

Arduino based Automatic Monitoring Environment by Sensors and GSM

Sara W. Abdulmajeed

Department of Network
Engineering,
Al-iraqia University, Baghdad, Iraq

Mothena F. Shaker

Department of Network
Engineering,
Al-iraqia University, Baghdad, Iraq

Zahraa K. Taha

Department of Network
Engineering,
Al-iraqia University, Baghdad, Iraq

ABSTRACT

Environment monitoring plays an important role in human life, so the collection of information about changes of environment is very important. In this paper, an Arduino based microcontroller system and many sensors are used to monitor the environment. The parameters of the environment to be monitored are chosen as temperature, humidity, water level and light. The values of these parameters are transmitted to a base station where they are being monitored, so that every person in the range of the system can check it over their smart phones and laptops as these parameters hold importance to everyone. In proposed system, when the parameters reaches a defined threshold value a message will be displayed in the printed circuit boards LCD, and a SMS will be sent to the user's mobile phone.

General Terms

Arduino Uno, GSM, LCD, Humidity sensor Light Dependent Resistance and water sensor.

Keywords

Arduino Uno, GSM, LCD, Humidity sensor Light Dependent Resistance and water sensor.

1. INTRODUCTION

The Arduino is a tiny computer system that can be programmed with your instructions to interact with various forms of input and output. The current Arduino board model, the Uno, is quite small in size compared to the average human hand [1]. Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs from personal computers. The microcontrollers are typically programmed using a dialect of features from the programming languages C and C++. In addition to using traditional compiler toolchains, the Arduino project provides an integrated development environment (IDE) based on the Processing language project [2].

Shield Arduino that can accept the modules from Simcom SIM900 (GSM/GPRS functionality) and SIM928 (functionality GSM / GPRS & GPS) and allows to make voice calls, send SMS or make connections to the Internet network. The availability of new GSM/GPRS & GPS library, allow great flexibility. For example is possible reading a received text message or start a call and make all other operations performed by a standard mobile phone [3].

Sensor collects information from various environment the

actuators takes place to produce the outcome for the collected information to know the status of that environment. The actuator can control the environment changes. Many sensors can be used with arduino for examples water sensor, humidity sensor etc.

A monitoring of environment is an instrument that measures and records parameters using sensors without intervention of humans. The measured parameters can be stored or can be transmitted via GSM.

In a monitoring environment each sensors consists of processing capability (one or more microcontroller), may contain multiple types of memory (program, data and flash memories), have a Zigbee, GSM, RF transceiver. Wireless communication is widely used in electronics. The design to implement for monitoring the environment using many sensors which manages information [4].

2. ADRIANO UNO [1]

microcontroller is an easy to use yet powerful single board computer that has gained considerable traction in the hobby and professional market. The Arduino is open-source, which means hardware is reasonably priced and development software is free. This guide is for students in ME 2011, or students anywhere who are confronting the Arduino for the first time. The Duemilanove board features an Atmel ATmega328 microcontroller operating at 5 V with 2 Kb of RAM, 32 Kb of flash memory for storing programs and 1 Kb of EEPROM for storing parameters. The clock speed is 16 MHz, which translates to about executing about 300,000 lines of C source code per second. The board has 14 digital I/O pins and 6 analog input pins. There is a USB connector for talking to the host computer and a DC power jack for connecting an external 6-20 V power source, for example a 9 V battery, when running a program while not connected to the host computer. Headers are provided for interfacing to the I/O pins using 22 g solid wire or header connectors .Show in figure(1)

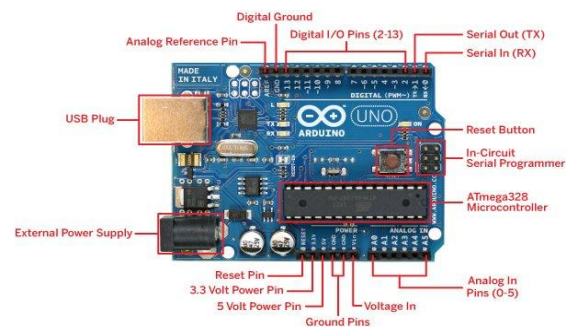


Fig. 1: Arduino uno

3. GSM/ GPRS SIM900

The Arduino GSM Shield 2 allows an Arduino board to connect to the internet, make/receive voice calls and send/receive SMS messages. The shield uses a radio modem M10 by Quectel. It is possible to communicate with the board using AT commands. The GSM library has a large number of methods for communication with the shield as shown in figure 2.

