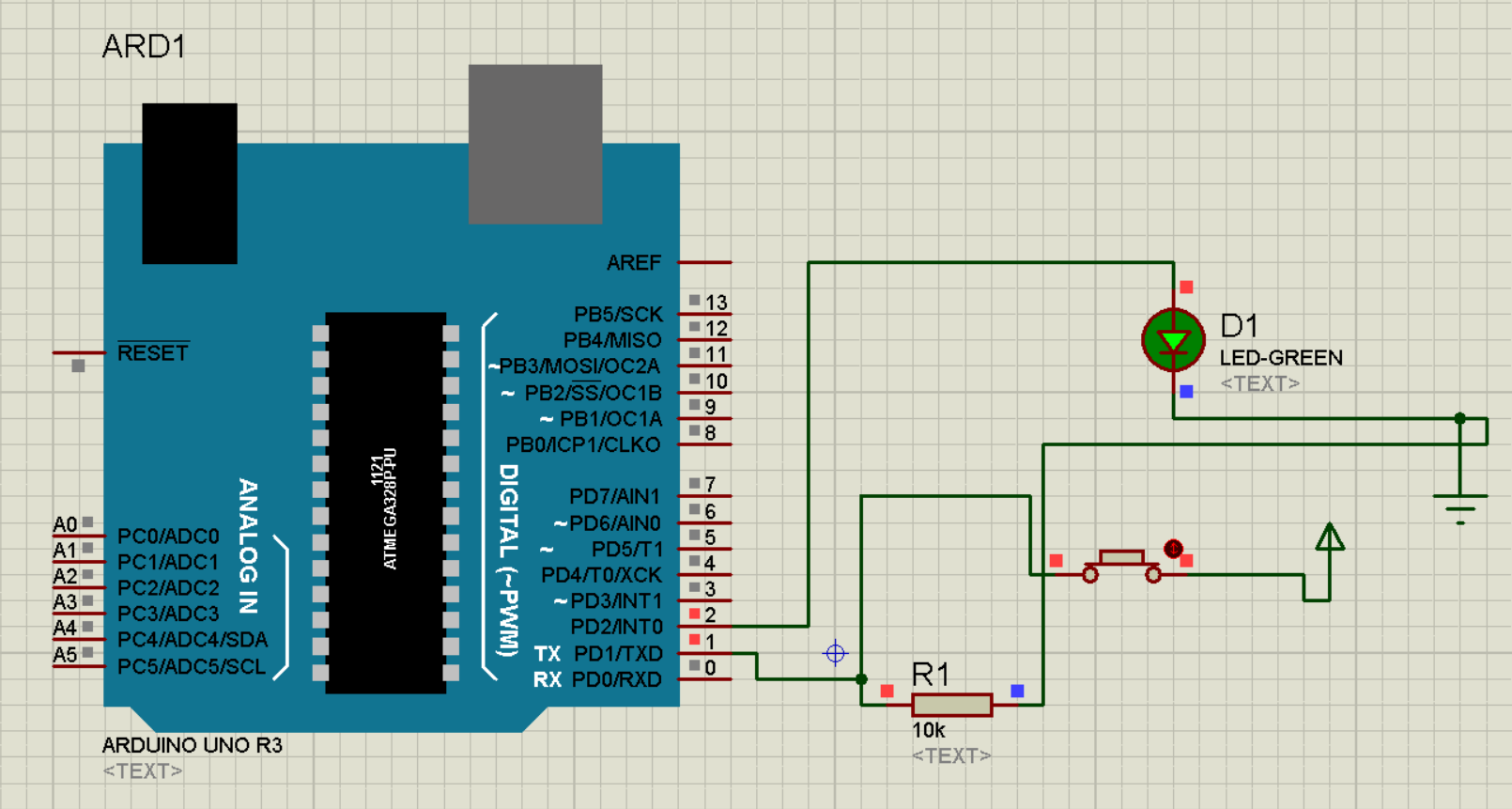
delay(100);

