Ijesm
Consulting, help, relaxation
INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES
&
MANAGEMENT

GSM MODEM BASED DATA ACQUISITION SYSTEM

Vandana Pandya*and Deepali Shukla
Medicaps Institute of Technology & Management, Indore, (M.P.)

Abstract

GSM Modem Based Data Acquisition is a process control system that enables a site operator to monitor and control processes that are distributed among various remote sites. This project is designed to monitor various parameters like humidity, rain fall, wind direction, temperature, light intensity etc. This system saves time and money by eliminating the need for service personnel to visit each site for inspection and data collection. They are used in all types of industries, from electrical distribution systems, to food processing, to facility security alarms.

Key-words: Modem, Data, Industries

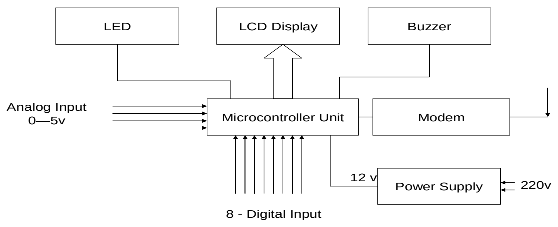
Introduction

This system uses AVR microcontroller ATmega 644P as its brain. Eight channel ADC receives analog data from various sensors connected to it and converts the data into digital data. This digital data will be sensed by the microcontroller and the system continuously monitors the data condition. This system is interfaced with a GSM modem. Whenever an abnormal condition occurs, this system senses the hazard and sends caution message to mobile numbers using SMS every ten minutes. This data is displayed on LCD every second. Using this system, the operator can monitor the signals from anywhere in the world as it works on GSM network.

The GSM modem is connected to microcontroller using RS232 interface. Whenever an SMS is sent to the GSM modem, the GSM modem receives the data and sends to microcontroller. After receiving the signal from the microcontroller it processes the data and sends the read data to the preprogrammed mobile number through GSM modem. The collected data is formed a database and stored in a pc.

The received data is displayed on the LCD. 16X 2 LCD is provided for user interface. This project uses regulated 12V, 1A SMPS.

Block Diagram



GSM Modem Based Data acquisition System Working

- This system uses AVR microcontroller ATmega 644P as its brain.
- Eight channel ADC receives analog data from various sensors connected to it and converts it to digital.
- This digital data is sensed by the microcontroller and the system continuously monitors the data condition.
- This system is interfaced with a GSM modem.
- The system monitors the conditions continuously and sends caution message to mobile numbers using SMS every ten minutes.
- This data is displayed on LCD every second. Using this system, the operator can monitor the signals from anywhere in the world as it works on GSM network.
- At the receiving end the SMS is received by the modem and processed by the microcontroller and the data is collected

*** Corresponding Author**

by a central server and formed a data base and stored in a pc.

- Mobile Switching Centre) is used for short messages originated with a mobile on that network. The SMS-GMSC role is similar to that of the GMSC, whereas the SMS-IWMSC provides a fixed access point to the Short Message Service Centre.

GSM Modem

A GSM modem **is** a specialized type of modem which accepts a SIM card, and operates over a subscription to a mobile operator, just like a mobile phone. From the mobile operator perspective, a GSM modem looks just like a mobile phone. When a GSM modem is connected to a computer, this allows the computer to use the GSM modem to communicate over the mobile network. While these GSM modems are most frequently used to provide mobile internet connectivity, many of them can also be used for sending and receiving SMS and MMS messages.

GSM modem must support an "extended AT command set" for sending/receiving SMS messages. GSM modems can be a quick and efficient way to get started with SMS, because a special subscription to an SMS service provider is not required. In most parts of the world, GSM modems are a cost effective solution for receiving SMS messages, because the sender is paying for the message delivery.

To begin, insert a GSM SIM card into the modem and connect it to an available USB port on your computer.

