



International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Arduino Based Automatic Plant Watering System

¹S. V. Devika, ²Sk. Khamuruddeen, ³Sk. Khamurunnisa, ⁴Jayanth Thota, ⁵Khalesha Shaik

¹²Associate Professor, Dept. of ECE, HITAM, Hyderabad, India

³MSC 2nd Year, Department Of Electronics, HRD, Hyderabad, India

⁴IV B.Tech, Department of ECE, HITAM, India

⁵IV B.Tech, Department of ECE, ANRH, Hyderabad, India

Abstract: Watering is the most important cultural practice and most labor intensive task in daily greenhouse operation. Watering systems ease the burden of getting water to plants when they need it. Knowing when and how much to water is two important aspects of watering process. To make the gardener works easily, the automatic plant watering system is created. There have a various type using automatic watering system that are by using sprinkler system, tube, nozzles and other. This project uses watering sprinkler system because it can water the plants located in the pots.

This project uses Arduino board, which consists of ATmega328 Microcontroller. It is programmed in such a way that it will sense the moisture level of the plants and supply the water if required. This type of system is often used for general plant care, as part of caring for small and large gardens. Normally, the plants need to be watered twice daily, morning and evening. So, the microcontroller has to be coded to water the plants in the greenhouse about two times per day. People enjoy plants, their benefits and the feeling related to nurturing them. However for most people it becomes challenging to keep them healthy and alive. To accommodate this challenge we have developed a prototype, which makes a plant more self-sufficient, watering itself from a large water tank and providing itself with artificial sunlight. The pro-To type reports status of its current conditions and also reminds the user to refill the water tank. The system automation is designed to be assistive to the user. We hope that through this prototype people will enjoy having plants without the challenges related to absent or forgetfulness.

Keyword's— Automatic Watering System, Arduino-board

I. INTRODUCTION

Irrigation is the artificial application of water to the land or soil. It is used to assist in the growing of agricultural crops, maintenance of landscapes, and re vegetation of disturbed soils in dry areas and during periods of inadequate rainfall. When a zone comes on, the water flows through the lateral lines and ultimately ends up at the irrigation emitter (drip) or sprinkler_heads. Many sprinklers have pipe thread inlets on the bottom of them which allows a fitting and the pipe to be attached to them. The sprinklers are usually installed with the top of the head flush with the ground surface. When the water is pressurized, the head will pop up out of the ground and water the desired area until the valve closes and shuts off that zone. Once there is no more water pressure in the lateral line, the sprinkler head will retract back into the ground. Emitters are generally laid on the soil surface or buried a few inches to reduce evaporation losses.

Healthy plants can transpire a lot of water, resulting in an increase in the humidity of the greenhouse air. A high relative humidity (above 80-85%) should be avoided because it can increase the incidence of disease and reduce plant transpiration. Sufficient venting or successive heating and venting can prevent condensation on plants surfaces and the greenhouse structure. The use of cooling systems during the warmer summer months increases the greenhouse air humidity. During periods with warm and humid outdoor conditions, humidity control inside the greenhouse can be a challenge. Greenhouses located in dry, dessert environments benefit greatly from evaporative cooling systems because large amounts of water can be evaporated into the incoming air, resulting in significant temperature drops.

Since the relative humidity alone does not tell us anything about the absolute water holding capacity of air, a different measurement is sometime used to describe the absolute moisture status of the soil. The vapor pressure deficit is a measure of the difference between the amount of moisture the air contains at a given moment and the amount of moisture it can hold at that temperature when the air would be saturated. Pressure deficit measurement can tell us how easy it is for plants to transpire: higher values stimulate transpiration (but too high can cause wilting), and lower values inhibit transpiration and can lead to condensation on leaf and greenhouse surfaces.

In the mid 20th century, the advent of diesel and electric motors led to systems that could pump groundwater out of major aquifers faster than drainage basins could refill them. This can lead to permanent loss of aquifer capacity, decreased water quality, ground subsidence, and other problems.

Apart from all these problems and failures, there has been a considerable evolution in the methods to perform irrigation with the help of technology. The application of technology in the areas of irrigation has proven to be of great help as they deliver efficiency and accuracy.

II. BLOCK DIAGRAM & WORKING

There are two functional components in this project. They are the moisture sensors and the motor/water pump. Thus the Arduino Board is programmed using the Arduino IDE software. The function of the moisture sensor is to sense the level of moisture in the soil. The motor/water pump supplies water to the plants.

