LATHA MATHAVAN ENGINEERING COLLEGE
Run by Karuppiah Pillai Theivanai Ammal Educational Trust
(An ISO 9001: 2015 Certified Institution)
KIDARIPATTI POST, ALAGARKOIL VIA, MADURAI - 625 301

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Subject Code: CS3691
Subject Name: Embedded system and IoT Lab
Year / Semester: III / VI

PREPARED BY
Dr.S.MURUGAN , HOD/CSE

LIST OF EXPERIMENTS

EX No : 1 ARITHMETIC OPERATIONS IN 8051 USING KEIL
Date :

AIM:

To write the Assembly Language Programs for performing the following Arithmetic operations.

a) 8–bit binary addition
b) 8–bit binary subtraction
c) 8–bit binary multiplication
d) 8–bit binary division

APPARATUS/COMPONENTS REQUIRED:

- Microcontroller Kit

PROBLEMSTATEMENT:

Write an ALP in8051 microcontroller to perform 8-bit arithmetic operations for the numbers stored in memory locations 4500H and 450H and store the results in 4600H and 4601H.

ALGORITHMS:

8-BITADDITION

1. Start.
2. Initialize carry register to zero.
3. Store data1 in accumulator.
4. Store data2 in any of R0– R7 registers.
5. Load the data pointer with external memory address.
6. Add the contents of register to that of accumulator.
7. If carry is generated, increment the carry register.
8. Store the contents of Accumulator, which is the sum in memory.
9. Move the carry register contents to Accumulator and store the same in memory.
10. Stop.

8-BITSUBTRACTION

1. Start.
2. Initialize Borrow register to zero.
3. Storedata1inaccumulator.
4. Storedata2inanyofR0– R7 registers.
5. Load the data pointer with external memory address.
6. Subtract the contents of register from that of accumulator.
7. If borrow is generated, increment the borrow register.
8. Store the contents of Accumulator, which is the difference in memory.
9. Move the borrow register contents to Accumulator and store the same in memory.
10. Stop.

8-BITMULTIPLICATION

1. Start.
2. Storedata1inAccumulator.
3. Storedata2inBregister.
4. Load the datapointer with external memory address.
5. Multiply A and B registercontents.
6. Move the A and B register contents to memory.
7. Stop.

8-BITDIVISION

1. Start.
2. Storedata1inAccumulator.
3. Storedata2inBregister.
4. Load the datapointer with external memory address.
5. Divide A and B register contents.
6. Move the A and B register contents to memory.
7. Stop.

Program:

Addition:

LABEL	MNEMONICS	COMMENTS
	MOV A,#Data1	Move data 1to A reg.
	ADD A,#Data2	Add data 2 and A reg content
	MOV DPTR,#4500	Move the data in Address to data pointer
	MOVX@DPTR,A	Move A reg to data pointer
HERE	SJMP HERE	Jump on HERE

OUTPUT:

ASCII Array		Binary(Hexa)Array	
INPUT		OUTPUT	
ADDRESS	DATA	ADDRESS	DATA
4101		4500	
4104			

Subtraction:

LABEL	MNEMONICS	COMMENTS
	MOV A,#Data1	Move data1toA reg.
	SUBBA,#Data2	Subtract data 2 and A reg content
	MOV DPTR,#4500	Move the data in address to data pointer
	MOVX@DPTR,A	Move A reg to data pointer
HERE	SJMP HERE	Jump on HERE

OUTPUT:

ASCIIArray		Binary(Hexa)Array	
INPUT		OUTPUT	
ADDRESS	DATA	ADDRESS	DATA
4101		4500	
4104			

Multiplication:

LABEL	MNEMONICS	COMMENTS
	MOVA,#0A	Data is moved to A reg
	MOV B,#88	Data is moved to B erg
	MUL A,B	Multiply data in A, B
	MOV DPTR,#4000	Initialize memory pointer
	MOVX@DPTR,A	Move A reg content to Immediate DPTR
	INCDPTR	Increment DPTR
	MOVA,B	Move B reg contents to A reg
	MOVX@DPTR,A	Move A reg content to Memory pointer
HERE	SJMPHERE	Loop is terminated

