SciencePG
Science Publishing Group

A Microcontroller-Based Remote Embedded SCADA Control System With the Aid of a 32-Bit Microcontroller

Mostefa Ghassoul[*], Hajer Hafedh Ebrahim, Fatima Abdul Hussain Yusuf

Chemical Engineering, University of Bahrain, Manama, Bahrain

Email address:
mghassoul@uob.edu.bh (M. Ghassoul), hajar.hafedh@outlook.com (H. H. Ebrahim), fatemay@prudentsolutions.net (F. A. H. Yusuf)
[*]Corresponding author

Abstract: Supervisory control and data acquisition are well established control and visualization systems in industry. SCADA systems are DCS, PLC, soft PLC, industrial computer or microcontroller based. Some of those types of control are suitable for certain industries and are not for others. DCS is suitable in process control such as chemical, petrochemical, pharmaceutical industry and alike. PLCs are more suitable for food industry, beverages, car industry etc. Microcontrollers are emerging as a competitive option in small open SCADA system. All those types of control are based on data sharing using different types of networks such as Fieldbus, Profibus, Modbus and high speed ethernet. Those nets are interconnected using hard cabling which renders the communication distance very limited. One option is the use of public services such as internet but the risk is too high due to hacking, sabotage just to name few. This is demonstrated by the various conferences held over the world discussing protective ways against such practices. To overcome those problems including the cabling problem, one can call upon wireless communication with the aid of microcontroller and a SMART PHONE. This paper presents the design of a remote embedded SCADA system based on a 32 bit Microchip microcontroller. The controller has three main tasks to carry out. It controls a heat exchanger rig. It drives a multimedia graphic expansion board and it communicates with the SMARTPHONE using GSM where it interacts with the process either in normal mode or emergency mode. The proposed scheme has several advantages. It can communicate to any user independent of distance and time. It could be easily customized to allocate small and probably middle industry. On top of the cost effectiveness, the system could be expanded in local network to accommodate more complex applications.

Keywords: SCADA, PIC32 Microcontroller, Cell Phone, Remote Control, PID Controller, Heat Exchange, Machine to Machine

1. Introduction

Automation control has become the collarbone of any control system especially in large scale industries, where distributed control system (DCS), programmable logic controllers (PLC) and industrial computers (IP) play a pivotal role either in controlling the process or visualizing it, either locally or remotely. This has many advantages. Good quality of the product, competitive cost, high level of safety and a very good turn over in the long run. Unfortunately, due to its complexity and high cost, this type of control would not be convenient to control small applications. With the

tremendous technology development, a better way is to use a customized low-cost microcontroller based around which a SCADA system could be developed, with the use of public cell phone, where the microcontroller is used as a local controller and the cell phone as a remote supervisory system. Many good works in the area have been reported in literature. Das & Al. [1] proposed an authentication technique to safely control home appliances. Being authenticated, the cell-phone based system relays commands to a microcontroller, that takes over and performs pre-programmed tasks. Goel & Al. [2] developed an integrated wireless SCADA system for monitoring & accessing the performance of remotely situated device parameter such as temperature, pressure, humidity on

real time basis, whereas Ali & AL. [3.] developed a human-machine -Interface set up based on cellular phone to monitor temperature. Ali & Al. [3] developed an automated setup for the measurement of temperature of any industrial environment. They claim that the hardware requirements are significantly curtailed using the microcontroller's modern embedded features like A/D conversion, display interface, hardware and software interrupts, and communication protocols for interfacing cellular phones and GSM modules. Li & AL. [4] described the design and realization of GSM short message for alarming system through AT command. Security office can deal with emergency cases through finding information displayed on the screen of the mobile cell. Mbaocka [5] proposed a design of a system that dictates the presence of GSM signal from unauthorized user in restricted areas and districts that user from service. Wale & Al. [6] proposed a monitoring SCADA system to diagnose and forecast strategy of the functioning state of power system. Rehna & Al. [7] presented the design & implementation of low cost Cell Phone Detection System with a "Denial of Service" technique jamming. Cheng & Al. [8] presented a study where they integrated mobile phones, wearable devices, temperature and human motion detectors as smart sensors for enabling smart air conditioning control. Smart sensors obtain feedback, especially occupants' information, from mobile phones and wearable devices placed on human body. Sairam [9] has designed an intelligent mobile vehicle checking system implementing the functions of video capturing, intelligent plate distinguishing GPS positioning and wireless transmission. Irannejad & Al. [10] presented a device for remote sending the cathodic protection in formation of oil and gas pipelines and applying it to SCADA system via GSM. Malikamber & Al. [11] presented a paper which focuses on developing IEEE C37.1 Standard comprising of a combination of hardware and software to monitor and control various parameters in industry. The scope of the project extends from providing a system status on a single machine (usually PC) to a distant place using IEEE 802.3 LAN protocol and Web based access. The system is also capable of controlling devices in the industry. Mascare & Al. [12] developed a project to remotely monitor burner processes using SCADA. The process was monitored either locally or remotely with the aid of GSM in case operator not present on site or otherwise. He should be kept aware of ongoing process, faults occurring, the air fuel ratio values etc. A GE Fanuc PLC was used in the control process. Joshi & Al. [13] proposed a system work at remote location for supervisory control and data acquisition with the help of appropriate Programmable Logic Controller (PLC). Main aim to design automatic system implemented using ABB 564 Ethernet PLC having control builder plus and panel builder tools. Keeli & AL. [14] developed a remote SCADA system to manifest this real-time situation, a temperature recording system for a remote plant set-up is developed by Visual Studio. Temperature sensors are interfaced to a microcontroller. Information collected from the sensors is continuously sent over 2.4GHz trans-receiver wirelessly to

the microcontroller which is then received at the corresponding 2.4 GHz USB type trans-receiver connected to a PC / Laptop. In the computer "DAQ System" (software) is loaded which takes the collected data and presents them on PC / Laptop's front panel, and also logs them into the database. Shingre & Al. [15] developed a SCADA system to monitor and control several physical parameters based on cellular phone using GPRS or WAP to enhance the performance of the equipment. Ghadage & Al. [16] wrote a paper presenting an ANDROID phone as supervisory controller and PC used as base station to store data. Communication is set by blue tooth application protocol. Mc Connell & AL. [17] described and evaluated a telemetry system based on GSM mobile phone technology that may provide mark-recapture data for single year survivorship studies on gray seal on the costal area of Scotland. Ozdemir & AL [18] presented a paper on a Java-enabled mobile phone has been used as a client in a sample SCADA application in order to display and supervise the position of a sample prototype crane. The paper presents an actual implementation of the on-line controlling of the prototype crane via mobile phone. The wireless communication between the mobile phone and the SCADA server is performed by means of a base station via general packet radio service _GPRS_ and wireless application protocol _WAP_.

