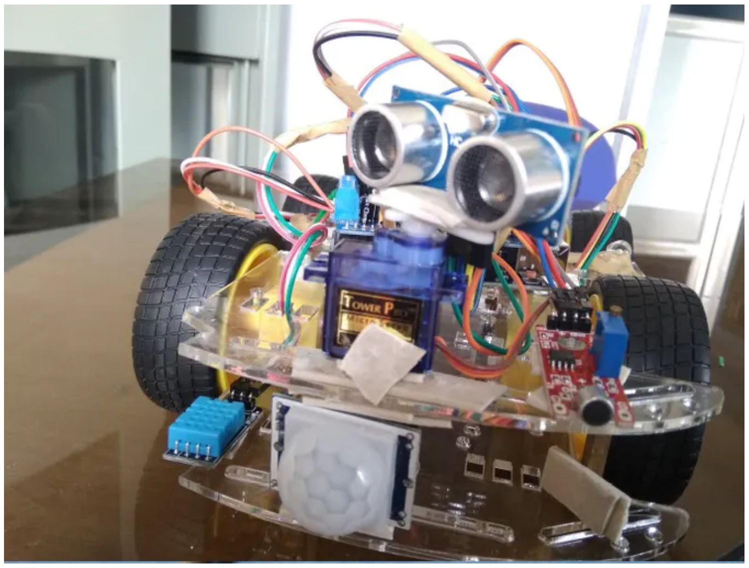
MINI TASK - 1

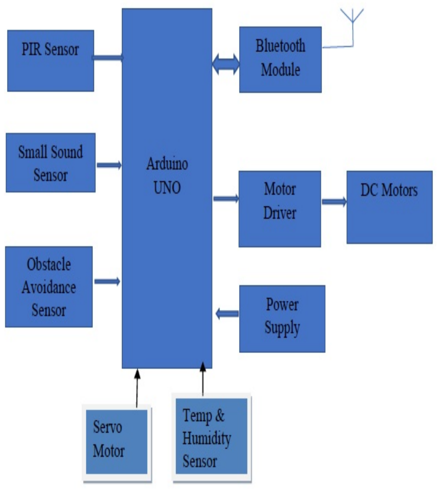
Project 1:

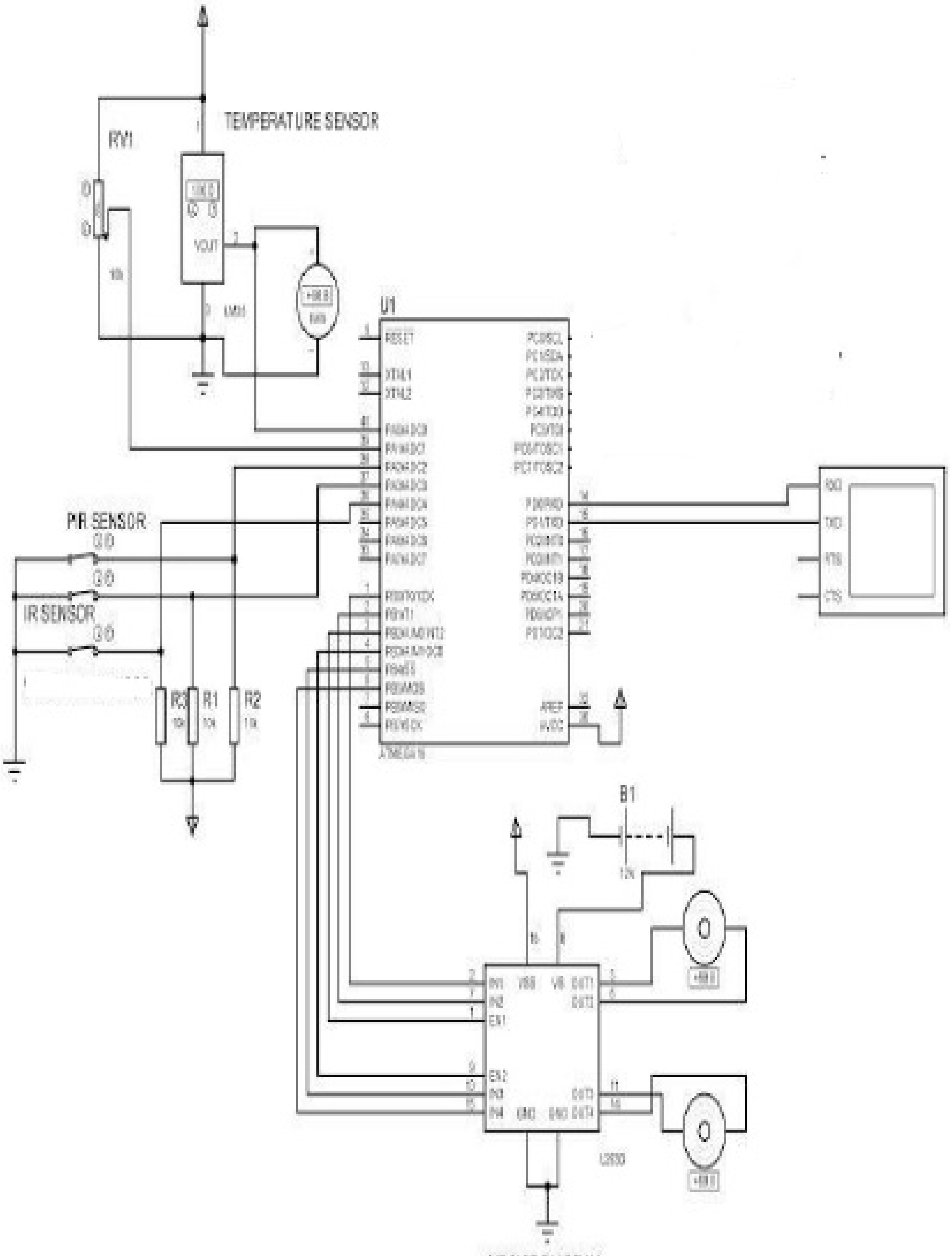
Human detection Robotics Systems using Arduino:
This System design a mobile rescue robotic Vehicle system based on Arduino to help the people on time which are trapped in natural calamity like disaster, earthquake, floods etc.It gives timely & accurately reflect dynamic situation of human in disaster region like in the underground regions to control room, so that rescue team of Experts & doctors can be sending to the victim’s location for primary treatment and can be sent to the safe place or hospital. The entire process takes place within a few seconds as the system is controlled by a Arduino unit. PIR sensors are passive infrared sensors which detects movement of people with the help of changes in the infrared (heat) levels emitted by surrounding objects. The human body emits thermal radiation at a wavelength of about 10 microns. It is received and manipulated by the PIR sensor to detect human beings. It operates at 5V DC. The motion of the human being can be detected by checking for a sudden change in the surrounding IR patterns. Obstacle sensor detects the obstacle and sends the analog signals to the Arduino. Arduino is programmed to guide the robot automatically depending on the obstacle detected and to send the human being information to remote control place through the Bluetooth Technology. The Data is received in the Base Station(Control Center). Analyzing the data the Rescue team can take necessary steps to rescue the trapped human Beings.

This project can be further improved by connecting a camera and running DL models to identify the humans. We can improve their code by creating a for-loop and taking larger data and using moving time average algorithms we can reduce noise and we can get more precise

location of the human. We can improve our bot without investing any extra money but rather making slight changes in our code.
More about this project can be found out using this link.