OUTPUT:

ASCIIArray		**Binary(Hexa)Array**	
INPUT		**OUTPUT**	
ADDRESS	**DATA**	**ADDRESS**	**DATA**
4101		4500	**(LSB)**
4104		4501	**(MSB)**

Division:

LABEL	MNEMONICS	COMMENTS
	MOV A,#08	Data is moved to A reg
	MOV B,#04	Data is moved to B erg
	DIVA,B	Divide A by B
	MOV DPTR,#4000	Initialize memory pointer
	MOVX@DPTR,A	Move A reg content to Immediate DPTR
	INCDPTR	Increment DPTR
	MOVA,B	Move B reg contents to A reg
	MOVX@DPTR,A	Move A reg content to Memory pointer
HERE	SJMPHERE	Loop is terminated

OUTPUT:

ASCII Array		Binary(Hexa)Array	
INPUT		OUTPUT	
ADDRESS	DATA	ADDRESS	DATA
4101		4500	
4104		4501	

CONCLUSION:

Thus Assembly Language Programs for performing the arithmetic operations –8 bit addition, subtraction, multiplication and division were executed using 8051 Keil compiler and output verified.

EX.NO:2 DATE:	**Test data transfer between registers and memory Using Keil**

AIM:

To write an assembly language program to transfer 5 data bytes.

ALGORITHM:

- Clear carry.
- Load R0 with any desired data.
- Load R1 with any desired data.
- Load R3 with the value of 5.
- Observe the incrementing values.
- Stop the program.

Program:

```
Org 00h
Mov R0, #30H
Mov  R1,#40H
Mov R3, #05
Mov A2@R0
Up:Mov@R1,A
INC R0
INC R1
DJNF R1,Up
END
```

RESULT:

Thus the 8051 ALP for data transfer is executed using Keil compiler.

EX.NO:3 DATE:	**Logical operations Using 8051 Keil**

AIM:

To perform logical operation using
8051microcontroller AND, OR & EX-OR.

ALGORITHM:

- ➢ Get the input value and store data in the accumulator.
- ➢ Get the second values and store the B register.
- ➢ Logical operation to perform the given number
- ➢ Store the output value in memory.

Program:

```
Clr c
Mov A, #07
ANL A, #03
Mov Ro, A
Clr c
Mov A, #07
ORL A, #03
Mov R1, A
Clr c
```

```
Mov A, #07
XRL A,#03
Mov R2,A
Clr c
Mov A,#07
CPL A
INC
MovR3,A
END
```

OUTPUT:

RESULT:

Thus the 8051 ALP for data transfer is executed using Keil compiler

EX.NO:4 DATE:	**Write Basic and arithmetic Program Using Embedded C.**

AIM:

To write an Arithmetic program to add, Subtract,multiply and divide two8-bit numbers using Embedded C Programming for 8051 microcontroller.

Addition Program ALGORITHM:

- Assign any desired 8-bit data to variable x.
- Assign an other desired 8-bit data to another variable y.
- Add two 8-bit numbers and store in another variable z.
- Store the result in Port 0

Subtraction program ALGORITHM:

- Assign any desired 8-bit data to variable a.
- Assign an other desired 8-bit data to another variable b.
- Subtract two 8-bit numbers and store in another variable c.
- Store the result in Port1

Multiplication program ALGORITHM:

- Assign any desired 8-bit data to a variable d.
- Assign another desired 8-bit data to another variable e.
- Multiply two 8-bitnumbers and store in another variable f.
- Store the result in Port 2

Division program ALGORITHM:

- Assign any desired 8-bit data to a variable p.
- Assign another desired 8-bitdata to another variable q.
- Divide two 8-bit numbers and store in another variable r.
- Store the result in Port 3
- Stoptheprogram.

