<u>CONTENTS</u>

EMBEDDED SYSTEM

As its name suggests, Embedded means something that is attached to another thing. An embedded system can be thought of as a computer hardware system having software embedded in it. An embedded system can be an independent system or it can be a part of a large system. An embedded system is a microcontroller or microprocessor-based system which is designed to perform a specific task. For example, a fire alarm is an embedded system, it will sense only smoke.

An embedded system has three components −

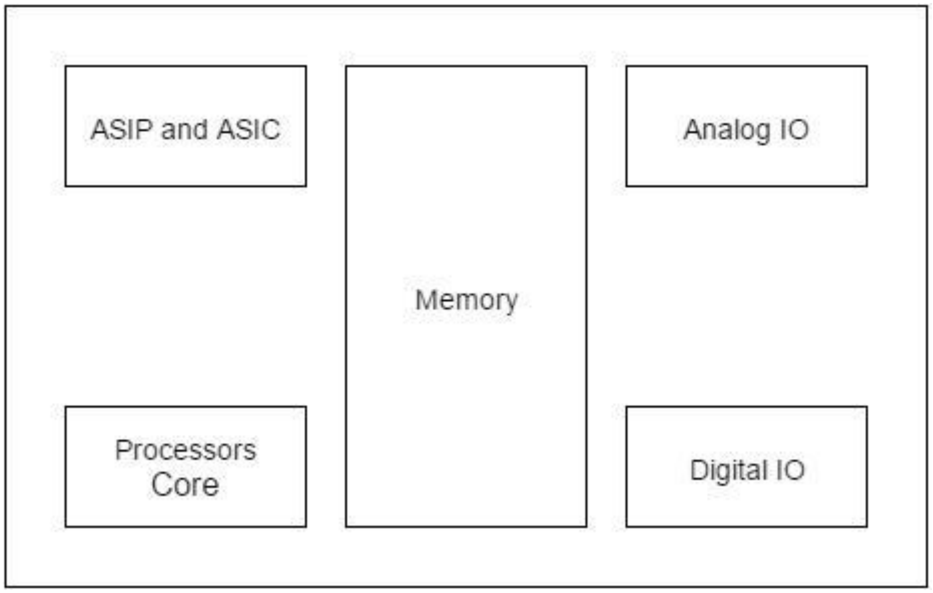
- It has hardware.
- It has application software.
- It has Real Time Operating system (RTOS) that supervises the application software and provide mechanism to let the processor run a process as per scheduling by following a plan to control the latencies. RTOS defines the way the system works. It sets the rules during the execution of application program. A small-scale embedded system may not have RTOS.

So we can define an embedded system as a Microcontroller based, software driven, and reliable, real-time control system.

Characteristics of an Embedded System

- Single-functioned − an embedded system usually performs a specialized operation and does the same repeatedly. For example: A pager always functions as a pager.
- Tightly constrained − All computing systems have constraints on design metrics, but those on an embedded system can be especially tight. Design metrics is a measure of an implementation's features such as its cost, size, power, and performance. It must be of a size to fit on a single chip, must perform fast enough to process data in real time and consume minimum power to extend battery life.
- Reactive and Real time − Many embedded systems must continually react to changes in the system's environment and must compute certain results in real time without any delay. Consider an example of a car cruise controller; it continually monitors and reacts to speed and brake sensors. It must compute acceleration or de-accelerations repeatedly within a limited time; a delayed computation can result in failure to control of the car.
- Microprocessors based − It must be microprocessor or microcontroller based.
- Memory − It must have a memory, as its software usually embeds in ROM. It does not need any secondary memories in the computer.
- Connected − It must have connected peripherals to connect input and output devices.

- HW-SW systems − Software is used for more features and flexibility. Hardware is used for performance and security.



Advantages

- Easily Customizable
- Low power consumption
- Low cost
- Enhanced performance

Disadvantages

- High development effort
- Larger time to market

MICROCONTROLLER

A microcontroller (or MCU for microcontroller unit) is a small computer on a single integrated circuit. In modern terminology, it is a system on a chip or SoC. A microcontroller contains one or more CPU’S (processor cores) along with memory and programmable input output peripherals. Program memory in the form of Ferroelectric RAM, NOR flash or OTP ROM is also often included on chip, as well as a small amount of RAM. Microcontrollers are designed for embedded applications, in contrast to the microprocessors used in personal computer or other general-purpose applications consisting of various discrete chips.

