IoT Skill Approach

BY- Er. Ishwar Rathod.
SCSIT, SUAS, Indore.

Reading Temperature from Resistance Sensor

How an analog sensor and a sensing resistor with associated Electronics enables the sensing & measuring of temperature.

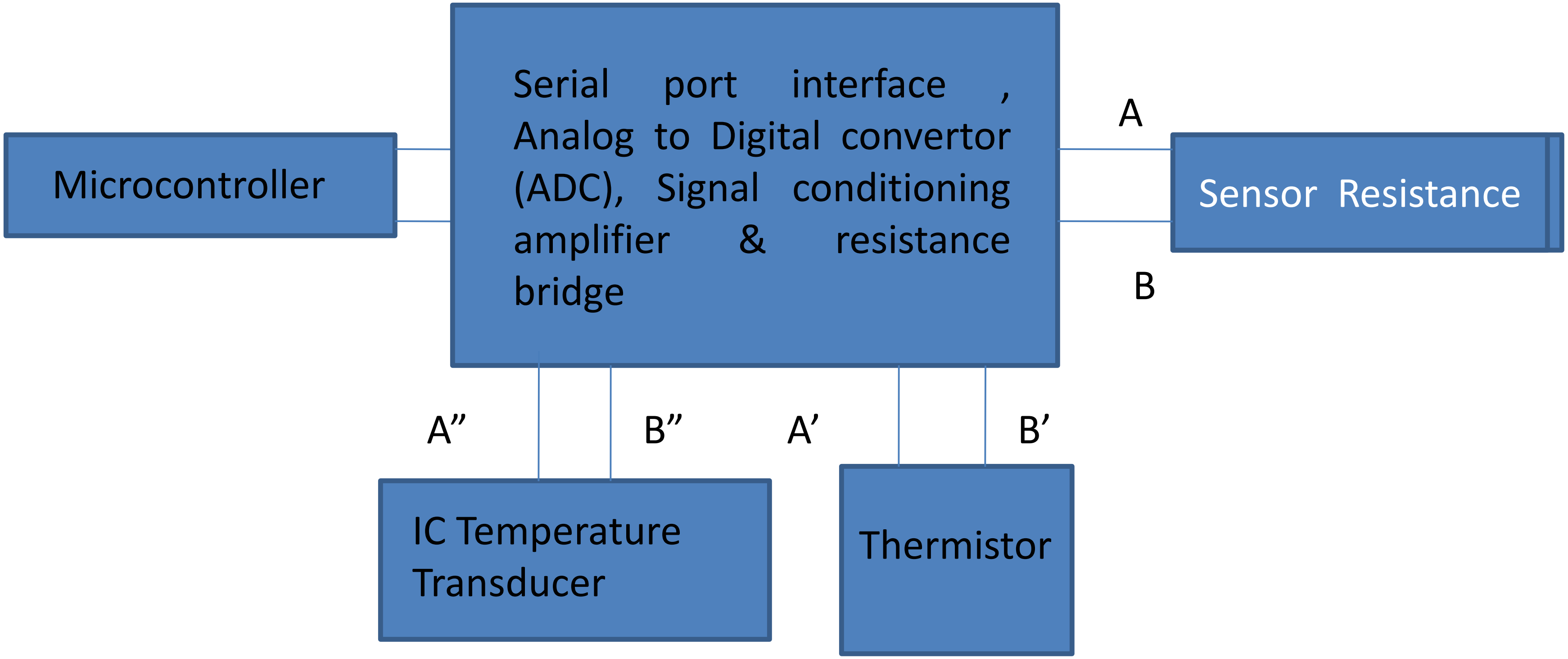
- **Problem:** Assume an electronic circuit with a sensing component resistance as a physical object. How are temperature sensed?
- **Solution:** A resistor in the form of a wire or a component can be part of the electronic circuit. Ohm's law states that resistance remains constants only as long as physical conditions remain same. The resistor function as a sensor when its value changes measurably within the required temperature range for sensing.

Reading Temperature from Resistance Sensor

- The measurements can be first made using two standard or reference temperature points, such as 0°C & 100°C.
- An equation or table can be prepared for the sensing component resistance R values as a function of the temperature T in °C.
- When changes are linearly related to the change in the physical environment then the equation is used. When changes are nonlinearly or exponentially related to the change in the physical environment, then use of table is preferred.
- When temperature, such as of oil or coolant plate in an M2M or IoT device in an automobile needs to be sensed then a simple electronic circuit uses the sensing component at the sensing object.
- The associate computing device calculates the temperature value at the measuring instance.

Below figure shows a circuit consisting of a resistance bridge.

The bridge has one sensing resistor at the sensing object & three fixed (standard) resistors. The figure shows a microcontroller using an electronics circuit with a port connected to sub-circuit s, serial port interface, ADC, Signal conditioning amplifier & resistance bridge.



Reading Temperature from Resistance Sensor

- **Microcontroller serial port connected to sub-circuit-serial port interface, ADC, Signal conditioning amplifier, resistance bridge & sensor resistance outputs A ad B. Alternatively circuit connects to a thermistor output at A’ & B’ or IC based temperature transducer output at A” & B”**
- A transducer induces current or voltage. The output changes as per a change in the physical energy at input. An IC based circuit for a temperature transducer induces current in the output according to the heat energy represented by the temperature.
- Microcontroller is a computing device which reads the input at its ports, saves the reading in memory & then the reading is used for computations & communication.

Note: on the basis of above understanding & referring the IoT BOOK of Dr. Rajkal students have to prepare & add in assignment the below.

1. Reading from capacitive sensor.
2. Reading from diode sensor.
3. Reading from transistor-based sensor.
4. Examples of sensors.

1. Home Automation System Using a Simple Android App

- Nowadays, people have smartphones with them all the time. So it makes sense to use these to control home appliances. Presented here is a home automation system using a simple Android app, which you can use to control electrical appliances with clicks or voice commands? Commands are sent via Bluetooth to Arduino Uno. So you need not get up to switch on or switch off the device while watching a movie or doing some work.

Home automation: circuit and working

- The home automation circuit is built around an Arduino Uno board, Bluetooth module HC-05 and a 3-channel relay board. The number of channels depends on the number of appliances you wish to control. Arduino Uno is powered with a 12V DC adaptor/power source. The relay module and Bluetooth module can be, in turn, powered using a board power supply of Arduino Uno. Author’s prototype is shown in Fig. 1. Connection details for each appliance are shown in Fig. 2.