Program:

```
#include<reg51.h>
Void main(void)
{
  Unsigned char x,y,z,a,b,c,d,e,f,p,q,r; //define variables
   //addition
x=0x03; //first 8-bit number
y=0x04;//second 8-bit number
P0=0x00;//declare port0 as output port
z=x+y; // perform addition
P0=z;//display result on port0

//subtraction
a=0x03; //first 8-bit number
b=0x04;//second 8-bit number
P1=0x00;//declare port1 as output
port
 c=b-a; // perform subtraction
P1=c;//display result on port1

//multiplication
d=0x03; //first 8-bit number
e=0x04;//second 8-bitnumber
P2=0x00;//declare port2 as output
port
 f=e*d; // perform multiplication
P2=f; //display result on port 2

//division
p=0x03; //first 8-bit number
q=0x04;//second 8-bit
number
```

P3=0x00;//declare port3 as output
port r=q/p; // perform division
P3=r;//display result on port 3
while(1);}

Output:

INPUT		OUTPUT	
ADDITION			
DATA1		PORT0	
DATA2			
SUBTRACTION			
DATA1		PORT1	
DATA2			
MULTIPLICATION			
DATA1		PORT2	
DATA2			
DIVISION			
DATA1		PORT3	
DATA2			

RESULT:

Thus the 8051 C–Programming for Addition, Subtraction, Multiplication and Division of two 8 bit numbers is executed in Keil.

EX.No: 5	LED BLINKING USING ARDUINO

AIM:

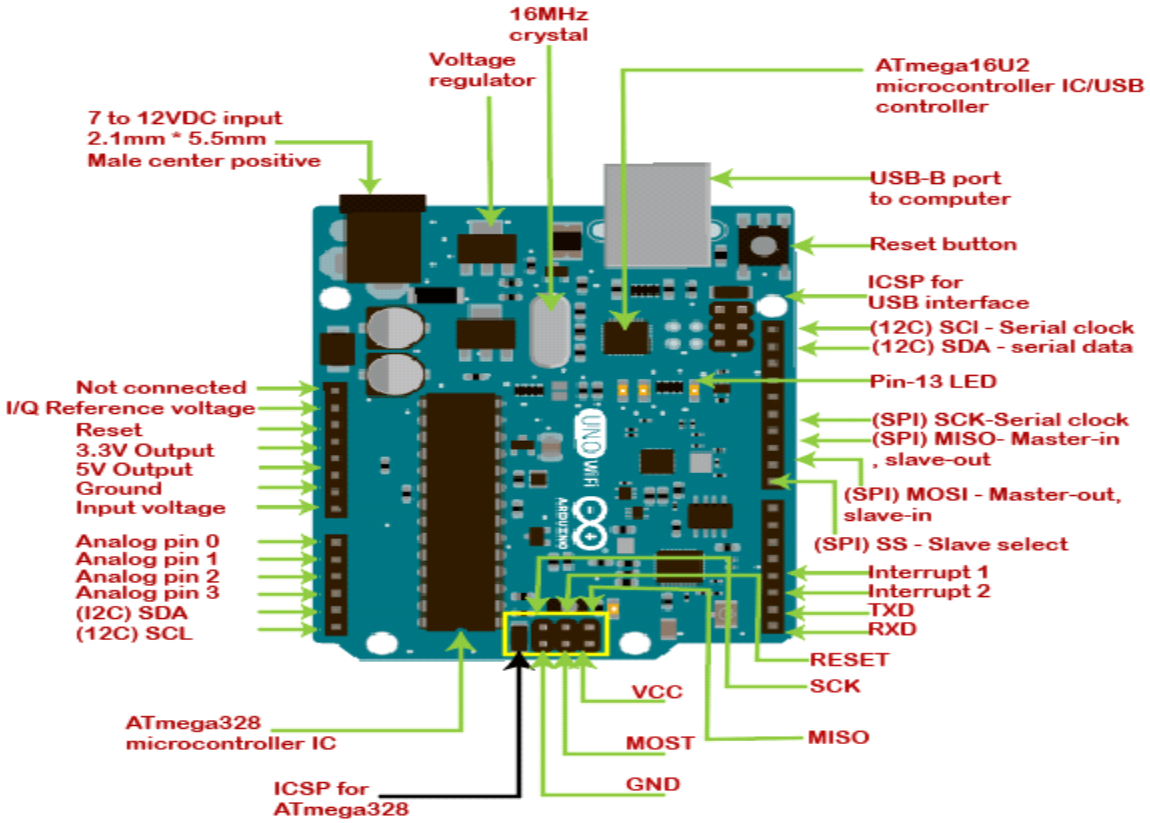
To Blink LED using Arduino Board.