Microcontrollers are used in automatically controlled products and devices, such as automobile engine control systems, implantable medical devices, remote controls, office machines, appliances, power tools, toys and other embedded systems. By reducing the size and cost compared to a design that uses a separate microprocessor, memory, and input/output devices, microcontrollers make it economical to digitally control even more devices and processes. Mixed signal microcontrollers are common, integrating analog components needed to control non-digital electronic systems.

Some microcontrollers may use four-bit words and operate at frequencies as low as 4 kHz, for low power consumption (single-digit mill watts or microwatts). They will generally have the ability to retain functionality while waiting for an event such as a button press or other interrupt; power consumption while sleeping (CPU clock and most peripherals off) may be just nano watts, making many of them well suited for long lasting battery applications. Other microcontrollers may serve performance-critical roles, where they may need to act more like a digital signal processor (DSP), with higher clock speeds and power consumption.

AVR MICROCONTROLLER

AVR is a family of microcontrollers developed by Atmel beginning in 1996. These are modified Harvard architecture 8-bit RISC single-chip microcontrollers. AVR was one of the first microcontroller families to use on-chip flash memory for program storage, as opposed to one-time programmable ROM, EPROM, or EEPROM used by other microcontrollers at the time.

AVR microcontrollers find many applications as embedded systems; they are also used in Arduino line of open source board designs.

History

The AVR architecture was conceived by two students at the Norweign Institute of Techonology (NTH), Alf-Egil Bogen and Vegard Wollan

The original AVR MCU was developed at a local ASIC house in Trondheim Norway, called Nordic VLSI at the time, now Nordic Semiconductor, where Bogen and Wollan were working as students. When the technology was sold to Atmel from Nordic VLSI, the internal architecture was further developed by Bogen and Wollan at Atmel.

Norway, a subsidiary of Atmel. The designers worked closely with compiler writers at IAR Systems to ensure that the AVR instruction set provided efficient Compilations of high level languages .

Atmel says that the name AVR is not an acronym and does not stand for anything in particular. The creators of the AVR give no definitive answer as to what the term "AVR" stands for. However, it is commonly accepted that AVR stands for Alf and Vegard's RISC processor.

There are six different series of AVR

1.tinyAVR – the ATtiny series

2. mega AVR – the ATmega series

3. XMEGA – the ATxmega series

4. Application-specific AVR

5.FPSLIC (AVR with FPGA)

6. 32-bit AVRs

These six series are classified according to their Flash, EEPROM, and SRAM, Internal data memory, Internal registers, GPIO ports, MCU speed.

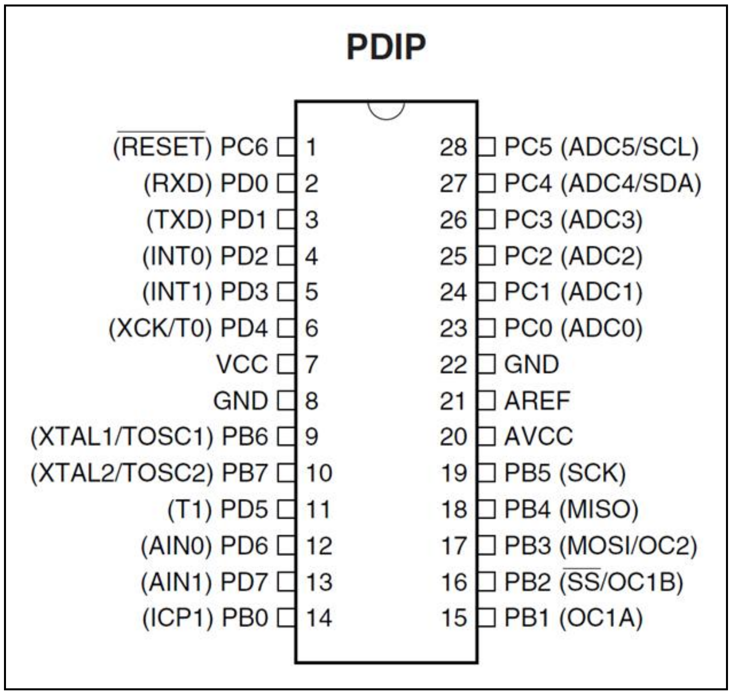
Some mostly used AVR microcontrollers are:-

ATmega8 microcontroller, ATmega16 microcontroller, ATmega32 microcontroller and ATmega328 microcontroller.