This paper describes the design of a remote SCADA system to control a heat exchanger, where the control is done locally using a Microchip microcontroller PIC32, and at the same time, it is interfaced to a UBLOX Machine to Machine (M2M) card. A SMART card is inserted into The M2M so that communication could be established with any cell phone using Global System for Mobile communication (GSM). To display the SCADA dynamic graphs and trends as well as soft buttons, a clever little multimedia expansion board (MEB) is interfaced to the microcontroller. The mimics of the heat exchanger are generated using EDRAW-MAX. To convert those mimics to machine language for the microcontroller to execute, a tool known as Graphics Display Designer (GDD) integrated in the Microchip MPLAB IDE platform is called upon to convert the dynamic graphics into a C subroutine. Another important advantage, that any control scheme such as MPC, adaptive, fuzzy, neural network could be developed in Simulink then converted into C files which could easily be fired into the microcontroller online.

2. Heat Exchanger Process

The proposed system was tested using a lab heat exchanger. The exchanger consists of a closed circuit primary hot water system (the heating fluid), supplying heat to a secondary water system (the process fluid) fed from the mains. A plate type heat exchanger transfers the heat from the primary to the secondary system. The primary source of hot water is a 70 liters sump tank fitted with two 3kW immersion heaters controlled by a thermostat. Water is pumped from the tank through the valve and flow meter system, thence through the heat exchanger and back to the sump tank for reheating. Two flow meters of the Rota meter (variable area)

type are fitted in the primary hot water system for monitoring the hot water flow. The Water circuit is shown in Figure 1. The hot water flowing into the exchanger is controlled by an electro-pneumatically operated control valve (V_c). A platinum resistance sensor of type resistance temperature dependent (RTD) measures the process fluid outlet temperature. This temperature is compared with the set point and the error is fed to a PID controller. The control variable is then converted to 4-20 mA which feeds the current to

pressure converter (I to P) whose output pressure controls the valve which in turn controls the flow.

A rise in the outlet temperature will produce an increase in the value of the current signal, and hence an increase in the control air pressure supplied to the valve. The control valve will thus partially close, decreasing the primary hot water flow rate, thus reducing the heat supply to the exchanger. A fall in the outlet temperature will increase the valve opening, thus more hot water.
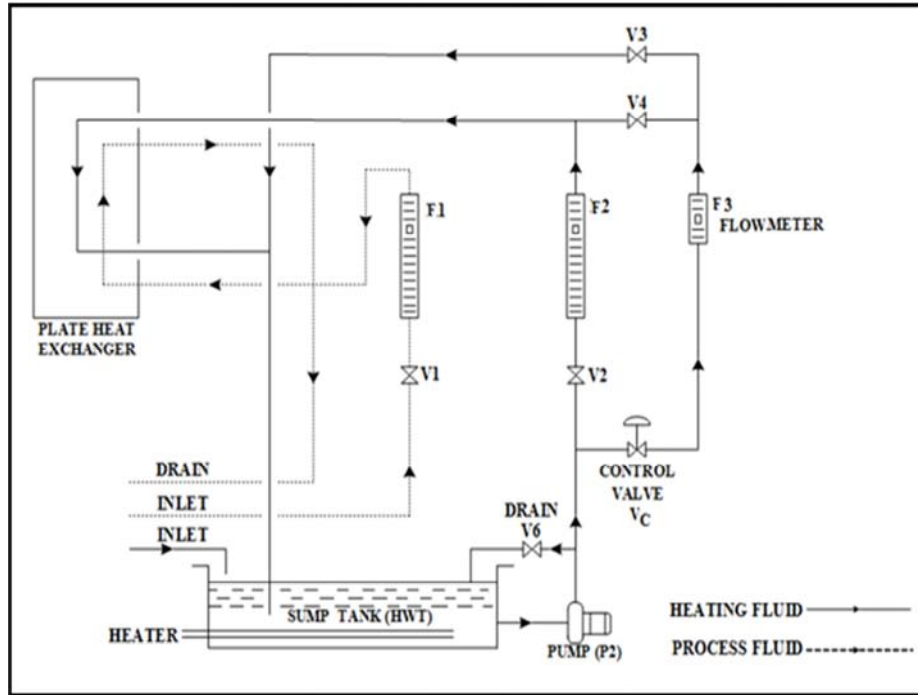

Figure 1. Schematic diagram of the heat exchanger.

3. System Hardware Design

As shown in Figure 2, the system is based on a32bit PIC microcontroller from Microchip, mounted on a starter kit. The controller has three roles to play. The first one is to control the rig, when set to automatic mode from the cell phone, where it reads the temperature at the outlet of the secondary circuit of the heat exchanger, which is picked up by an RTD connected across a 4mA current generator to produce an output voltage proportional to temperature. This voltage is fed to the microcontroller through one of the 10bit dual slope analogue channels where it is compared with the set point to produce the error signal. The error is fed to the PID controller. Microcontrollers do not generally have any digital to analog

converter which is required to drive the control valve. To overcome this problem, an external DAC is used. Two options are possible a parallel DAC or serial DAC. Parallel DAC is not good option because it requires two output ports. Those ports may be required for other applications. A better option is the use of serial DAC especially when high speed conversion is not required. The serial DAC is driven as a synchronous slave by the microcontroller which is configured as the master. In this mode, only three lines are required; a serial clock (SCL), a chip select $(\overline{CS})$ and serial data pin (SDI). The analog output is converted into current using voltage to current converter (V to I) where a 0 to 5V produces 4 to 20 mA. This current is applied to I to P converter which produces the standard 3 to 15 PSI required to drive the control valve.
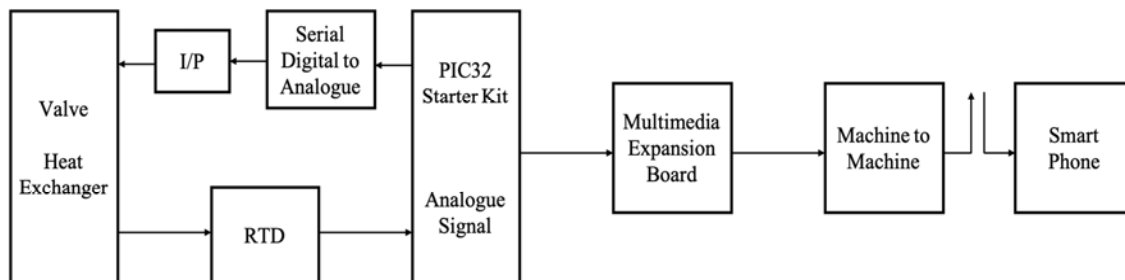

Figure 2. SCADA hardware design.