Features of SIM300 GSM Module

Designed for global market, SIM300 is a Tri-band GSM engine

Works on frequencies EGSM 900 MHz, DCS 1800 MHz and PCS 1900 MHz.

SIM300 features GPRS multi-slot class 10/ class 8 (optional) and supports the GPRS coding schemes

CS-1, CS-2, CS-3 and CS-4.With a tiny configuration of 40mm x 33mm x 2.85mm

SIM300 can fit almost all the space requirements in your applications, such as smart phone, PDA phone and other mobile devices

This GSM modem is a highly flexible plug and play quad band GSM modem

interface to RS232.

- Supports features like Voice, Data/Fax, SMS,GPRS and integrated TCP/IP stack.
- Control via AT commands(GSM 07.07,07.05 and enhanced AT commands)
- Use AC – DC Power Adaptor with following ratings · DC Voltage : 12V /1A
- Current Consumption in normal operation 250mA, can rise up to 1Amp while transmission.

Interfaces

- RS-232 through D-TYPE 9 pin connector,
- Serial port baud rate adjustable 1200 to115200 bps (9600 default)
- BRK connector for MIC & SPK, SIM card holder
- Power supply through DC socket
- SMA antenna connector and Murata Antenna (optional)
- LED status of GSM / GPRS module

AT Commands

- AT Commands are used to control a modem. AT means Attention. Every command line starts with "AT". These are of two types : Basic and Extended.
- **ATEO – Echo off**
- **ATE1- Echo on**
- **ATD –Call a dial no.**

Syntax : ATD 9479555640

- ATDL- Redial last telephone no.
- ATA- Answer an incoming call
- ATH-Disconnect existing connection
- AT+CMGS-To send SMS

Syntax: AT+CMGS="9479555640" Press enter
Type text and press ctrl+z

- **AT+CMGR – To read SMS**

Syntax : AT+ CMGR=1 ; reads first SMS in sim card

- **AT+CMGD – To delete SMS**

Syntax : AT+CMGD = 1 ; deletes first SMS in sim card.

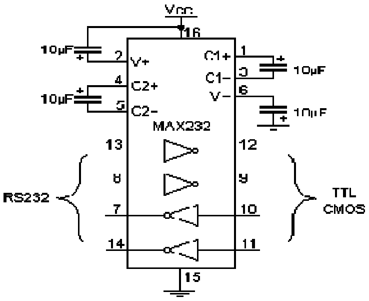
Hardware - The Arduino

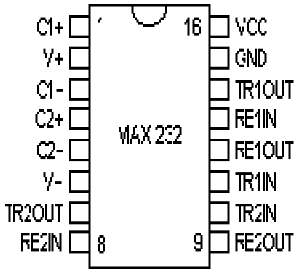
Arduino is a tool for the design and development of embedded computer systems, consisting of a simple open hardware design for a single-board microcontroller, with embedded I/O support and a standard programming language. An Arduino is a tool for making computers that can sense and control more of the physical world.

Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the Arduino programming language(based on Wiring) and the Arduino development environment (based on Processing).

RS 232 Circuit

Since RS232 is not compatible with microcontrollers we need a voltage converter to





Arduino software

The Arduino IDE is a cross-platform application written in Java which is derived from the IDE made for the Processing programming language and the Wiring project. It is designed to introduce programming to artists and other newcomers unfamiliar with software development.

It includes a code editor with features such as syntax highlighting, brace matching, and automatic indentation, and is also capable of compiling and uploading programs to the board with a single click. There is typically no need to edit Makefiles or run programs on the command line.

The Arduino IDE comes with a C / C++ library called "Wiring" (from the project of the same name), which makes many common input/output operations much easier. Arduino programs are written in C/C++, although users only need to define two functions in order to make a runnable program:

setup() – a function run once at the start of a program which can be used for initializing settings, and

loop() – a function called repeatedly until the board is powered off.