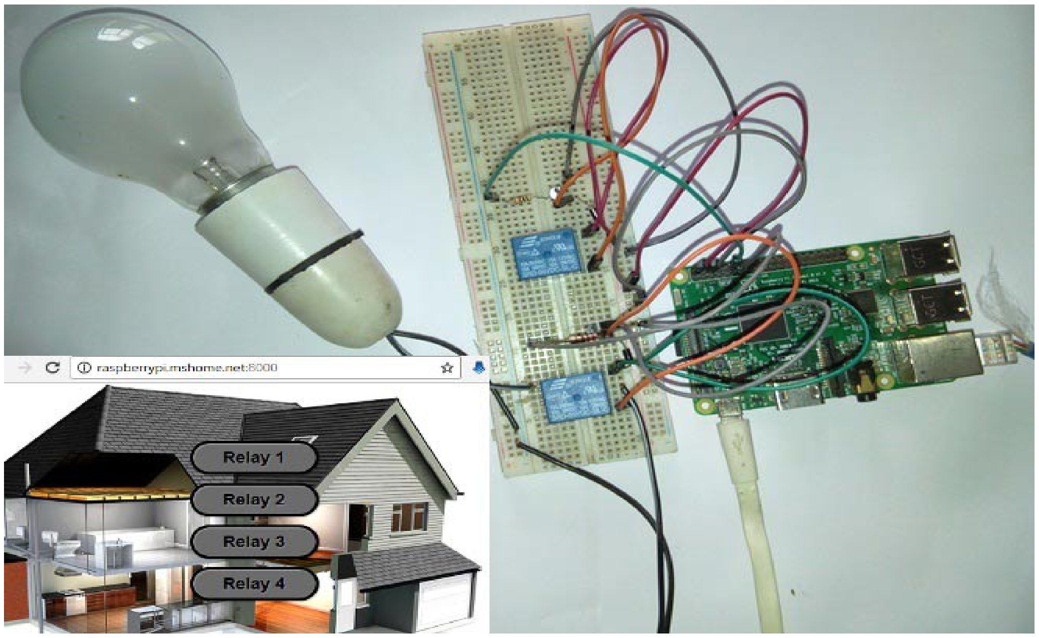


CIRCUIT DIAGRAM

Project 2:

IoT Based Web Controlled Home automation using Raspberry PI:

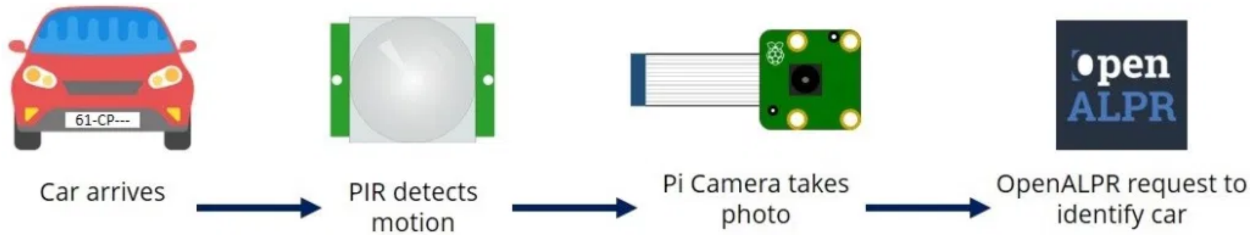
In this project we are going to explore the possibility of **controlling AC appliances with the click of buttons on a webpage using internet**. Using this **IoT based home automation system**, you can control your Home appliances from anywhere in the world. This web server can be run from any device which can run HTML applications, like Smart Phone, tablet, computer etc.
In this project we don't even need to code in python.
All we need is to use WebIOPi framework, notepad++ in pc and filezilla to copy files from pc to raspberry pi especially the web app files.
We have to install **the WebIOPi framework** which will help us to handle the communication between the webpage and raspberry pi.
We can beautify the webpage and add all the files using css and other files.
We can improve this project by using python Flask framework. The tasks would have been easier.



Project 3:

Car Plate recognition using Raspberry pi and Node-RED:

In this project we will be going to recognize car plates using a software called OpenALPR(Automatic License Plate Recognition) that is an API We can use to identify car plates and car models based on the image. We will use PIR sensor to detect the car. When the sensor detects a motion the Raspberry Pi camera will take the picture which inturn sends a request to OpenALPR with the car photo to be identified which will return the plate number ,color and model name. After identifying a car, we'll do some verifications, and if we found an authorized car, we'll trigger an event (that can be open the garage, for example)



Improvisations can be made by using other sensors like Hall effect sensor, Ultrasound sensor and Infrared sensor and which can fuse all the sensors' values to get an optimum solution.link

Project 4:
Alexa(Echo) with controlled Voice controlled Relay using ESP-32
In this project Alexa will respond to certain commands and helps us switch on or off lights.
We will be using 433 MHz RF wall panel switch which is used to remotely control the lights. To control ESP32 with alexa we need to use FauxmoESP library. This library emulates a Belkin Wemo device, allowing you to control your ESP32 or ESP8266 using this protocol.
We can learn more about this project using this link.

Project 5:

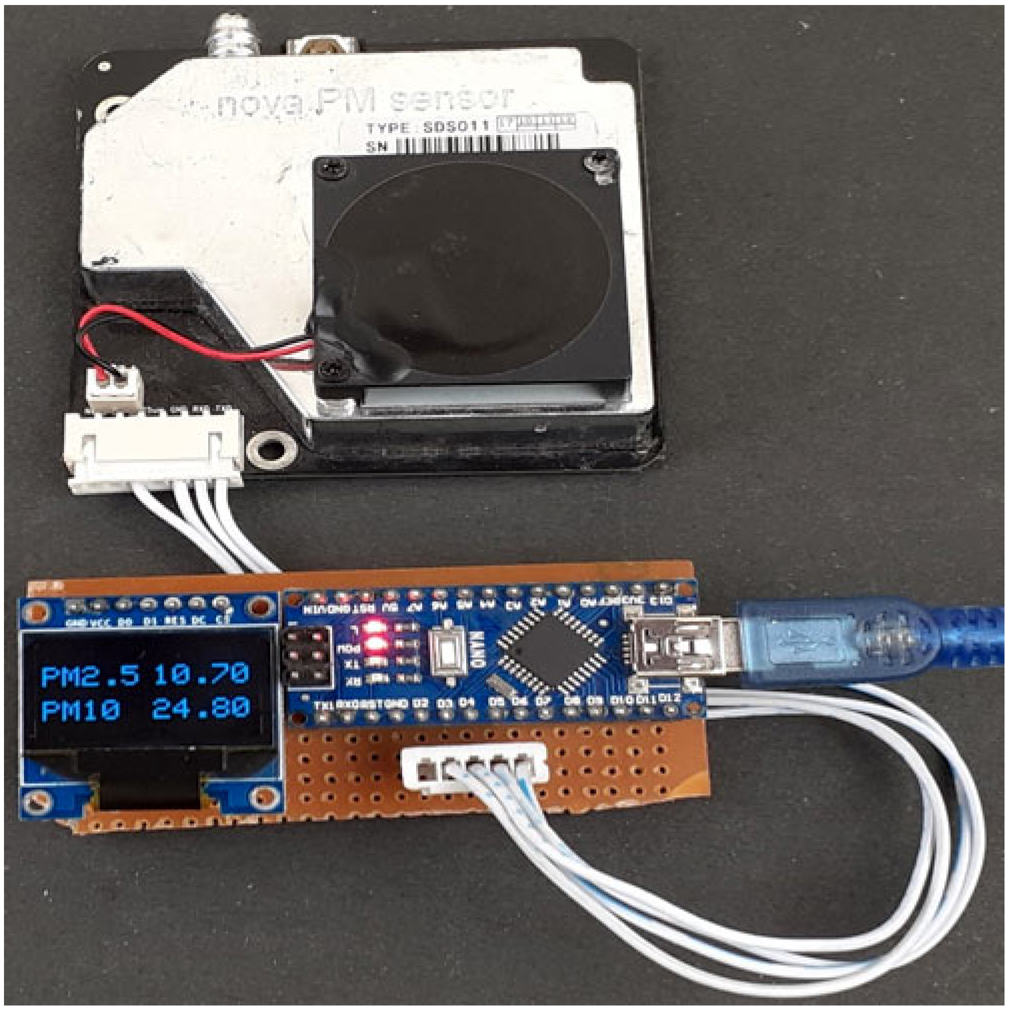
Speech Recognition using Arduino Nano:

