Smart Bin Using GSM Modem

Mabel R. Luis[1] Diksha S. Lotlikar[2] Chrislin L. Fernandes[3] Maria Samantha Cardoso[4]
[1,2,3]Student [4]Assistant Professor
[1,2,3,4]Department of Electronics and Telecommunication Engineering
[1,2,3,4]Don Bosco College Of Engineering, Fatorda, 403602, Goa, India

Abstract— The waste segregation system will sort the waste into its different components. This segregated waste is then collected in a smart bin equipped with GSM technology to indicate via SMS the status of each bin. The bins are built on a microcontroller based platform Arduino ATmega 2560 which is interfaced with GSM module. The ultrasonic sensor is used for distance measurement. The threshold distance is the difference in height at which the sensor is placed and the level of garbage in the bins. Whenever the difference between the sensor and the waste falls below the threshold value the GSM module is activated to send an alert signal to the concerned personal through an SMS. As soon as an SMS alert is received the concerned authority can place orders to the workers for emptying the bins before they overflow.

Key words: Arduino ATmega 2560, GSM/GPRS SIM900A module, ultrasonic sensor

I. Introduction

An increase in population leads to increase urbanization and an increase in the amount of goods produced to satisfy various human needs. This increasing urbanization is a major cause of the large amount of waste generated. Overflowing waste bins have become a common sight in a number of cities. These overflowing bins are an ideal breeding ground for bacteria, insects and vermin causing various diseases. Hence smart dustbin is a system which can eradicate this problem or at least reduce it to a minimum level [1].

Fig. 1: Overflowing Garbage Bin

II. Block Diagram

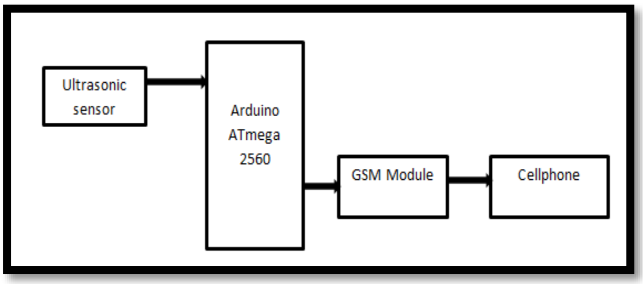


Fig. 2: Block diagram for Smart Bin

Interfacing is done by connecting receiver pin of the module to transmitter pin of the board and transmitter pin of the module to receiver pin of the board. The VCC and GND pins of the ultrasonic sensor are connected to the VCC and GND pins of the Arduino board respectively. The Echo and Trigger pins of the ultrasonic sensor are connected to the digital pins of the Arduino board.

III. Flow Chart

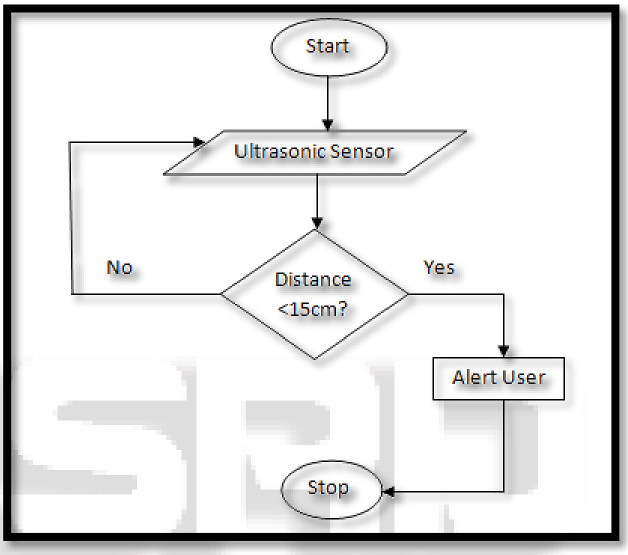


Fig. 3:

A. *Ultrasonic sensor:*

It is to find the distance from the object. Ultrasonic distance sensor uses a sound transmitter and a receiver. The ultrasonic sensor sends out a high frequency sound pulse and then times how long it takes for the echo of the sound is approximately 341m/s in air. The ultrasonic sensor uses this information along with the sound pulse to determine the distance of an object [2]. Some objects might not be detected by the ultrasonic sensor. This is because some objects are shaped or positioned in such a way that the sound wave bounces off the object but are deflected away from the ultrasonic sensor. It is also possible for the objects to be too small to reflect enough of the sound wave back to the sensor to be detected. Other objects can absorb the sound wave all together, which means that there is no way for the sensor to detect them accurately. The accuracy of ultrasonic sensor can be affected by the temperature and humidity in the air; however this change in accuracy is negligible. Ultrasonic sensors use sound instead of light for ranging so that they can be used outside in bright sunlight whereas IR sensor cannot be used [3].