The code written in Arduino not be seen by a standard C++ compiler as a valid program, so when the user clicks the "Upload to I/O board" button in the IDE, a copy of the code is written to a temporary file with an extra include header at the top and a very simple main() function at the bottom, to make it a valid C++ program.

Since the IDE is pin oriented, we can quickly achieve our desired logic and build a working model

Functions used in Arduino

pinMode()

Configures the specified pin to behave either as an input or an output.

pinMode(pin, mode)

pin : the number of the pin whose mode you wish to set mode: either input or output.

digitalRead()

Reads the value from a specified digital pin, either HIGH or LOW.

digitalRead(pin)

pin : the number of the digital pin you want to read (*int*)

digitalWrite()

Write a HIGH or a LOW value to a digital pin

analogRead()

Reads the value from the specified analog pin. The Arduino board contains a 6 channel ,10-bit analog to digital converter.

This means that it will map input voltages between 0 and 5 volts into integer values between 0 and 1023. This yields a resolution between readings of: 5 volts / 1024 units or, 4.9 mV per unit.

analogWrite()

Writes an analog value (PWM wave) to a pin. After a call to **analogWrite()**, the pin will generate a steady square wave of the specified duty cycle.

The frequency of the PWM signal is approximately 490 Hz

Serial Communication

The Serial port (USART) is used for communication between the Sanguino board and a computer or other devices.

The Sanguino has two serial ports :

- Serial1 on pins 19 (RX) and 18 (TX)
- Serial2 on pins 17 (RX) and 16 (TX)
- Serial.begin(int baud)
- Serial.available( )
- Serial.read( )
- Serial.println( )

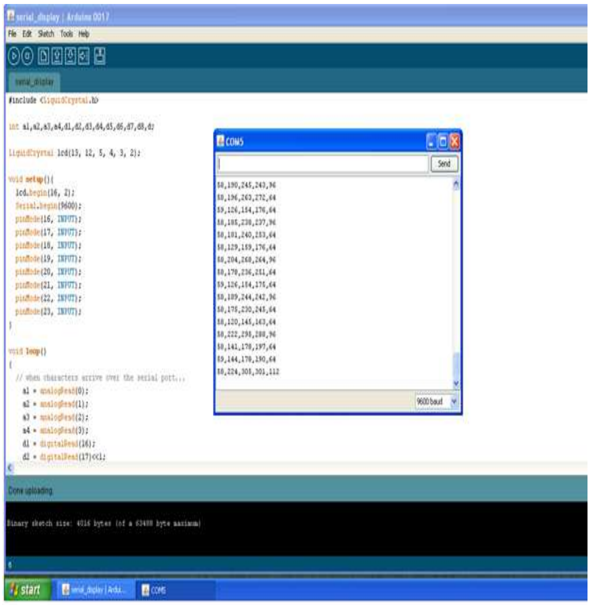
Conclusion

The design was successfully dissected into various modules. Each module with the exception of the database system has been implemented. The integration of the individual modules was achieved successfully.

1. The GSM Modem is tested using hyperterminal on pc and run the AT commands.
2. Read four analog channels and 8 bit digital and display it on serial port, LCD and send it through SMS
3. Receive SMS and display it on LCD.

4. Received SMS display on pc using VisualC#.net and form a database (MS Access) to store it.

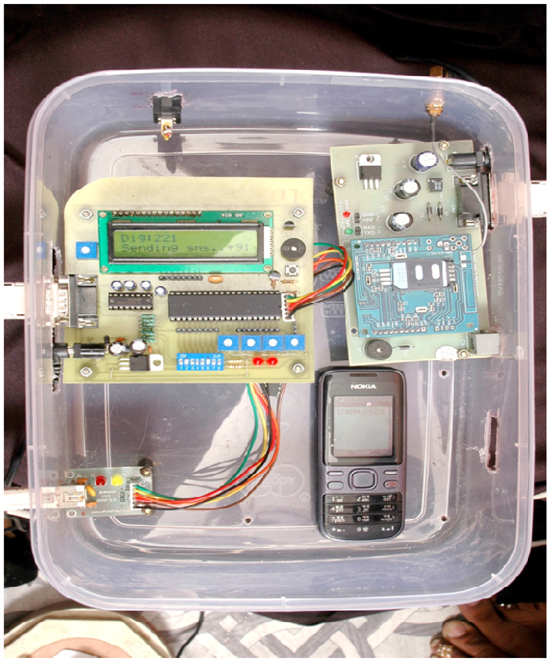
All the results were categorically measured. All the GUI functionality discussed in the report is operational

