GSM BASED REMOTE TERMINAL UNIT OF THE FLOOD WARNING AND CONTROL SYSTEM

by

Zai Shyong KWAH

(UB No.06029418)

Submitted in partial fulfillment of the requirements for the degree of

Bachelor of Engineering (Hons) in

Electrical & Electronic Engineering 3+0

Supervisor: Mr. Kwong Chiew Foong

School of Engineering, Design and Technology

University of Bradford

January 2010
Declaration and Copyright

I, Victor Kwah Zai Shyong, hereby declare that the project entitled GSM Based Remote Terminal Unit of the Flood Warning and Control System report submitted for the INTI University College and University of Bradford, is my own original work and has not previously been submitted to any institutions. All authors quoted are indicated and acknowledged by means of a comprehensive list of references.

Victor Kwah Zai Shyong
“GSM Based Remote Terminal Unit of the Flood Warning and Control System” is the title of this project. This project is designed to give early flood warning where the water level of a certain place is monitored remotely. The water level data is then sent to the flood monitoring station via SMS. At the monitoring station, the warning will be given once the water level has achieved different level.

This system is named “Alerter”. By implementing one of the GSM technologies which is SMS, “Alerter” can be used by anyone at anywhere and anytime to monitor whatever you want on-time remotely, water level in this case.

This project enlightens the people how advance technologies are especially in communication that every single motion of everything can be monitored through GSM technologies. It proves no boundary to communicate between two different devices at two different places at the same time.

Human has evolved from time to time and to cope with the unpredictable disasters. Thus, technologies should always been used wisely to avoid the disasters laden on them. Tsunami has brought unnecessary lost of lives and property. Hence, this system is developed to reduce lose of unnecessary lives and properties.
ACKNOWLEDGEMENT

First of all, I would like to express my gratitude to my Project Advisor especially the leader who is Mr Kwong Chiew Foong and the members who are Mr Lai Tian Fat and Ms Zuraidah for spending their precious time getting thrown by tons of questions throughout the time.

Secondly, I would like to thank Munich from Wolfgang (http://www.wrankl.de/) for the guidance and answering my questions all over the time.

My many thanks to Zuraidah, previous BEEUB Head of Programme and Ms Chen Wan-Yu the current BEEUB Head of Programme.

Last but not least, I would like thank my parents for their constants and unconditional supports, and providing every single cent for everything I have today.
# Table of Contents

Declaration and Copyright ........................................................................................................... 2  
Abstract ....................................................................................................................................... 3  
ACKNOWLEDGEMENT ................................................................................................................. 4  
List of Figures ................................................................................................................................. 9  
List of Tables ................................................................................................................................. 9  
Chapter 1 ....................................................................................................................................... 10  
Introduction ................................................................................................................................. 10  
1.1 Project Description .................................................................................................................. 10  
1.2 Background Information ........................................................................................................ 11  
1.3 Project Specification ............................................................................................................... 12  
1.4 Problem Statement ................................................................................................................. 12  
1.5 The Sub problems ................................................................................................................... 12  
1.6 Delimitations .......................................................................................................................... 13  
1.7 Assumptions ........................................................................................................................... 14  
1.8 Report Organization .............................................................................................................. 15  
Chapter Two ................................................................................................................................. 16  
Literature Review ......................................................................................................................... 16  
2.1 Introduction ............................................................................................................................ 16  
2.2 GSM Background .................................................................................................................. 17  
2.3 GSM's History ....................................................................................................................... 17  
2.4 GSM's Architecture and it’s Essential Components ............................................................. 18  
  2.4.1 Mobile Station (MS) ......................................................................................................... 18  
  2.4.2 International Mobile Station Equipment Identity (IMEI) ............................................... 19  
  2.4.3 International Mobile Subscriber Identity (IMSI) ............................................................. 19  
  2.4.3 Mobile Subscriber ISDN Number (MSISDN) ................................................................. 20
2.4.4 Base Station Subsystem (BSS) ................................................................. 20
2.4.5 Base Station Controller (BSC) ............................................................... 20
2.4.6 Base Transceiver Station (BTS) .............................................................. 20
2.4.7 Network Switching Subsystem (NSS) .................................................... 21
2.4.8 Mobile Switching Centre (MSC) ............................................................ 21
2.4.9 Home Location Register (HLR) ............................................................. 21
2.4.10 Visitor Location Register (VLR) ............................................................ 21
2.4.11 Authentication Centre (AC) ................................................................. 22
2.4.12 Equipment Identity Register (EIR) ......................................................... 22
2.4.13 Operation and Maintenance Subsystem (OMSS) .................................... 22
2.5 GSM Services ......................................................................................... 23
2.5.1 Bearer Services .................................................................................... 23
2.5.2 Teleservices ......................................................................................... 23
2.5.3 Voice ................................................................................................... 24
2.5.4 Short Message Service (SMS) ............................................................... 24
2.5.5 Supplementary Services ..................................................................... 24
2.6 PIC16F876A .......................................................................................... 24
2.6.1 PIC16F876A Memory Organization ..................................................... 25
2.6.2 Flash Program Memory ....................................................................... 25
2.6.3 RAM Data Memory ............................................................................. 25
2.6.4 Data EEPROM (Electrically Erasable Programmable Read Only Memory) .................................................................................. 25
2.6.5 Input Output Ports ............................................................................... 25
2.6.6 Current in PIC ..................................................................................... 25
2.7 USART .................................................................................................... 26
2.8 RS232 ..................................................................................................... 27
2.8.1 RS232 DB9 Loopback ......................................................................... 27
Chapter 5 - Conclusion and Recommendations .......................................................... 64