We are using ATmega8 microcontroller and Arduino both for this particular project.

ATmega8 Microcontroller

It is a member of AVR microcontroller family. Let us see its features functions of different pins.



Memory: It has 8 Kb of Flash program memory (10,000 Write/Erase cycles durability), 512 Bytes of EEPROM (100,000 Write/Erase Cycles). 1Kbyte Internal SRAM.
I/O Ports: 23 I/ line can be obtained from three ports; namely Port B, Port C and Port D.
Interrupts: Two External Interrupt source, located at port D. 19 different interrupt vectors supporting 19 events generated by internal peripherals.
Timer/Counter: Three Internal Timers are available, two 8 bit, one 16 bit, offering various operating modes and supporting internal or external clocking.
SPI (Serial Peripheral interface): ATmega8 holds three communication devices integrated. One of them is Serial Peripheral Interface. Four pins are assigned to Atmega8 to implement this scheme of communication.
USART: One of the most powerful communication solutions is USART and ATmega8 supports both synchronous and asynchronous data transfer schemes. It has three pins assigned for that. In many projects, this module is extensively used for PC-Micro controller communication.
TWI (Two Wire Interface): Another communication device that is present in ATmega8 is Two Wire Interface. It allows designers to set up a commutation between two devices using just two wires along with a common ground connection, As the TWI output is made by means of open collector outputs, thus external pull up resistors are required to make the circuit.

Analog Comparator: A comparator module is integrated in the IC that provides comparison facility between two voltages connected to the two inputs of the Analog comparator via External pins attached to the micro controller.
Analog to Digital Converter: Inbuilt analog to digital converter can convert an analog input signal into digital data of 10bit resolution. For most of the low end application, this much resolution is enough.

Arduino Uno

Arduino Uno is a microcontroller board based on 8-bit ATmega328P microcontroller. Along with ATmega328P, it consists other components such as crystal oscillator, serial communication, voltage regulator, etc. to support the microcontroller. Arduino Uno has 14 digital input/output pins (out of which 6 can be used as PWM outputs), 6 analog input pins, a USB connection, A Power barrel jack, an ICSP header and a reset button.

The 14 digital input/output pins can be used as input or output pins by using pinMode(), digitalRead() and digitalWrite() functions in arduino programming. Each pin operate at 5V and can provide or receive a maximum of 40mA current, and has an internal pull-up resistor of 20-50 KOhms which are disconnected by default. Out of these 14 pins, some pins have specific functions as listed below:

Serial Pins 0 (Rx) and 1 (Tx): Rx and Tx pins are used to receive and transmit TTL serial data. They are connected with the corresponding ATmega328P USB to TTL serial chip.

External Interrupt Pins 2 and 3: These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.

PWM Pins 3, 5, 6, 9 and 11: These pins provide an 8-bit PWM output by using analog Write () function.

SPI Pins 10 (SS), 11 (MOSI), 12 (MISO) and 13 (SCK): These pins are used for SPI communication.

In-built LED Pin 13: This pin is connected with an built-in LED, when pin 13 is HIGH – LED is on and when pin 13 is LOW, it’s off.