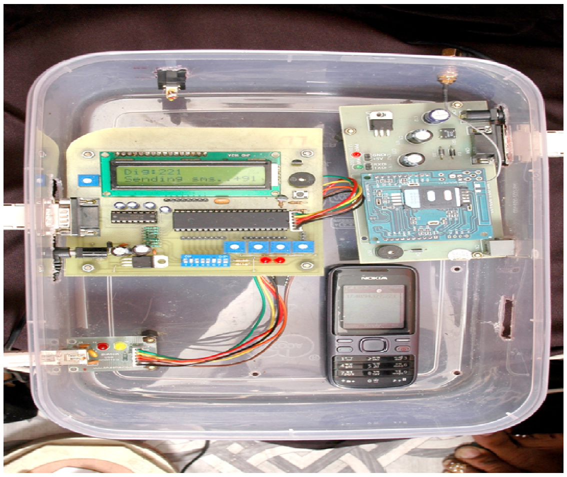


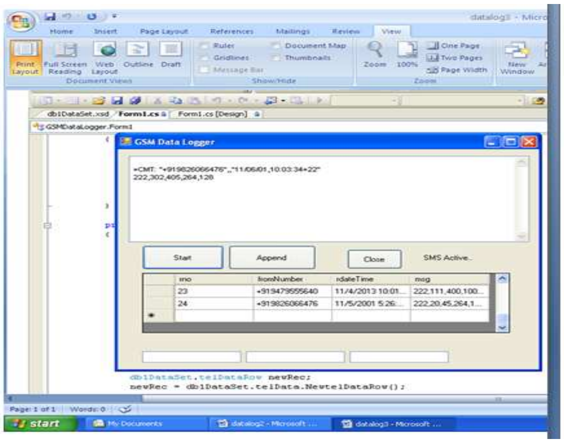
```
#include <LiquidCrystal.h>

int a1,a2,a3,a4,d1,d2,d3,d4,d5,d6,d7,d8,d;

LiquidCrystal lcd(13, 12, 5, 4, 3, 2);

void setup(){
  lcd.begin(16, 2);
  Serial.begin(9600);
  pinMode(16, INPUT);
  pinMode(17, INPUT);
  pinMode(18, INPUT);
  pinMode(19, INPUT);
  pinMode(20, INPUT);
  pinMode(21, INPUT);
  pinMode(22, INPUT);
  pinMode(23, INPUT);
}

void loop()
{
  // when characters arrive over the serial port...
    a1 = analogRead(0);
    a2 = analogRead(1);
    a3 = analogRead(2);
    a4 = analogRead(3);
    d1 = digitalRead(16);
    d2 = digitalRead(17)<<1;
```

```
58,190,245,243,96
58,196,263,272,64
59,126,154,176,64
58,185,238,237,96
58,181,240,253,64
58,129,159,176,64
58,204,268,264,96
58,178,236,251,64
59,126,184,175,64
58,189,244,242,96
58,175,230,245,64
58,120,145,163,64
58,222,298,288,96
58,141,178,197,64
59,144,178,190,64
58,224,308,301,112
```

```
Done uploading.

Binary sketch size: 4016 bytes (of a 63488 byte maximum)
```