The hardware is pretty straightforward. It requires an Arduino uno and a microphone amplifier. The microphone amplifier captures the sound and transfers it to Arduino which in turn runs the algorithm inside. The learning part is done by the PC.
To analyze the utterances, first divide each sample utterance into 50 ms segments. Think of dividing a single spoken word into its different syllables. Like analyzing the “se-” in “seven” separate from the “-ven.” 50 ms might be too long or too short to capture each syllable cleanly, but hopefully, that gives you a good mental picture of what program is doing. We then calculate the energy of 5 different frequency bands, for every segment of every utterance. We implement it using 5 digital band pass filters, allowing us to more easily compute the energy of the signal in each frequency band. The energy of each frequency band for every segment is then sent to a PC where a custom-written program creates “templates” based on the sample utterances generated.The crux of his algorithm is comparing how closely the energy of each frequency band for each utterance (and for each segment) is to the template. The PC program produces a .h file that can be compiled directly on the Nano.link

Project 6:

Air Quality Monitor using Arduino:

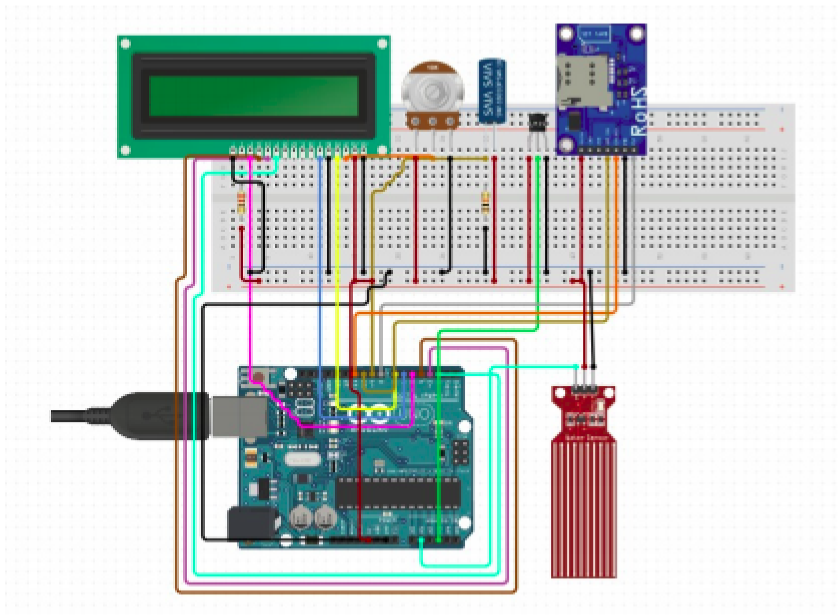
It is used to give PM10 and PM2.5 levels in air. Nova PM Sensor SDS011 is used to determine the air quality. The sensor uses a laser diode and a photodiode to detect and count particles, while a fan moves air through the system. If you aren't up on pollution metrics, PM2.5 is a count of very fine particles (under 2.5 microns) and PM10 is a count of particles for 10 microns.
We can use MQ-2 Gas sensor to determine whether the air present has lot of smoke or other harmful gases. link



Project 7:

IoT based Bridge Health Monitoring System and alerting System:

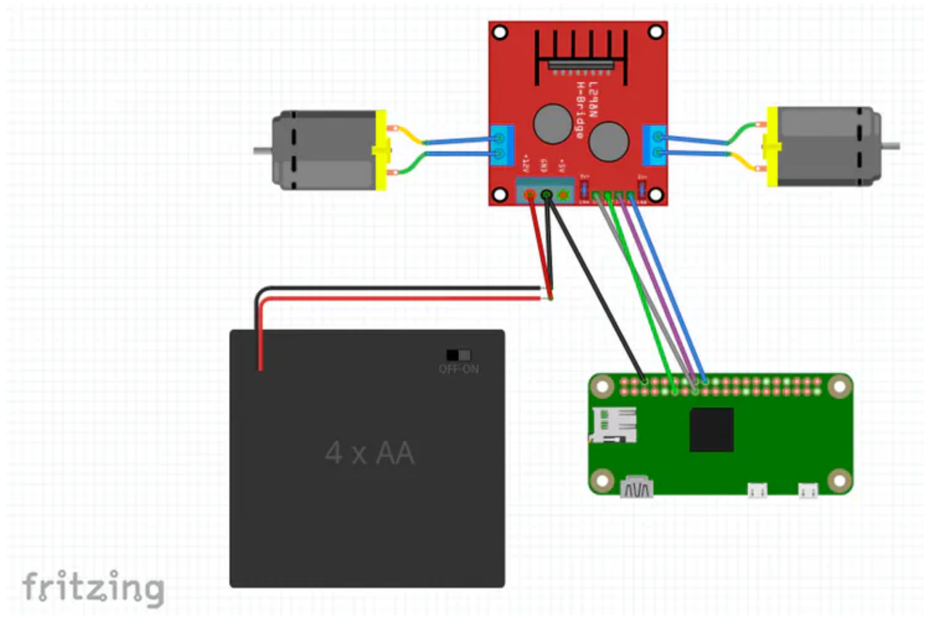
The aim of this project is to support the construction of an efficient HealthMonitoring System for ensuring the safety, using-life of bridges, preventing the collapse affairs, protecting people lives, environment, and reducing unnecessary finance expenses.This report presents an effective method for application on vibration, temperature, and water level signal analysis in Bridge Health Monitoring.The data collected from sensors can be transported through GSM Module to the cloud storage which is the Thingspeak cloud. The various signals from sensors are collected through a wireless sensor network. These signals are very important in Bridge HealthMonitoring System because the variation of bridge signals indicates the changing of bridge structure state. link



Project 8:

Internet controlled RC car with HD video streaming using Raspberry PI:

The project is to build a car which is controlled by web and gives us live video stream option.We can access the car if we are connected to the same IP and we should type **http://raspberry_ip:8000/** as our web address in our web browser. The car requires an L298 as a motor driver to control the motors which are responsible for the movement of the car. The schematic for connecting the motor drivers, Rpi and motors is as shown below:



Link

Project 9:
Car Parking Guard Circuit Using Infrared Sensor:

This circuit helps the person in the driving seat in such a way that it gives an alarm if there is any obstacle or a wall while parking or while driving in reverse. It is very useful in our real life.