The following equation governs the control signal:

$$C_{out} = K(error + \frac{ACCerror}{\tau_i} + \tau_d(error - errorold)) \quad (1)$$

Where C_{out} is the control variable, K is the gain, error is the difference between the set point and the measured variable, ACCerror is the accumulated error, τ_i is the integral time constant, τ_d is the derivative time constant and errorold is the previous calculated error. This signal is outputted through an external 12bit serial converter which produces 0 to 5 volts. This voltage is then converted to a 4 to 20 mA current, which in turn drives a current to pressure converter which controls the control valve. The second role of the microcontroller is the communication with the cell phone through the M2M PICtail daughter board developed by U-BLOX for Microchip. This module is capable of sending short messages to any SMARTPHONE using different communication systems such as GPRS, GPS and GSM. For the application in hand, the latter was used to make calls or send short messages (SMS) in both ways. To ensure the security of the phone calls and SMS, GSM uses a Subscriber Information Modules (SIM) card inserted on the back slot of the PICtail board.

The third role is the SCADA which is based on the PIC32 microcontroller as a processing power interfaced to a Multi Media Expansion board. The Expansion board is used to display the dynamic mimics, the trends as well as the commands from the SMARTPHONE through soft buttons. The mimics are generated using EDRAW MAX. The heat exchanger mimic is shown in Figure 3. The mimic is then uploaded into MPLAB IDE, the PIC platform, using Graphic Display Designer (GDD) where it creates a PIC32 C language program which could be integrated into the main program, compiled and downloaded into the controller.
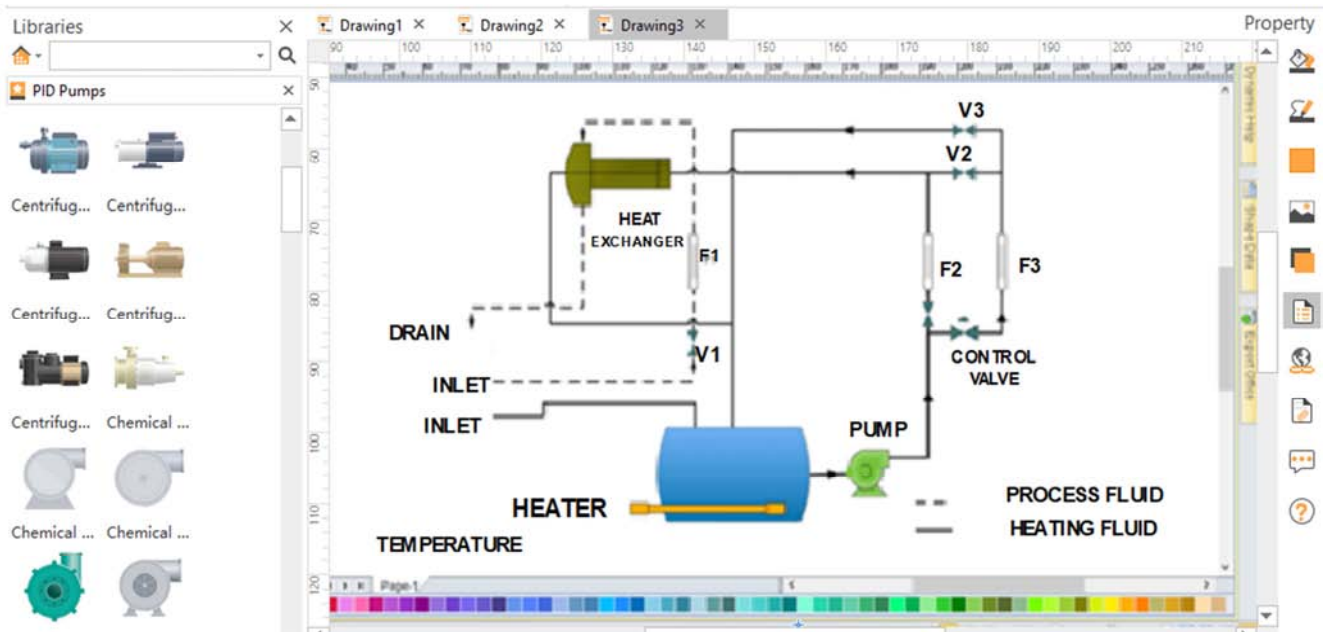

Figure 3. Heat exchanger Mimic in EDRAW MAX software.

4. A Word on Edraw Max

Edraw max program is a software which has numerous applications. It is provided with a set of libraries which can be used according to the application needed, for example it contains library with certificates templates for creating certificates, it has flow chart main symbols to make it easier to make one, and most importantly for our application, it has a set of the process units and instruments so a mimic diagram of a process can be developed. As seen in figure3, most of process control and measurement components such as valves, pumps, motors, meters, annunciators, switches etc. are all available in the software's library.