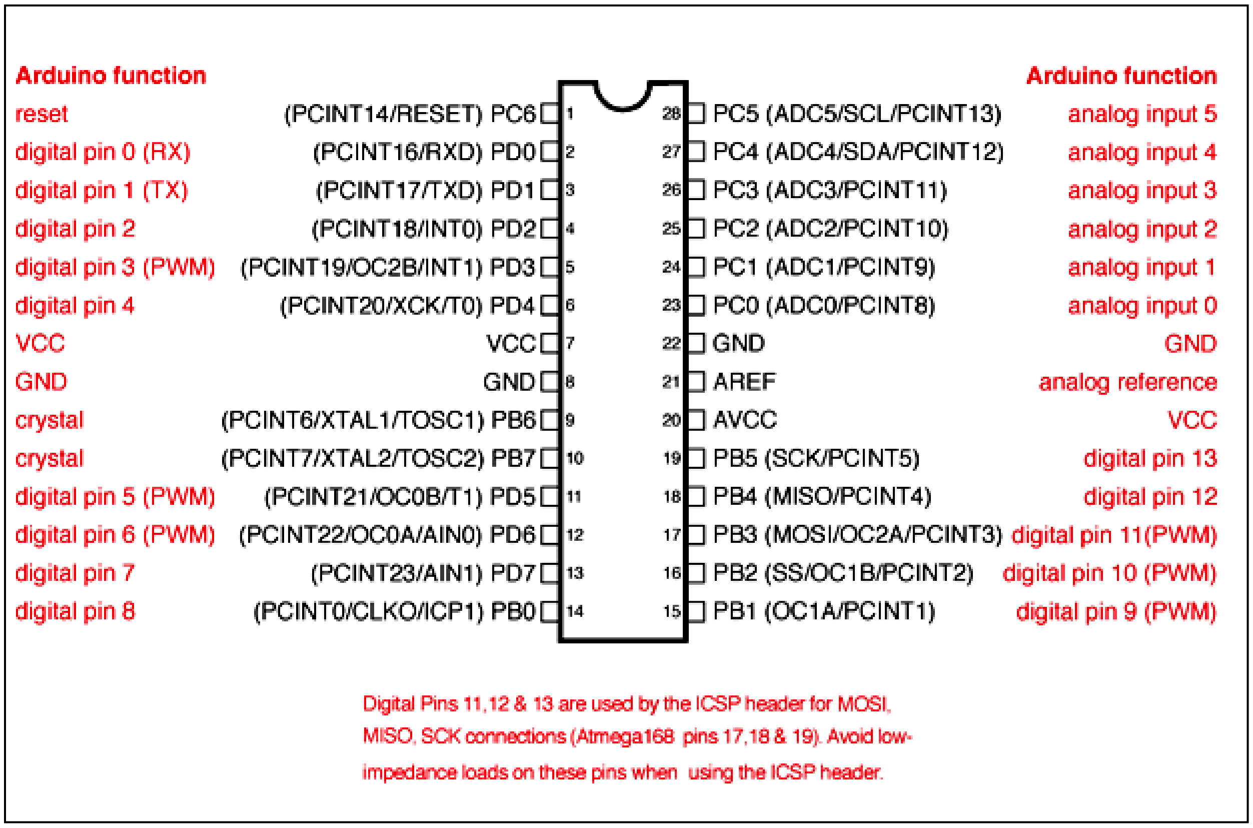
Along with 14 Digital pins, there are 6 analog input pins, each of which provide 10 bits of resolution, i.e. 1024 different values. They measure from 0 to 5 volts but this limit can be increased by using AREF pin with analog Reference () function.

Analog pin 4 (SDA) and pin 5 (SCA) also used for TWI communication using Wire library.

Arduino Uno has a couple of other pins as explained below:

AREF: Used to provide reference voltage for analog inputs with analog Reference() function.

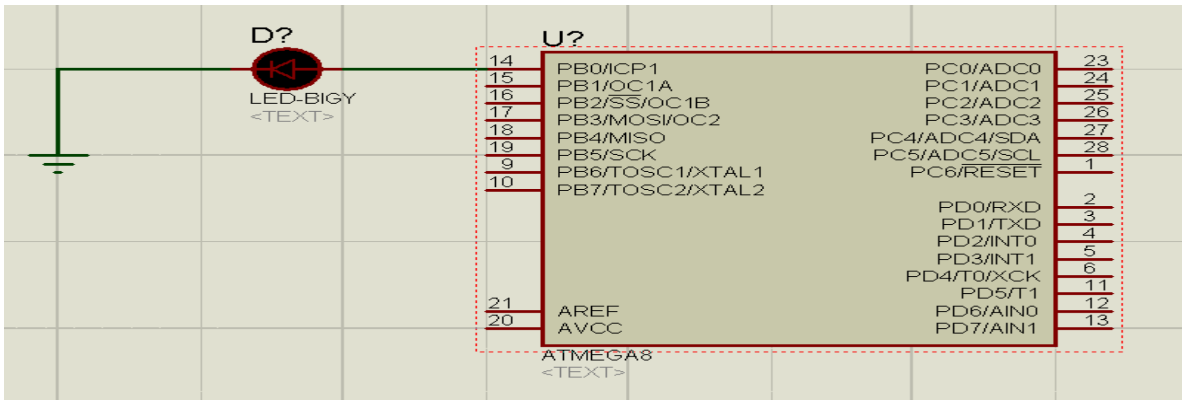
Reset Pin: Making this pin LOW, resets the microcontroller.