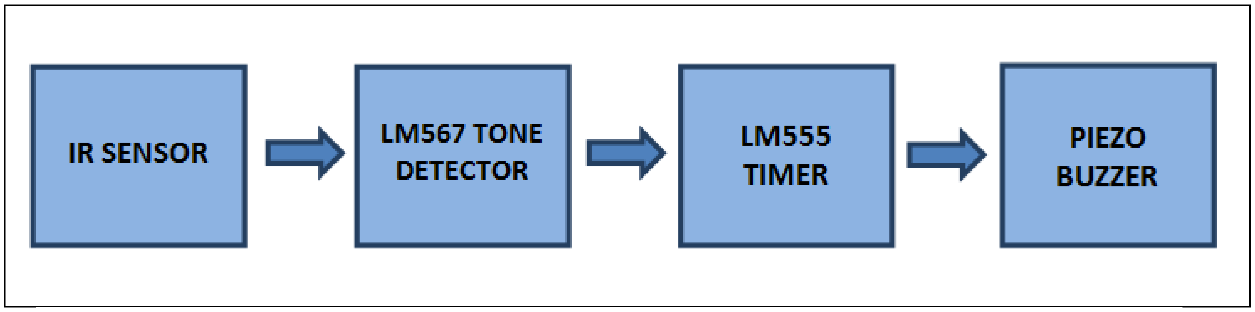
<u>Working:</u>

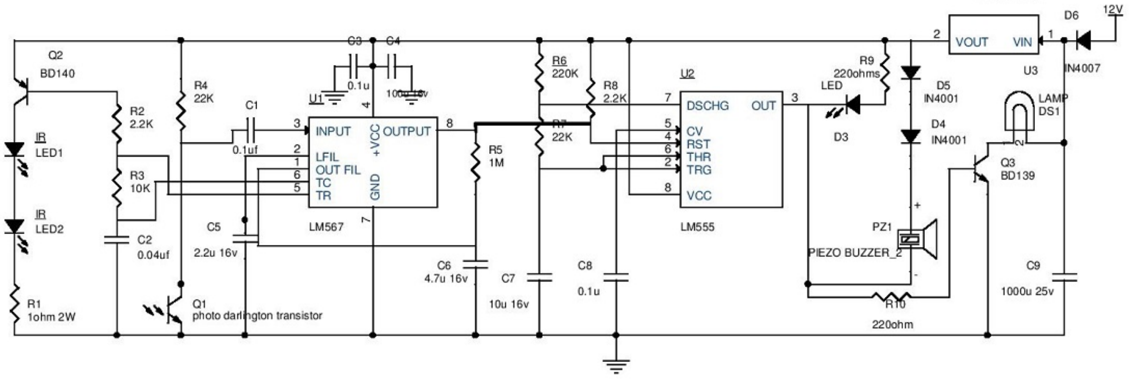
The reverse indicator light supply is given to the 7805 regulator to give 5V to the rest of the circuit. The diode D6 is used to eliminate the reverse current and wrong supply polarity.

The IR sensor will detect the obstacle within 100cm, if there is any obstacle it will sense and give information to the tone detector which will enable the LM555 timer to generate a PWM for the buzzer. The LM555 will generate the pulse which helps to buzz the buzzer so the driver can understand that there is an obstacle.

The main components used are:

- LM567
- LM555
- IR Sensor
- Photo Darlington Transistor



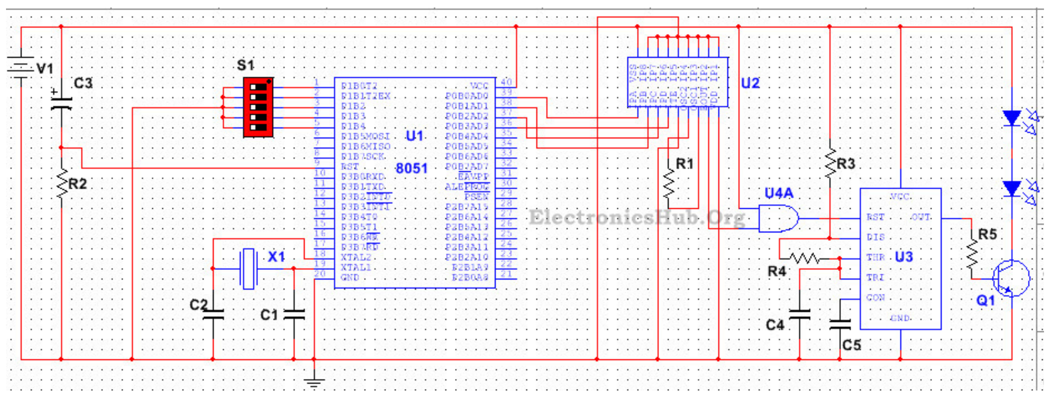


Project 10:
5 Channel IR Remote Control System using Microcontroller:

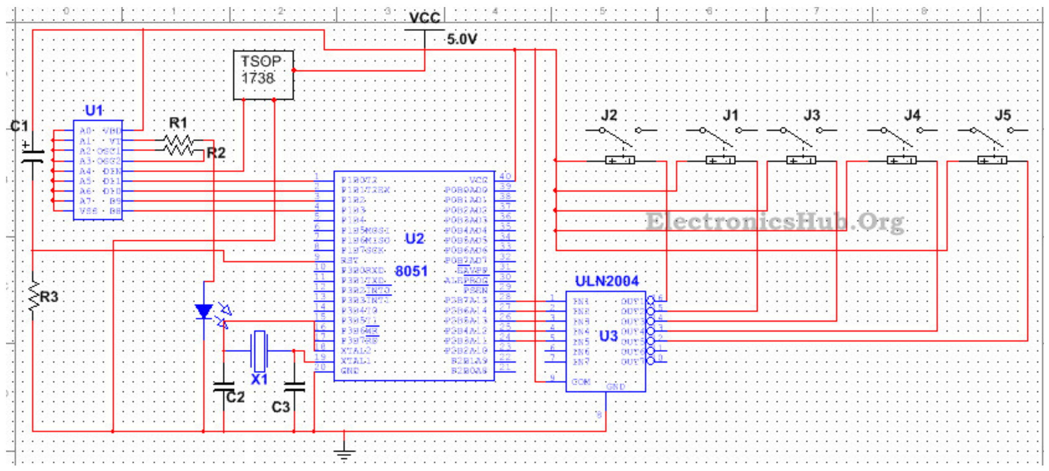
This project is aimed to design and demonstrate a simple 5 channel remote control system to drive five loads. This circuit works on the principle of IR communication.

The circuit works on the principle of IR communication. IR communication involves transmitting signals using infrared signals as the carrier. The input signal from switches is processed by the microcontroller, encoded by the encoder, modulated and transmitted by the transmitter. At the receiver, the modulated signal is demodulated by the IR receiver, decoded by the decoder and processed by the microcontroller to control the output loads.

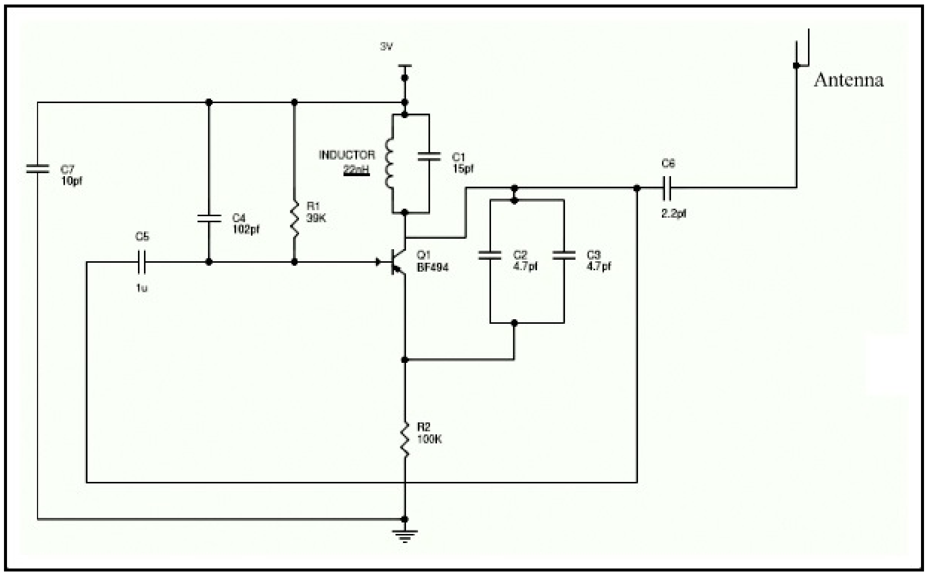
More details can be found in the link.