Parameters	Comments
Supply Voltage	5V
Global Current Consumption	15mA
Ultrasonic Frequency	40KHz
Maximal Range	400cm

Minimal Range	3cm
Trigger Pulse Width	10μs

Table 1: Specification of ultrasonic sensor

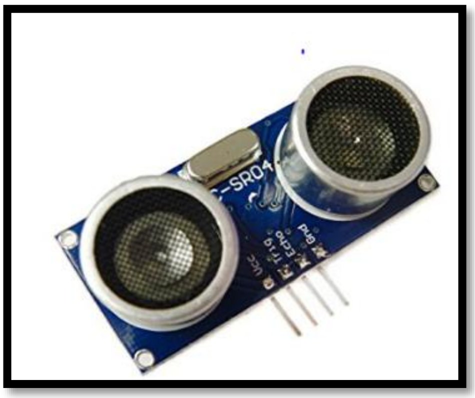


Fig. 4: Ultrasonic Sensor

B. GSM Modem:

The GSM modem can either be connected to the PC or a serial port directly or to any microcontroller through MAX 232. It can be used to send and receive SMS or make/receive voice calls. It can also be used in GPRS mode to connect to the internet and do many applications for data logging and control. In GPRS mode we can also connect to any remote FTP server and upload files for data logging. This GSM modem is highly flexible plug and play quad band SIM900A GSM module for direct and easy integration to RS-232 application [4].

C. SIM900A Dual band GSM/GPRS TTL Module:

SIM900A works on frequency 900/1800MHz. It has a selectable interfacing voltage, which allows connecting 5V and 3.3V microcontroller directly without any level conversion chips. The baud rate is configurable from 9600-115200 through AT command. The GSM/GPRS module is having internal TCP/IP stack to enable to connect with the internet via GPRS. It is suitable for SMS, voice as data transfer applications in M2M interface. The modem can operate at both TTL 3.3V and 5V logic level hence making it suitable for controllers like 3.3V or 5V and power the module with supply range (4.5-12V). This modem is meant for Asian countries only [5].

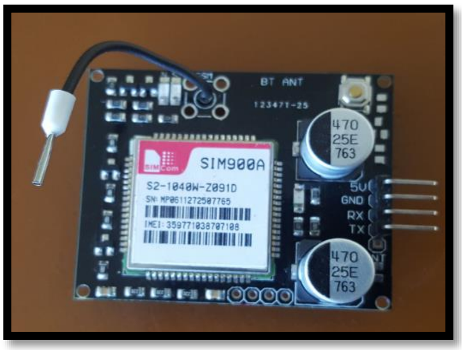


Fig. 5: SIM900A Dual band GSM/GPRS TTL Module

D. Arduino Board:

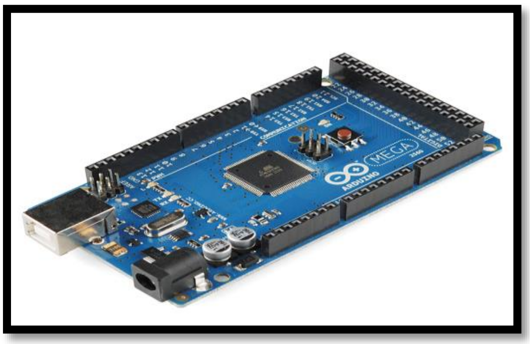


Fig. 6: Arduino ATmega 2560

The Arduino ATmega 2560 is a microcontroller board based on ATmega 2560. It has 54 digital input output pins,16 analog input, 4 UARTs a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. Connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery. Arduino board works at 5V power supply. The Mega 2560 is compatible with most shields designed for the uno and the former boards Duemilanove or Diecimila[6]. Arduino is preferred because it has more digital and analog pins and programming is easy.

Parameters	Comments
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	20 Ma
DC Current for 3.3V Pin	50 Ma
Flash Memory	256KB of which 8KB used by bootloader
SRAM	8KB
EEPROM	4KB
Clock Speed	16 MHz

Table 2: Specifications of Arduino 2560:

Power:

The Arduino Mega2560 can be powered via the USB connection or with an external power supply. The power source is selected automatically.

The power pins are as follows:

- VIN: The input voltage to the Arduino board when it's using an external power source.
- 5V: The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- 3.3V: A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50mA.
- GND: Ground pins.

Memory:

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the boot loader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the EEPROM library).

Input and output:

Each of the 54 digital pins on the Mega can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX). Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3) and 21(interrupt 2). These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.
- PWM: 0 to 13. Provide 8-bit PWM output with the analogWrite() function.
- SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Duemilanove and Diecimila.
- LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- •2C: 20(SDA) and 21 (SCL).Support I2C (TWI) communication using the Wire library (documentation on the Wiring website).The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and analogReference() function.
- There are a couple of other pins on the board:
- AREF: Reference voltage for the analog inputs. Used with analogReference().
- Reset: To reset the microcontroller bring this line LOW. Typically used to add a reset button to shields which block the one on the board.