Digital Pins 11,12 & 13 are used by the ICSP header for MOSI, MISO, SCK connections (Atmega168 pins 17,18 & 19). Avoid low-impedance loads on these pins when using the ICSP header.

INTERFACING OF LED

LED is a 2 terminal device one terminal is P and other terminal is N.



In this arrangement of led if we give PB0 high then LED will glow because N terminal of led is connect to ground (0).

For this we have to the controller and give particular HEXADECIMAL CODE to PORT B.

For finding HEX CODE for PORT B

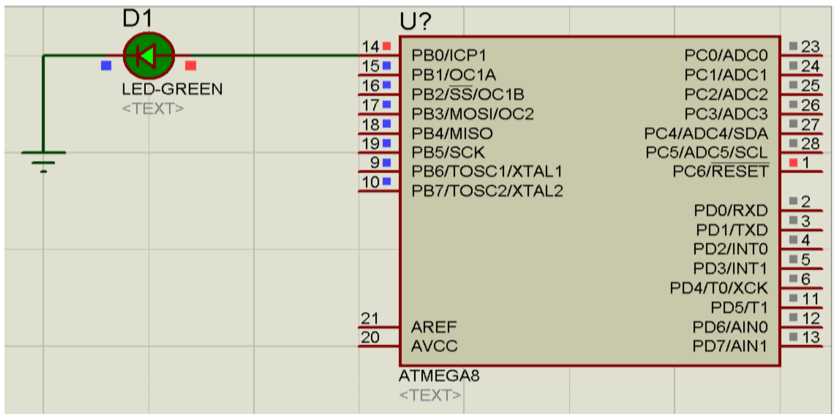
PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
0	0	0	0	0	0	0	1
	0			1			

If we give this data to PORT B

e.g PORT B=0x01;

Then LED will glow .

BY MAKING HEX CODE WE CAN GLOW ANY LED AT ANY PORT.



EXAMPLE

```
#include <avr/io.h>   // I/O ports

#include <util/delay.h>//delay function

int main(void)

{

DDRB=0xff; // it indicates port is used as output or input.

while (1)

  {

     PORTB=0x01; //hex code for port B

                         _delay_ms(10000); //delay time

  }

}
```

INTERFACING OF SERIAL COMMUNICATION IN ARDUINO

Serial communication on pins TX/RX uses TTL logic levels (5V or 3.3V depending on the board). Don't connect these pins directly to an RS232 serial port; they operate at +/- 12V and can damage your Arduino board.

Serial is used for communication between the Arduino board and a computer or other devices. All Arduino boards have at least one serial port (also known as a UART or USART): Serial. It communicates on digital pins 0 (RX) and 1 (TX) as well as with the computer via USB. Thus, if you use these functions, you cannot also use pins 0 and 1 for digital input or output.

You can use the Arduino environment's built-in serial monitor to communicate with an Arduino board. Click the serial monitor button in the toolbar and select the same baud rate used in the call to begin ().

The Arduino Mega has three additional serial ports: Serial1 on pins 19 (RX) and 18 (TX), Serial2 on pins 17 (RX) and 16 (TX), Serial3 on pins 15 (RX) and 14 (TX). To use these pins to communicate with your personal computer, you will need an additional USB-to-serial adaptor, as they are not connected to the Mega's USB-to-serial adaptor. To use them to communicate with an external TTL serial device, connect the TX pin to your device's RX pin, the RX to your device's TX pin, and the ground of your Mega to your device's ground.

The Arduino Due has three additional 3.3V TTL serial ports: Serial1 on pins 19 (RX) and 18 (TX); Serial2 on pins 17 (RX) and 16 (TX), Serial3 on pins 15 (RX) and 14 (TX). Pins 0 and 1 are also connected to the corresponding pins of the ATmega16U2 USB-to-TTL Serial chip, which is connected to the USB debug port. Additionally, there is a native USB-serial port on the SAM3X chip, Serial USB'.

The Arduino Leonardo board uses Serial1 to communicate via TTL (5V) serial on pins 0 (RX) and 1 (TX). Serial is reserved for USB CDC communication.