The shield uses digital pins 2 and 3 for software serial communication with the M10. Pin 2 is connected to the M10's TX pin and pin 3 to its RX pin [5].

You will need to acquire a SIM card for this shield. Works on frequencies 850 MHz, 900 MHz, 1800 MHz and 1900 MHz. The baud rate can be configurable from 9600-115200 through AT command [6].

- Requires an Arduino board (not included)
- Operating voltage 5V (supplied from the Arduino Board)
- Connection with Arduino Uno on pins 2, 3 (Software Serial) and 7 (reset).

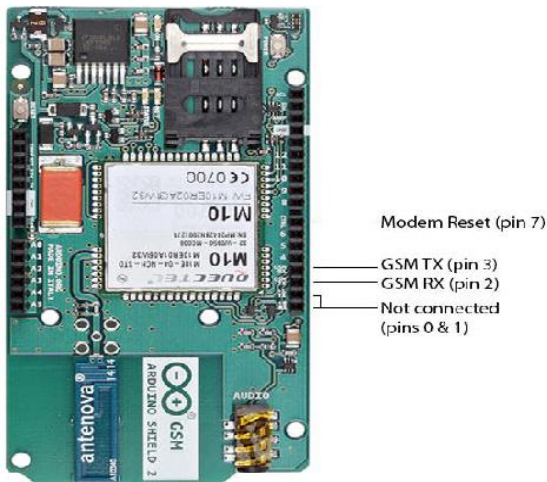


Fig. 2: GSM/GPRS

4. HUMIDITY SENSOR (DHT11)

Humidity sensor in figure 3 works on the principle of relative humidity and gives the output in the form of voltage. This analog voltage provides the information about the percentage relative humidity present in the environment [7]. The AM2302 is a wired version of the DHT11, in a large plastic body. It is a basic, low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air, and spits out a digital signal on the data pin (no analog input pins needed). It's fairly simple to use, but requires careful timing to grab data.

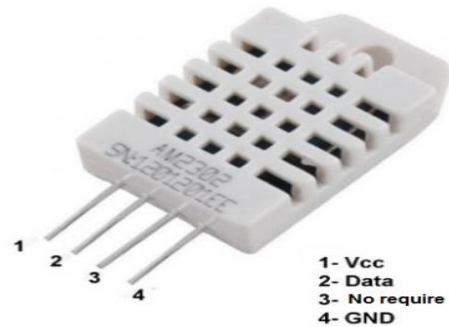


Fig. 3: AM2302 Humidity sensor

Item	Measurement Range	Humidity Accuracy	Temperature Accuracy	Resolution	Package
DHT11	20-90%RH 0-50 °C	±5%RH	±2°C	1	4 Pin Single Row

From data sheet of sensor, this sensor can read the humidity between 20-90 and temperture between 0-50C. There is more information about this sensor in the data sheet [8].

5. WATER SENSOR

Water sensor brick is designed for water detection, which can be widely used in sensing rainfall, water level, and even liquid leakage.



Fig.4: Water Sensor

Connecting a water sensor to an Arduino is a great way to detect a leak, spill, flood, rain, etc. It can be used to detect the presence, the level, the volume and/or the absence of water. While this could be used to remind you to water your plants, there is a better Grove sensor for that. The sensor has an array of exposed traces, which read LOW when water is detected.

Water sensor has three terminals - S, Vout(+), and GND (-). Connect the sensor as follows

- Connect the +Vs to +5v on your Arduino board.
- Connect S to any digital pin number on Arduino board.
- Connect GND with GND on Arduino.

