Design Requirement for the Prototype Version of the “AWSOM Base Station” by the Senior-Engineers’

Design Requirement for the Prototype “AWSOM Base Station”

Design Requirement for the Prototype "AWSOM Base Station"

1. Introduction

The 'Senior-Engineers' were approached on how to monitor a Hand Pumped Wells in a Rural Location.

There is a need to monitor hand pumped wells in rural locations without any connection to the water pumped or mechanical connection to the pump. In this rural location there is limited infrastructure but there is a GSM Service, so it was decided to monitor the Well Pumps by means of an Arduino Microprocessor with an attached GSM Telephone Module (SIM800L). Each Well Head Module transmits a Text SMS to a Base Station Module via the available Network Service Provider.

The aim is to maintain a web site that display the last fourteen days of each monitored well's activity on a Web Site. The Trials Data can be viewed at http://www.wellmonitoringservice.co.uk/ (follow the Maps trail).

The AWSOM System consists of four parts:

a. Well Head Module

The Well Head Module uses an Arduino Microprocessor with a Doppler Radar and GSM Telephone Modules connected. It is powered by a NiMH Battery that is recharged by a Solar Cell. The Arduino monitors the activity around the Well by means of the Doppler Radar, keeping a count of the number of 30 Second intervals in which activity is detected. By assuming the distribution of Men, Women and Children, this number can be converted into Litres by using an EXCEL Worksheet.

The Arduino monitors the output of the solar cell; on an interrupt as voltage is detected (sun shining), the "Day Number" is incremented by One; on an interrupt when no voltage is detected, a SMS is sent.

The SMS contains the "Doppler Radar Activity Number", the "Day Number" and other engineering data.

b. Base Station Module

The Base Station Module uses an Arduino Microprocessor with a GSM Telephone Module connected. It is Mains powered and is connected to a Workstation by a USB Cable.

The Base Station responds to commands serially transmitted over the USB Serial Link, with the reply being returned.

The Base Station can be controlled by using a "Serial Terminal Emulator", but the team have developed an EXCEL Workbook (enhanced with Visual Basic Code) to automate this activity.

2. An Automated EXCEL Workbook

This Workbook must be adapted for each individual application, some of the more obvious parameters are:

- FTP Account Details for the Website Server
- Identities of the Well Head Modules
- The Number of Base Stations controlled by the Workbook
- Details of the text to send to receive a Balance Enquire; how to deal with the Balance Text (With the SIMs used in the Prototype, it is necessary to send a SMS at least once every three months to maintain the Service).

If you would like to discuss how the EXCEL Workbook could help you, please contact the team at well.monitoring.team@outlook.com

3. Changes during the Development Phase (on the Prototype)

a. Phone Voltage Monitoring

The Schematic shows a resistor divider to monitor the SIM800L's Voltage on Arduino Pin A3. This was not implemented as the SIM800L's Command "AT+CBC"

b. Liquid Crystal Display

The method of announcing the progress was to use the inbuilt Window's SPEECH Application. It was felt that this could possible be intrusive, so provision was made to turn off these announcements and a "20 Char by 4 Line Serial Connected LCD" was incorporated into the design. The LCD is connected to the Arduino Pins A4 and A5.

4. Usage

The SIM800L that was used is a "Bare Die" mounted on to a Printed Circuit Board with a limited number of connections made available. Whilst the AT Command Set is extensive, the replies are not really capable of "Human Interpretation" so an Arduino is used to receive the Command and pass it to the SIM800L and then parse the reply into a suitable format.

a. From the “EXCEL Workbook”

The EXCEL Workbook will send the command to the Serial Output Buffer; wait until the reply has been processed and then read the Serial Input Buffer. This data is then processed.

b. “Standalone Usage”

The “AWSOM Base Station” can also be operated by a Serial Terminal Emulator. “Serial Debug Assistant” was available free from the WINDOWS 10 STORE, so it was downloaded and used. Please refer to the “Stand Alone Usage Guide” for more information.

5. Commands

a. ACK?