Fig. 1: Author's prototype

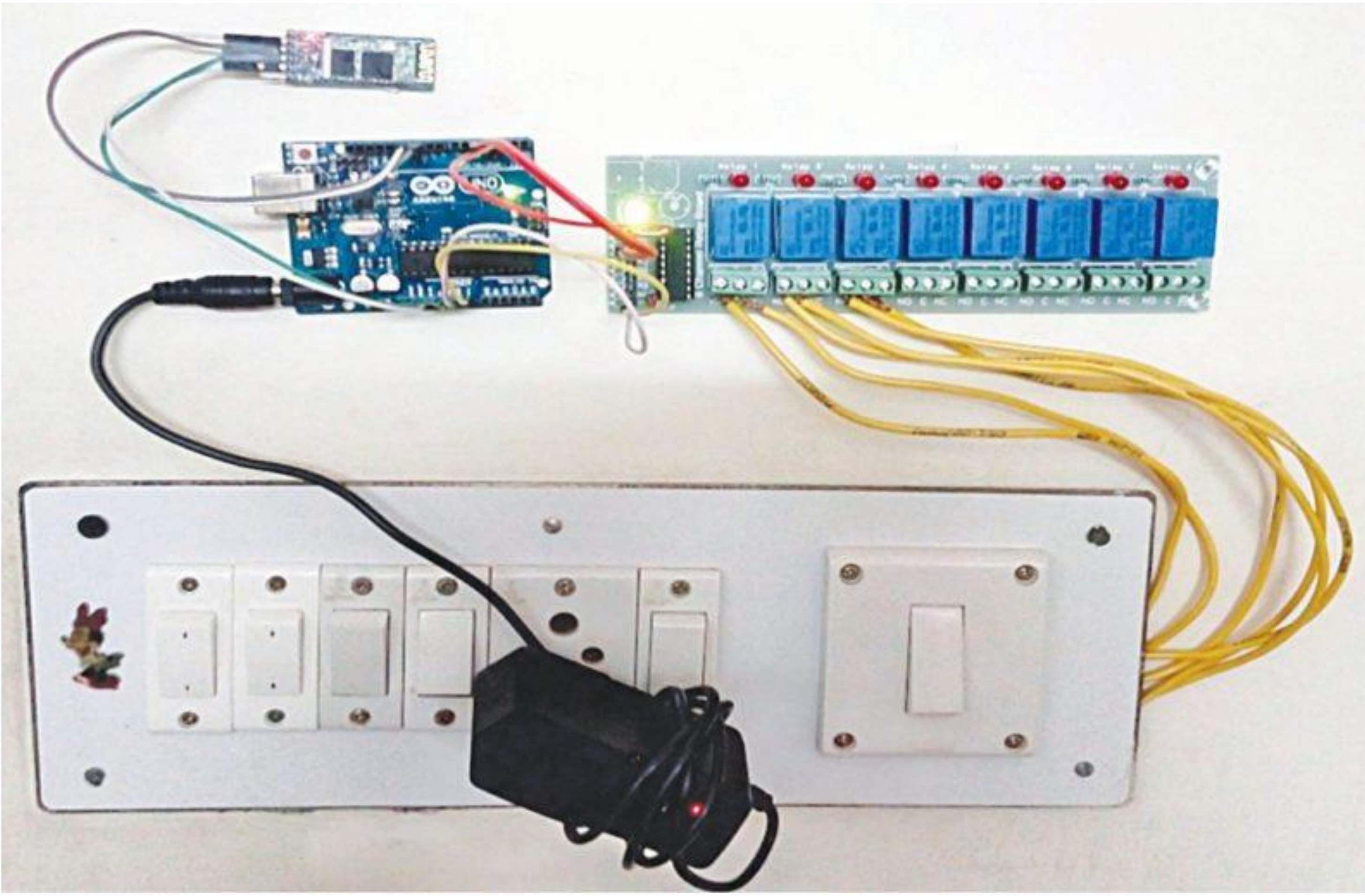
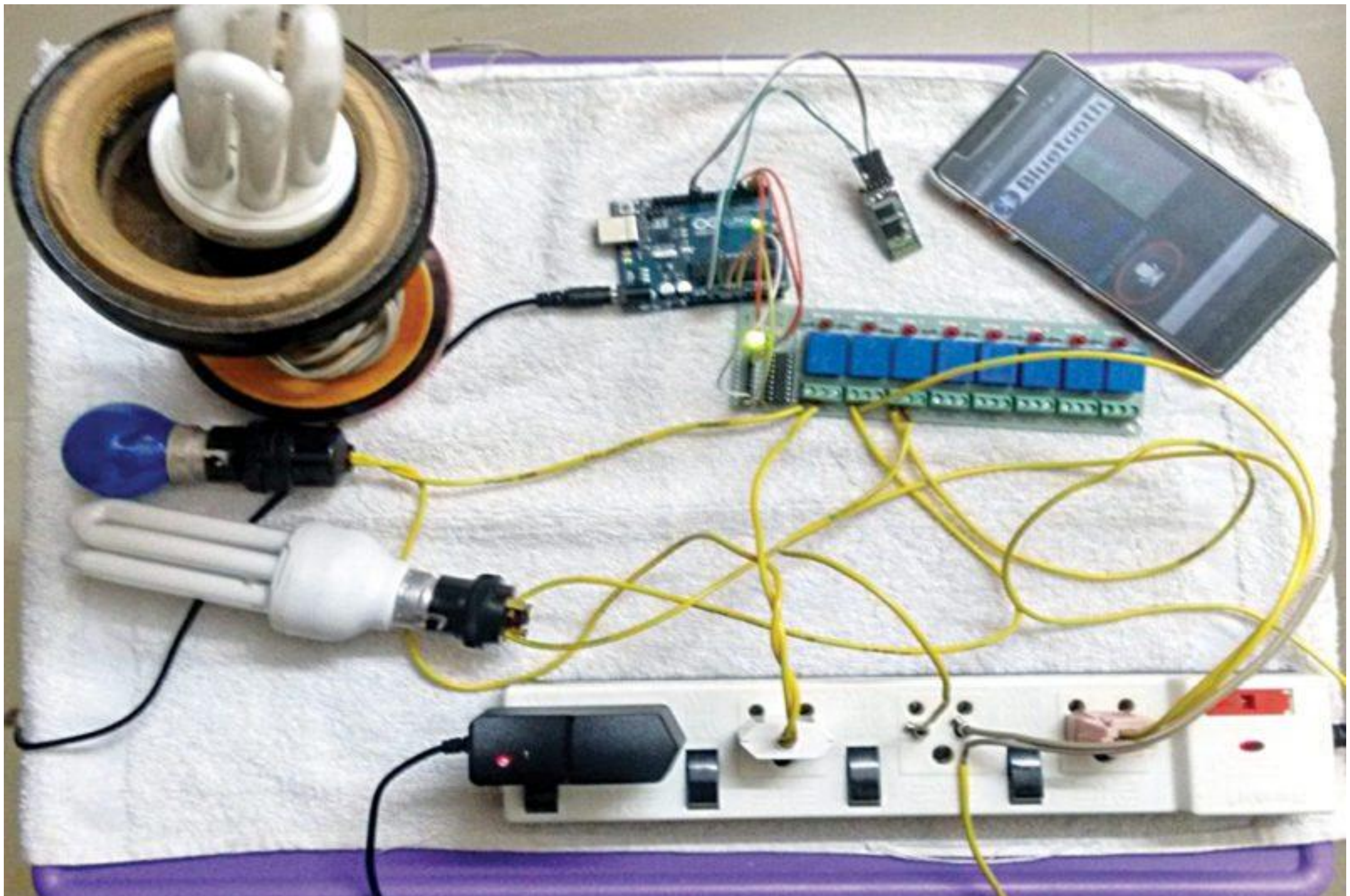


Fig. 2: Connections for the appliances



Bluetooth module &Relay module

- Bluetooth module used in this project is HC-05 (Fig. 4), which supports master and slave mode serial communication (9600-115200 bps) SPP and UART interface. Using these features it can communicate with other Bluetooth-enabled devices like mobile phones, tablets and laptops. The module runs on 3.3V to 5V power supply.
- A relay allows you to turn on or turn off a circuit using voltage and/or current much higher than what Arduino could handle. Relay provides complete isolation between the low-voltage circuit on Arduino side and the high-voltage side controlling the load. It gets activated using 5V from Arduino, which, in turn, controls electrical appliances like fans, lights and air-conditioners. An 8-channel relay module is shown in Fig. 5.

Arduino Uno board

- Arduino is an open source electronics prototyping platform based on flexible, easy-to-use hardware and software. It is intended for artists, designers, hobbyists and anyone interested in creating interactive objects or environments.
- Arduino Uno is based on ATmega328 microcontroller (MCU). It consists of 14 digital input/output pins, six analogue inputs, a USB connection for programming the on board MCU, a power jack, an ICSP header and a reset button. It is operated with a 16MHz crystal oscillator and contains everything needed to support the MCU. It is very easy to use as you simply need to connect it to a computer using a USB cable, or power it with an AC-to-DC adapter or battery to get started. The MCU on board is programmed in Arduino programming language using Arduino IDE.
- In this home automation project circuit, Pins 10 and 11 of Arduino are connected to pins T_{XD} and R_{XD} of the Bluetooth module, respectively, as shown in Fig. 6.
- Pins Gnd and Vcc of the Bluetooth module are connected to Gnd and +3.3V of Arduino board respectively. Pins 2, 3 and 4 are connected to the three relays (RL1, RL2 and RL3) of the relay board. Pins Vin and Gnd of the relay board are connected to pins Vin and Gnd of Arduino board, respectively.

Fig. 4: Bluetooth module

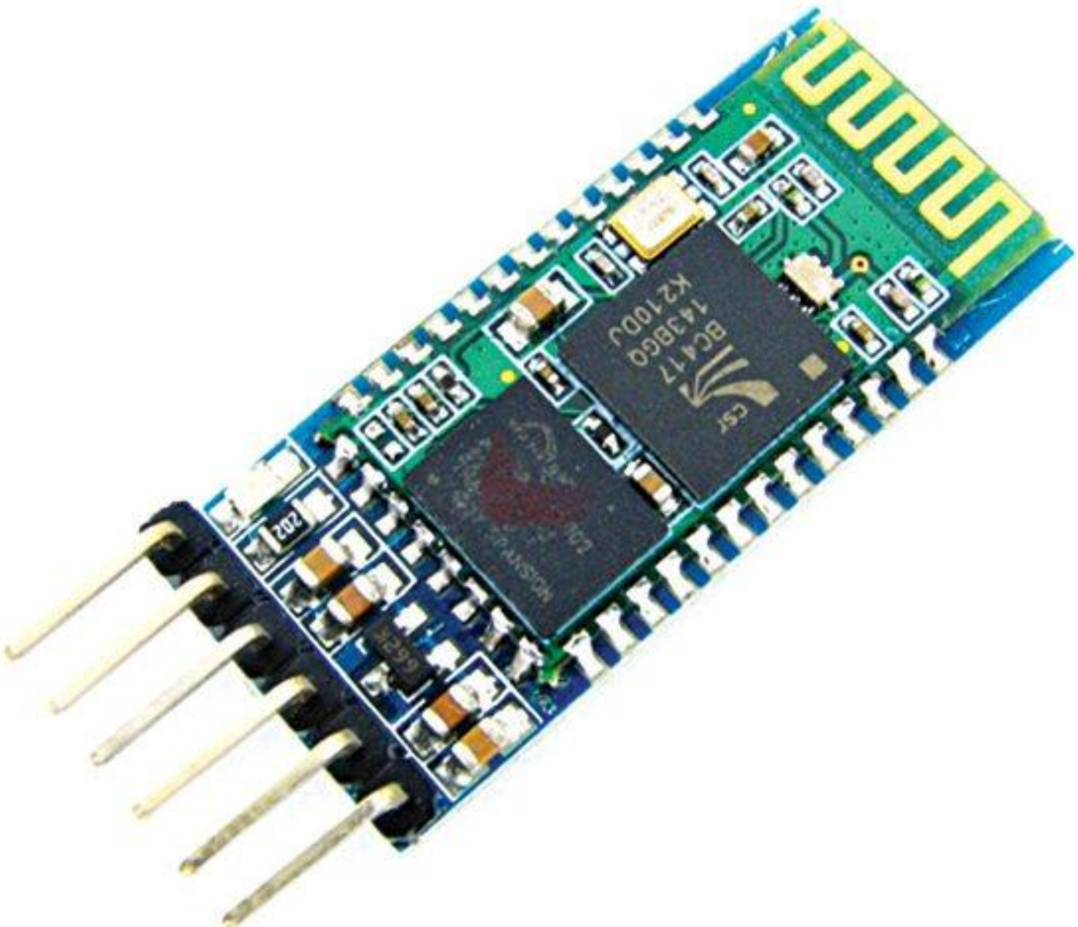


Fig. 5: An 8-channel relay module

- ***Note.*** Vin is usually used to give input power, but since we are supplying 12V to Arduino using an adaptor, we can use Vin pin on Arduino to power the 12V relay module.