6. LIGHT DEPENDENT RESISTANCE (LDR)

A light-dependent resistor (LDR) is a light controlled variable resistor. An LDR is a component that has a (variable) resistance [10]. The light dependent resistor resistance changes with intensity of light, with increase in light intensity the resistance offered by the sensor decreases and with decrease in light intensity the resistance offered by the sensor increases. Hence it acts as variable resistor with change in light intensity. These helps in finding the amount of light intensity at that instant of time and thus helping in regulating the lighting of our lighting system accordingly [11].

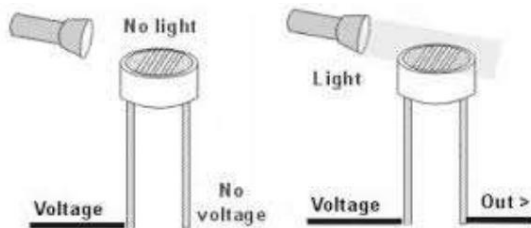


Figure 5: Light Depend Resistance with/ without light

7. LIQUID CRYSTAL (LCD) [12]

LCD modules that display characters such as text and numbers are the most inexpensive and simplest to use of all LCDs. They can be purchased in various sizes, which are measured by the number of rows and columns of characters they can display. Some include a backlight and allow you to choose the color of the character and the background color. Any LCD with an HD44780- or KS0066-compatible interface should work with your Arduino.

The LiquidCrystal library works with all LCD displays that are compatible with the Hitachi HD44780 driver. There are many of them out there, and you can usually tell them by the 16-pin interface.

With the LCD setup complete, clear the LCD's display with the following:

```
lcd.clear();
```

Then, to position the cursor, which is the starting point for the text,

use this:

```
lcd.setCursor(x, y);
```

Here, x is the column (0 to 15) and y is the row (0 or 1). So, for example,

to display the word text, you would enter the following:

```
lcd.print("text");
```



Fig. 6: LCD demonstration.

The circuit:

- * LCD RS pin to digital pin 12

- * LCD Enable pin to digital pin 11

- * LCD D4 pin to digital pin 5

- * LCD D5 pin to digital pin 4

- * LCD D6 pin to digital pin 3

- * LCD D7 pin to digital pin 2

- * LCD R/W pin to ground

10 * K resistor:

- * ends to +5V and ground

- * LCD VO to variable resistance

1. BLOCK DIAGRAM AND WORKING

A new architectural model was developed for the monitoring of the environment and data is transmitting by GSM. Figure 7 illustrates the workflow of sensor data transmission.