Transmitter Circuit



Project 11:
Mobile Jammer Circuit:
A Mobile Jammer Circuit or a Cell Phone Jammer Circuit is an instrument or device that can prevent the reception of signals by Mobile Phones. Basically, a Mobile Jammer Circuit is an RF Transmitter, which broadcasts Radio Signals in the same (or similar) frequency range of the GSM Communication.
This circuit is used to block the signals of cell phones within the range of 100 meters. This circuit can be used in TV transmission and also for remote controlled toys or play things.



More Information found at link

Project 12:
Auto Intensity Control of Street Lights:
This is a simple circuit that automatically controls the intensity of street lights which is designed using microcontrollers and LEDs.

Circuit Principle:

The main principle of this project is to Control the intensity of street lights using PWM.Peak hours of a particular area are calculated and accordingly PWM signal is adjusted by microcontroller to increase or decrease the intensity of street lights.

These peak hours can be calculated by considering parameters like traffic density,time, and light intensity of the environment.

Advantages:

- Power wastage can be reduced.
- Using LED array reduces the cost.
- Using RTC and LDR produces accurate results.

Details will be explained in the next task.

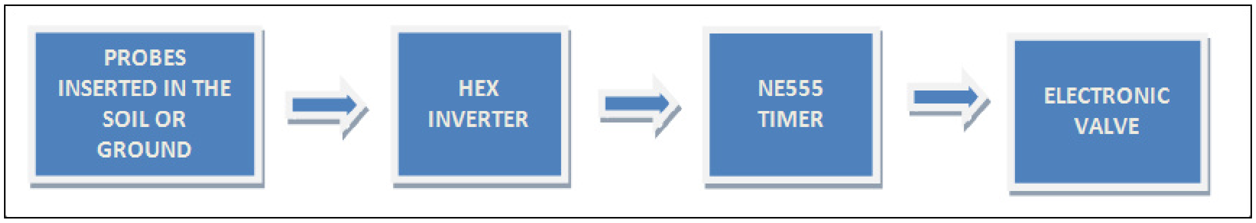
Project 13:

Automatic Plant Irrigation System:

This project circuit is more useful in watering plants automatically without any human interference. It is more useful when the owner is not present in the home for a few days.

We use the basic concept in this circuit i.e. soil have high resistance when it is dry and has very low resistance when it is wet.We insert two probes in the soil in such a way that that they will conduct when the soil is wet and they will not conduct when the soil is dry. So, when the probes do not conduct, system will automatically detect this condition with the help of HEX inverter which will become high when the input is low.HEX inverter will trigger the NE555 Timer and this NE555 timer will trigger another NE555 which is connected to the output of first NE555. Now the second NE555 which is configured as astable multivibrator will help to switch on the Electric valve and as result, it will allow the water to flow to the soil.

When the water wet the soil, probes will again conduct and make the output of 7404 low which will make the first NE555 to low and also drive the remaining circuit to low. So, automatically it will switch off the valve.



More information can be found at link

Project 14:

Smoke Detector Alarm Circuit:

Here is the circuit showing a smoke detector alarm. This can also be used to indicate the fire. This circuit uses a smoke detector and an LM358 Comparator.

Working:

LM358 acts as a comparator in this circuit. The inverting terminal of LM358 is connected to POT so that the sensitivity of the circuit can be adjusted.

The output of LM358 is given to an LED as an indicator although a buzzer can be used as an alarm. The non-inverting terminal of LM358 is connected with the output of the smoke sensor.

Initially, when the air is clean, the conductivity between the electrodes is less, as the resistance is in the order of 50KΩ. The inverting terminal input of the comparator is higher than the non-inverting terminal input. The indicator LED is OFF.

In the event of fire, when the sensor is filled with smoke, the resistance of the sensor falls to 5KΩ and the conductivity between the electrodes increases.

This provides a higher input at the non-inverting terminal of the comparator than the inverting terminal and the output of the comparator is high. The alarming LED is turned ON as an indication of the presence of smoke.

More information can be found at link

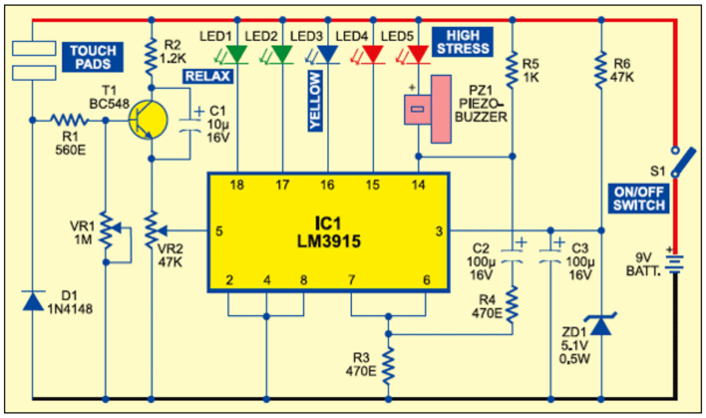
Project 15:

Stress Meter:

This stress monitor lets you assess your emotional pain. It can indicate three levels of stress: Relaxed, Moderate, and High levels with the help of a series of LEDs (a combination of green, yellow, and red). If the stress is very high, it gives a visual indication through the LED display along with a warning beep. The gadget is small enough to be worn around the wrist.

The gadget is based on the principle that the resistance of the skin varies in accordance with your emotional states. If the stress level is high the skin offers less resistance, and if the body is relaxed the skin resistance is high. The low resistance of the skin during high stress is due to an increase in the blood supply to the skin. This increases the permeability of the skin and hence the conductivity for electric current.

This property of the skin is used here to measure the stress level. The touchpads of the stress meter sense the voltage variations across the touch pads and convey the same to the circuit. The circuit is very sensitive and detects even a minute voltage variation across the touch pads.



Mini-Task 2:

Internet Controlled RC-car with HD Streaming using Raspberry Pi:

Problem Statement: Live Stream a video of the surroundings of the car and controlling it using other devices using wireless mode of transmission(web)

There is a huge setback with using “internet” in this project because it can mean anyone in the entire world can control the car ,that is it can be easily hacked and since its video streaming it may lead to privacy issues also. I would rather suggest to use the concept of intranet because it will be within a closed network only trusted people can use them. However it is not so fool proof but still it reduces most of our disadvantages to a large extent. Using intranet reduces our internet cost also as we won’t really require to pay a price for our network providers for using intranet. We can extend our Car range by using a certain number of repeaters or single repeater which follow IEEE Wifi protocol for the required range. We can attach a repeater to our Rpi and then it will extend the range for us and we can control it from a longer distance. Usually transmitting via Wifi signals adds noise to our video. So, it would be better if we include some noise filtering algorithms before streaming the video on a server. Doing this we will get a more clear video.
In conclusion, my idea is cost-effective and secure.

Debugging:

The Motor controlling part sequence is :

RaspberryPI → motor driver

The video Transmission part sequence is :
RaspberryPI cam → RaspberryPI → Server

We can easily deduce that if we aren't getting the right output the problem lies in these key parts only.

If all the modules light up then we have to get down to the software part,If they don't that means we would have either shorted it or power connections aren't proper.