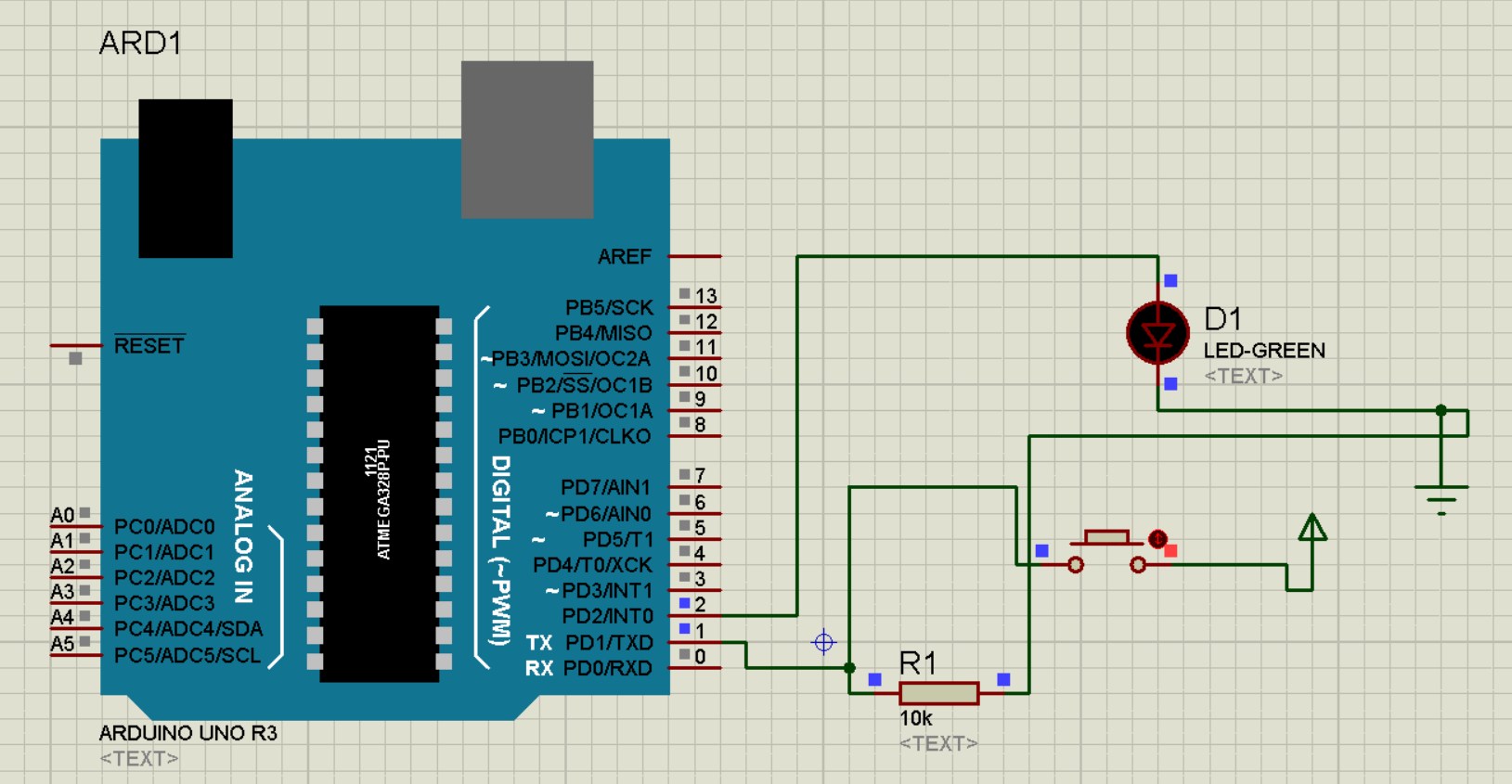
}

}

*Simulation:*







6).Differentiate the switch code in Arduino UNO & PIC16F877A controller.

**Aim:**

**Difference :**

The way of declaration of the input pin to switch in both Arduino and PIC16F877A are different.

In Pic we need to initialize the switch first

In Arduino we directly take the switch as the input device such that we write the code like

PinMode(pin\_no,INPUT);//pin\_no is the pin to which the switch is connected.