5.1 Introduction ............................................................................................................. 64
5.2 Design Conclusions ............................................................................................... 64
5.3 Problems Encountered and Solutions ................................................................... 64
  5.3.1 Controller board .............................................................................................. 64
  5.3.2 LCD .................................................................................................................. 65
  5.3.3 DB-9 RS-232 connector ................................................................................... 65
5.4 Recommendations and Future Work .................................................................... 65
  5.4.1 PCB board ........................................................................................................ 65
  5.4.2 LCD .................................................................................................................. 65
  5.4.3 Modem ............................................................................................................. 65
  5.4.4 Water Level sensor .......................................................................................... 65
  5.4.5 VB6 software ................................................................................................. 66
5.5 Conclusion ............................................................................................................. 66

Reference ..................................................................................................................... 67

Appendix ....................................................................................................................... 70

  1. Circuit Board Schematics [1] ................................................................................ 70
  2. Sensor Circuit [16] ................................................................................................. 71
  4. Design at UV board .............................................................................................. 95
List of Figures

Figure 2-1: GSM’s Architecture [2] ................................................................. 18
Figure 2-2: Operation and Maintenance Subsystem [3, 18] ............................ 22
Figure 2-3: Bearer services, teleservices and supplementary services [3, 18] 23
Figure 2-4: RS232 Protocol [8] ...................................................................... 26
Figure 2-5: DB-9 pins in male and female connector [8] ............................... 26
Figure 3-1: Overall Approach [19] .................................................................. 31
Figure 3-2: Detailed Design of Sub Systems of the Whole System ................. 33
Figure 3-3: Connection Description of HyperTerminal ................................. 34
Figure 3-4: Setting the COM Port Number in HyperTerminal ....................... 35
Figure 3-5: Port Setting at HyperTerminal .................................................... 36
Figure 3-6: Example of sending AT Command ............................................... 37
Figure 3-7: Sensor Signal Amplifier Circuit Diagram and Simulation [16] ......... 39
Figure 3-8: Circuit Diagram for Voltage Regulator 7805 and its sub-components 40
Figure 3-9: Circuit Diagram for PIC16F876A and Its Sub Components [1] ..... 41
Figure 3-10: Circuit Diagram for Interfacing LCD with PIC16f876A [1] ......... 42
Figure 3-11: Circuit Diagram Interfacing MAX-232 and DB-9 with PIC16F876A [1] 44
Figure 3-12 - Flow Chart for Water Level Sensing System ......................... 46
Figure 3-12 - Flow Chart for Water Level Sensing System ......................... 48
Figure 4-1: Main Board of the “Alerter” ......................................................... 51
Figure 4-2: Outlook of Sensor System .......................................................... 52
Figure 4-3: Outlook of Whole System ........................................................... 53
Figure 4-4: Setting Up The Hardware ............................................................ 54
Figure 4-5: Response Sending AT Command Using Microcontroller .......... 55
Figure 4-6: Another Response Sending AT Command Using Microcontroller 56
Figure 4-8: Monitoring Comm. Port through Bus Hound .............................. 58
Figure 4-9: Results Showing Water Level at Different Area through LCD ...... 60
Figure 4-10: Results Showing Water Level at Different Area in Visual Basic 6.0 Environments-61