Debugging Raspberry PI camera:

Insert the Raspberry PI camera and install the module to the Raspberry PI and if the setup fails the problem is with the connections or most probably with the camera.
Now type :

```
$ sudo raspi-config
```

And use arrow keys and enable camera

To check whether the camera is working properly or not connect it to a TV or something and run this on your Raspian:

```
raspistill -o output.jpg
```

It will first show us a preview of the image and seconds later it will be taking the snap and it will store as output.jpg
If this doesn't work then the Rpi camera is not working properly we would rather think of changing the Rpi camera.

Debugging Motor Driver:

There is a direct program by Adafruit to check whether our Motor Drivers are working properly or not.
Perform the following commands:

```
$wget
https://raw.githubusercontent.com/adafruit/Adafruit_Learning_System_Guides/master/Raspberry_Pi_Stepper_Motors/Raspberry_Pi_Stepper_Motors.py
```

```
sudo python3 ./Raspberry_Pi_Stepper_Motors.py
```

Enter a delay (5 is a good value) and then a number of steps (512 is a full rotation).

So we can check whether its working or not. If its not working then try making the connections proper and running it again, If it doesn't then replace the motor driver.

Test for RaspberryPI:

If the IC appears to be heated up that means the Raspberry pi has been short circuited and we need to replace it. Some times you need to download a new version of Raspian and flashing it into a new SD card and then it might work proper. There are applications like PIDoctor which we can install and monitor the health of our RaspberryPI.

Auto Intensity Control of Street Lights:

Problem statement: To save energy by calculating the traffic density at different times of the day and thereby adjusting the intensity of street lights.

Ideation, Planning and Implementation:

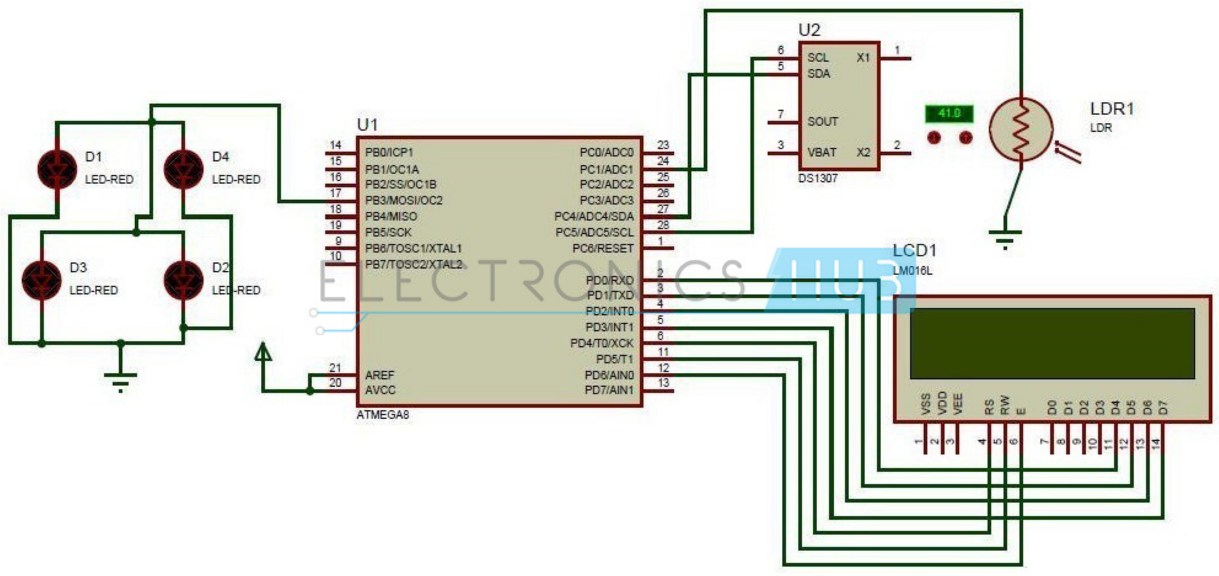
This project can be done by following two pipelines.

Pipeline 1: Auto Intensity Control of Street Lights using ATmega8

Circuit Principle:

The main principle of this project is to Control the intensity of street lights using PWM .Peak hours of a particular area are calculated and accordingly PWM signal is adjusted by microcontroller to increase or decrease the intensity of street lights. These peak hours can be calculated by considering parameters like traffic density,time, light intensity of the environment.

Circuit:



The auto intensity control of street lights circuit is simple but it requires more coding part. This circuit consists of Atmega8 controller, DS1307, LDR, Relay and LEDs.

LDR: LDR is used for calculating the light intensity of the environment .The light dependent resistor is connected to ADC1 (PC1) pin of the micro controller. The analog light value is converted to digital value using ADC.

RTC: Current time is calculated using RTC. Real time clock has 8 pins out of which SCL and SDA are connected to PC5 andPC4 pins respectively. SCL is serial clock while SDA is serial data RTC is I2C compatible, where I2C means inter integrated circuit. One bit of data is transmitted on data bus for each clock cycle.

LCD: LCD is the display used for displaying time which is read from RTC IC. Interfacing of LCD in 4-bit mode is shown in circuit diagram. D4-D7 pins of LCD are connected to PD0-PD3 pins of microcontroller.

RS pin of LCD is connected to PD4 pin of micro controller. RW and Enable pins are connected to PD5 and PD6 pins of controller.

LED array is number of high power LEDs connected in series. It is connected to PWM pin of the microcontroller.

Working:

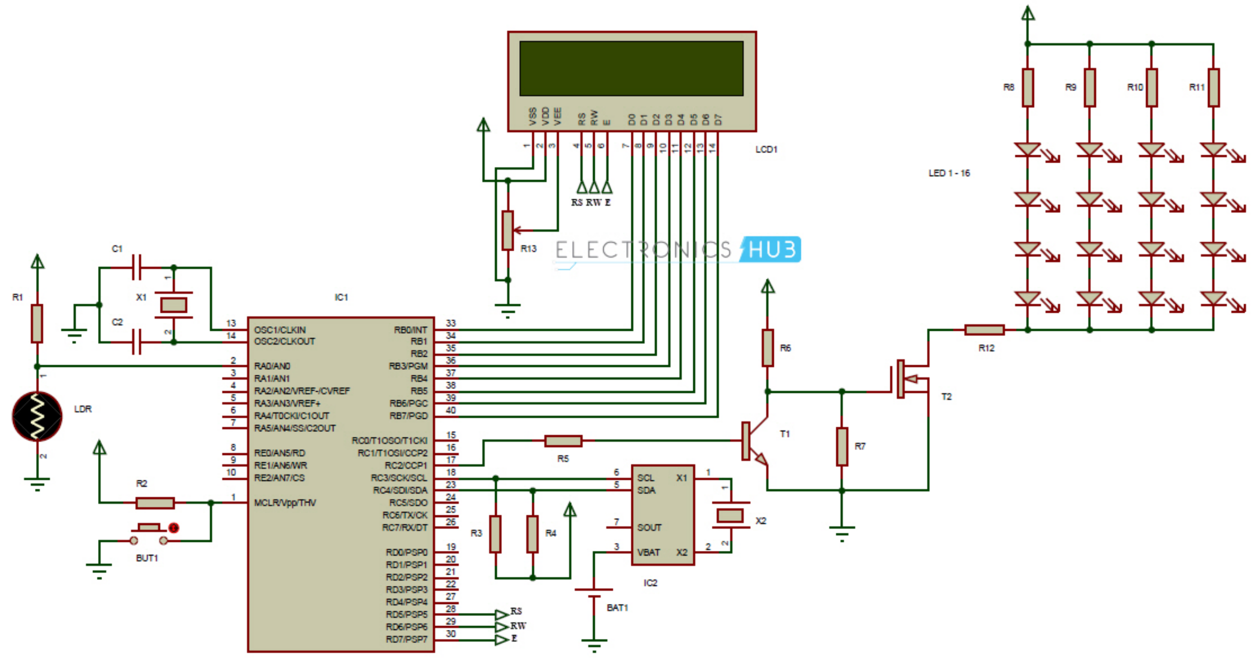
1. Initially power the circuit.
2. Time is displayed on the LCD display.
3. Place the LDR in darkness as the street lights switches on only when there is no light on LDR.
4. Now check the time if the time is between 9 pm to 2 am street light glows with full intensity.
5. From 2 pm intensity of the lights slowly starts decreasing and finally in early morning it glows with least intensity. When the light is sensed by the LDR lights are switched off automatically.