**Program**::

#include <1 SWITCH 1 LED.h>

#bit sw=0X06.1

#bit tsw=0x86.1

#bit led=0x07.0

#bit tled=0x87.0

void main()

{

sw=0;

tsw=1;

led=0;

tled=0;

while(TRUE)

{

if(sw==1)

{

led=1;

}

else

{

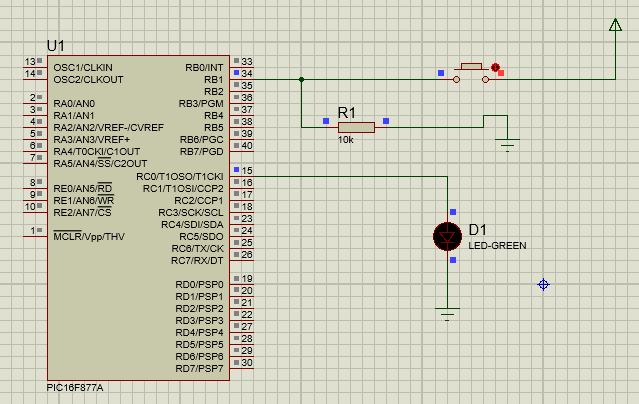
led=0;

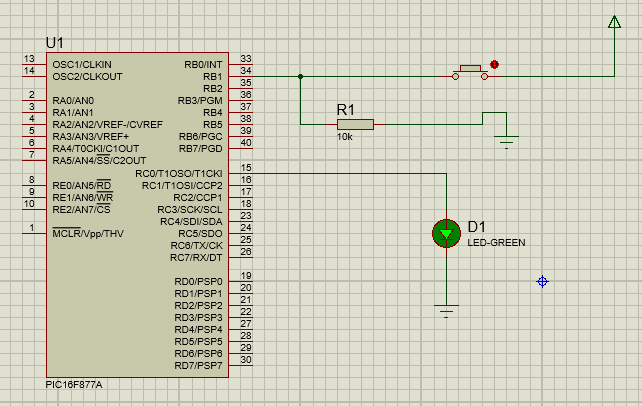
}

}

}

**SIMULATION OUTPUT:**





void setup() {

pinMode(1,INPUT);

pinMode(2,OUTPUT);

}

void loop() {

int a=digitalRead(1);

if(a==1)

{

digitalWrite(2,1);

delay(100);