APPARATUS/COMPONENTSREQUIRED:

- Arduino UNO Board
- LED

CIRCUITDIAGRAM:

To build the circuit, connect one end of the resistor to Arduino pin13. Connect the long leg of the LED (the positive leg, called the anode) to the other end of the resistor. Connect the short leg of the LED (the negative leg, called the cathode) to the Arduino GND, as shown in the diagram and the schematic below. Most Arduino boards already have an LED attached to pin 13 on the board itself. If you run this example with no hardware attached, you should see that LED blink. The value of the resistor in series with the LED may be of a different value than 220 ohm; the LED will lit up also with values up to 1K ohm.



PROGRAM:

After you build the circuit plug your Arduino UNO board into your computer, start the Arduino Software (IDE) and enter the code below. You may also load it from the menu File/Examples/01.Basics/Blink. The first thing you do is to initialize pin 13 as an output pin with the line

pinMode (13,OUTPUT);

In the main loop, you turn the LED on with the line:

digitalWrite(13, HIGH);

This supply 5 volt to pin13.That creates a voltage difference across the pins of the LED, and lights it up. Then you turn it off with the line:

digitalWrite(13,LOW);

That takes pin13 back to 0volts, and turns the LED off. In between the on and the off, you want enough time for a person to see the change, so the delay() commands

Tell the board to do nothing for1000 milliseconds, or one second. When you use the delay () command, nothing else happens for that amount of time. Once you've understood the basic examples, check out the Blink Without Delay example to learn how to create a delay while doing other things.

Once you've understood this example,check out the Digital Read Serial example to learn how read a switch connected to the board.

```
/*
 Blink
  Turn on an LED for one second, turn off for one second, repeatedly. Most Arduinos have an on-board LED you can control. On the Uno and
  Leonardo, it is attached to digital pin13.
/the set up function runs once when you press reset or power the board
void setup() {
//initialize digital pin 13 as an output.
pinMode(13,OUTPUT);
}
//the loop function runs over and over again forever
```

```
void loop() {
digitalWrite(13, HIGH);   //turn the LED on(HIGH is the voltage level)
delay(1000);              // wait for a second
digitalWrite(13, LOW);    //turn the LED off by making the voltage LOW
delay(1000);              // wait for a second
}
```

CONCLUSION:

Thus the LED blinking using Arduino board is done and verified.

QUESTIONS:

1. WhatismeantbyPinMode?
2. Definebootloader.

EX.No:6	Reading Temperature using Arduino with Temperature Sensor

AIM:

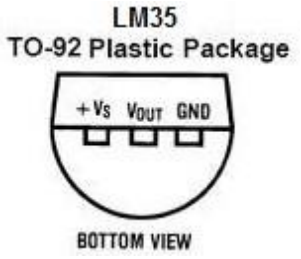
To Read the room temperature using temperature sensor LM35 with Arduino UNO Board.

APPARATUS/COMPONENTSREQUIRED:

- Arduino Board(AnyVersion)
- LM35 Temperature Sensor
- USB Cable
- Computer with Arduino IDE Software

THEORY:

Connections:



The voltage output of LM35 is connected to the analog input A0 of the arduino. The voltage at this pin will be proportional to the temperature and this voltage is read using analogRead function. The analogRead function will read the voltage (in a range 0 to 5) at a particular analog input pin and converts it into a digital value between 0 and 1023. For example, if 29°C is the temperature, the output of LM35 will be 290mV.The result of the analogRead function will be 290mV/(5/1023) =59.