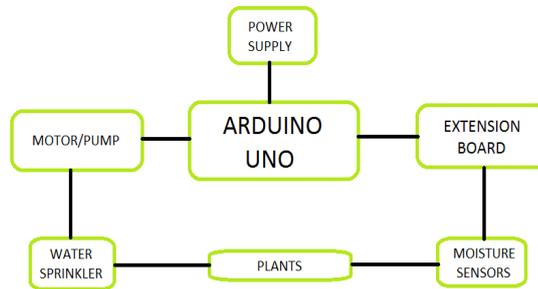


Figure 2.1 Automatic Plant Watering Block Diagram

This project uses Arduino Uno to controls the motor. Follow the schematic to connect the Arduino to the motor driver, and the driver to the water pump. The motor can be driven by a 9 volt battery, and current measurements show us that battery life. The Arduino Board is programmed using the Arduino IDE software. The moisture sensor measures the level of moisture in the soil and sends the signal to the Arduino if watering is required. The motor/water pump supplies water to the plants until the desired moisture level is reached.

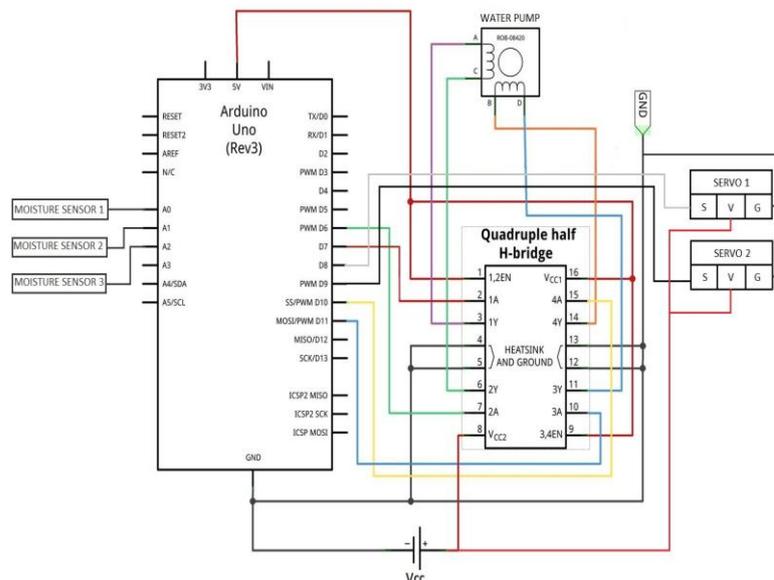


Figure 2.2 Automatic Plant Watering Schematic diagram

2.1 ARDUINO UNO:

The Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

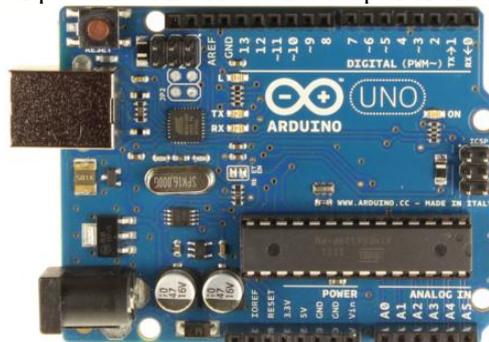


Fig 2.1.1 Arduino Uno

The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

Table 2.1.1 Arduino Specifications

FEATURE	SPECIFICATION
Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by boot loader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

2.2 Programming:

The Arduino Uno can be programmed with the Arduino software. Select "Arduino Uno from the Tools > Board menu (according to the microcontroller on your board). For details, see the reference and tutorials.

The ATmega328 on the Arduino Uno comes preburned with a boot loader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files). We can also bypass the boot loader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details.

The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available. The ATmega16U2/8U2 is loaded with a DFU boot loader, which can be activated by:

- On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2.
- On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode.

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1). A Software Serial library allows for serial communication on any of the Uno's digital pins. The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus.

2.3. Physical Characteristics:

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Four screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

2.4. MOISTURE SENSOR:

Soil moisture sensors measure the water content in soil. A soil moisture probe is made up of multiple soil moisture sensors.

Technologies commonly used in soil moisture sensors include:

- Frequency domain sensor such as a capacitance sensor.

- Neutron moisture gauges, utilize the moderator properties of water for neutrons.
- Electrical resistance of the soil

In this particular project, we will use the moisture sensors which can be inserted in the soil, in order to measure the moisture content of the soil.