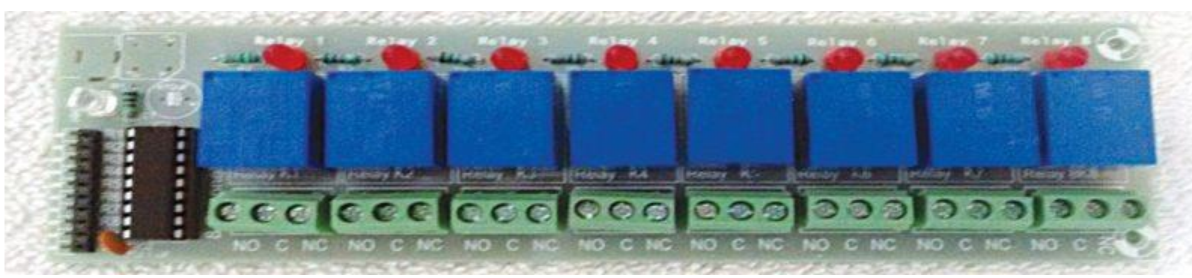
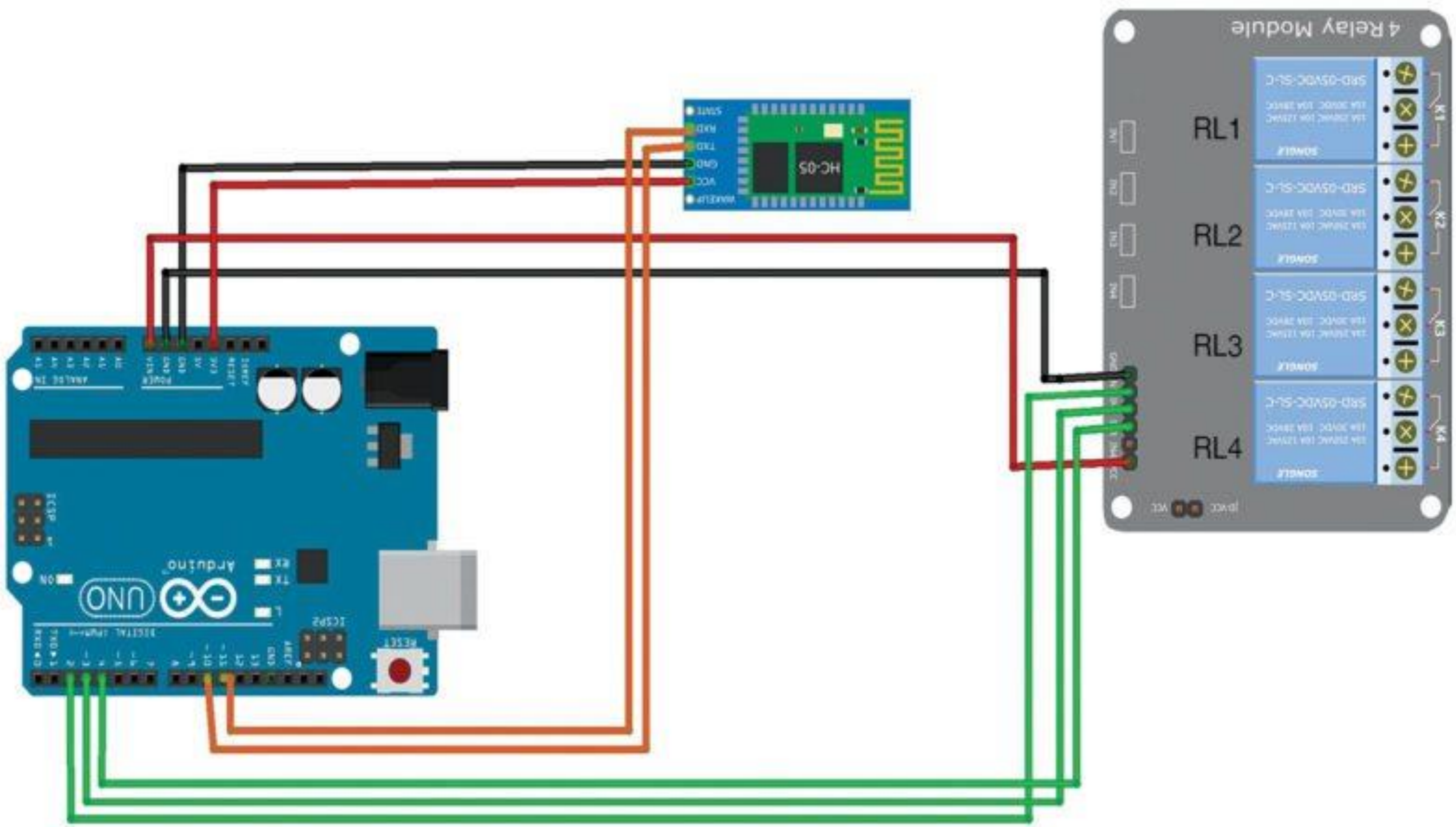


Fig. 6: Relay module connection



2.Assemble yourself a portable, four-channel multi-mode light controller using an Arduino Uno board and very few external components

- The fancy lights controller described here is built around the Arduino (an Open Source single-board microcontroller) platform that can be purchased in pre-assembled hardware form. The circuit is nothing but a portable four-channel, multi-mode digital light controller, realised using very few external components. Four LEDs are made to glow in different sequences and patterns, controlled from the Arduino board (Fig. 1).