PROGRAM:

```
Float temp;
Int temp Pin=0;
void setup()
```

```
{
  Serial.begin(9600);
}
Void loop()
{
  temp = analogRead (tempPin);
  temp = temp * 0.48828125;
  Serial.print("TEMPRATURE=");
  Serial.print(temp);
  Serial.print("*C");
  Serial.println();
  delay(1000);
}
Float temp;
Int temp Pin=0;
void setup()
{
  Serial.begin(9600);
}
voidloop()
{
  temp = analogRead (tempPin);
  temp = temp * 0.48828125;
  Serial.print("TEMPRATURE=");
  Serial.print(temp);
  Serial.print("*C");
  Serial.println();
  delay(1000);
}
```

TEMPERATURE CONVERSION:

(SUPPLY_VOLTAGEx1000/1024)/10 where SUPPLY_VOLTAGE is 5.0V (the voltage used to power LM35) = (5x1000/1024)/10=0.48828125.

1024 = 2^10 is the analog value represented by Atmega and 1000 is used to change the unit from V to mV.

Each 10 mV is directly proportional to 1 Celcius.

For the above case of 29 degree C, the result of analog Read function 59 multiplied by 0.48828125 gives 29ºC.

CONCLUSION:

Thus the temperature is measured by interface temperature sensor LM35 with Arduino Board.

QUESTIONS:

2. WhatistheuseofLM35?

3. How to interface temperature sensor with Arduino board?

TRAFFIC LIGHT CONTROL WITH ARDUINO UNO BOARD

Ex. No. 7
Date :

AIM:

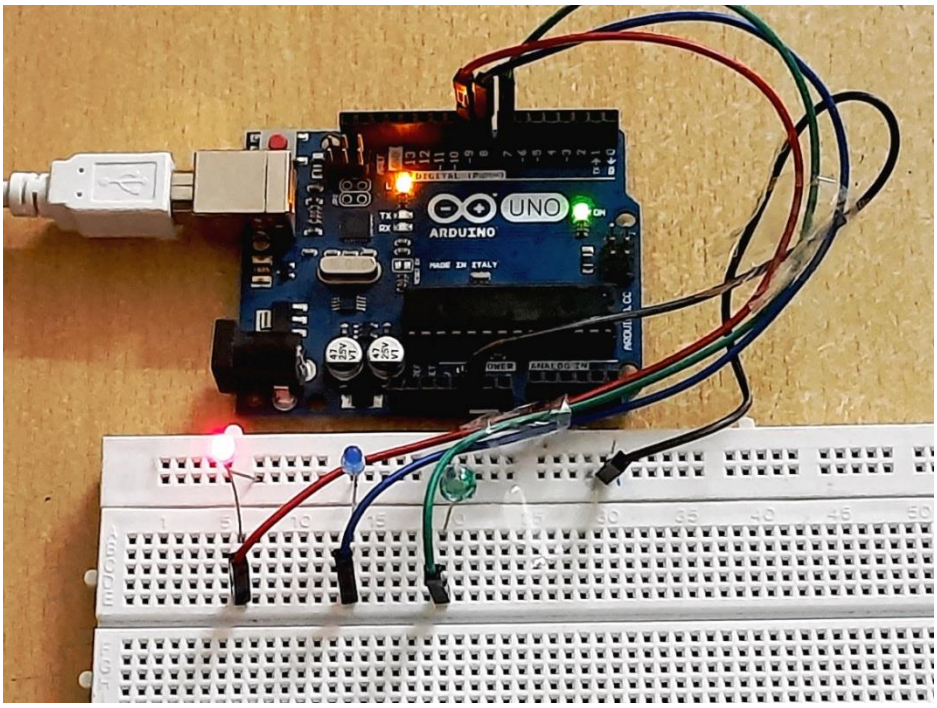
To Blink LED using Arduino Board.