5. Program Flow

The running program is developed in Microchip C, compiled under MPLAB platform to carry out a series of tasks [19]. To start with, the work space contains all the libraries containing the functions and loops used to set up and control the operations of the peripheral modules. It contains functions to control the M2M and the MEB, *alongside with the PIC32. Those libraries are* divided into a number of subroutines. Figure 4 shows all the C subroutines generated by the compiler, where Figure 4a presents the MPLAB project worshop, Figure 4b presents the header files called in the main whereas Figure 4c shows the GSM configuration. It also shows the header files required by the compiler to first configure the microcontroller PIC32MX795

(C32_config_mx795.h) and configuration of UBLOX which is the module communicating with the smart phone. This module is based on the LEON-G200 which is a Quad Band GSM/GPRS data and voice module. Communications to the module are through AT commands. The UART module on the PIC32 device handles the AT commands. The LEON-G200 handles the GPS communications to the NEO-6Q module. To configure this module, header files libcom.h, libubx.h, libacd.h and libubx_config.h are called. The module also contains 1 MB of non-volatile memory that can be used for storing local or Internet files. At the start, GSM network is configured using command ubxGetGsmNetworkOperator ( ). The control parameters (heater, pump and alarm) were then declared as integer and set to initial value zero (figure 5a). Next the mobile phone number is inserted so the concerned person will be contacted. The soft buttons for tank level alarm, set point as well as pump and heater are created. Depending on the alarm acknowledgement by the operator through the smart phone, both the pump and heater are either controlled locally or remotely. If for some reason the operator does not consult his phone and does not acknowledge the alarm, the microcontroller takes action by turning them off locally (Figure 5b). The different control strategies are shown in Figure 5c. The SMS is sent out using instruction ubxSendGsmShortMessage (number, sms1) (figure 6a). If the level in the tank is low, this will be indicated on the mimic as red led and at the same time, it sends out an SMS message using the stored phone number (Figure 6b). After receiving the alarm, and On acknowledgement (Figure 6c), the operator turns off the alarm remotely and turns back on both the pump and heater remotely (Figure 6d).

(a)
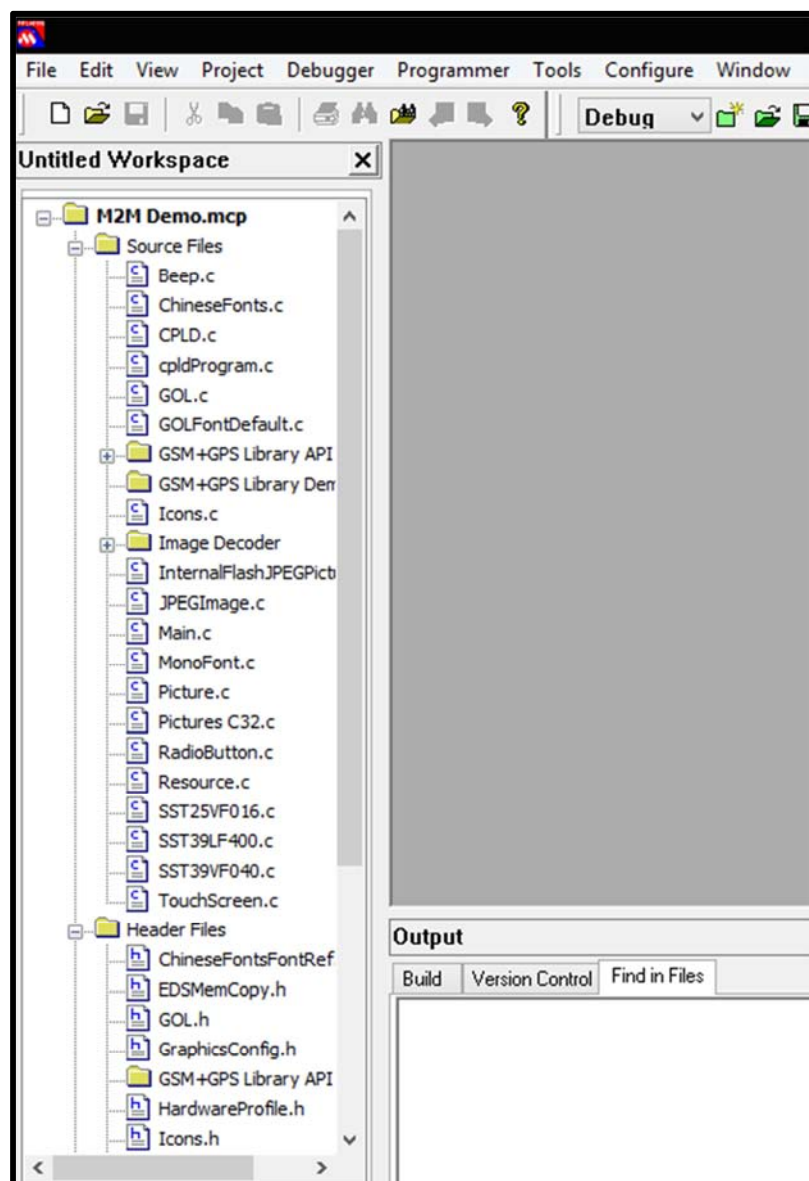
```c
// standard C libraries
#include <math.h>

// PIC32 peripheral library
#include <plib.h>

// GSM+GPS Library API
#include "c32_config_mx795.h"
#include "libp32.h"
#include "libcom.h"
#include "libubx.h"
#include "libacd.h"
 #include "libubx_config.h"

// GSM+GPS Library Demo
#include "Main.h"
#include "Fonts/ChineseFontsFontRef.l
#include "JPEGImage.h"
```

(b)

```c
{
int i,y;
static char temp[128];
int count=0;
static char buffer[COM_TX_BUFFER_SIZE + 1];
ubxConfigureBoard();
DelayMs(5000);    //Delay needed for proper GSM
ubxConfigureGsm();
ubxRegisterGsmNetwork();
if (ubxGetGsmNetworkStatus() == UBX_S_GSM_NETWORK_REGISTRED) {
                    while(1)  //Need to wait for GSM Service provider response
                    {
                            for(i=0; i<GSM_OPERATOR_STR_LENGTH-1; i++){stringTemp1[i]=(XCHAR)GSMoperatorStr[i];}
                            if(ubxGetGsmNetworkOperator(temp) == UBX_E_OK){
                            for(y=0; temp[y] != '\0'; y++){
                            stringTemp1[i]=(XCHAR)temp[y];
                            i++;
                            }

//                          if(count++ == 50) {OutTextXY(60,GetMaxY()-75,(XCHAR *)SimNotActiveStr); while(1);}

//                           OutTextXY(60,GetMaxY()-75, (XCHAR *)stringTemp1);
                            DelayMs(800); //Wait to try again

                            if(temp[y-1] > 73) break; //A charachter value was returned for a Servie Provider
                    }
                    }
                    acdSMSSetup();   //Setup SMS Reading
                    }
                    count=0;  //Reset the timeout counter
```

(c)

Figure 4. *GSM configuration Figure 4(a) MPLAB project workspace (b) Files included in main. (c) GSM configuration.*

```c
int alarm=0;
int alarm_update=0;
int pump=0;
int heater=0;
int yy=0;
int Hitemp_alarm=0;
int temp_update=0;
```