List of Tables

Table 2-1: DB9 Loopback Pins Configuration [8] ............................................ 27
Table 2-2: Assembly Code for Data Transfer [19, 22] ................................... 28
Table 2-3: Assembly Code for Arithmetic & Logic [10, 11, 19, and 22] .......... 28
Table 2-4: Common AT Command code [10, 11, 19, and 22] ....................... 29
Chapter 1

Introduction

There is nothing more difficult to take in hand, more perilous to conduct or more uncertain in its success than to take the lead in the introduction of a new order of things.

Niccolo Machiavelli (1469 - 1527), The Prince (1532)

1.1 Project Description

This project is to design and develop a flood warning system which is able to monitor the water levels at different areas using GSM technologies. This is as security and to protect the safety of the live stocks and peoples living beside the river or seas. There are many side usage through this system, which is not only flood monitoring. It can be designed to monitor the water level at places at sink and dam.

In this project, the system is divided into two categories which are hardware and software. The hardware covers 80% of the whole project while software covers 20%.

In order to provide a test-bed for the GSM Based Remote Terminal Unit of the Flood Warning and Control System, I have negotiated an agreement with MOBITEK System Sdn. Bhd. to provide a modem Q24 loan unit for this project.
1.2 Background Information

Prevention is better than cure. As we know that Malaysia is one of the monsoon countries which having Southwest Monsoon Rain from late May to September, and Northeast Monsoon Rain from November to March. Especially during the Northeast Monsoon Rain developed by the Siberian High and getting heated during moving from Northern Hemisphere to Southern Hemisphere, it brings heavy deadly rains which cause severe floods along the east coast states of Malaysia. The states affected are usually Kelantan, Terengganu, Pahang and East Johor in Peninsular Malaysia and also the east of Sarawak state. Therefore, a GSM Based Remote Terminal Unit of the Flood Warning and Control System is necessary to evacuate the people and live stock in time. [25]

There are many others technologies available to communicate with the water level monitoring system such as satellite technologies. However, compared to satellite monitoring, GSM technologies has better advantages by sending SMS to multiple recipients which is faster (more spontaneous), cheaper and more popular way. Therefore, crops, properties and live stocks can be saved in time.
1.3 Project Specification

This system was specified that it could:

1. Provide on-time communications
2. Can be used at anytime (flexibility), in anyplace (mobility), and by anyone (Easily Used)
3. Sending SMS
4. Low power usage
5. High Endurance

1.4 Problem Statement

The problem of this project is to design and build a system which can show the different water level at different areas remotely in the GUI.

1.5 The Sub problems

Sub problem 1

To come out with a flow chart with the main idea and then develop it’s detailed functions and protocols of each of constituents system.

Sub problem 2

To purchase components especially the modem that cost a fortune for a jobless undergraduate.

Sub problem 3

To test the circuits part by parts on the bread board and combines different parts together on breadboard
Sub problem 4

To combine the whole board and test its operations, performances and limitations

1.6 Delimitations

1. This project was limited to the design and the functions of the system which primarily focuses on the following aspects:

2. Mobility of monitoring water level

3. SMS monitoring

The study will not consider the following aspects:

1. Operating in rough situation

2. Places where no GSM coverage

3. Height accuracy
1.7 Assumptions

Assumption 1

It is assumed that the GSM Network Service provider will provide incessant good service throughout the time.

Assumption 2

It is assumed that SMS is delivered on time or no delay throughout the process.
1.8 Report Organization

This project report is the main documentation of the whole project. It contains technical data of the Alerter. This document composed of five different chapters.

Chapter One

This chapter gives an overview of the project. It discusses the functions, hypothesis, problems, specifications, delimitations, and assumptions made on this project.

Chapter Two

This chapter presents the literature review. It discusses the history, background, infrastructures and services of GSM. Others discussions in this chapter is PIC16F876A, AT Command, USART and Assembly Code.

Chapter Three

This chapter presents the system design and implementations. The design of the hardware and software is shown and the setting and programming is shown as well.