Limitations of this Circuit:

- Even though energy is saved if there are any vehicles after fixed time, intensity of the light is low.
- Maximum energy cannot be saved.

Pipeline 2:

Auto Intensity Control of Street Lights using a PIC Microcontroller



Component Description:

PIC16F877A

The microcontroller used in the circuit is a PIC16F877A. It is an 8 – bit microcontroller that reads the voltage across LDR and also checks the time in Real Time Clock IC. Based on the readings, the LEDs are switched on or off.

DS1307

It is a Real Time Clock IC. The communication between microcontroller and DS1307 is via I2C protocol. It provides clock and calendar with details like seconds, minutes, hours, day, date, month and year. Time can be set in either 12 hour mode or 24 hour mode and there is an indication of AM/PM.

Working:

We use both LDR and RTC in the circuit for the following reason: if only LDR is used, then there is no chance of saving any energy as the street lights will glow as soon as the intensity of light on LDR decreases and when the intensity increases, the street lights are turned off.

If only RTC is used, the street lights are turned on and off at preset time irrespective of the outside lighting conditions. When the device is turned on, RTC starts with the preset time in the code.

The microcontroller waits for the signal from LDR and when the intensity of light on LDR decreases, the output of the microcontroller is activated and the street lights start to glow. This event occurs only when the current time is in the range of preset time i.e. only after 5PM.

The lights continue to glow at full intensity up to 3 AM. When the time reaches 3 AM, the intensity of the street light gradually decreases and will turn off either at 6 AM or when the light on LDR in increasing, whichever is first.

Hence, we are going to choose pipeline 2- the auto intensity control of street lights is achieved with the circuit which has an LDR, an RTC, a PIC microcontroller and an LED array.

Mini-Task 3:

Debugging the project:

Auto Intensity Control of Street Lights:
The potential problems that we can face are defects in PIC16F877A and DS1307.

In-Circuit Debugging of PIC microcontroller (PIC16F877A):

We will be using mikroElektronika's PICFlash with mikroICD device in conjunction with the mikroC Pro for PIC compiler.

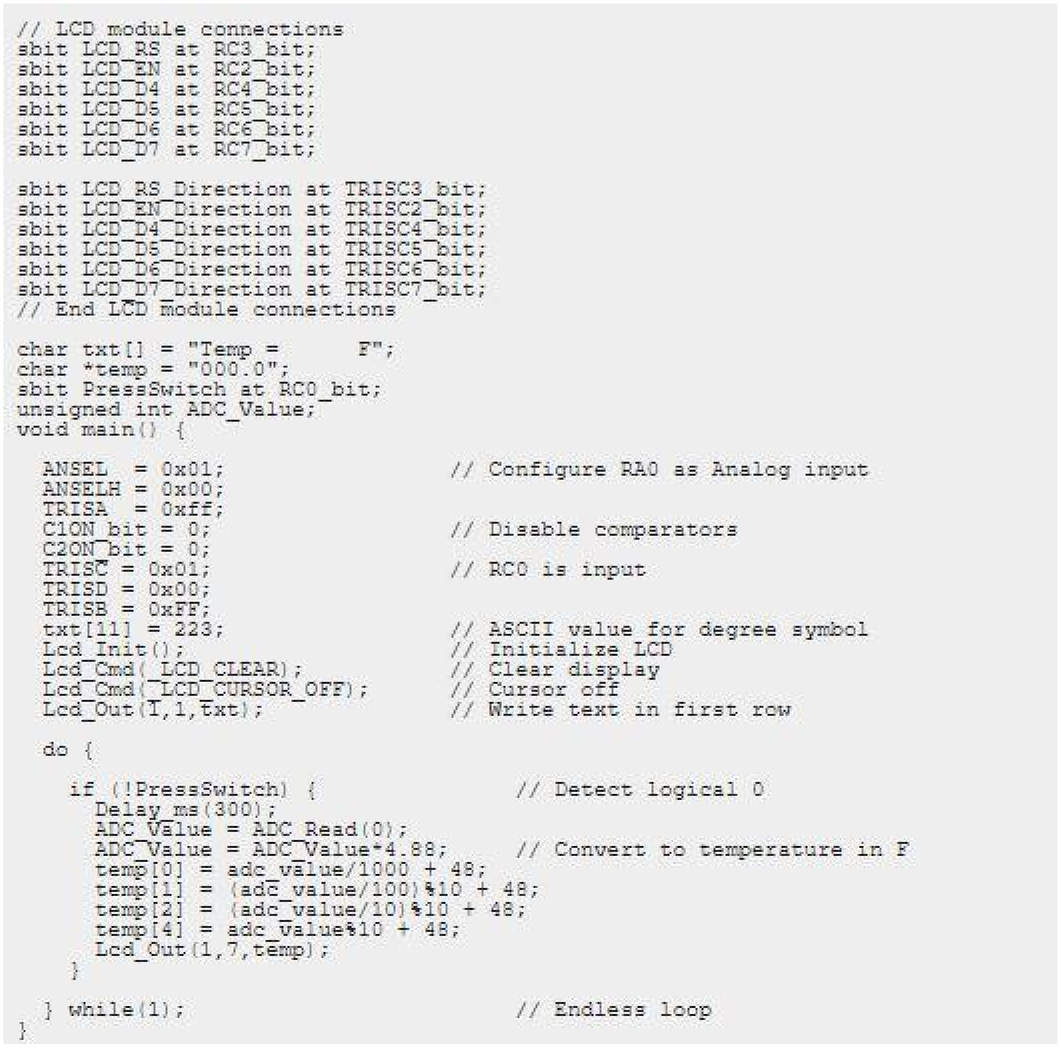
The PICFlash with mikroICD is a programmer and in-circuit debugger for PIC12, PIC16, and PIC18 series microcontrollers.

The first step of using the mikroICD is writing an application program in the chosen compiler, say mikroC Pro for PIC. The compiler's IDE provides debugging functions such as running the program step by step, pausing the program execution using breakpoints, examining the state of the internal registers, tracking the values of variables in the program, etc.

First of all create a new project with the following source code to read the LM34DZ output and display the measured temperature on the LCD screen.