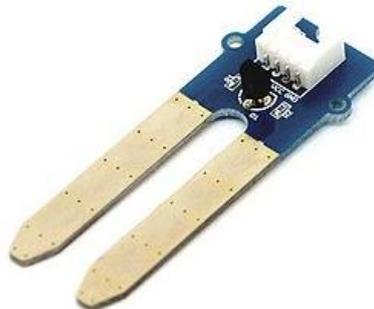


Fig 2.4.1 Moisture Sensor

Soil electrical conductivity is simply measured using two metal conductors spaced apart in the soil except that dissolved salts greatly alter the water conductivity and can confound the measurements. An inexpensive fix is to embed conductors in a porous gypsum block which releases calcium and sulphate ions to swamp the soil background level of ions. The water absorbed by the block is correlated with soil water potential over the range -60 to -600 kPa providing a tertiary indicator for use in medium to heavy soils. Non-dissolving granular matrix sensors are now available with a more exacting specification for the range 0 to -200 kPa and use internal calibration methods to offset variations due to solutes and temperature.

Methods for exploiting soil dielectric properties actually measure proxy variables that more or less include a component due to the soil electrical conductivity and are thus inherently sensitive to variations in soil salinity and temperature as well as water. Measurements are also affected by soil bulk density and the proportion of bound and free water determined by the soil type. Nevertheless, good accuracy and precision can be achieved under specific conditions and some sensor types have become widely adopted for scientific work.

In general, conversions from raw sensor readings to volumetric moisture content or water potential using secondary or tertiary methods tend to be sensor or soil specific, affected or precluded at high salinity levels and dependent on temperature. Research-grade instruments typically have laboratory measured accuracy worse than +/- 4% when relying on factory settings or as good as +/- 1% when calibrated for the specific soil. Sensors based on the TDR method seem to require least calibration but may be unsuitable for soils with very high salinity or clay content. There are no comparable laboratory specifications for granular matrix sensors, possibly because they are technically more difficult to calibrate, their response times are relatively slow and the output is hysteretic for wetting and drying curves.

Soil dielectric measurement is the method of choice for most research studies where expertise is available for calibration, installation and interpretation, but scope for cost reduction through sensor multiplexing is limited due to the possibility of stray capacitances. A lower manufacturing cost is possible through development of application specific integrated circuits (ASICs), though this requires a high level of investment. Multiple sensors are required to provide a depth profile and cover a representative area, but this cost can be minimized through use of a computer model to extend the measurements in a predictive way. Thus, by using the moisture sensors, the over-riding factor will be reliable, cost-effective sensors and electronic systems for accessing and interpreting the data.

2.5. WATER PUMP:

The water pump is used to artificially supply water for a particular task. It can be electronically controlled by interfacing it to a microcontroller. It can be triggered ON/OFF by sending signals as required. The process of artificially supplying water is known as pumping. There are many varieties of water pumps used. This project employs the use of a small water pump which is connected to a H-Bridge.



Fig 2.5.1 Water Pump

The pumping of water is a basic and practical technique, far more practical than scooping it up with one's hands or lifting it in a hand-held bucket. This is true whether the water is drawn from a fresh source, moved to a needed location, purified, or used for irrigation, washing, or sewage treatment, or for evacuating water from an undesirable location. Regardless of the outcome, the energy required to pump water is an extremely demanding component of water consumption. All other processes depend or benefit either from water descending from a higher elevation or some pressurized plumbing system.

2.6. SERVO:

A servo is a rotary actuator that allows for precise control of angular position, velocity and acceleration. It consists of a suitable motor coupled to a sensor for position feedback. It also requires a relatively sophisticated controller, often a dedicated module designed specifically for use with servomotors.



Fig 2.6.1 Servo

Servomotors are not a different class of motor, on the basis of fundamental operating principle, but use servo mechanism to achieve closed loop control with a generic open loop motor. In other words, a servomotor is just a regular motor with a sensor installed, typically to measure angular position during operation. A servomotor is a specific type of motor and rotary encoder combination that forms a servomechanism. This assembly may in turn form part of another servomechanism. The encoder provides position and usually speed feedback, which by the use of a PID controller allow more precise control of position and thus faster achievement of a stable position (for a given motor power).

Stepper motors are not considered as servomotors, although they too are used to construct larger servomechanisms. Stepper motors have inherent angular positioning, owing to their construction, and this is generally used in an open-loop manner, without an encoder.

Servomotors are used for both high-end and low-end applications, although the mid-range is generally handled by stepper motors. Most servomotors, at least under this name, are precision industrial components. However the very cheap radio control servo, because it combines a free-running motor and a simple position sensor with an embedded controller, also qualifies as a servomotor.