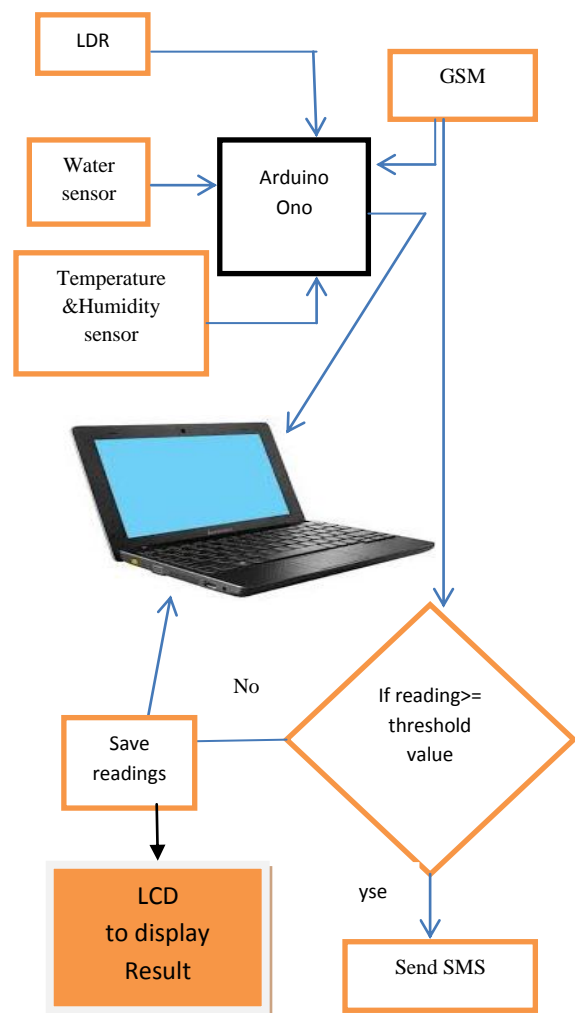


Fig. 7: Block diagram of the proposed system

Project:

```
#include <SoftwareSerial.h>
```

```
#include <LiquidCrystal.h>
```

```
LiquidCrystal lcd(12,11,5,4,3,2); // pins for RS, E, DB5, DB4, DB3, DB2
```

```
int water_pin=8;
int humidity_pin=2;
int light_pin=A0;
//SoftwareSerial GSM(7, 8);//Rx, TX of arduino
void setup()
{
  lcd.begin(16,2); // we tell the Arduino the size of the LCD //in
  columns and rows.
  pinMode(humidity_pin, INPUT);
  pinMode(water_pin, INPUT);
  GSM.begin(9600);
  Serial.begin(9600);
  GSM.println("AT+CMGF=1");
  delay(1000);
}
void loop()
{
  while(GSM.available()) {
    Serial.write(GSM.read());
  }
  int humidity_val = digitalRead(humidity_pin);
  int water_val = digitalRead(water_pin);
  int light_val = analogRead(light_pin);
  sendSMS();
}
delay(500);
}
void sendSMS() {
  Serial.print("Switch was turned ");
  Serial.println(state ? "on" : "off");
  GSM.println("AT+CMGS=\n"+64123456789\");
  delay(500);
  GSM.print("humidity_val =");
  GSM.println(humidity_val );
  GSM.print("water_val =");
  GSM.println(water_val );
  GSM.println("light_val" );
  GSM.println(light_val );
  GSM.println(state ? "on" : "off");
  GSM.write( 0x1a ); // ctrl+Z character
  delay(500);
```

```
}
// Then, to position the cursor, which is the starting point for
the text
lcd.setCursor(0,0);
//Displaying Text:
lcd.print(humidity_val);
  lcd.print(water_val)
  lcd.print(light_val);
  {
```

8. CONCLUSION

In agriculture, sensor plays major role in environmental monitoring like light, temperature, water and humidity. In our proposed method, the environment is monitored by sensors and the measured information's are transmitted to the user's mobile phone by means of GSM. The GSM is simple than Zigbee. It is more reliability than other wireless methods

9. REFERENCES

- [1] J. Boxball "Arduino Workshop: A Hands-On Introduction with 65 Projects" 1st edition, 2013
- [2] <https://en.wikipedia.org/wiki/Arduino>
- [3] https://store.open-electronics.org/GSMGPRSSHIELDv2?gclid=Cj0KEQjwn_3GBRDc8rCnup-1x8wBEiQAdw3OAVwIZ50JB5o8_La_Ut7g68-u2i1CBVPI6xpjoGzDI4saAhHX8P8HAQ
- [4] <https://www.arduino.cc/en/Main/ArduinoGSMShield>
- [5] S. Devika, SK. Khamuruddeen, J. Thota and K. Shaik "Arduino Based Automatic Plant Watering System" International Journal of Advance Research in Computer Science and Software Engineering, vol. 4, issue 10, October 2014.
- [6] H. Timmis "Practical Arduino Engineering", 2011.
- [7] P. Susmitha and G. Sowmyabala "Design and implementation of weather Monitoring and Controlling System", international Journal of Computer Application, vol. 79, No. 3, July 2014.
- [8] <https://arduino-info.wikispaces.com/DHT11-Humidity-TempSensor?responseToken=47d75760cbbb3fb177d30cc95c5351c4>
- [9] <http://www.instructables.com/id/How-to-use-a-Water-Level-Sensor-Arduino-Tutorial/>
- [10] P. S. Sathish and B. Chellaprabha "MONITORING THE PLANT GROWTH USING SENSOR NETWORK" ARPN Journal of Engineering and Applied Sciences, vol. 10, No. 7, April 2015.
- [11] D. Rath "Arduino Based: Smart Light Control System" International Journal of Engineering Research and General Science, vol4, issue 2, April, 2016.
- [12] J. Boxball "Arduino Workshop: A Hands-On Introduction with 65 Projects" 1st edition, 2013