PROJECT DESCRIPTION

We are realising this project (Automatic door sliding control system) using two ways using -

1. ATmega8 microcontroller and
2. Arduino based microcontroller.

1. ATmega Based

The project Automatic door sliding control system is basically having the control mechanism of identifying the presence of object radiating infrared rays (e.g. human beings).

The identification is directly investigated through PIR sensor. PIR sensor directly gives the binary data in HIGH or LOW after identification. HIGH is generated when it sense infrared rays, otherwise it remains LOW.

The output of the PIR sensor is directly controlled by the microcontroller and after that the door mechanism is operated by the controller itself according to the data provided to controller by PIR sensor.

Door sliding mechanism is controlled by the DC motor using motor driver IC (L293D).

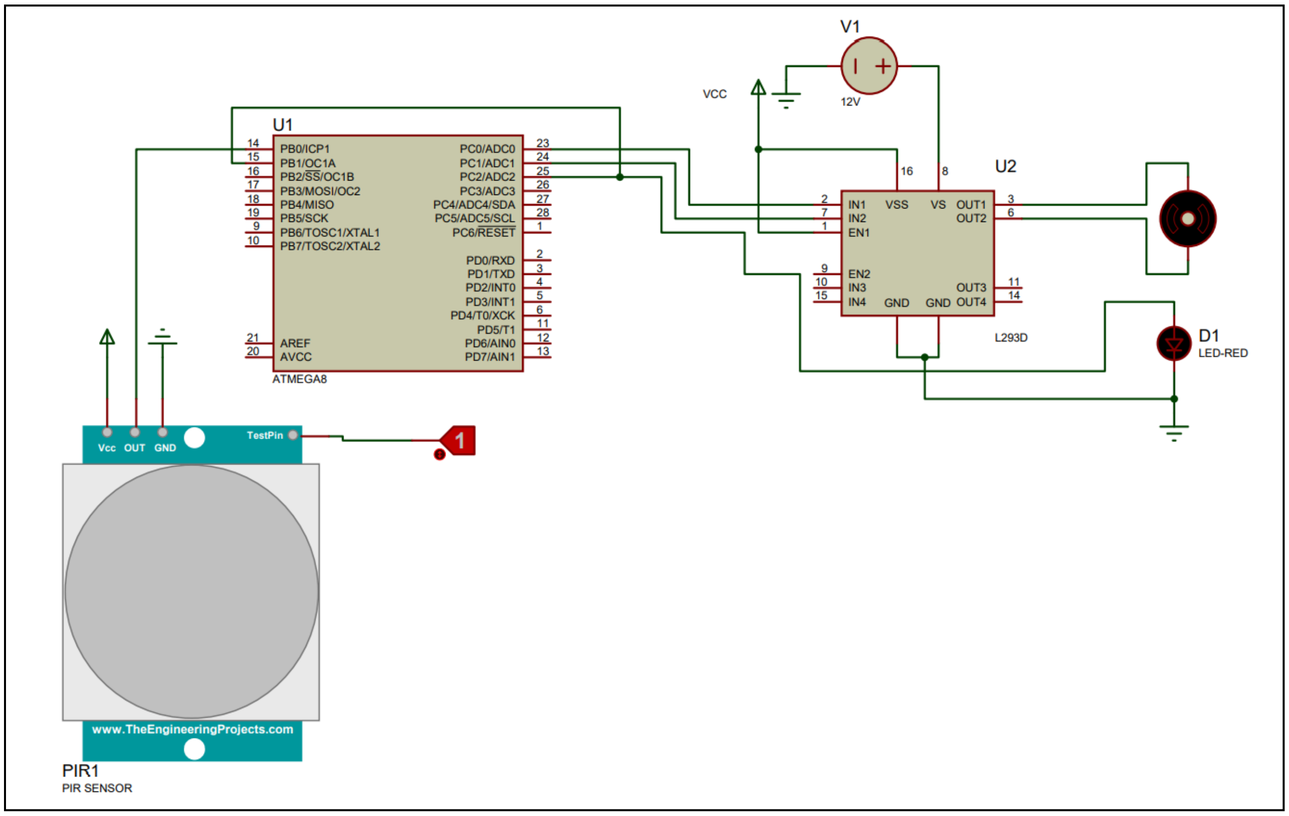
After detection of object motor will perform the function of door open, and after opening of door PIR sensor will again check the presence, if not then motor will perform the function of door close.