}

else

{

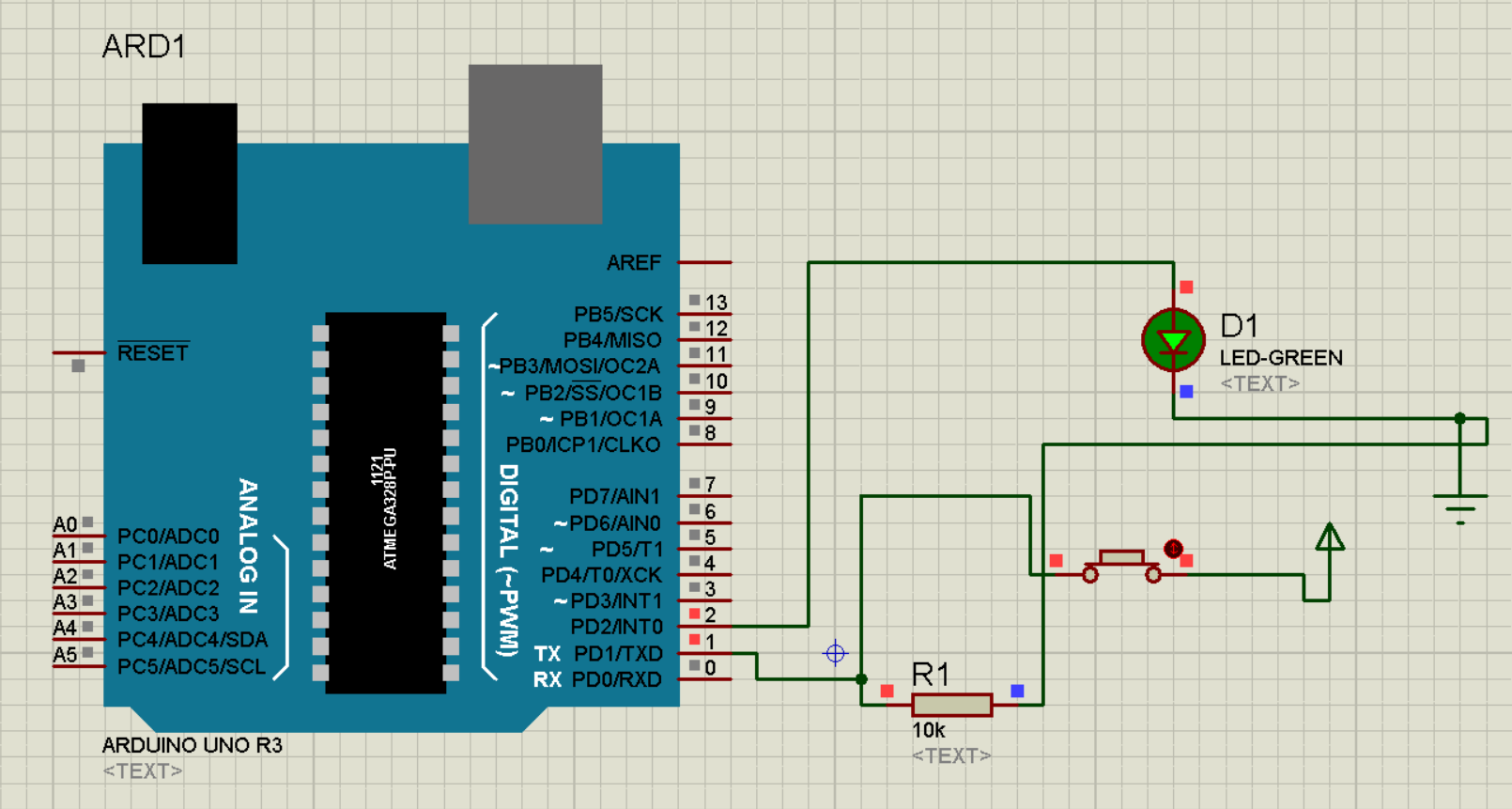
digitalWrite(2,0);

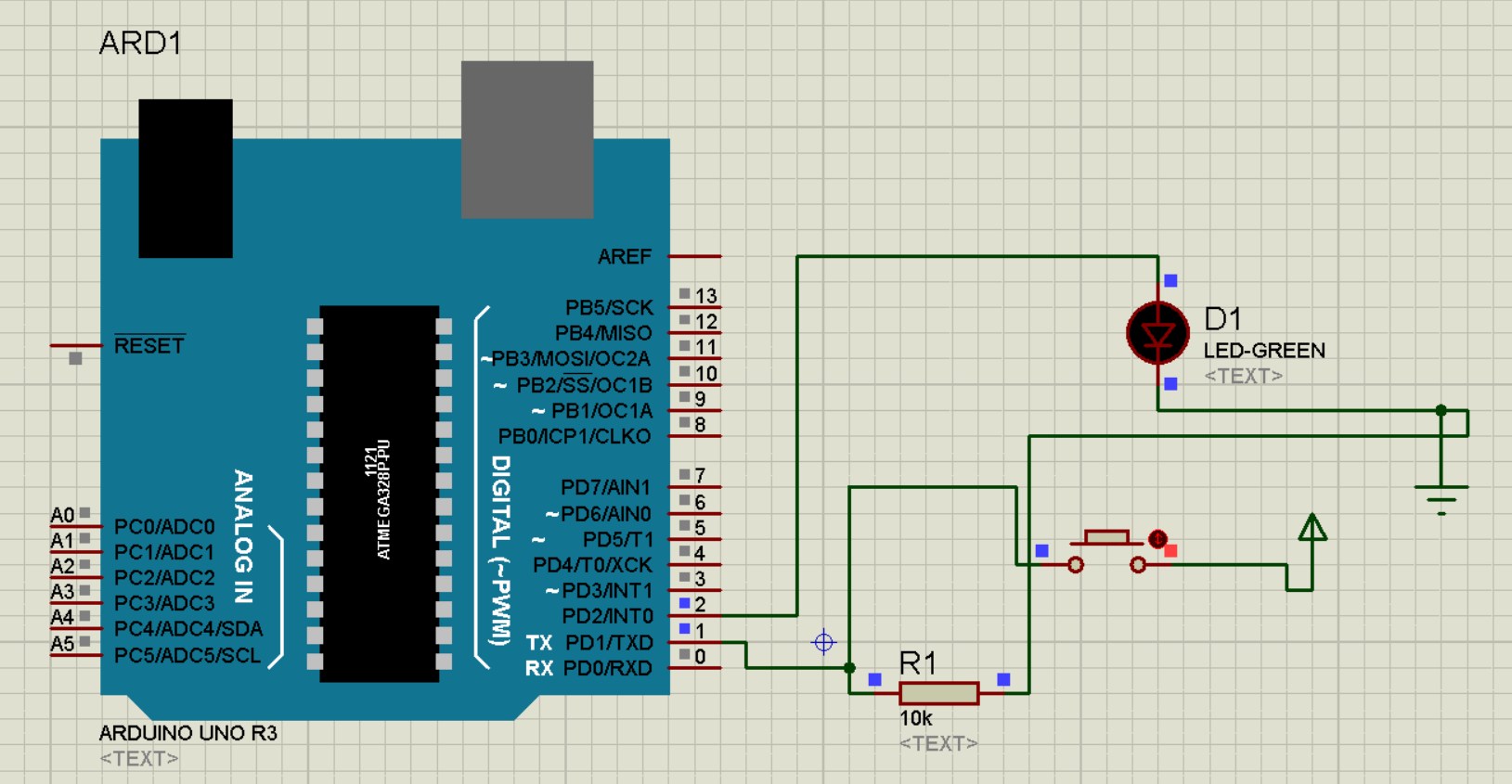
delay(100);