Chapter Four

This chapter presents the tests and its results of the output design of the whole system. It also covers the discussion on the test and its results.

Chapter Five

This chapter draws conclusion of each parts of the system and the combination of the system.
Chapter Two

Literature Review

Develop interest in life as you see it; in people, things, literature, music - the world is so rich, simply throbbing with rich treasures, beautiful souls and interesting people. Forget yourself.

Henry Miller (1891 - 1980)

2.1 Introduction

Human technologies have never halt evolving ever since human find its minor existence on this earth. Communications have never ceased its importance in alerting human of the dangerous lies ahead. Once, communications made between two areas are to protect their own ethnic group from being attacked by others. Humans gradually turn out to communicate in various ways and for different purposes in different languages at anytime, in anywhere and by anyone (not only human but even machines!) parallel with the rapid growth of human technologies. Therefore, it would be a waste to communicate without knowing the stories behind the development of communication. One of the vast contributions technologies in communications is none other than GSM technology.

In this chapter, literature review will be focused on GSM technology, its history, architectures, essential components, and its services. PIC 16f876A, RS-232, USART, Assembly Language, and AT Command will be discussed here too.
2.2 GSM Background

GSM also known as Global System for Mobile Communications is a set of ETSI standards specifying the infrastructure for a digital cellular service and initially developed and introduced in European countries during the late 80’s and early 90’s. Since GSM is the first known digital mobile telephony system, it has speedily gained recognition throughout the world, sharing portion of the market cake. It is estimated that 80% of the global mobile network market uses the GSM standard, except for Japan, which you can find nowhere GSM network in Japan. Now, GSM technologies has silently creep into over 3 billion peoples’ life and adopted in more than 212 countries and territories. [4]

2.3 GSM’s History

In 1982, GSM is known as Groupe Special Mobile which is formed by a group name called Conference of European Posts and Telegraphs (CEPT).

In 1987, Memorandum of Understanding (MOU) was agreed by 13 countries to establish the same telephony system across Europe.

In 1989, GSM responsibility was transferred to ETSI technical committee Special Mobile Group (SMG TC).

In 1990, GSM900 was frozen and DCS 1800 commences.

In 1991, first GSM Network launched and DCS1800 specifications finalized.

In 1992, SMS services were introduced and most networks go commercial.

In 1993, first roaming agreement was signed.

In 1994 and 1995, GSM Phase 2 was introduced.
In 1996, GSM was introduced in China and Russia. Prepaid-SIM was introduced as well.

In 1997, the first tri-band handset was produced.

In 1999, Wireless Application Protocol was introduced.

In 2000, GPRS goes commercial. [23]

2.4 GSM’s Architecture and it’s Essential Components

GSM can be divided into two categories which are Mobile Subscribers and the fixed built infrastructure. The fixed built network is then sub-divided into Base Station Subsystem, Switching and Management Subsystem and Maintenance Subsystem. [2]

2.4.1 Mobile Station (MS)

Mobile Station shortly is the combination of Mobile Equipment (ME) and Subscriber Identity Module (SIM). Together they provide various kinds of GSM services such as bearer services, tele-services and supplementary services. Each Mobile Equipment has its own unique identification which is known as International Mobile Equipment Identity (IMEI) while SIM has its own identification that is International Mobile Subscriber Identity (IMSI). [3, 18]
2.4.2 International Mobile Station Equipment Identity (IMEI)

The International Mobile Station Equipment Identity distinguishes each unique Mobile Station from others and is assigned to each of the Mobile Station by the equipment manufacturer. Then, they are registered by network operator and store them in the Equipment Identity Register (EIR). Through this, any reported stolen MS, obsolete or nonfictional equipment can be identified abruptly and thus denying services provided to the MS. IMEI can be divided into three categories within the EIR:

White List – The list which contains all the IMEI numbers of all equipment

Black List – The list which contains all the IMEI numbers of the suspended equipment. Network Operator exchanges this list periodically.

Gray List – The list which contains all the IMEI numbers which the malfunctioning equipment or the equipment which operating with obsolete software. The equipment however, still can be used but its use is reported to the operating personnel (end user).