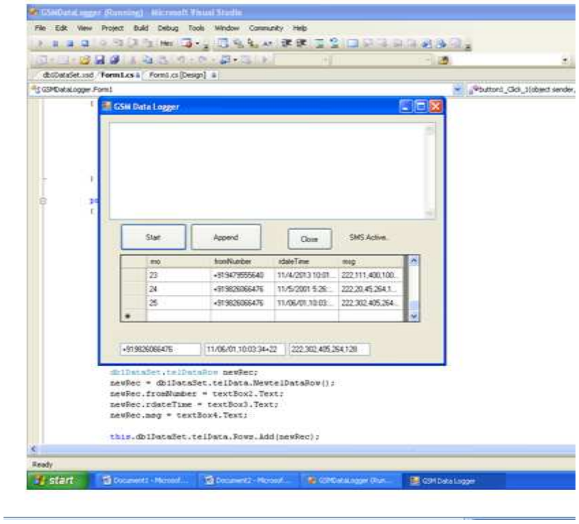

datalog3 - Micro
Home Insert Page Layout References Mailings Review View
Print Layout Full Screen Reading Web Layout Outline Draft
Document Views
Ruler Document Map
Gridlines Thumbnails
Message Bar
Show/Hide
Zoom 100% One Page Two Pages Page Width
Zoom
New Window
db1DataSet.xsd Form1.cs Form1.cs [Design]
GSMDataLogger.Form1
GSM Data Logger
Start Append Close
Page: 1 of 1 Words: 0
start My Documents datalog2 - Microsoft ... datalog3 - Microsoft ...

```
+CMT: "+919826066476",,"11/06/01,10:03:34+22"
222,302,405,264,128
```

SMS Active..

	rno	fromNumber	rdateTime	msg
	23	+919479555640	11/4/2013 10:01...	222,111,400,100...
	24	+919826066476	11/5/2001 5:26:...	222,20,45,264,1...
*				

```
db1DataSet.telDataRow newRec;
newRec = db1DataSet.telData.NewtelDataRow();
```


GSMDataLogger (Running) - Microsoft Visual Studio
File Edit View Project Build Debug Tools Window Community Help
db1DataSet.xsd Form1.cs Form1.cs [Design]
GSMDataLogger.Form1
button1_Click_1(object sender,
GSM Data Logger
Start Append Close
Ready
start Document1 - Microsof... Document2 - Microsof... GSMDataLogger (Run... GSM Data Logger

SMS Active..

	rno	fromNumber	rdateTime	msg
	23	+919479555640	11/4/2013 10:01...	222,111,400,100...
	24	+919826066476	11/5/2001 5:26:...	222,20,45,264,1...
	25	+919826066476	11/06/01,10:03:...	222,302,405,264...
*				

+919826066476 | 11/06/01,10:03:34+22 | 222,302,405,264,128

```
db1DataSet.telDataRow newRec;
newRec = db1DataSet.telData.NewtelDataRow();
newRec.fromNumber = textBox2.Text;
newRec.rdateTime = textBox3.Text;
newRec.msg = textBox4.Text;

this.db1DataSet.telData.Rows.Add(newRec);
```

References

1. www.arduino.cc
2. http://.www.bhasha.co.cc
3. Mobile Computing by Dr. Rajkamal Chapter 3
4. http://.www.bhasha.co.cc/pdf/sanguine-manual.pdf
5. http://www.developershome.com/sms
6. http://blog.nowsms.com/search/label/GSM %20Modem
7. www.ftdichip.com/driver/vcp.htm
8. http://www.sim.com
9. SMS Tutorial –How to use Microsoft hyperterminal to send AT commands
10. Hardware description of GSM modem simcom300 reference manual
11. Pdf : AT commands interface guide
12. http//www.iaeng.org/publication/WCECS 2008/WCECS2008-pp231-235.pdf
13. http://en.wikibooks.org/wiki/Serial_programming/MAX 232 Driver -receiver_
14. http://www.embetronics.com
15. http://en.wikipedia.org/wiki/Ssubscriber _Identity_Module