III. DESCRIPTION OF ATMEGA 328P MICRO CONTROLLER:

The ATmega48PA/88PA/168PA/328P is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega48PA/88PA/168PA/328P achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed

The ATmega48PA/88PA/168PA/328P provides the following features: 4K/8K bytes of In-System Programmable Flash with Read-While-Write capabilities, 256/512/512/1K bytes EEPROM, 512/1K/1K/2K bytes SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible Timer/Counters with compare modes, internal and external interrupts, a serial programmable USART, a byte-oriented 2-wire Serial Interface, an SPI serial port, a 6-channel 10-bit ADC (8 channels in TQFP and QFN/MLF packages), a programmable Watchdog Timer with internal Oscillator, and five software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, USART, 2-wire Serial Interface, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next interrupt or hardware reset.

In Power-save mode, the asynchronous timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except asynchronous timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the crystal/resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low power consumption.

The AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle.

The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port, and interrupt system to continue functioning. The Power-down mode saves the RAM contents but freezes the oscillator, disabling all other chip functions until the next interrupt or hardware reset. The device is manufactured using Atmel's high density non-volatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed In-System through an SPI serial interface, by a conventional non-volatile memory programmer, or by an On-chip Boot program running on the AVR core. The Boot program can use any interface to download the application program in the Application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega48PA/88PA/168PA/328P is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

The Boot program can use any interface to download the application program in the Application Flash memory. This allows very fast start-up combined with low power consumption.

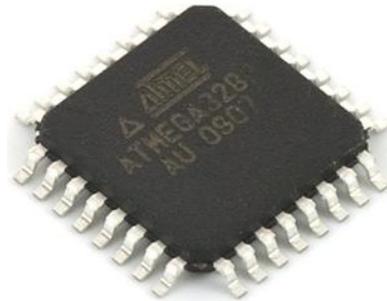


Fig 3.1. ATMEGA 328

Even though there are separate addressing schemes and optimized opcodes for register file and I/O register access, all can still be addressed and manipulated as if they were in SRAM.

In the ATMEGA variant, the working register file is not mapped into the data address space; as such, it is not possible to treat any of the ATMEGA's working registers as though they were SRAM. Instead, the I/O registers are mapped into the data address space starting at the very beginning of the address space. Additionally, the amount of data address space dedicated to I/O registers has grown substantially to 4096 bytes (0000₁₆–0FFF₁₆). As with previous generations, however, the fast I/O manipulation instructions can only reach the first 64 I/O register locations (the first 32 locations for bitwise instructions).

In most variants of the AVR architecture, this internal EEPROM memory is not mapped into the MCU's addressable memory space. It can only be accessed the same way an external peripheral device is, using special pointer registers and read/write instructions which makes EEPROM access much slower than other internal RAM.

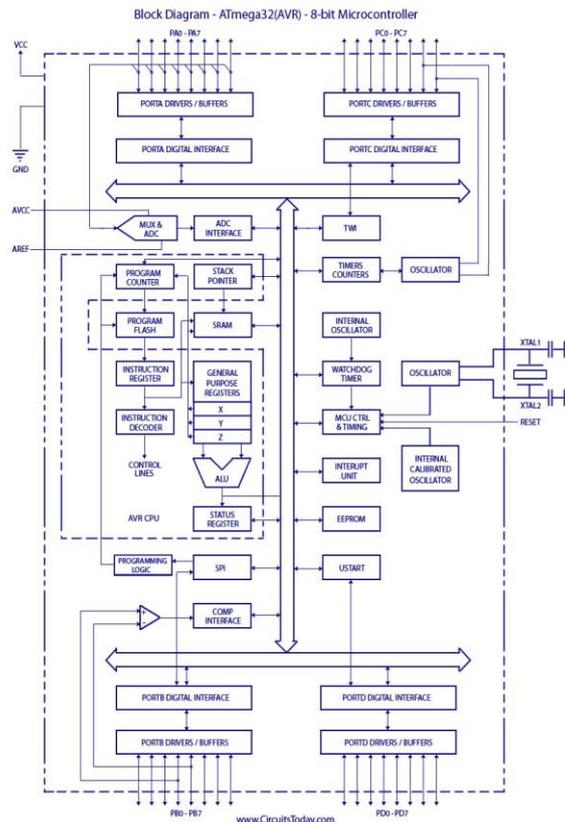


Figure 3.2 : Pin configuration

IV. ARDUINO IDE TOOL

The open-source Arduino environment makes it easy to write code and upload it to the i/o board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing, avr-gcc, and other open source software.

Steps for using Arduino IDE:

Step 1: Get an Arduino board and USB cable

In this tutorial, we assume you're using an Arduino Uno You also need a standard USB cable (A plug to B plug): the kind you would connect to a USB printer, for example.