APPARATUS/COMPONENTSREQUIRED:

- Arduino UNO Board
- LED – 3 No
- USB CABLE
- COMPUTER

CIRCUITDIAGRAM:

To build the circuit, connect one end of the resistor to Arduino pin13. Connect the long leg of the LED (the positive leg, called the anode) to the other end of the resistor. Connect the short leg of the LED (the negative leg, called the cathode) to the Arduino GND, as shown in the diagram and the schematic below. Most Arduino boards already have an LED attached to pin 13 on the board itself. If you run this example with no hardware attached, you should see that LED blink. The value of the resistor in series with the LED may be of a different value than 220 ohm; the LED will lit up also with values up to 1K ohm.



PROGRAM:

After you build the circuit plug your Arduino UNO board into your computer, start the Arduino Software (IDE) and enter the code below.

```
Int red =2;

Int yellow =4;

Int green =6;

Void setup( )

{

pinMode (red,OUTPUT);

pinMode(yellow, OUTPUT);

pinMode(green, OUTPUT);

}

Voidloop ( )

{

 digitalWrite(red, HIGH);

digitalWrite(yellow, LOW);

digitalWrite(green, LOW);

delay (1000);

digitalWrite(yellow, HIGH);

digitalWrite(red, LOW);

digitalWrite(green, LOW);

delay (500);

digitalWrite(green, HIGH);
```

```
digitalWrite(green, HIGH);

digitalWrite(yellow, LOW);

digitalWrite(red, LOW);

delay (1000);

}
```

CONCLUSION:

Thus the traffic light control using Arduino board is executed and verified.

QUESTIONS:

3. What is meant by PinMode?
4. Define boot loader.

EX.NO:8 DATE:	Communication with IoT devices

Aim:

To communication with IOT devices Using Arduino Uno board via GSM and Bluetooth.

Apparatus:

S.No.	Apparatus	Range/Rating	Quantity
1	Universal Board		1
2	Arduino board		1
3	Bluetooth		1
4	Zigbee		
5	GSM board		
6	12V Adaptor		1
7	Power jack		1
8	USB Cable		1
9	Jumper Wires		Required

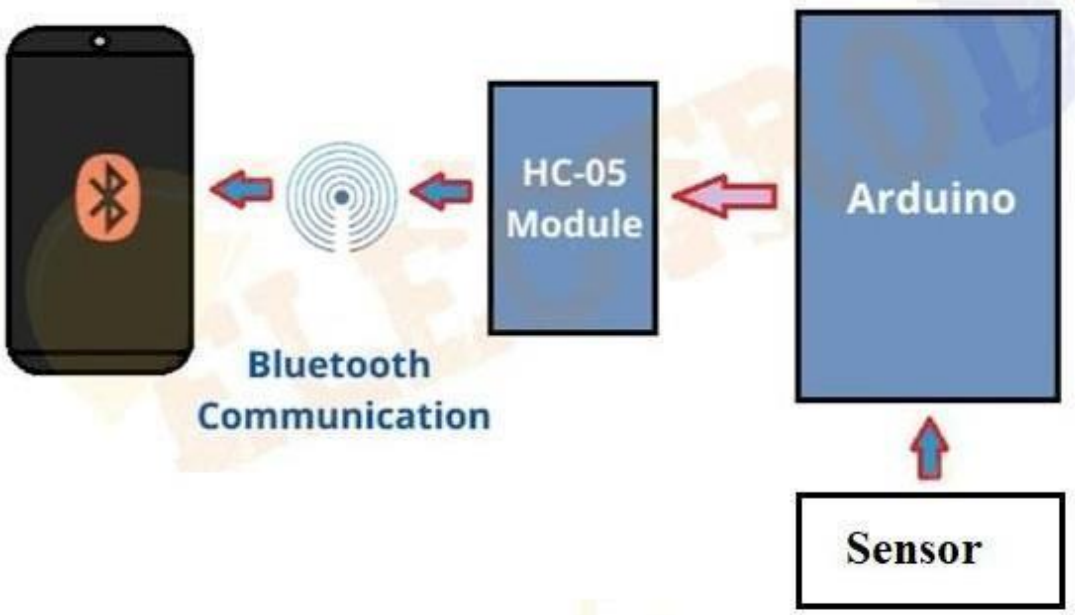
Hardware Procedure:

- Connect LM35 or LDR to ArduinoUno pin of A0.
- Read the sensor value from the Arduino pin A0.
- Power jack is connected to the Arduino.
- USB connector is connected toArduino Uno to monitor.
- Connect the Bluetooth or Zigbee or GSM board with Arduino Uno.
- Check the output from the development board.