(a)

```
char number[25 + 1] = {'0','0','0','0','0','0','0','0'};
```

(b)

```
if (alarm==0 && yy== 1 && temp_update == 0){
 if (GetState(GOLFindObject(overview_pump), RB_DISABLED)){
    ClrState(GOLFindObject(overview_pump), RB_DISABLED);                    // reset disable
    SetState(GOLFindObject(overview_pump), RB_DRAW);
 }
if (GetState(GOLFindObject(overview_heater), RB_DISABLED)){
ClrState(GOLFindObject(overview_heater), RB_DISABLED);                      // reset disable
    SetState(GOLFindObject(overview_heater), RB_DRAW);
 }

if (pump==1){
if (GetState(GOLFindObject(overview_pump), RB_CHECKED)){
    ClrState(GOLFindObject(overview_pump), RB_CHECKED);
    SetState(GOLFindObject(overview_pump), RB_DRAW_CHECK);
pump=0;
 }
else {
SetState(GOLFindObject(overview_pump),RB_CHECKED| RB_DRAW);

pump=0;
 }
 }
if (heater==1){
if (GetState(GOLFindObject(overview_heater), RB_CHECKED)){
    ClrState(GOLFindObject(overview_heater), RB_CHECKED);
    SetState(GOLFindObject(overview_heater), RB_DRAW_CHECK);
heater=0;
 }
else {
SetState(GOLFindObject(overview_heater),RB_CHECKED| RB_DRAW);

heater=0;
 }
 }
TurnLEDOff(LED_5);
 }
```

(c)

Figure 5. *(a) initialization (b) Phone number definition of pump and heater (c) Control algorithm with different switching setting.*

```
switch(GetObjID(pObj))
{
    case ID_BUTTON_NEXT:
        if(objMsg == BTN_MSG_RELEASED)
        {
            switch(principalPage)
            {
            case START_NO_CONFIG:
                //ubxConfigureBoard();
                //principalPage = SELF_TEST;

                 sprintf(sms1, "low tank level");
            ubxSendGsmShortMessage(number, sms1);
 TurnLEDOn(LED_5);

alarm =1;
```

(a)

```
if (alarm==1){
SetState(GOLFindObject(overview_pump),RB_DISABLED | RB_DRAW);
SetState(GOLFindObject(overview_heater),RB_DISABLED | RB_DRAW);
pump=0;
heater=0;
}
```

(b)

```
if(strncmp ("A", SMSMessage,5) == 0){
 TurnLEDOff(LED_5);
alarm =0;
}
```

(c)

```
else if(strncmp ("P", SMSMessage,5) == 0){
 pump=1;
}
else if(strncmp ("H", SMSMessage,5) == 0){
heater=1;
}
```

(d)

Figure 6. *(a) Sending out an SMS message (b) Local control takes over and switches off both pump and heater (c) SMS acknowledgement (d) Turning pump and heater on.*

6. Results and Discussions

Two control schemes are implemented; normal mode and emergency mode. In emergency mode either the tank level is too low or the temperature is too high. In the case of tank level low, an emergency SMS is sent out saying "Low level". The user acknowledges it by sending back an "A" message. Then sends a "P" message. This toggles the pump operation between on and off. If the temperature is too high, the controller sends out an SMS message "High temperature". The user sends back an acknowledgement signal, and an "H" command to turn the heater off. In Normal mode, the control is either run automatically or manually. In the manual mode, an SMS message "M" (manual mode) is sent to the controller with the required valve opening ratio "Vxx" so that the valve opens accordingly. In automatic mode, the user sends a command "U" (automatic mode) together with the desired set point "S0000", then the microcontroller takes over using the PID to bring the temperature to the received set point. The scheme is shown in the flowchart below (figure.7), and it was tested in the following sequence:

The first step is to develop an HMI at both ends and connect several applications through a low level local network in a master slave topology; then establishing the wireless communication through the master. When the program was downloaded, the process graphics were on the MEB screen, along with three soft buttons, one for testing the alarming monitor system, which is the square button with the word (LEVEL) on, as well as the two red circles which are buttons to indicate if the device is on or off, and to be used for local control. The heater was locally turned on, the soft button for the heater was pressed, and to indicate that the

heater is on, a small white circle appears in the middle of the red circle. The pump was also turned on locally by pressing the pump circle. Similarly for the heater, a white circle appears to indicate that the pump is working (figure.8). The (LEVEL) alarm soft button was then pressed; the local alarm indicator was turned on (LED5) and at the same time, the pump and heater were both turned off, where the circles turned red again indicating that the pump and heater are both off (Figure 9a) . At the same time, an SMS is received on the phone cell, through the number that was previously stored in the program to indicate the emergency initiated in the process (Figure 9b). To notify the system that the person concerned has read the alarming SMS, an acknowledgment message is sent out to the GSM modem. When the modem receives the SMS, it passes it through to the PIC. The latter decodes it, and since the message is an acknowledgement one, the microcontroller turns off the alarm and the local alarm indicator (LED is automatically turned off indicating alarm acknowledgement. After the acknowledgement SMS, it is possible to remotely open the pump and heater, by sending two separate messages, one with 'H' to open the heater, and another with 'P' to open pump as shown in figure10.