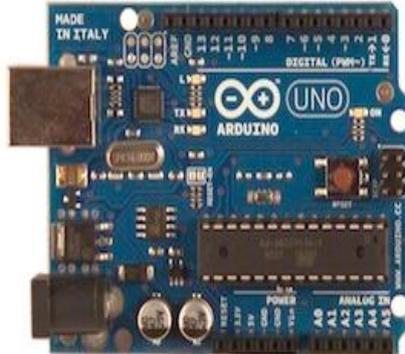


Fig:4.1. Arduino

Step 2 : Download the Arduino environment

Get the latest version from the download page. When the download finishes, unzip the downloaded file. Make sure to preserve the folder structure. Double-click the folder to open it. There should be a few files and sub-folders inside.

Step 3 : Connect the board

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either the USB connection to the computer or an external power supply. If you're using an Arduino Diecimila, you'll need to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it's on the two pins closest to the USB port.

Connect the Arduino board to your computer using the USB cable. The green power LED (labelled PWR) should go on.

Step 4 : Install the drivers

Installing drivers for the Arduino Uno or Arduino Mega 2560 with Windows7, Vista, or XP

Step 5: Launch the Arduino application

Double-click the Arduino application. (Note: if the Arduino software loads in the wrong language, you can change it in the preferences dialog. See the environment page for details.)

Step 6: Open the blink example

Open the LED blink example sketch: File > Examples > 1.Basics > Blink.

Step 7: Select your board

You'll need to select the entry in the Tools > Board menu that corresponds to your Arduino.

Step 8: Select your serial port

Select the serial device of the Arduino board from the Tools | Serial Port menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu; the entry that disappears should be the Arduino board. Reconnect the board and select that serial port.

Step 9 : Upload the program

Now, simply click the "Upload" button in the environment. Wait a few seconds - you should see the RX and TX leds on the board flashing. If the upload is successful, the message "Done uploading." will appear in the status bar.

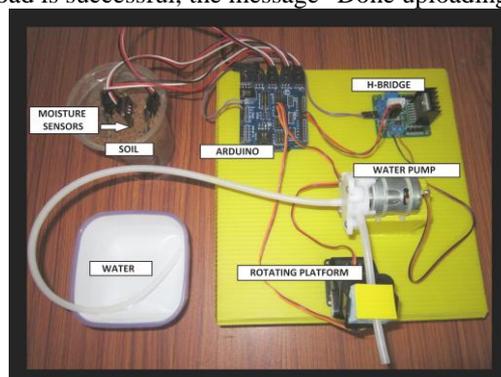


Fig:4.2. Arduino Based
Automatic Plant Watering System

V. CONCLUSION

Thus the “**ARDUINO BASED AUTOMATIC PLANT WATERING SYSTEM**” has been designed and tested successfully. It has been developed by integrated features of all the hardware components used. Presence of every module has been reasoned out and placed carefully, thus contributing to the best working of the unit. Thus, the Arduino Based Automatic Plant Watering System has been designed and tested successfully. The system has been tested to function automatically. The moisture sensors measure the moisture level (water content) of the different plants. If the moisture level is found to be below the desired level, the moisture sensor sends the signal to the Arduino board which triggers the Water Pump to turn ON and supply the water to respective plant using the Rotating Platform/Sprinkler. When the desired moisture level is reached, the system halts on its own and the Water Pump is turned OFF. Thus, the functionality of the entire system has been tested thoroughly and it is said to function successfully.

REFERENCES

- [1] Design and construction of automatic power changeover system, by Jonathan Gana Kolo
- [2] The 8051 Micro controller and Embedded Systems, by Muhammad Ali Mazidi
- [3] Micro processor Architecture, Programming & Applications, by Ramesh S. Gaonkar
- [4] Fundamentals Of Micro processors and Micro computers, by B.Ram
- [5] Embedded system, by Raj Kamal

AUTHORS:

- [1] **Mrs. S. V. Devika** Working as an Associate Professor in Hyderabad Institute of Technology & Management, her area of interest is communications, VLSI & Antenna theory. She can be reached at chinn.devika@gmail.com.
- [2] **Mr.SK. Khamuruddeen** working as an Associate Professor in Hyderabad Institute of Technology & Management, His area of interest is VLSI & System Design. He can be reached at khd485@gmail.com.
- [3] **Mrs. SK.KHAMURUNNISA Pursuing MSC. Her area** of interest in research areas related to embedded system, image processing.
- [4] **Mr. Jayanth Thota** he is good student of ECE, HITAM College, Hyderabad. He is active members, in technical education along the student related motivation activities. his area of interest is communications, VLSI & Antenna theory
- [5] **Mr. Khalesha Shaik** He is active members, in technical education along the student related motivation activities. his area of interest is communications, VLSI & Antenna theory