}

}

*Simulation:*





7).Write an Arduino code to interface LCD for displaying the name.

**Aim:**

**Program**::

#include<LiquidCrystal.h>

LiquidCrystal LCD(13,12,11,10,9,8);

void setup() {

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

LCD.

begin(16,2);

}

void loop() {

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

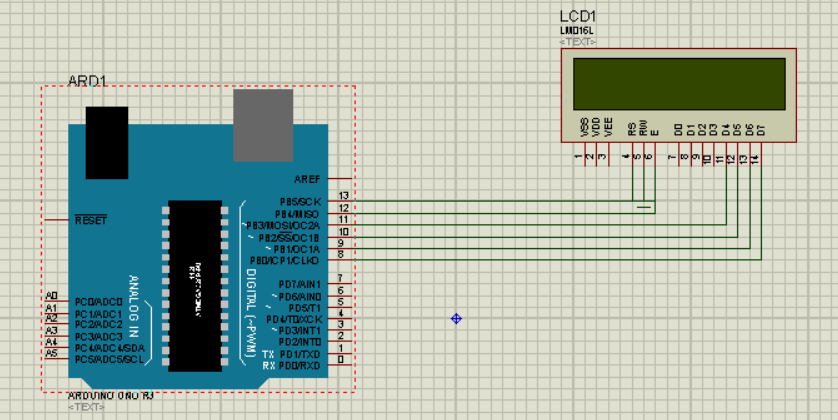
LCD.setCursor(5,0);

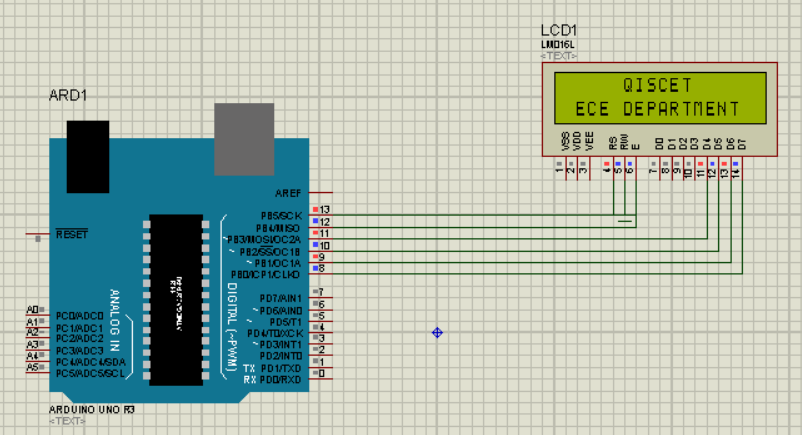
LCD.print("QISCET");

LCD.setCursor(1,1);

LCD.print("ECE DEPARTMENT");

}





8). Write an Arduino code to display the person count inside the room.