Fig. 1: Arduino board

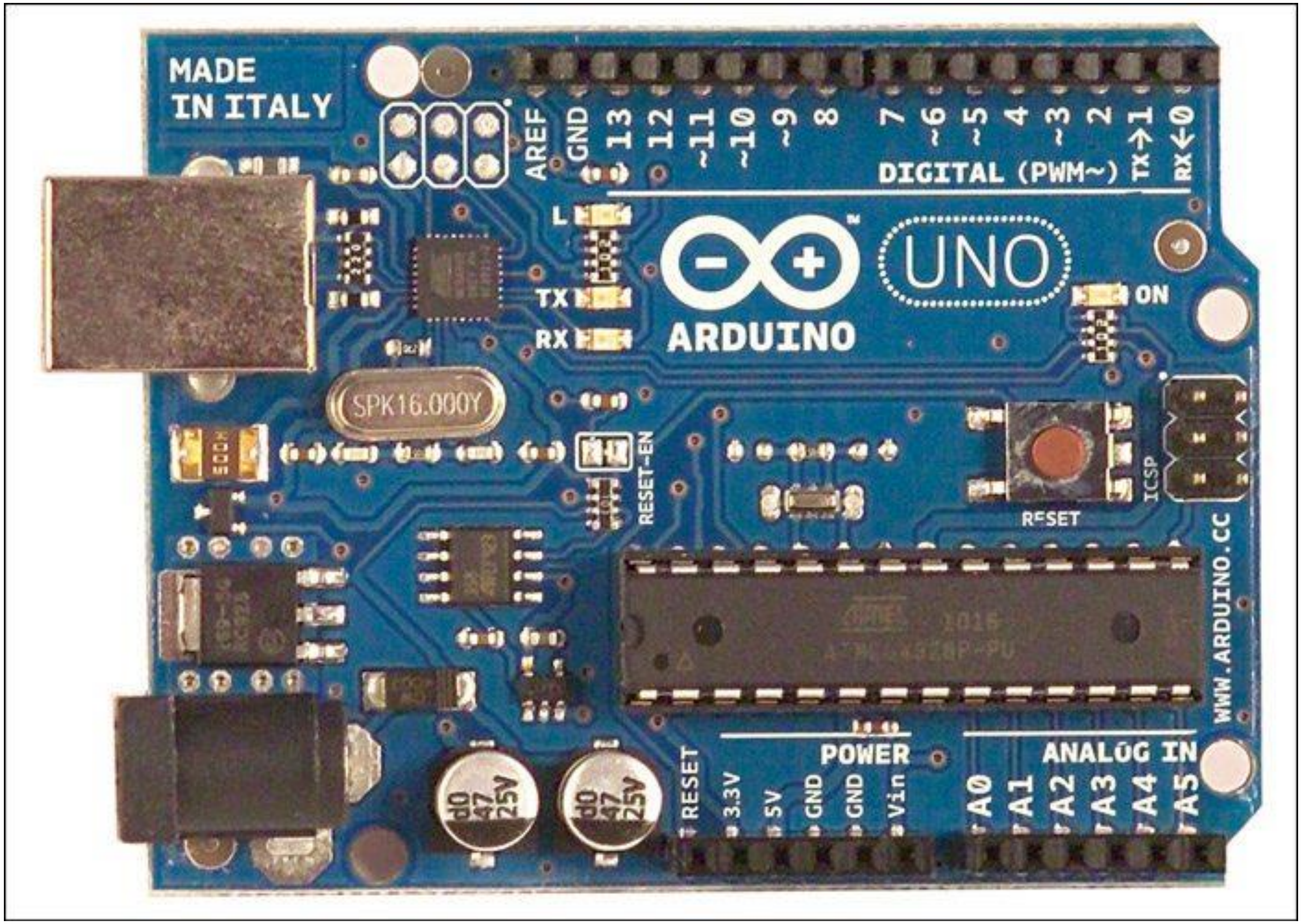
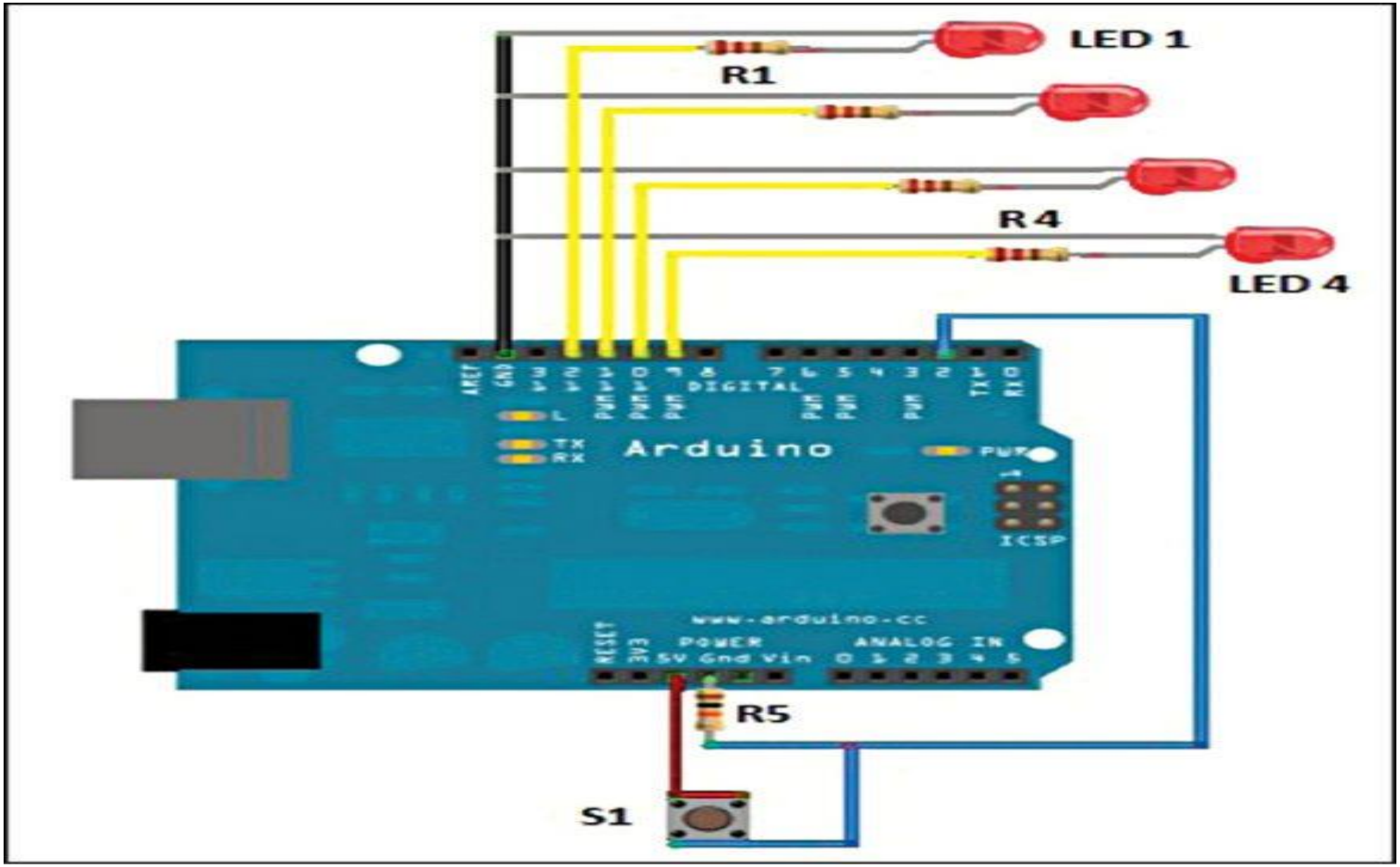


Fig. 2: Fritzing image of fancy lights controller



Fritzing image of fancy lights controller

- For this project, in addition to an Arduino UNO board, you need a USB cable (A to B), PC running the Arduino IDE (0022), components (four LEDs and 220-ohm resistors, a 10-kilo-ohm resistor and a switch),

Fritzing image of fancy lights controller

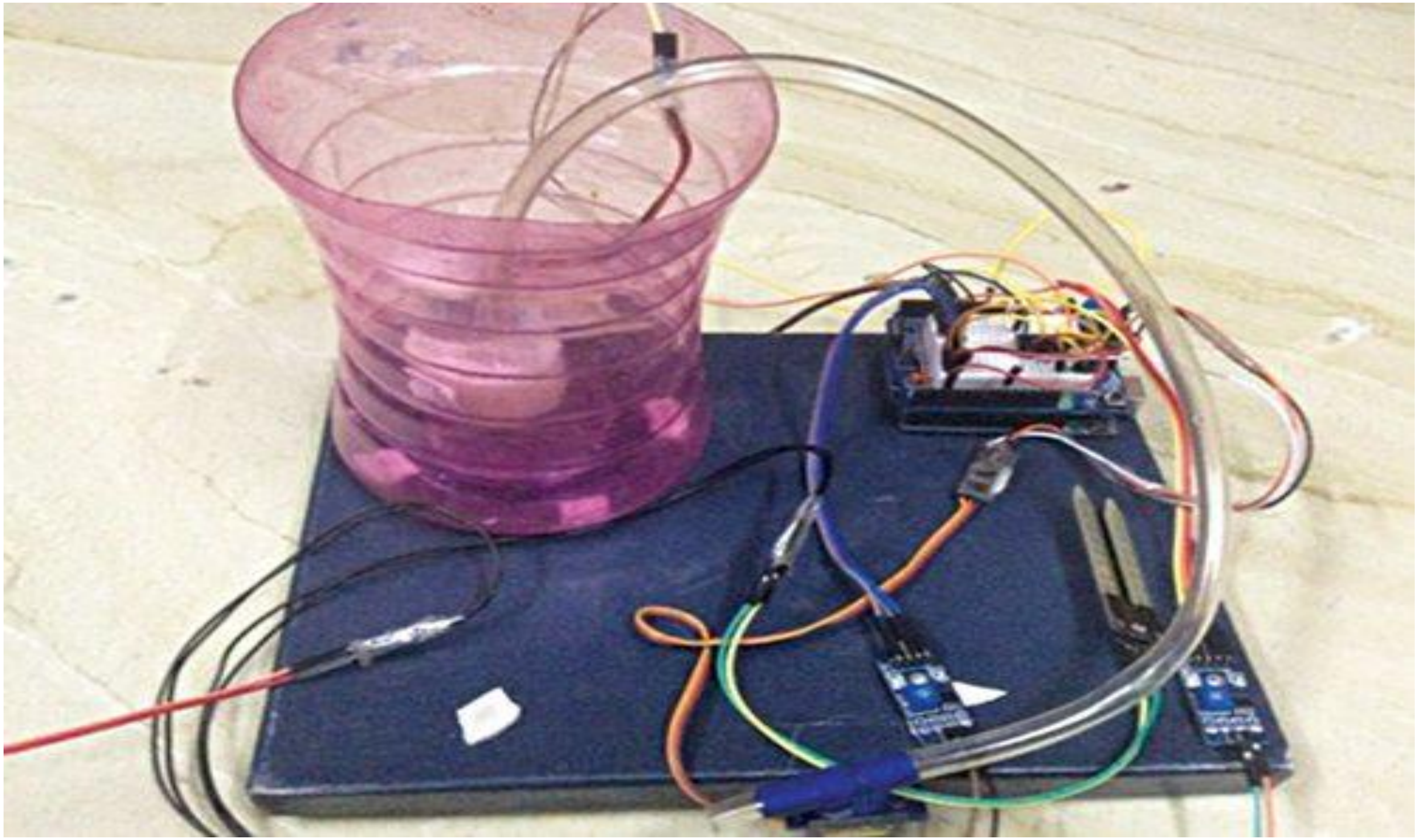
- accessories (wires, connectors and battery), and the code or sketch for the MCU.
- In the prototype, four 5mm red LEDs (LED1 through LED4) are used at the output. However, the same output lines can also be used to control high-voltage incandescent lamp strings with the help of suitable solidstate light switch modules. A push-to-on microswitch (S1) is the one and only output mode selector in the circuit!
- The screenshot of fancy lights controller wiring generated using Open Source Fritzing software is shown in Fig. 2. The table shows the function of switch S1 and corresponding output on LEDs (LED1 through LED4).