Reset:

Rather than requiring a physical press of the reset button before an upload, the Arduino Mega2560 is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega2560 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the boot loader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

Over current protection:

The Arduino Mega has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical characteristics:

The maximum length and width of the Mega PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case.

IV. Software

The IDE (Integrated Development Environment) is a special program running on your computer that allows you to write sketches for the Arduino board in a simple language modeled after the processing language. All we need to do is write a sketch using the IDE software and upload it onto the board. The code is written in C language and is passed to the avr-gcc compiler that translates the code top a language understandable to the microcontroller.

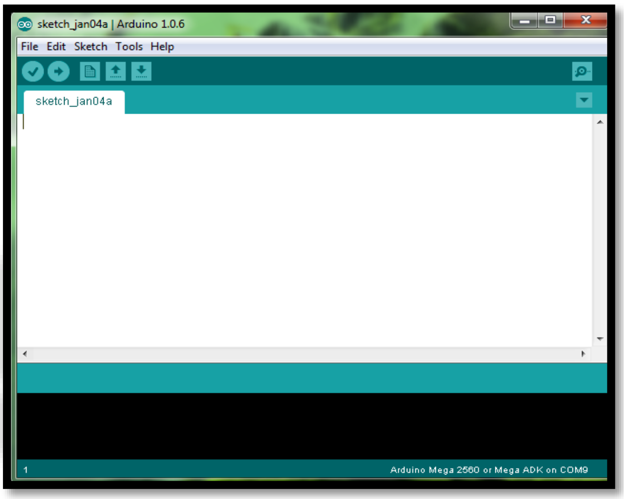

sketch_jan04a | Arduino 1.0.6
File Edit Sketch Tools Help
sketch_jan04a
1
Arduino Mega 2560 or Mega ADK on COM9

Fig. 7: Snapshot of Text Editor

V. Result

The observations are as follows:

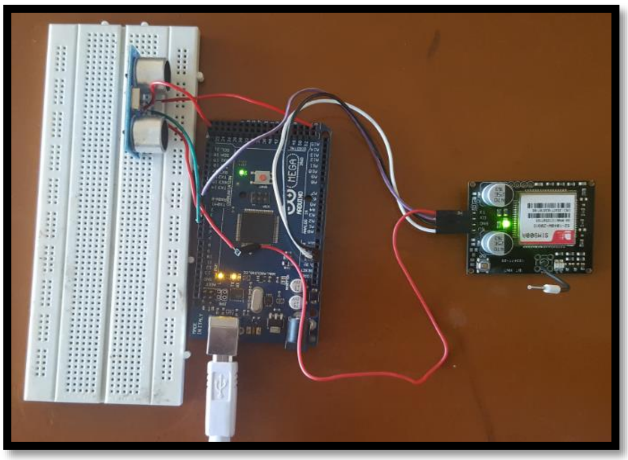
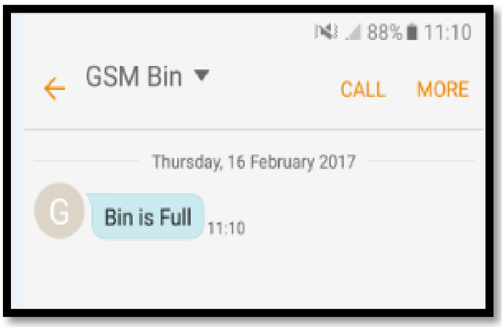


Fig. 8: Circuit Connections



Bin is Full 11:10

Fig. 9: Snapshot of Message Received

VI. Conclusion

We have implemented waste management system by using smart dustbins to check if the dustbin is full using a GSM module where the user will receive an SMS stating the dustbin is full. By implementing this proposed system cost reduction, resource optimization, effective usage of smart bin can be done.

VII. Future work

This system can also be implemented using IOT. In this system the dustbins are provided with low cost embedded device which helps in tracking the level of the garbage bins and an unique ID will be provided for every dust bin so it is easy to identify which bin is full. These details can be accessed by the concern authorities from their place with the help of Internet.

References

[1] Monika KA, Nikitha Rao, Prapulla SB, Shobha G, “Smart Dustbin-An Effective Garbage Monitoring System”, in International Journal of Engineering Science and Computing, Volume 6, Issue 6, 2016.
[2] http://www.ultrasonicdistancesensor
[3] http://educaion.rec.ri.emu.edu
[4] Smart Dustbin, “National conference on Product Design”, July 2016
[5] http://www.rhydolabz.com
[6] http://www.arduino.cc