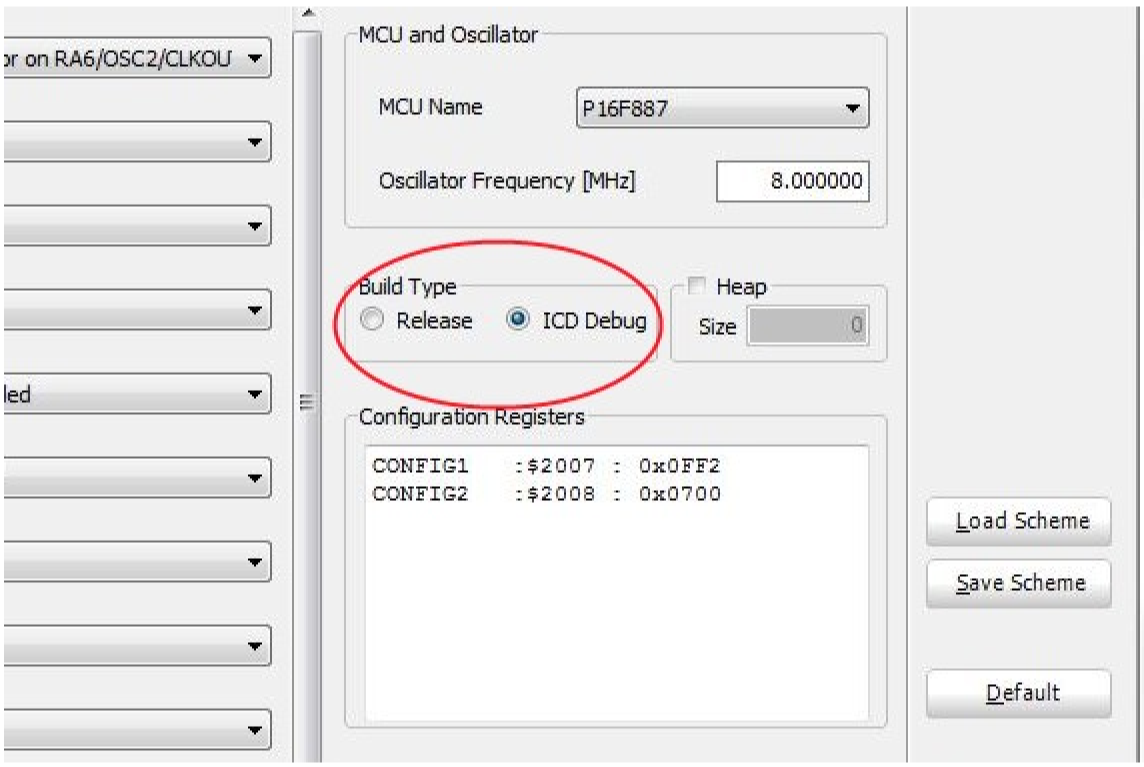
For example:

```
/*

 * Project name:

     Testing mikroICD debugging tool with PIC16F887
 * Copyright:

     (c) Rajendra Bhatt

 * Test configuration:

     MCU:             PIC16F887

     Oscillator:      HS, 10.0000 MHz

*/



// LCD module connections

sbit LCD_RS at RC3_bit;

sbit LCD_EN at RC2_bit;

sbit LCD_D4 at RC4_bit;

sbit LCD_D5 at RC5_bit;

sbit LCD_D6 at RC6_bit;

sbit LCD_D7 at RC7_bit;



sbit LCD_RS_Direction at TRISC3_bit;

sbit LCD_EN_Direction at TRISC2_bit;
sbit LCD_D4_Direction at TRISC4_bit;

sbit LCD_D5_Direction at TRISC5_bit;

sbit LCD_D6_Direction at TRISC6_bit;

sbit LCD_D7_Direction at TRISC7_bit;

// End LCD module connections



char txt[] = "Temp =      F";

char *temp = "000.0";

sbit PressSwitch at RC0_bit;

unsigned int ADC_Value;

void main() {



  ANSEL  = 0x01;                 // Configure RA0 as Analog input

  ANSELH = 0x00;

  TRISA  = 0xff;

  C1ON_bit = 0;                  // Disable comparators

  C2ON_bit = 0;
  TRISC = 0x01;                    // RC0 is input

  TRISD = 0x00;

  TRISB = 0xFF;

  txt[11] = 223;                   // ASCII value for degree symbol

  Lcd_Init();                      // Initialize LCD

  Lcd_Cmd(_LCD_CLEAR);             // Clear display

  Lcd_Cmd(_LCD_CURSOR_OFF);        // Cursor off

  Lcd_Out(1,1,txt);                // Write text in first row


  do {



    if (!PressSwitch) {                // Detect logical 0

      Delay_ms(300);

      ADC_Value = ADC_Read(0);

      ADC_Value = ADC_Value*4.88;      // Convert to temperature in F

      temp[0] = adc_value/1000 + 48;

      temp[1] = (adc_value/100)%10 + 48;
      temp[2] = (adc_value/10)%10 + 48;

      temp[4] = adc_value%10 + 48;

      Lcd_Out(1,7,temp);

    }



  } while(1);                        // Endless loop

}
```



```
// LCD module connections
sbit LCD_RS at RC3_bit;
sbit LCD_EN at RC2_bit;
sbit LCD_D4 at RC4_bit;
sbit LCD_D5 at RC5_bit;
sbit LCD_D6 at RC6_bit;
sbit LCD_D7 at RC7_bit;

sbit LCD_RS_Direction at TRISC3_bit;
sbit LCD_EN_Direction at TRISC2_bit;
sbit LCD_D4_Direction at TRISC4_bit;
sbit LCD_D5_Direction at TRISC5_bit;
sbit LCD_D6_Direction at TRISC6_bit;
sbit LCD_D7_Direction at TRISC7_bit;
// End LCD module connections

char txt[] = "Temp =      F";
char *temp = "000.0";
sbit PressSwitch at RC0_bit;
unsigned int ADC_Value;
void main() {

  ANSEL  = 0x01;                 // Configure RA0 as Analog input
  ANSELH = 0x00;
  TRISA  = 0xff;
  C1ON_bit = 0;                  // Disable comparators
  C2ON_bit = 0;
  TRISC = 0x01;                  // RC0 is input
  TRISD = 0x00;
  TRISB = 0xFF;
  txt[11] = 223;                 // ASCII value for degree symbol
  Lcd_Init();                    // Initialize LCD
  Lcd_Cmd(_LCD_CLEAR);           // Clear display
  Lcd_Cmd(_LCD_CURSOR_OFF);      // Cursor off
  Lcd_Out(1,1,txt);              // Write text in first row

  do {

    if (!PressSwitch) {                // Detect logical 0
      Delay_ms(300);
      ADC_Value = ADC_Read(0);
      ADC_Value = ADC_Value*4.88;      // Convert to temperature in F
      temp[0] = adc_value/1000 + 48;
      temp[1] = (adc_value/100)%10 + 48;
      temp[2] = (adc_value/10)%10 + 48;
      temp[4] = adc_value%10 + 48;
      Lcd_Out(1,7,temp);
    }

  } while(1);                          // Endless loop
}
```

To start debugging, select the *Start Debugger* option from the *Run* drop-down menu or use the function key [F9].


r on RA6/OSC2/CLKOU

led

MCU and Oscillator

MCU Name: P16F887

Oscillator Frequency [MHz]: 8.000000

Build Type: Release [✓ ICD Debug]

Heap

Size: 0

Configuration Registers

```
CONFIG1    :$2007 : 0x0FF2
CONFIG2    :$2008 : 0x0700
```

Function Key	Description
[F9]	Start up debugger
[F6]	Run or pause debugger
[Ctrl+F2]	Stop debugger
[F7]	**Step in:** Execute the current program line, then halts. If the program line executed calls another routine, the debugger steps into the subroutine and halts after executing the first instruction within it.
[F8]	**Step over:** Execute the current program line, then halts. If the program line executed calls another routine, the debugger will not step into it; instead the whole subroutine will be executed and the debugger halts at the first instruction following the call.
[Ctrl+F8]	**Step out:** Execute all remaining program lines within the subroutine. The debugger halts immediately upon exiting the subroutine.
[F4]	Run to cursor: Execute the program until reaching the cursor position.
[F5]	Toggle breakpoint: It allows you to turn a breakpoint on or off at the current cursor position in the program.
[Shift+F4]	Show/Hide breakpoints: To view all the breakpoints in the program, use this option.
[Ctrl+Shift+F5]	Clear all breakpoints