**Aim:**

**Program**::

9).Write an Arduino to display the ADC value in LCD.

**Aim:**

**Program**::

#include<LiquidCrystal.h>

LiquidCrystal LCD(13,12,11,10,9,8);

void setup() {

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

LCD.

begin(16,2);

pinMode(A0,INPUT);

}

void loop() {

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

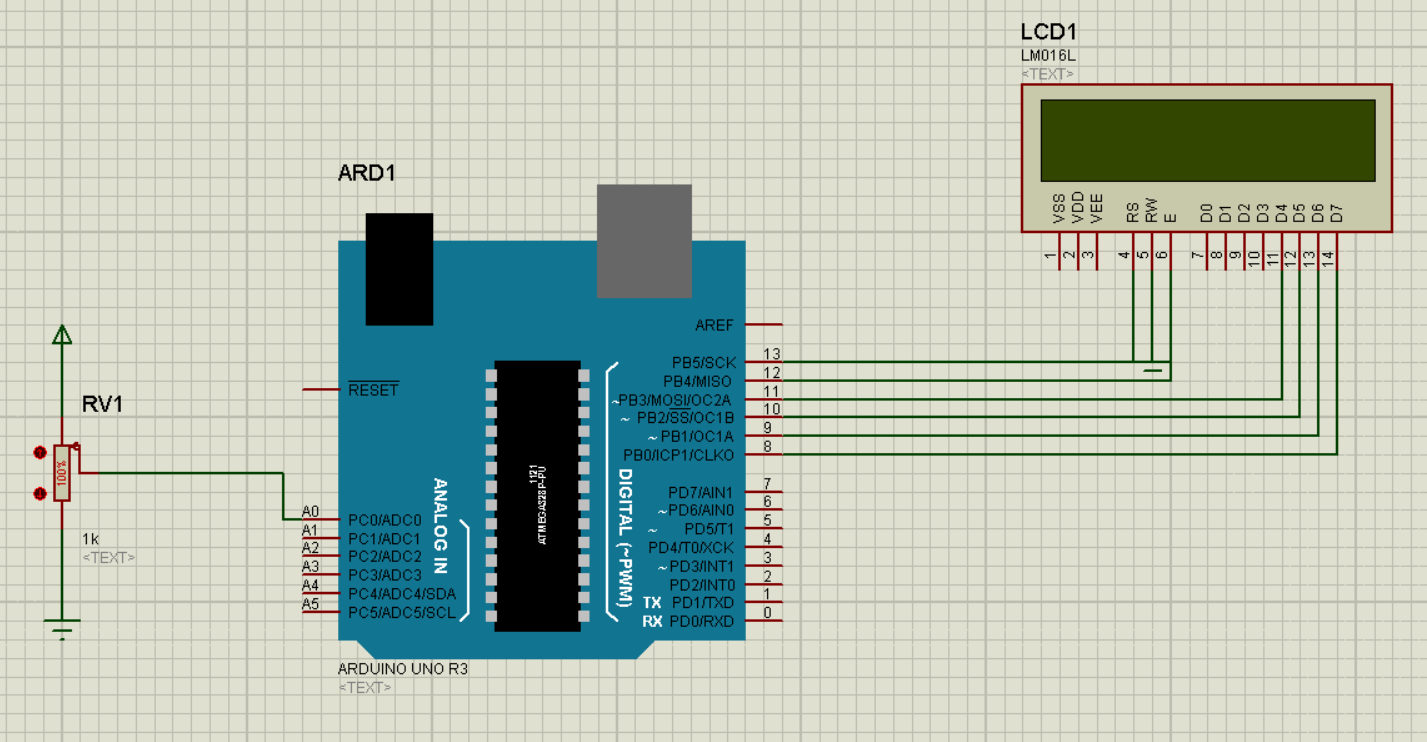
int a=analogRead(A0);