Fig. 3: Accessories required for the project

3.AUTOMATED PLANT WATERING SYSTEM

- During summers, most people are too lazy to water the potted plants on their rooftop gardens every day. Explained in this section is a simple and exciting automatic plant watering system that you can build yourself in just a few hours. It is an Arduino based automatic plant watering system that uses a soil moisture sensor. The author's prototype is shown in Fig. 1.

Author's Prototype



Automatic plant watering system circuit and working

- The circuit diagram of the automatic plant watering system is shown in Fig. 2. The circuit comprises an Arduino UNO board, a soil moisture sensor, a servo motor, a 12V water pump and an L293D (IC1) motor driver IC to run the water pump.
- You can power the Arduino board using a 7V to 12V wall wart or plug-in adaptor or solar panel. You need a separate 12V battery or power supply or solar panel for the pump motor.
- **Soil moisture sensor**
- Two types of soil moisture sensors are available in the market—contact and non-contact sensors. A contact soil sensor (as shown in Fig. 3) is used in this project because it has to check soil moisture to measure the electrical conductivity.

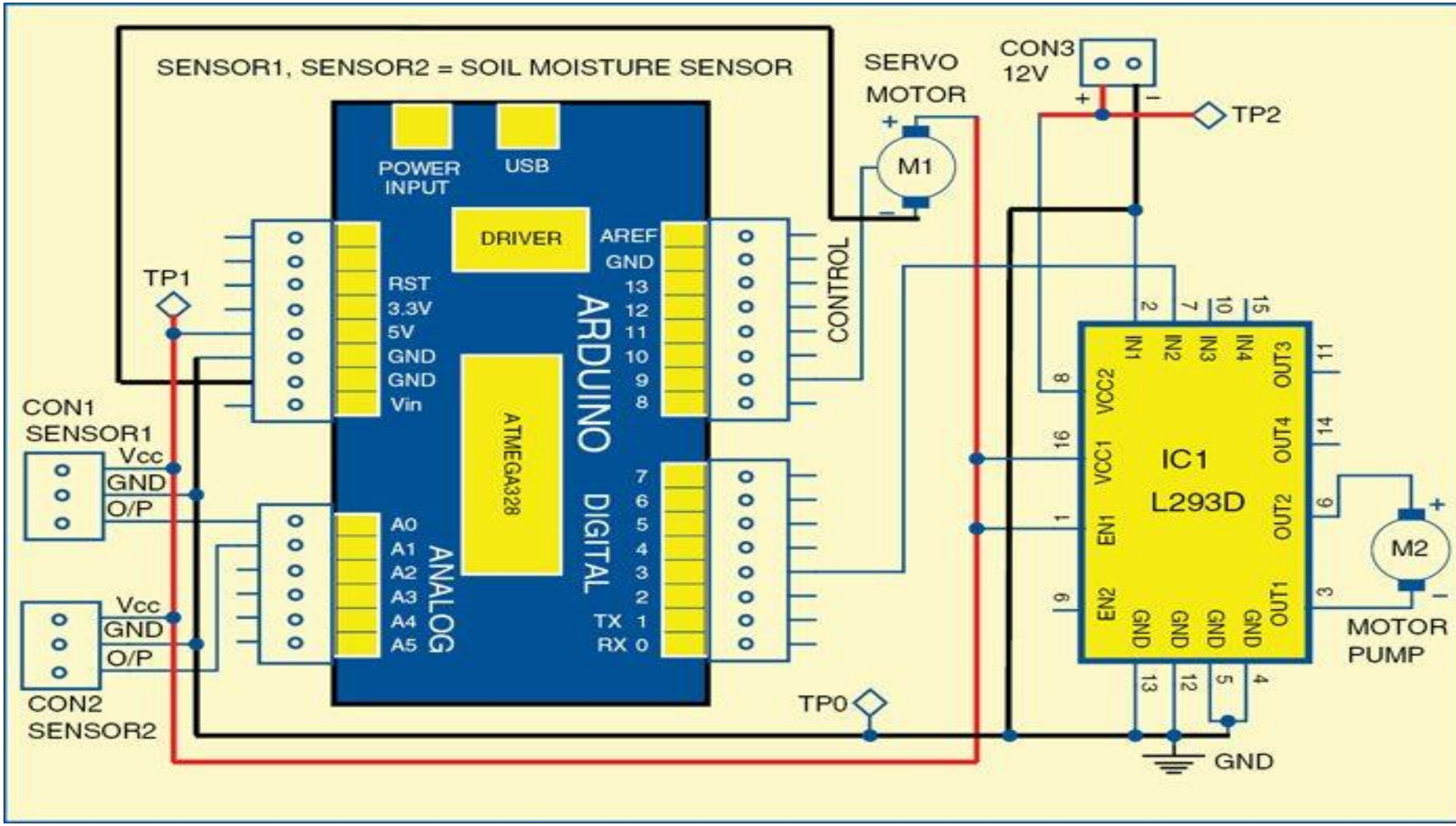
Test Points

Test point	Details
TP0	0V
TP1	5V
TP2	12V

Automatic plant watering system circuit and working

- The moisture sensor provides an analogue output, which can easily be interfaced with Arduino. In this project, two sensors can be connected to analogue pins, A0 and A1, of the Arduino board. Each sensor has four pins (Vcc, Gnd, Ao and Do) available for interfacing with the Arduino board. Here, digital output pin (Do) is not used. The water pump and servo motor are controlled by Arduino connected to digital pins 3 and 9, respectively. That is, the servo motor signal control pin is connected to pin 9 of the Arduino board.
- The program in the Arduino reads the moisture value from the sensor every 20 seconds. If the value reaches the threshold value, the program does the following three things:
- It moves the servo motor horn, along with the water pipe fixed on it, toward potted plant, whose moisture level is less than the predetermined/ threshold level.
- It starts the motor pump to supply water to the plant for a fixed period of time and then stops the water pump (refer Fig. 4).
- It brings back the servo motor horn to its initial position.

Fig. 2: Circuit diagram of the automatic plants watering system



Components

PARTS LIST	
Semiconductor:	
IC1	- L293D motor driver
	- Soil moisture sensor
	- Arduino UNO board
Miscellaneous:	
M1	- Servo motor
M2	- 12V DC motor pump
CON1, CON2	- 3-pin connector
CON3	- 2-pin connector
	- Water container
	- Small flexible water pipe
	- 12V battery
	- 7-12V power adaptor

Software program

- The program is written in Arduino programming language. The code is well commented and is easy to understand. Compile the autowatering.ino code and upload it to the microcontroller, using Arduino IDE version 1.

Fig. 3: Soil moisture sensor (contact type)

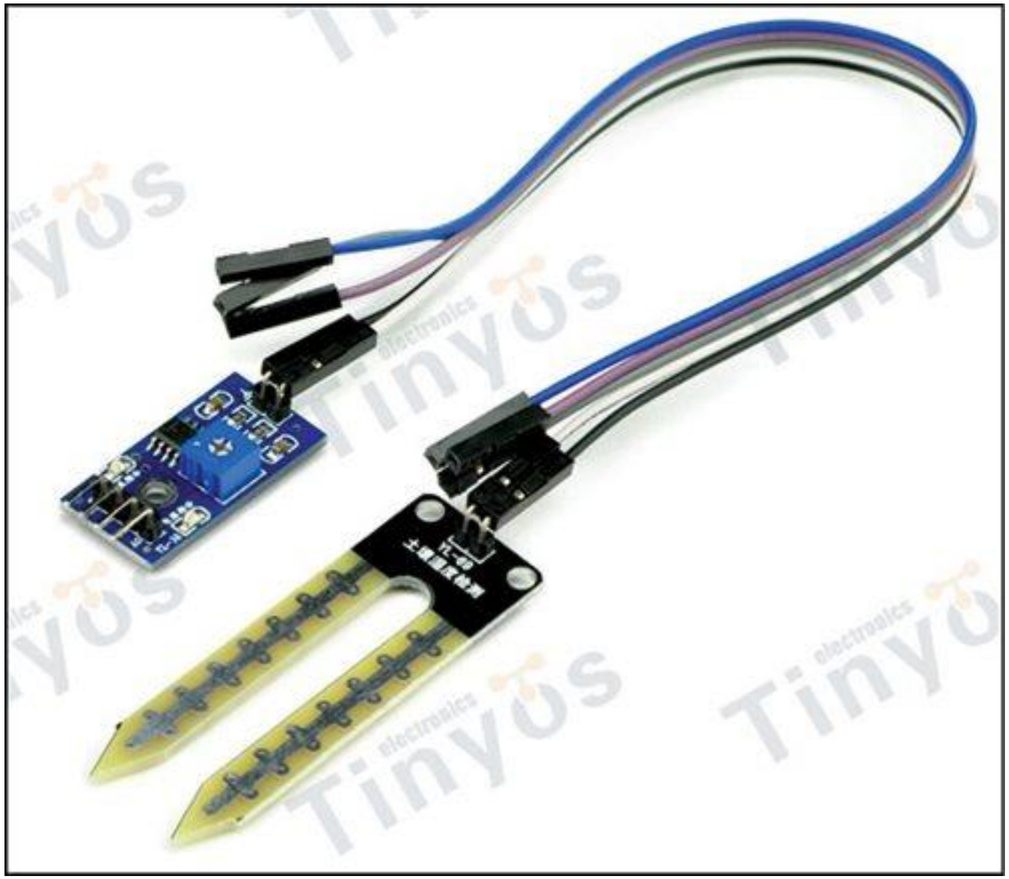
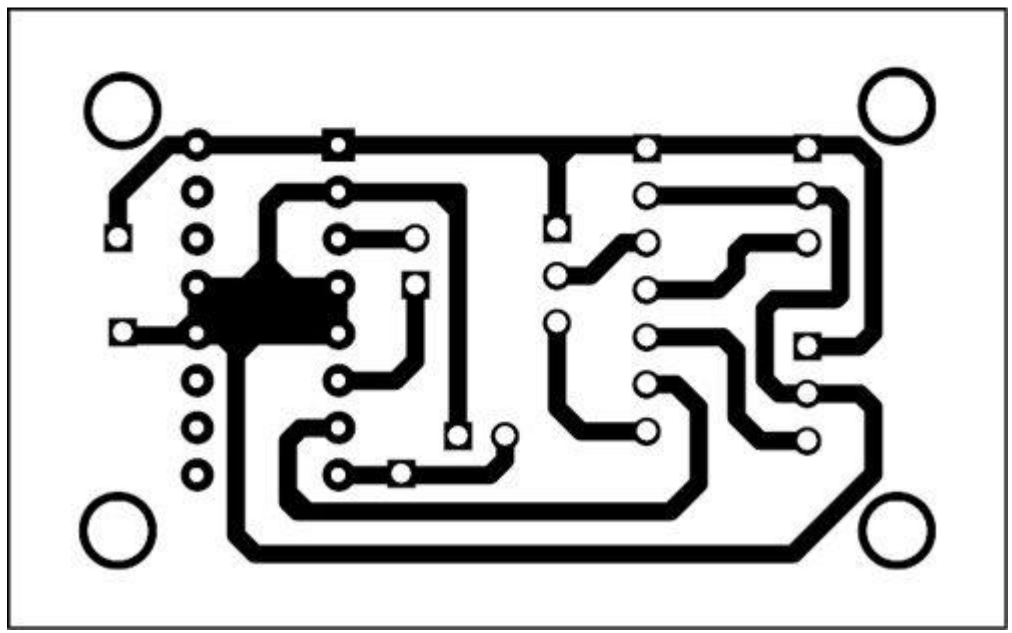