[3, 18]

2.4.3 International Mobile Subscriber Identity (IMSI)

Each and every subscriber of the mobile network service provider will receive a unique identifier which is known as International Mobile Subscriber Identity. IMSI is stored in SIM and used to identify SIM when MS is operated. This is to prevent SIM card from being duplicated by having only one unique IMSI subscribe to the network operator at one time. This is to ensure the privacy of the mobile subscriber and correct billing for the services. [3, 18]
2.4.3 Mobile Subscriber ISDN Number (MSISDN)

MSISDN is the ‘phone number’ of a Mobile Station. Each SIM can have only one IMSI but can have multiple MSISDN (depending on the SIM) for the selection of different services (voice, data, fax and etc.) [3, 18]

2.4.4 Base Station Subsystem (BSS)

Base Station Subsystem is obligated for managing and maintaining the radio network and is controlled by an MSC. An MSC can control multiple BSS at a time. BSS covers a comparatively large geographical area which insides located many cells. BSS comprises of two, which is Base Transceiver Station (BTS) and Base Station Controller (BSC). BSS is providing the radio resources which are the radio channel allocation and quality of the radio connection. [3, 18]

2.4.5 Base Station Controller (BSC)

BSC is the central network component which controls the radio network. It establishes connection between the MS and the NSS (Network Switching Subsystem) and performs calling operation through the group switch of the BSC. Since the ME might changing from cells to cells, the BSC is responsible for most of the handovers, performing precise handover decision based on the calculations report sent by the MS during a call. The data from the BTS, Transcoders and BSC are collected in the BSC and forwarded to NMS to perform statistical calculations, and thus quality of network and status can be obtained. [3, 18]

2.4.6 Base Transceiver Station (BTS)

BTS is the hardware architecture responsible for managing the air interface and reducing the transmission problems since the air interface is very susceptible towards disturbances. Almost more than 120 parameters used in designing BTS to solve the problems such as handovers, paging, radio power level control, and identification of BTS. [3, 18]
2.4.7 Network Switching Subsystem (NSS)

Network Switching Subsystem is the switching centre where databases store the data needed for routing and providing service such as call control, charging information, mobility management, signaling and subscriber data handling. It is the combination of different network elements such as MSC, VLR, HLR, AC and EIR. [3, 18]

2.4.8 Mobile Switching Centre (MSC)

MSC is responsible of all the switching functions of a fixed network switching node. It performs call origin and destination identification for mobile station or fixed line and for different type of call. Besides, MSC identifies the destination, the origin of a call and also the type of call. It is also responsible for setting up, supervising, and clearing connections for every call. It initiates paging to locate a particular MS in case of mobile terminate or moving to another cell. It collects the charging data too. [3, 18]

2.4.9 Home Location Register (HLR)

HLR is responsible for storing every entry of every subscriber. Each mobile subscriber ISDN number has their respective “home” which is HLR. HLR stores relevant temporary information and permanent data of all subscribers permanently registered in the HLR. Besides, HLR also periodically tracks the location of the subscribers. In Nokia implementation, two of the network elements, which are Authentication Centre (AC) and Equipment Identity Register (EIR), are located in the HLR. [3, 18]

2.4.10 Visitor Location Register (VLR)

VLR is a database responsible for storing information about the subscribers currently locating in the service area of the MSC/VLR such as identification numbers (IMEI, IMSI and etc.) of the subscribers and the services that be provided to the subscriber. The information of the subscribers at VLR is temporary and is held only when the subscriber is within its
service area. The information stores temporary at VLR is such as the address to every subscriber’s HLR. [3, 18]

2.4.11 Authentication Centre (AC)
For security purposes, AC is created to provide security information to the network in order to verify the SIM across the interfaces. [3, 18]

2.4.12 Equipment Identity Register (EIR)
EIR is responsible for IMEI checking. When performing this operation, the MS will provide the IMEI number which consists of type approval code, final assembly code and serial number of the MS. EIR contains three lists which have been discussed above that are white list, black list and grey list. [3, 18]

2.4.13 Operation and Maintenance Subsystem (OMSS)

Figure 2-2: Operation and Maintenance Subsystem [3,18]

OMSS controls and maintains the ongoing network operation. OMC initiates and monitor s network control functions. Some of its functions are shown below:

1. Commercial and administration operation (end terminals, subscribers, charging, statistics)
2. Security Management

3. Network operation, configuration, and performance management

4. Maintenance tasks [3, 18]

2.5 GSM Services

GSM Services can be categorised into three parts which are bearer services, teleservices and supplementary services.