Software Procedure:

- Click on Arduino IDE
- Click on file
- Click on New
- Write a Program as per circuit Pin connections
- Click on Save
- Click on Verify
- Click on Upload the code into Arduino Uno by using USB cable.

BLOCK DIAGRAM BLUETOOTH INTERFACING

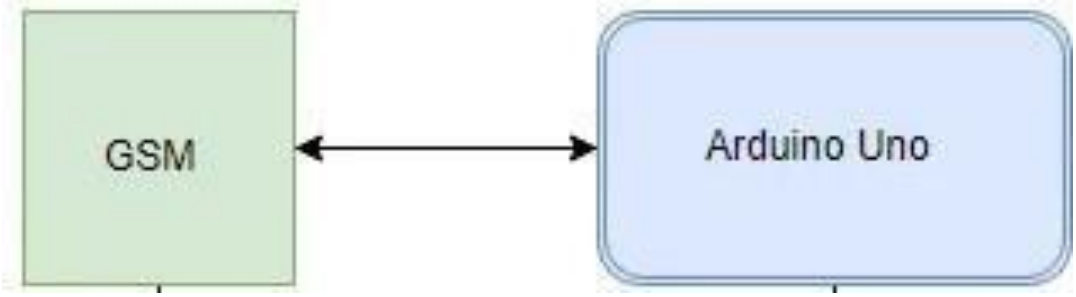


Program:

Communication using Bluetooth HC05–Arduino Uno with Mobile App (IoT Device)

```
intval;
voidsetup()
{
Serial.begin(9600);
pinMode(A0,INPUT);
}
voidloop()
{
val=analogRead(A0);
Serial.print("Value =");
Serial.println(val);
delay(500);
}
```

BLOCK DIAGRAM GSM INTERFACING



Program:

```
#define sw111
int swstate1;
void setup()
{
Serial.begin(9600);
 pinMode(sw1,INPUT);
}
```

```
voidloop()
{
swstate1=digitalRead(sw1);

delay(500);


if(swstate1==
1)
{
Serial.println("sending SMS");
SendMessage();
delay(1000);
}

  else

  {

    Serial.println("Waiting for Emergency switch");

  }

  delay(500);
  }

 voidSendMessage()

 {

  Serial.println("AT");

  //Sets the GSM Module
  in Text Mode

   delay(100);

  Serial.println((char)13);
  // ASCII code of enter
  delay(1000);

  Serial.println("AT+CMGF=1");
  //Sets the GSM Module in Text
  Mode

  delay(100);
```

```
 Serial.println((char)13);
 //ASCII code of enter

 delay(1000);
 // Delay of 1000
 milliseconds or 1 second

 Serial.println("ATE=0");
                                    /
 /Sets the GSM Module in
 Text Mode delay(100);
Serial.println
 ((char)13);/
 / ASCII
 code of
 enter
 delay(1000
 );
 Serial.println("AT+CMGS=\"+919994085790\"\r");
 //Replace x with mobile number
 delay(1000);

 Serial.println("CS3691–EMBEDDEDSYSTEMS AND IOT LAB");
 //The SMS text   you want to send delay(100);


   //mySerial.println
 ("ATD+60XX
 XXXXXXX;");
 Serial.println((ch
 ar)26);//ASCII
 code of
 CTRL+Zdelay(50
 00);

 Serial.println("ATD+919994085790;");//


 Replacex with mobile
   numberdelay(1000);

}
```

```
voidRecieveMessage()
{
Serial.println("AT+CNMI=2,2,0,0,0");//ATCommandtoreceivealive
 SMSdelay(1000);
}
```

RESULT:

Thus communication with IOT devices Using ArduinoUno board via GSM and Bluetooth is completed.