In between the door closing function if there is the presence again then the door will open automatically without closing completely.

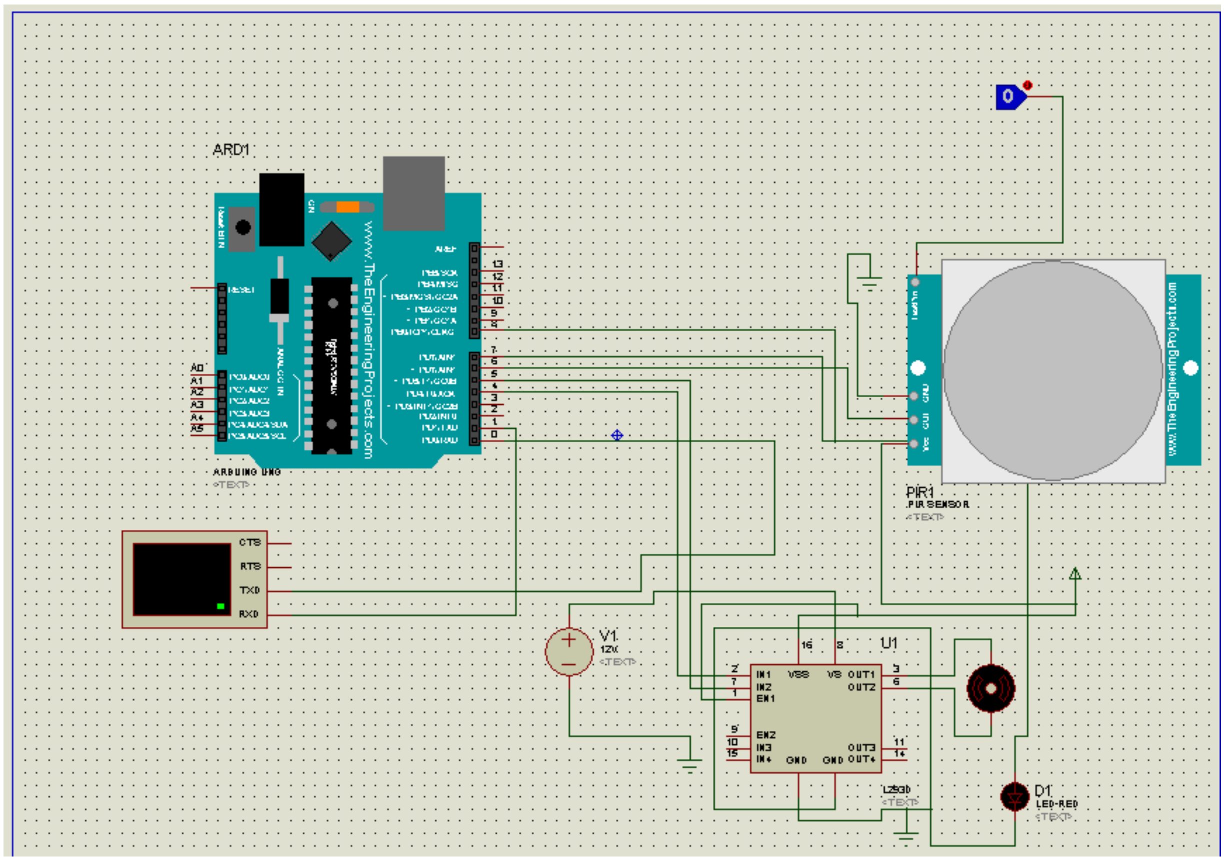
2. Arduino Based

The control mechanism using arduino is similar to ATmega using PIR sensor and motor driver IC. In addition to the above method we have introduced a new functioning using virtual terminal (serial communication - USART, Bluetooth communication). In this new function we are providing the statistics/data of opening and closing activities directly to user using Bluetooth communication.