Fig. 4: Motor pump

Fig. 5: Actual-size PCB layout of the circuit



Construction and testing

- The sensor will calibrate by itself once it is kept in the soil and the threshold value will be shown on the serial monitor in Arduino. Serial debugging is available in this program. Comment out if you do not wish to use the serial monitor.
- An actual-size, single-sided PCB layout of the automatic plant watering system is shown in Fig. 5 and its component layout in Fig. 6.
- Assemble the components on the PCB to minimise errors. Alternatively, you can assemble them on a breadboard or Arduino prototyping shield or a general-purpose PCB. Upload the code to Arduino UNO board and install the sensors in the soil of the potted plants. Do not immerse the sensors fully inside the soil.
- Install the pump in a water container (refer Fig. 7) that can hold a few litres of water. attach the water pipe on the servo motor horn as shown in Fig. 8.

Fig. 6: Component layout of the PCB

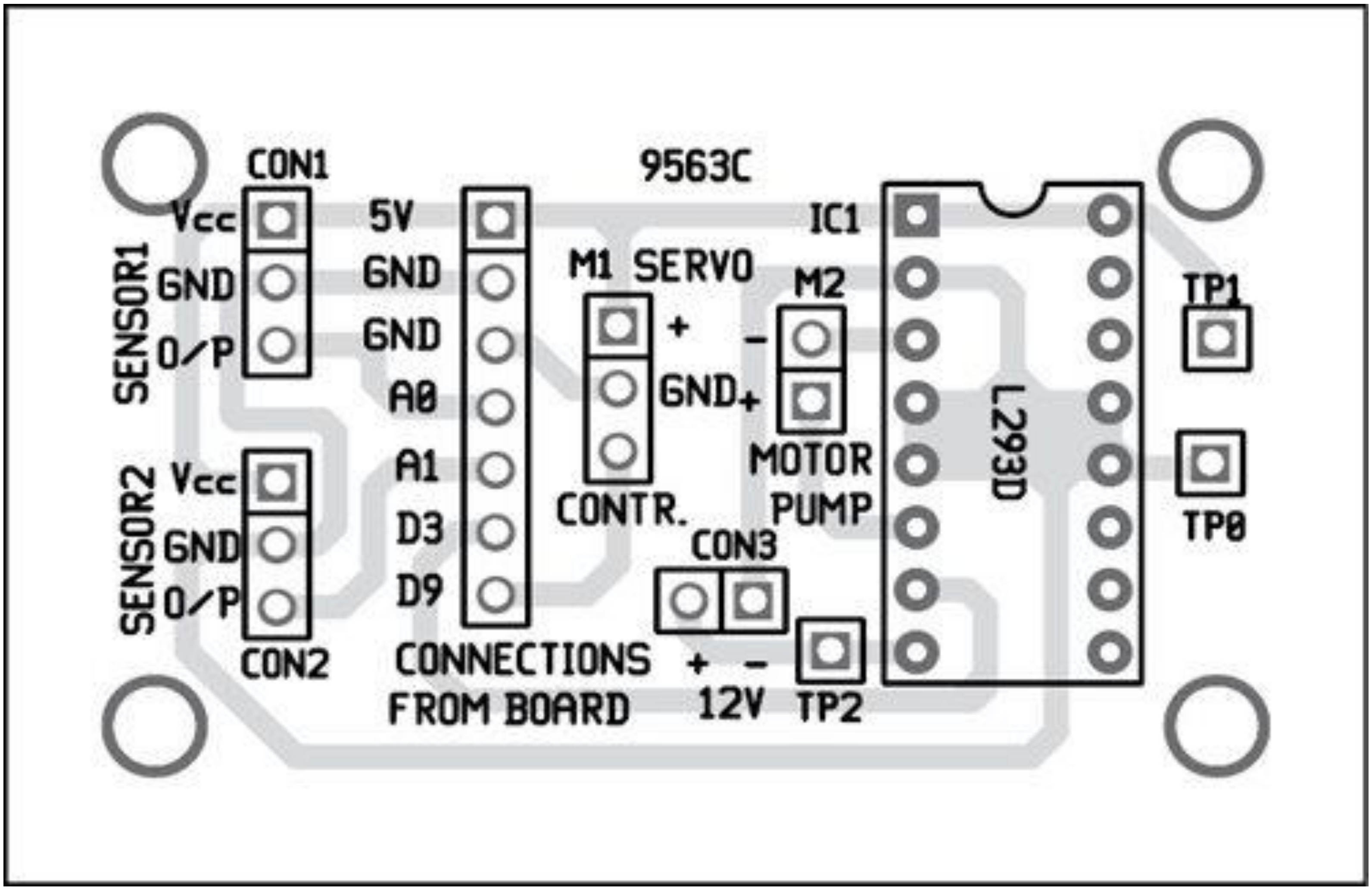


Fig. 7: Installing water pump in the container

Fig. 8: Attaching the pipe on the servo horn

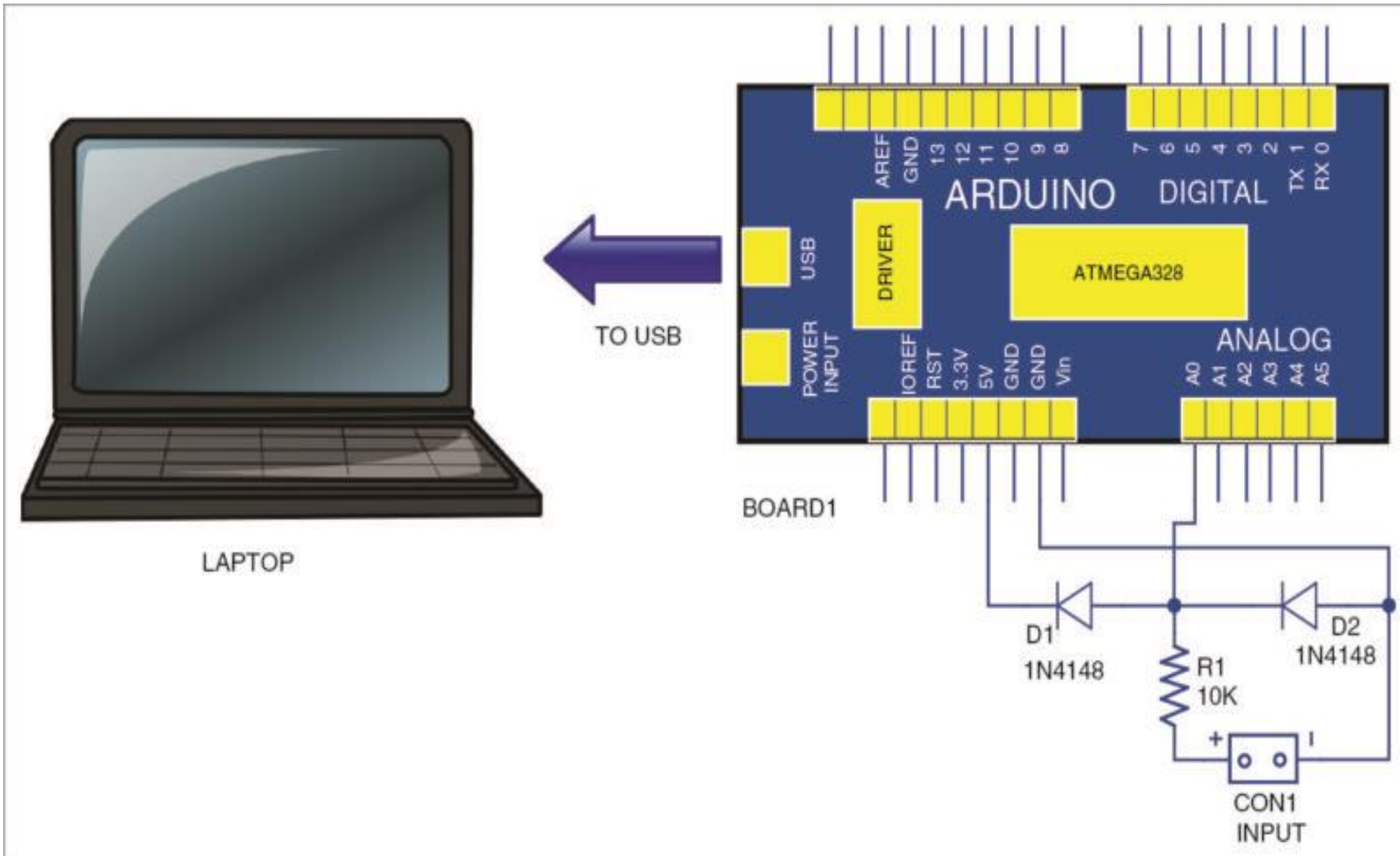
- Before powering the circuit on, you need to keep in mind the following macro definitions in the code:
- Changing the angle of rotation of the servo horn toward the first pot and second pot. The default values are 70 degrees and 145 degrees.
- Changing the watering time according to the size of the pot. The default values are five seconds and eight seconds.
- Changing the threshold value according to your need. The default value is 600.
- Place the flower pots where the pipe from the servo motor horn can easily reach them. When the moisture level dips below 600, servo horn rotates at an angle of 70 degrees. That is, after servo motor horn moves 70 degrees toward the first pot, the motor pump will be on for five seconds and then stop automatically. Then, the servo returns to its original position. Similarly, if you are using a second sensor, the servo motor horn will move to 145 degrees to the second biggest pot, motor pump will be on for eight seconds and then stop automatically. The servo returns to its original position.