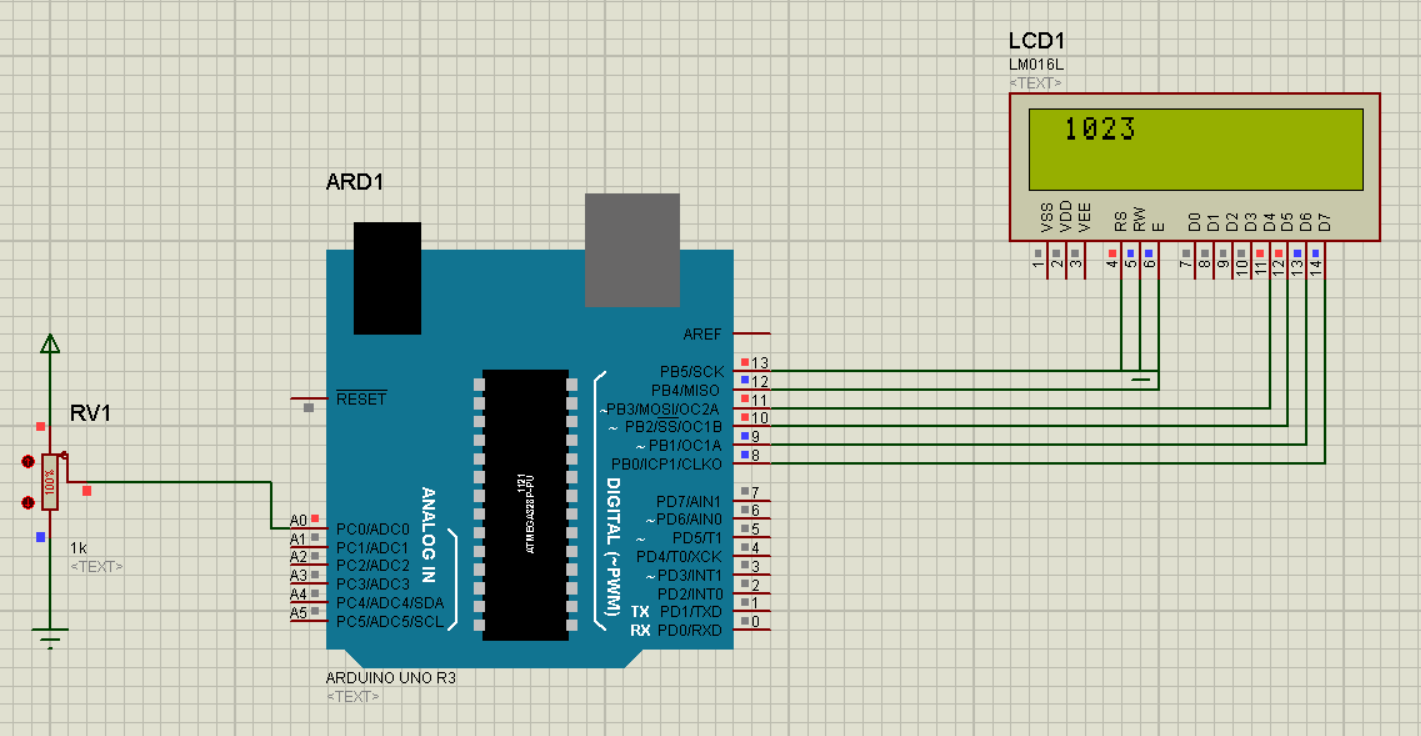
LCD.setCursor(1,0);

LCD.print(a);

}

*Simulation:*





10). Write a PIC16F877A code to transmit the data in virtual terminal.