Interfacing LED with RasperryPi RP2040

Ex. No. 9

Date :

Aim: To interface LED with Rasperry Pi RP 2040

S.No.	Apparatus	Range/Rating	Quantity
1	Universal Board		1
2	RP2040		1
6	Micro B Type cable		1
7	Power jack		1
8	USB Cable		1
9	Jumper Wires		Required

To Interface LED with Raspberry pi RP2040

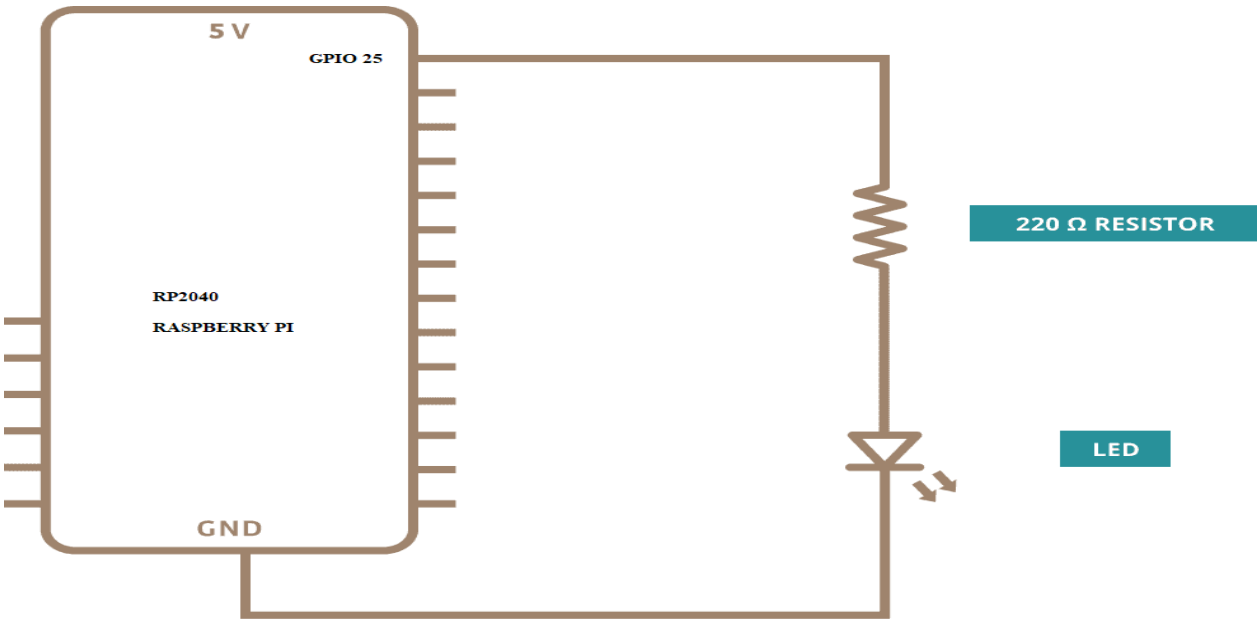
Hardware Procedure:

- Connect LED to GPIO 25
- Power jack is connected to the Arduino.
- USB connector is connected to RP2040 to monitor.

Software Procedure:

- Click on Thonny
- Click on file
- Click on New
- Write a Program as per circuit Pin connections
- Click on Save
- Click on Verify
- Click on Upload the code into RP2040 by using USB cable.

BLOCKDIAGRAM LEDINTERFACINGWITHRP2040



Program:

```
Import time
From machine import Pin
led=Pin(25,Pin.OUT)       #create LED object from pin13,  Set Pin13 to output

whileTrue:
  led.value(1)          #Set led turn on
  time.sleep(1)
  led.value(0)          #Set led turn off
  time.sleep(1)         #delay(1sec)
```

Result :

Thus, Interfacing of LED with Raspberry pi RP2040 was successfully done.