- **Further application**
- Using the Arduino UNO board, you can water six different potted plants. By adding a few more lines in the code, you can water even more plants—by using the Arduino Mega 2560 board which has more analogue input pins.
- You can also add an Ethernet or Wi-Fi shield and use the Twitter library, which will tweet from your plants side to send messages like: I need water, the tank is empty, refill the tank, thanks for the water, and so on.
- A 16×2 LCD can be added to indicate moisture levels.
- You can also enable the circuit to refill the tank after a few days, depending on the volume of the tank.

4.PC-based Oscilloscope Using Arduino

- Oscilloscopes are an essential tool for electronics hobbyists and professionals to verify that their designs would work as expected. PC based Oscilloscope score over standalone oscilloscopes due to their compact size, low cost and ability to do offline analysis.
- Here we describe how you can make your own oscilloscope at a very low cost using your PC and an Arduino board as the hardware for signal acquisition. You can use this oscilloscope to capture frequency signals up to 5kHz. The Arduinoboard, the heart of the oscilloscope, reads the values from its inbuilt analogue-to-digital converter (ADC) and pushes these to the PC via USB port. We have provided here an Arduino sketch, which you can compile and load directly to the Arduino. You also need to install an executable file or application in your Windows PC. This application works as the front-end to plot input signals as waveforms on your computer screen.
- The Arduino board consists of Atmel's AVR microcontroller, which can be 8-, 16- or 32-bit based on the type of the board. For this project, you can use any variant of the Arduino as hardware. The AVR microcontroller has an inbuilt ADC. In the project, we use pin A0 to capture the input signal. The captured input signal is fed to UART via UART-USB converter in the Arduino to the PC. A virtual COM port is created by Windows whenever the Arduino connects to the PC. A Windows-based application developed using NI LabWindows opens up the virtual COM port and starts plotting signals visually using Graph libraries.

Fig. 1: Circuit of the PC-based oscilloscope using Arduino



Construction

- The sampling speed of the oscilloscope is limited by the baud rate of the UART. The Arduino sketch is coded to read the ADC using ISR, and the UART baud rate is configured at 115200, which sends data at 85µs intervals. This gives an effective sampling rate of 12kSa/s.

- The PC scope set-up is quite simple and straightforward as shown in Fig. 1. The Arduino board connects to your laptop or PC via the USB cable. Any external power supply for the board is not required as the board is powered by the USB only. Connect switching diodes (D1 and D2) as input protection circuit to pin A0 of the Arduino's ADC. You need Arduino sketch (pcscope.ino) and PC software or executable file (PCScope.exe) in order to use this circuit. Install PCScope.exe program (developed by author) in your Windows PC and open the application. Next, open the Arduino sketch from Arduino IDE and compile the sketch. Connect the Arduino board to the PC and flash the sketch into the microcontroller on the Arduino board.

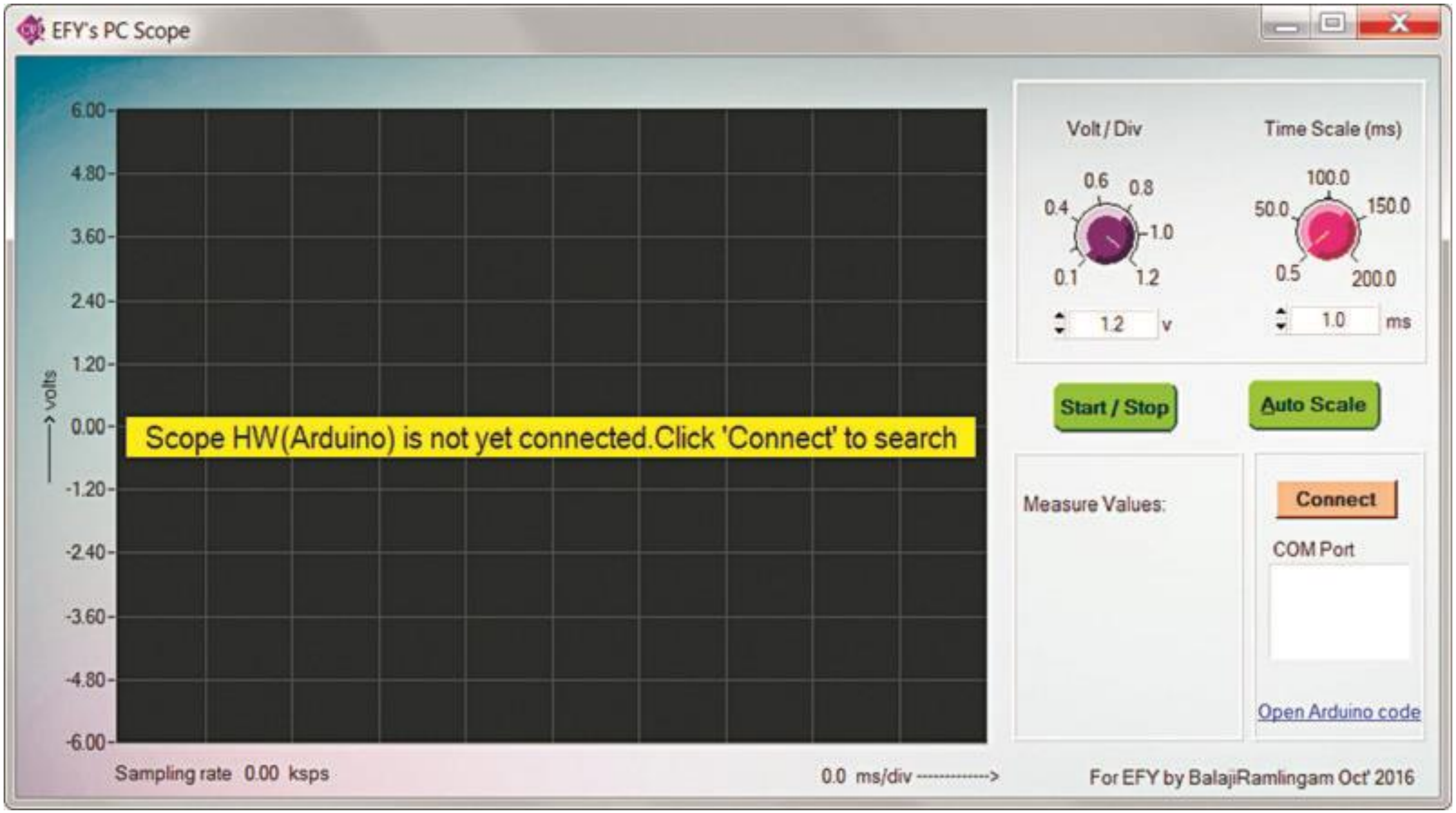
Software

- The ADC of Arduino can measure voltages up to 5V. So it is advisable to add a small protection circuit to limit the input voltage to 5V and clamp the negative voltage. A low-power, fast-switching diode like 1N4148 can be used to protect the input pin. Connect a 10-kilo-ohm resistor in series with the input. It will work as a current limiter in case the input goes beyond 5V. Additional voltage dividers can be used in case you need to measure voltages higher than 5V.
- Arduino sketch. The sampling rate of this PC scope application is limited by the rate at which the data is sent to the PC. Baud rate of 115000 gives time interval of around 85 µs. It is important to get the ADC signals much before this time to get reliable data plotting. The sketch reads pin A0 of Board1 and sends to UART at 115200 baud rate. At this speed, bytes of the input are pushed at time intervals of around 85µs.