To test the system even further, the alarm was again triggered, and an SMS was received on the cellphone; this time the alarm SMS was not acknowledged; but instead, an SMS for opening the heater was sent (figure 11a). The system received it, and the PIC decodes it, however no action was taken, and the system didn't react at all, instead as expected, the (LED5) was still on indicating the alarm, and the heater was still off and didn't open (figure 11b). The next step was to acknowledge the request again and test if the system will react to the sent message after the previous one and it did.
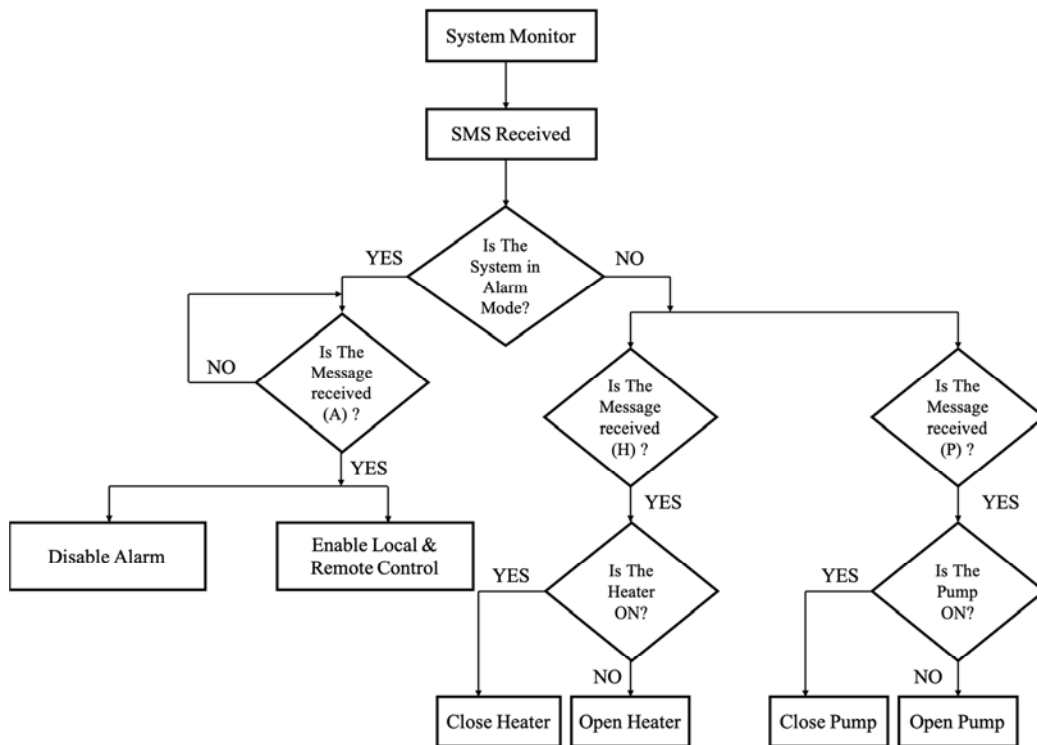
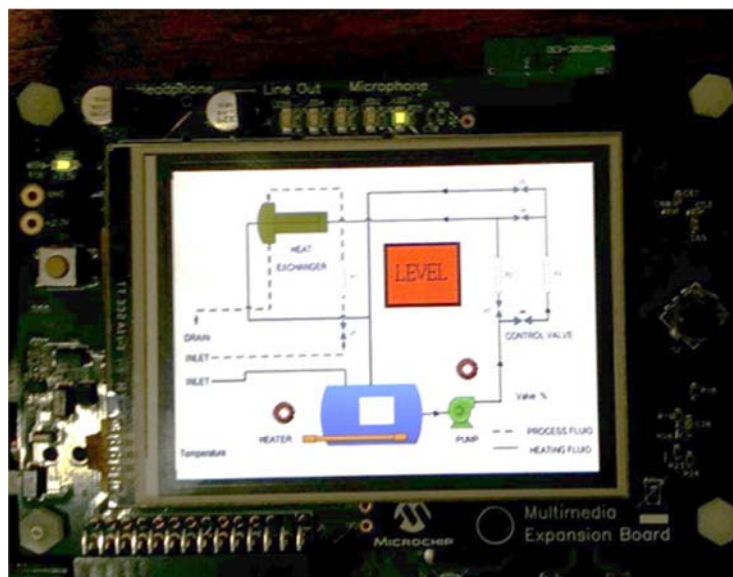**Figure 7.** *Flowchart showing the different steps of the process.*

Figure 8. *White circles in the graphics indicating the on state for the swith instruments.*
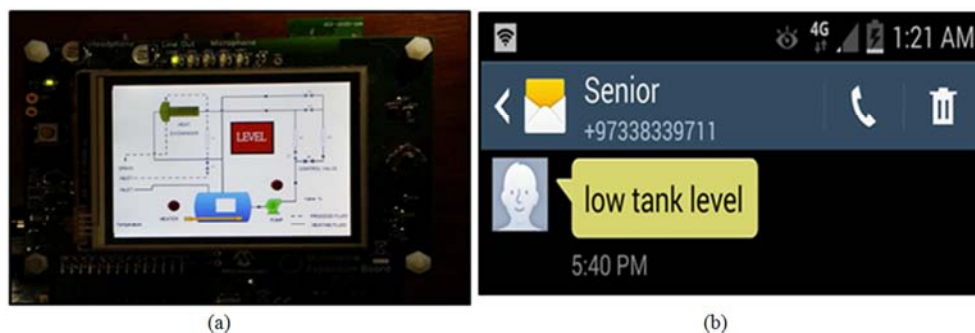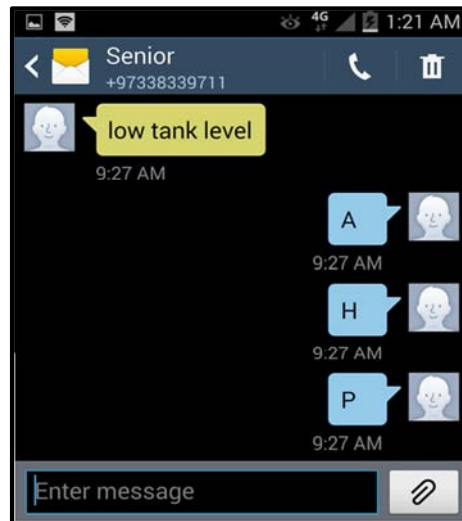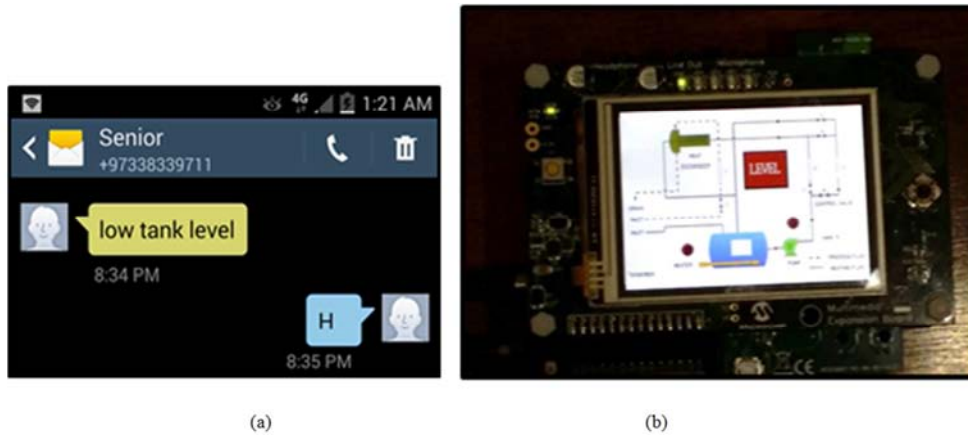