PROTEUS LAYOUT (USING ATMEGA8)



PROTEUS LAYOUT (USING ARDUINO)



EMBEDDED C - CODE FOR PROJECT

Code for ATmega Microcontroller

```
#include <avr/io.h>
#include<util/delay.h>
int main()
{
        DDRB=0x00;
        DDRC=0xff;
         while(1)
         {
                if((PINB&0x03)==01)
                {
                        open();
                }
          if ((PINB&0x03)==02)
          {
                        close();
          }
         }
}
void close()
{
        PORTC=0x05;
        for (int i=0;i<=9;i++)
        {
                _delay_ms(200);
                if ((PINB&0x03)==03)
                {
                        open_interrupt(i);
                        return;
                }
        }
        PORTC=0x00;
}
void open()
{
        PORTC=0x06;
        _delay_ms(2000);
        PORTC=0x04;
}
void open_interrupt(int x)
{
        PORTC=0x06;
        for (int y=0;y<x;y++)
        {
                _delay_ms(200);
        }
        PORTC=0x04;
}
```

Code for Arduino

```
const int in1 = 4;

const int in2 = 5;

const int pirin = 6;

const int ledout = 7;

const int ledin = 8;


void setup()

{

  pinMode(in1, OUTPUT);

  pinMode(in2, OUTPUT);

  pinMode(pirin, INPUT);

  pinMode(ledout, OUTPUT);

  pinMode(ledin, INPUT);

  Serial.begin(9600);

}


void loop()

{

     int countopen=0,countclose=0;

     while(1)

     {

     if(digitalRead(pirin)==HIGH)

     {
      if(digitalRead(ledin)==LOW)
      {
        open_door();
        countopen=countopen+1;
      }
    }
    if (digitalRead(pirin)==LOW)
    {
      if(digitalRead(ledin)==HIGH)
      {
        close_door();
        countclose=countclose+1;
      }
    }
    Serial.print(countopen);
    Serial.print(" - Open Count & , ");
    Serial.print(countclose);
    Serial.print(" - Close Count");
    Serial.println();
    delay(1000);
    }
}


void close_door()
```

```
{
 digitalWrite(in1,HIGH);
 digitalWrite(in2,LOW);
 digitalWrite(ledout,HIGH);
 for (int i=0;i<=9;i++)
 {
   delay(100);
   if (digitalRead(pirin)==HIGH)
   {
     if(digitalRead(ledin)==HIGH)
     {
       open_interrupt_door(i);
       return;
     }
   }
 }
 digitalWrite(in1,LOW);
 digitalWrite(in2,LOW);
 digitalWrite(ledout,LOW);
}

void open_door()
{
  digitalWrite(in1,LOW);
  digitalWrite(in2,HIGH);
  digitalWrite(ledout,HIGH);
  delay(1000);
  digitalWrite(in1,LOW);
  digitalWrite(in2,LOW);
  digitalWrite(ledout,HIGH);
  return;
}

void open_interrupt_door(int x)
{
  digitalWrite(in1,LOW);
  digitalWrite(in2,HIGH);
  digitalWrite(ledout,HIGH);
  for (int y=0;y<x;y++)
  {
    delay(100);
  }
  digitalWrite(in1,LOW);
  digitalWrite(in2,LOW);
  digitalWrite(ledout,HIGH);
  return;
}
```

SOFTWARES USED

- PROTEUS (Virtual Design and Real Simulation)

  The Proteus Design Suite is a proprietary software tool suite used primarily for electronic design automation. The software is used mainly by electronic design engineers and technicians to create schematics and electronic prints for manufacturing printed circuit boards.
- ARDUINO IDE

  Arduino IDE is open source software that is mainly used for writing and compiling the code into the Arduino Module. It is official Arduino software.
- ATMEL STDIO 7.0

  Studio 7 is the integrated development platform (IDP) for developing and debugging all AVR® and SAM microcontroller applications. The Atmel Studio 7 IDP gives you a seamless and easy-to-use environment to write, build and debug your applications written in C/C++ or assembly code.

REFERENCES

- https://www.tutorialspoint.com/embedded_systems/ to get the information about embedded systems.
- https://www.circuitstoday.com/avr-atmega8 to get information about ATmega8 microcontroller.
- https://en.wikipedia.org/wiki/Arduino_Uno to get information about Arduino Uno
- https://drive.google.com/file/d/1Eyr2- 68qpmV8rrHIxUj6J6OeE-37GZAK/view?usp=sharing International Journal of Scientific Engineering and Technology Research (Volume.07, IssueNo.12)
- Code of the Project has been written by ourselves with best of our knowledge and not copied from anywhere.