**Aim:**

**Program**::

#include <UART1.h>

#byte SPBRG=0x99

#byte TXREG=0x19

#byte RCREG=0x1A

#bit TXIF=0x0c.4

#bit RCIF=0x0c.5

#byte TXSTA=0x98

#byte RCSTA=0x18

void transmit(char a)

{

TXREG=a;

while(TXIF==0);

TXIF=0;

}

void main()

{

int i;

char data[]="QISCET\_ECE";

TXSTA=0x26;

RCSTA=0x90;

SPBRG=129;

while(1)

{

for(i=0;data[i]!=0;i++)

{

transmit(data[i]);

}

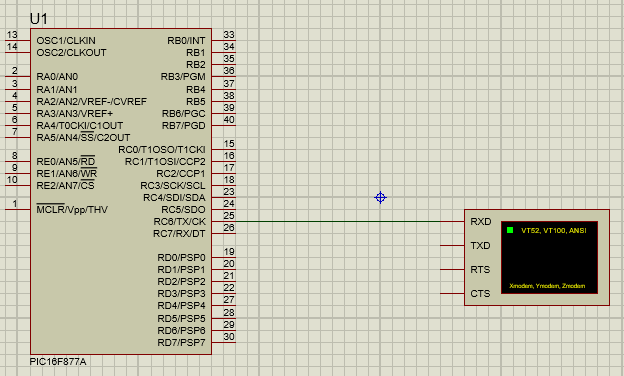
transmit('\r');

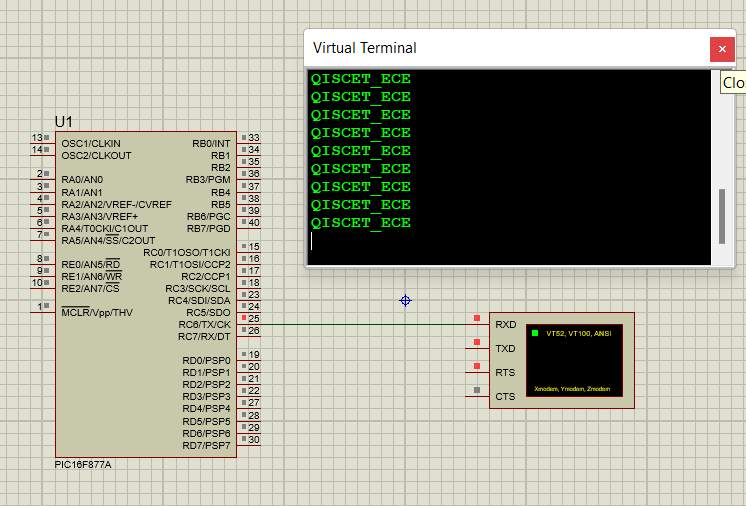
delay\_ms(500);

}

}

**SIMULATED OUTPUT:**





11).Write an Arduino code to turn on the load based on the temperature sensor value.

**Aim:**

**Program**::

12).Write an Arduino code to control the load according to the data received from the HCO5 Bluetooth module.

**Aim:**

**Program**::

char Incoming\_value = 0;

void setup()

{

Serial.begin(9600);

pinMode(13, OUTPUT);

}

void loop()

{

if(Serial.available() > 0)

{

Incoming\_value = Serial.read();

Serial.print(Incoming\_value);

Serial.print("\n");

if(Incoming\_value == '1')

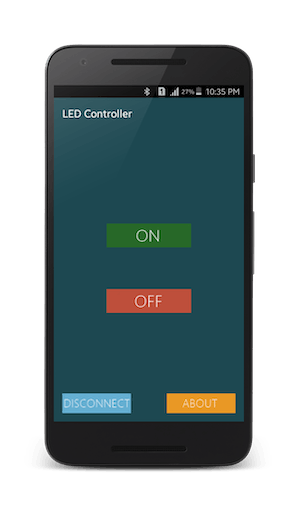
digitalWrite(13, HIGH);

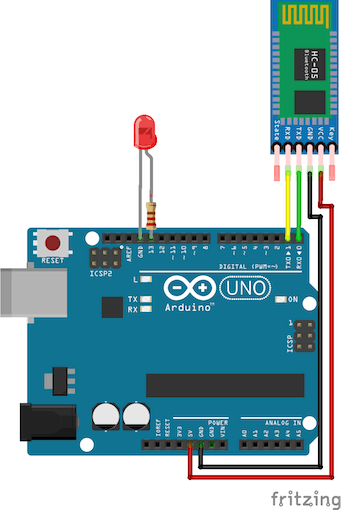
else if(Incoming\_value == '0')

digitalWrite(13, LOW);

}

}





*Done By :*

***Velidi Bola ShankaR\_***

*ROLL NO :: 20491A0402*