Figure 9. *(a) Local alarm indicator turned on (b) SMS was received on the phone number that was stored in the program.*

Figure 10. *Acknowledging the emergency & Remotely controlling the heater and pump.*


(a)   (b)

Figure 11. *(a) Command sent to system without acknowledgement (b) No action taken without the acknowledgement msg.*

At the beginning of the test, the control scheme was in automatic mode and is indicated by the letter (A) on the blue tank (Figure 12). After the phone cell receives this message, it turns the scheme to manual by sending the command "M" as shown on the right of Figure 13a, the command was followed by sending another SMS, "V25" to set the valve opening to 25%. The change immediately takes effect as shown in Figure 13b at the right, where it is indicated to the right of the pump.
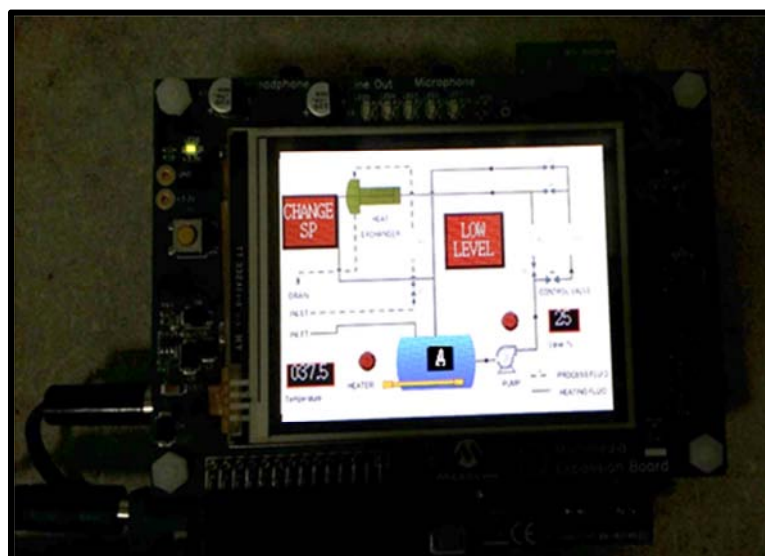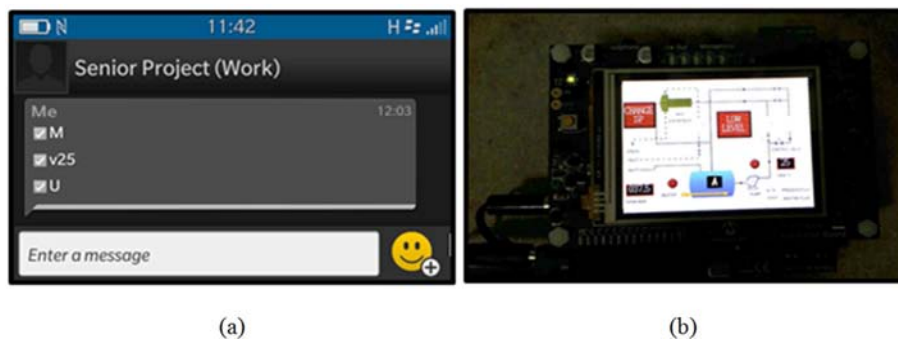

**Figure 12.** *Automatic mode control.*
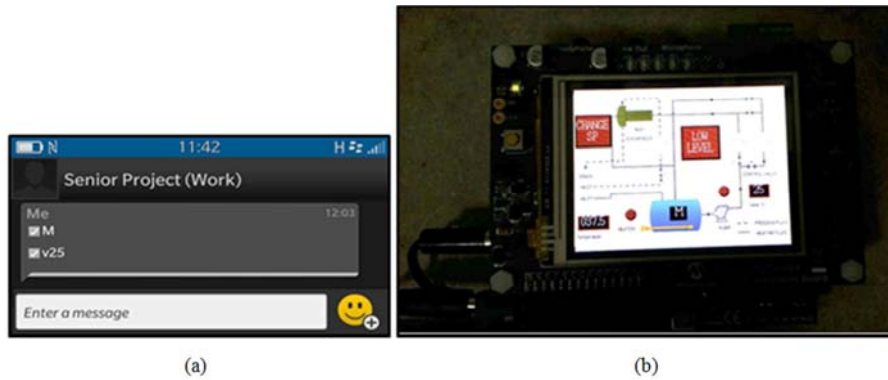
Figure 13. *(a) Manual controlling- setting the Valve at25% opening (b) Valve opened 25%.*

To check the automatic mode, an SMS command "U" was sent to the microcontroller (figure 14a). This is indicated on the mimic by the letter "A" on the blue tank (figure 14b) and the new set point is set to the desired value, in this case, it is 25.


Figure 14. *(a) Automatic control: SMS was sent to the control system (b) System control turned to automatic control mode.*

To test the emergency mode, the temperature was allowed to rise until it reached 80 degrees Celsius, the microcontroller sent out an SMS message to the cell phone indicating "High temperature" (Figure 15a), also at the same time the local alarm LED5 turned ON (Figure 15b). The user acknowledges the alarm and turns off both the heater and the pump. A similar test was carried out for "Low level" and it worked as expected. This is demonstrated in Figure 16.
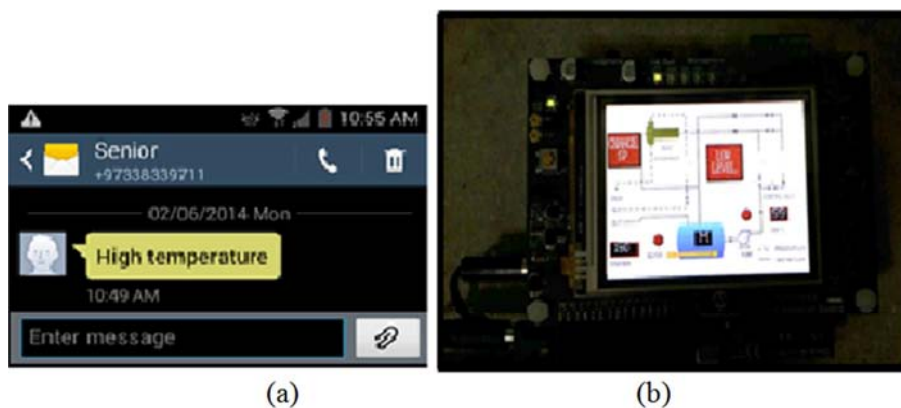
Figure 15. *(a) High temperature Alarm SMS (b) Alarm LED5 on.*
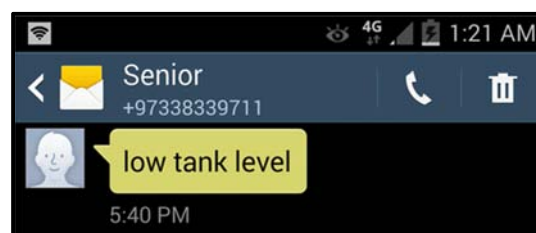
Figure 16. *Low level alarm SMS.*