Software

- By default, the ADC configuration of the Arduino gives samples every 116µs. So here the ADC is configured with additional lines of code to get samples faster than 85µs by setting the prescaler to 16. With this, you get ADC conversion every 20µs, which is much faster than the UART data transfer rate.
- PC software. As stated earlier, the front-end PC software for signal acquisition and processing is developed using NI LabWindows. The serial port data is captured through Arduino at regular time intervals and plotted as a graph on the screen using the Plot function library. The display points along X-axis are calculated based on the user-defined time scale. The Y-axis range is set using the voltage selection control.

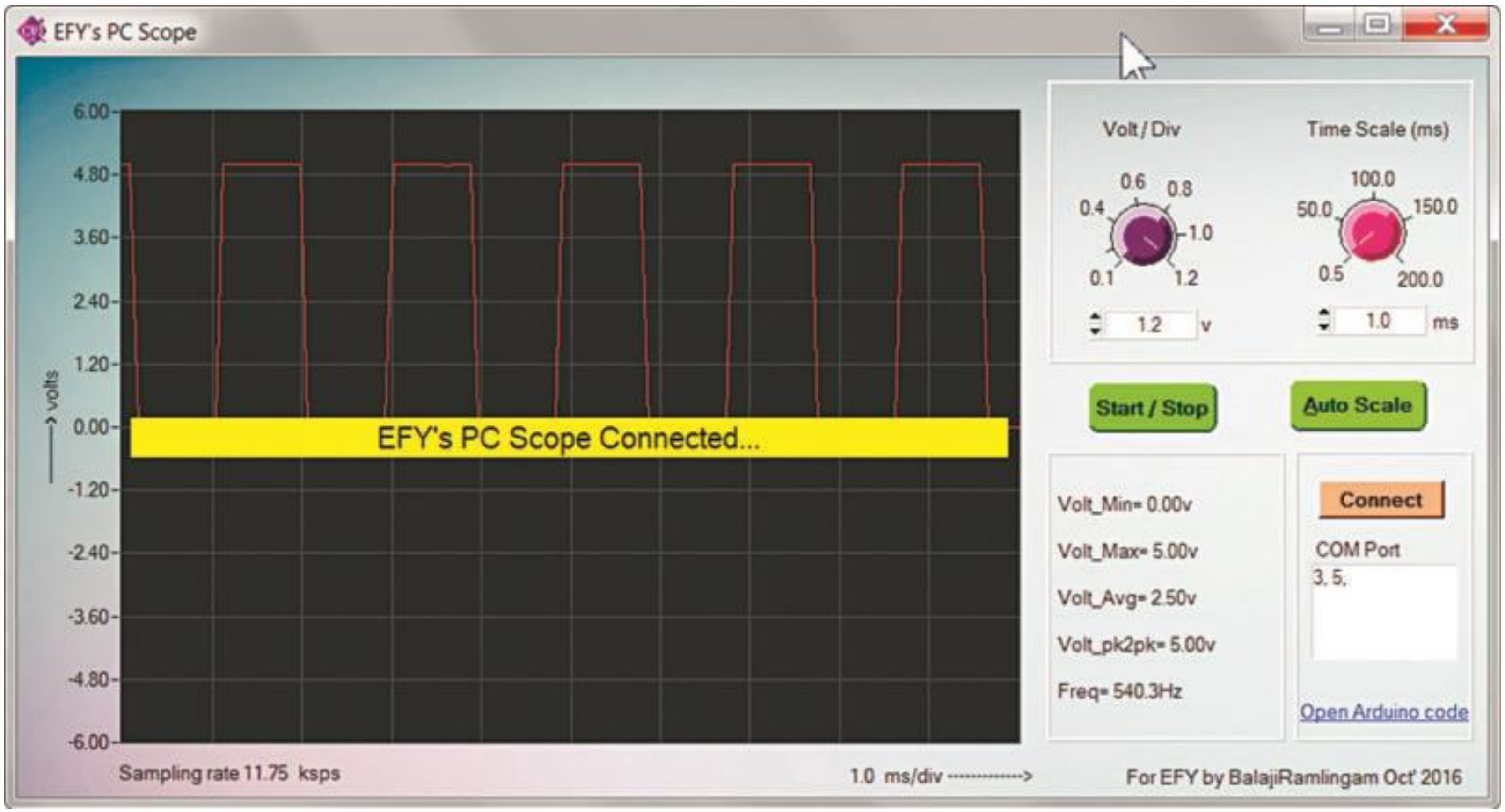
FIG.2Message on the screen when the PC-based scope is run for the first time



Testing

- After installing the PC scope application, click ‘Connect’ button on your PC screen to connect to the Arduino board (Fig. 2). When the board gets connected to your PC, you will get a confirmation message for three seconds as shown in Fig. 3.

FIG.3Message after the hardware successfully connects to the PC



- Feed any square wave input of up to 5kHz at CON1. The software must plot its output waveform on your PC. Square and triangular output waveforms of 525Hz and 530Hz captured on the screen during testing are shown in Figs 4 and 5, respectively. Similarly, you can feed rectangular or pulse inputs (but not sine waves) to get output waveforms.

Fig. 4: Test signal of 525Hz square waveform captured on the screen

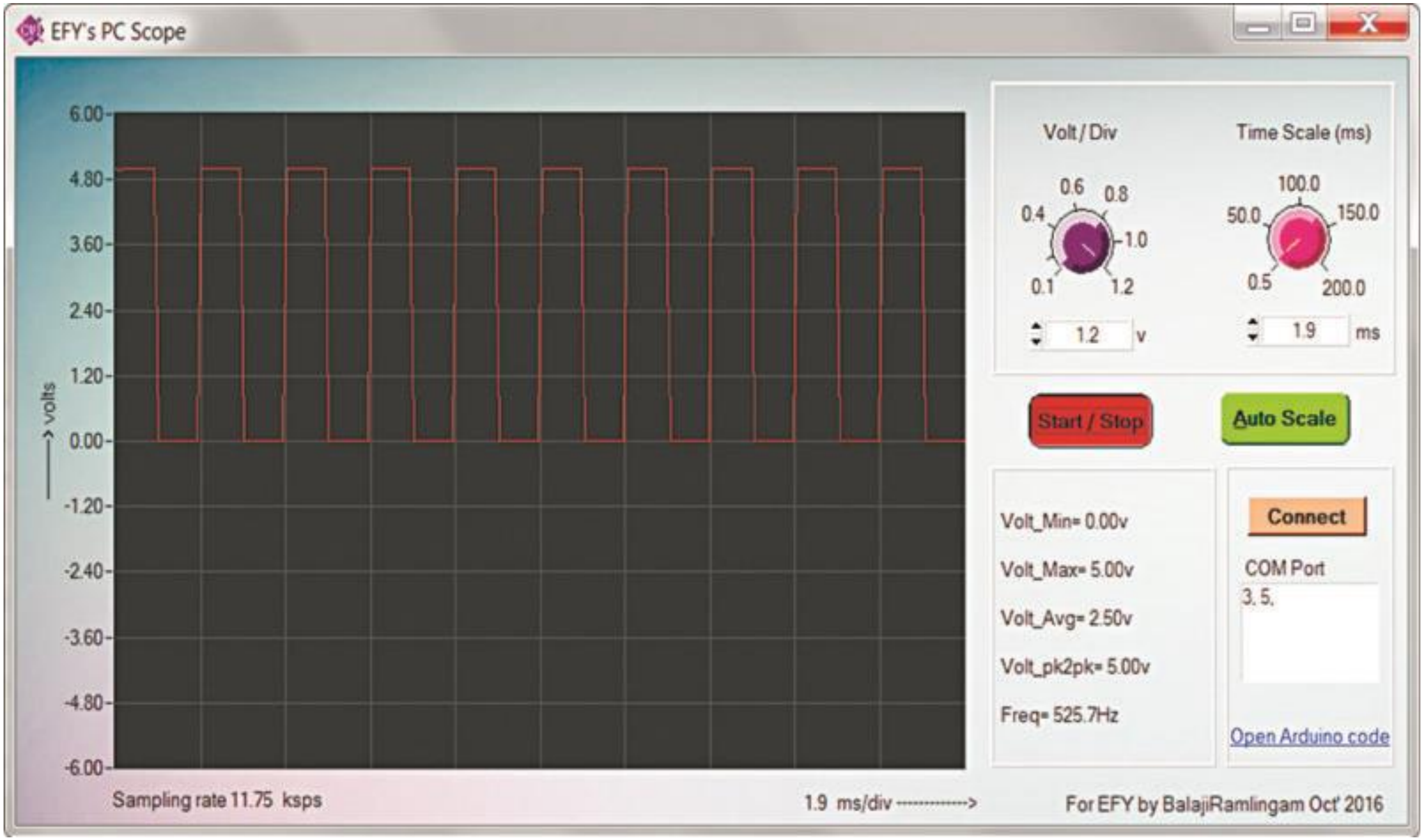


Fig. 5: Test signal of 530Hz triangular waveform captured on the screen