The number of the Base Station is stored as a constant within the code. The acceptable values are 1 to 4, this is due to the use of legacy code in the EXCEL Workbook. This is to enable up to four Base Stations to correctly function in different Workbooks on one Workstation.

b. VERIFY()

Returns current status of the Mobile Phone Module

I. Status

Where the command "AT+CGREG" is used to find the network registration status

Where the reply is “Code=n” where the values for ‘n’ are:

“0” means that no network is connected
“1” means that the phone is registered to a home network
“2” means that the phone is not registered to a network but trying to connect
“3” means that the registration is denied (this is assumed to be because that the SIM Card has expired)
“4” means that the network registration is UNKNOWN
“5” means that the phone is registered to a roaming network

II. subscriber number

Where the command "AT+CNUM" is used to find the subscriber number. If no subscriber number is found than it is assumed that there is no SIM Card fitted and VERIFY() will return “NO SIM”.

III. battery status

Where the command "AT+CBC" is used to find the voltage applied to the phone module (shown as millivolts)

IV. rssi

Where the command "AT+CSQ" is used to find the Received Signal Strength Indicator (RSSI) value from the phone module. Acceptable values are:

0	-115 dBm or less

31	-52dbBm or greater
99	not known or not detectable

V. ber

Where the command "AT+CSQ" is used to find the BIT Error Rate (BER) value from the phone module. Acceptable values are:

1	0% errors

7	40% errors
99	not known or not detectable

c. FIND(n)

The Arduino does not have enough capacity in memory locations to count (and read) all the available 70 messages stored in the SIM800L Module. It is not possible to use the command AT+CMGL="ALL" to count the total number of texts stored.

When texts are cleared, the remaining texts are not automatically reassigned into the now empty slot, this command must take this into account. This appears to happen if the SIM800L Module is powered down for some time.

This Command, FIND(n), reads the text number received by using AT+CMGR=n after setting the SIM800L Module to “Text mode” by using command AT+CMGF=1. The length of the reply is determined and if it is less than 25 characters long (short reply), it is determined to have been deleted and the next message is read. This is repeated until either 5 short replies have been read or a long reply has been read. The last reply is transmitted. Included in this reply is the number of the text. To optimise the time taken, the number of times the texts checked is limited to 5. It is planned for the EXCEL Workbook to then resend the command with the returned number until a long text is received. If the number of the text requested has not be stored, then an ERROR message will be received. This will enable all texts to be confirmed as deleted or available and read.

d. CLEAR(n)

This command has one passed parameter that is the number of the text that is to be cleared. Just enter it as a number, e.g. CLEAR(1), there is no imposed upper limit to the value of the passed parameter. The Phone’s memory will be cleared of the text related to the passed parameter.

e. SEND(telNo,text)

This command has two passed parameters that are (the mobile number that the text is to be sent too and the text to be sent separated by a “,” that the coding uses together with both brackets to read the parameters. The mobile number can either start with “0” or “+44” (or any other international code).

Send a text to the telephone number with the text as detailed in the passed parameters using:

a. “AT+CMGF=1” to set the phone module to text mode
b. “AT+CMGS” to send the text

f. NETLIGHT CONTROL

This command has a query as well as an action. The NETLIGHT informs the user as to the status of the SIM800L’s Network Status. It can be turned ON or OFF (when ON it ‘flashes’ dependent upon the Network Connection).

For the query; enter NETLIGHT? and send. The reply will be either NETLIGHT=0 or NETLIGHT=1

For the command; enter NETLIGHT(1) or NETLIGHT(0) and send. the reply will be either NETLIGHT=0 or NETLIGHT=1

g. COUNT()

A partial solution was found to count the number of messages in the Message Store was found. The command "AT+CPMS?" (Preferred SMS Message Storage) was used to interrogate the number of messages stored. The default configuration is to have all three stores configured to hold RECEIVED SMS’, but the reply will not count above 20, but is useful to know now many messages are stored. if the number read is '20' then the separator is changed from '=' to '>'

h. DISPLAY(line,text)

By using the Arduino Library <LiquidCrystal_I2C.h>, the logical way to display information is by an individual line. In the passed parameters, the 'line' refers to which line the text will be displayed; range is 0 to 3 and 'text' is the text to be displayed; the text is truncated at 20 characters.

i. LINE_CLEAR(line)

This will cause the passed line number to be cleared

j. ALL_CLEAR()

This will clear all the display