7. Conclusions

It could be concluded that for small and perhaps medium industries, a remotely microcontroller-based SCADA system could be more appropriate to implement. This will not only reduce the control system's cost significantly compared with classical ones such as PLC or DCS, but it is also simple to implement and more secure as well, due to the fact that it uses a subscriber Information Module which cannot be easily invaded by intruders. The system is also capable of handling any emergency cases when operators are not available. The system could be extended by implementing any control strategy available in Simulink where microchip has provided a set of device block sets which allow the control scheme to be converted into C files which could easily be programmed into the microcontroller and online.

Nomenclature

SCADA: Supervisory Control And Data Acquisition GDD: Graphic Design Display

PIC32: 32bit Peripheral Interface Controller
MPC: Model Predictive Control
GSM: Global System for Mobile communication
RTD: Resistor Temperature Dependence
PID: Proportional Integral and Derivative controller
IC: Industrial Computer
DCS: Distributed Control System
A/D: Analog to Digital converter
PLC: Programmable Logic Controller
GPRS: General Packet Radio Services
MEB: Multimedia Expansion Board
I to P Current to Pressure converter
M2M: Machine to Machine
AT: ATtention command set
USART: Universal Synchronous Asynchronous Receiver Transmitter

References

[1] Das C. K., Sanaullah M., Sarower H. M. G. and Hassan M. M. "Development of a Cell Phone based Remote Control System: An Effective Switching System for Controlling Home and Office Appliances" International Journal of Electrical & Computer Sciences IJECS-IJENS Vol: 09 No 10, 2009, pp 23-29.

[2] Goel A. & Mishra R. S." Remote Data Acquisition Using Wireless - Scada System" International Journal of Engineering (IJE), Vol. 03, Issue 01, March2009, pp 58-65.

[3] Ali H., Ali A., Syed R. H., Khan A. & Khan I. "SCADA implementation of industrial temperature automation" International Journal of Computer & Network Security, Vol.11, No 08, PP.145-150, AUG. 2011.

[4] Li F. & Hu S. "The design and implementation of intelligent mobile message alarming system" 2nd International Conference on Electronic & Mechanical Engineering & Information Technology EMEIT-2012, pp 66-70.

[5] Mbaocha C. C. "Design and implementation of intelligent mobile phone detector" Academic Research International Vol. 03, No. 01, July 2012, pp 478-483.

[6] Wale S. S. & Patil S. B." GSM or X10 based SCADA system for industrial automation" International Journal of Research in Engineering and Technology, Vol. 02 Issue: 12, Dec-2013, 801-804.

[7] Rehna V. J., Kehveshan J. S., Hasrsha K. & Vinay V. " Cell phone detection and jamming system for GSM 900MHz and 1800MHz frequency band" International Journal of Advanced Trends in Computer Science and Engineering Vol. 03, No. 04, 2014, pp 29-32.

[8] Cheng C. & Lee D. (2014) " Smart Sensors Enable Smart Air Conditioning Control" Sensors 2014, 14, 11179-11203.

[9] Sairam K. " Design of Intelligent Mobile Vehicle Checking System Based on ARM7" International Journal for Scientific Research and Development Vol. 02, No. 05, 2014, pp 428-430.

[10] Irannejad M. & Iraninjad M. "Remote monitoring of oil pipelines cathodic protection system via GSM and its application to SCADA system" International Journal of Science & Research Vol. 03, No. 05, May 2014, pp 1619-1622.

[11] Malikamber M. A & Tamhankar S. G. " Implementing SCADA System for Industrial Environment Using 'IEEE C37.1' Standards" IEEE C37.1: I EEE Standard for SCADA and Automation Systems.

[12] Maskare S., Pendse S., Kamble P.& Kamble V. S. " GSM Based PLC-SCADA Burner Control System" International Journal for Scientific Research & Development, Vol. 3, Issue 01, 2015, pp 930-932.

[13] Joshi J. G. & Ahire D. D." Data Acquisition System Using Programmable Logic Controller" International Journal of Innovative Science, Engineering & Technology, Vol. 2 Issue 10, October 2015, pp 39-43.

[14] Keeli A., Teja E. P., Devi N., & Venkateswarlu G. "Wireless SCADA for Industrial Automation" International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering Vl. 04, Issue 04, April 2015, pp 2024-2030.

[15] Shingre N., Nagwevar R., Roy R. , Shendkar T. & Kaur R. "Review on mobile phone based SCADA for industrial automation" International Journal of Technical Research and Applications Issue 39 (KCCEMSR) (March 2016), PP. 1-4.

[16] Ghadage A., Kadam R., Kamathe P. & Bhuvad P. "Industrial UTOMATION USING MOBILE SCADA" International Journal of Technical Research and Applications, Special Issue 39 (KCCEMSR) (March 2016), pp 36-38.

[17] McDonnell B. & Beaton R. "Phoning hom e-A new GSM mobile phone telemetry system to collect mark-recapture data" Marine Mammal Science, Vol. 20, pp.274-283, Apr. 2004.

[18] Ozdenir E. & Karacor M. "Mobile phone based SCADA for industrial automation" ISA Transactions, Vol. 45, No 01, pp. 67-75, Jan. 2006.

[19] A. Folts "Using PIC32 MCUs to Develop GSM/GPRS/GPS Solutions" Microchip datasheet No AN1373, 2011, 1-34.