![Diagram of GSM Services](image)

**Figure 2-3: Bearer services, teleservices and supplementary services [3, 18]**

2.5.1 Bearer Services

Bearer services are a service which offers the fundamental technical capability for the transmission of binary data. GSM Bearer services can transport with circuit-switched or packet switched data rates of 300-9600 bit/s and 13kbit/s bearer service for voice. [3, 18]

2.5.2 Teleservices

Teleservices has many categories such as voice, fax transmission, Short Message Service, MHS access, Videotex access and Teletext transmission. [3, 18]
2.5.3 Voice

A teleservice make use the capabilities of a Bearer Service to transfer data, defining which capabilities are needed and how they should be set up. Teleservices can be divided into regular telephone service (TS11) and emergency service (TS12). Regular telephone service is the most basic teleservice supported by GSM while emergency service can be operated by dialing three digits without SIM. [3, 18]

2.5.4 Short Message Service (SMS)

Another teleservice which was given high priority in the service execution strategy, which is very successful, is the service which is able to send or receive short messages at the mobile station. [3, 18]

2.5.5 Supplementary Services

GSM supplementary services can be categorized into phase 1 and phase 2. GSM Phase 1 has introduced supplementary services such as call forwarding and call restriction (call barring). In GSM Phase 2, besides adding more functions in call forwarding, it introduces services such as number identification, community of interest charging, additional information transfer, call completion and multiparty conference calling. [3, 18]

2.6 PIC16F876A

MICROCHIP produces PIC16F87A which is programmable CMOS FLASH-based 8-bit microcontroller. It has 28 pins with A, B, and C port and operating in the voltage range from 2 to 5.5V. PIC16F876A features 256 bytes of EEPROM data memory, with an ICD, 2 comparators, 5 channels of 10-bit ADC, 2 capture/compare/PWM functions. Its synchronous serial port can be configured as either 3-wire Serial Peripheral Interface (SPI) or the 2-wire
Inter-Integrated Circuit (PC) bus and a Universal Asynchronous Receiver Transmitter (USART). [5, 6]

2.6.1 PIC16F876A Memory Organization
It can store up to 8K Flash Program Memory, 368 bytes Data memory and 256 bytes EEPROM Data Memory. [5, 6]

2.6.2 Flash Program Memory
Memory location which contains the program that had been written after we compile and burn into PIC using programmer. [5, 6]

2.6.3 RAM Data Memory
RAM memory type which is used to store Special Function Register (SPR) and General Purpose Register (GPR). This is the memory location to store the variable and will be deleted when we turn off the PIC. [5, 6]

2.6.4 Data EEPROM (Electrically Erasable Programmable Read Only Memory)
It is a non-volatile memory location which stores the variables after we turn off the PIC. [5, 6]

2.6.5 Input Output Ports
It allows you define every pin at the port as input output. Pins at the ports can be configured to be input by setting TRIS to 1 while configuring to output by setting TRIS register to 0. [5, 6]

2.6.6 Current in PIC
The output current of PIC can up to 25mA while the allowed input current is up to around 200mA. [5, 6]
2.7 **USART**

USART is a module used to interface with serial communication using MAX232 equivalent chip. [5, 6]
2.8 RS232

RS232 is an asynchronous communication protocol that lets you transmit and receive data between DTE (Data terminal Equipment) and DCE (Data Circuit Terminating Equipment) such as modem. It is the EIA/TIA (Electronic Industries Alliance/ Telecommunications Industry Association) that defines physical and electrical characteristics of the RS-232 interface.

RS-232 is active low voltage driven interface and operates at between -12V and +12V for which signal is LOW or 0 when voltage is higher than +3 Volt and signal is HIGH or 1 when voltage is lower than -3 Volt. For any voltage falling between +3 Volt and -3 Volts are considered ‘dead area’ or indeterminate value. TIA defines the power level for short circuit protection to be 100mA, however most RS-232 drivers will provide lower short circuit protection. [7, 8]

2.8.1 RS232 DB9 Loopback

Figure 1-4: RS232 Protocol [8]

Figure 2-5: DB-9 pins in male and female connector [8]
One of the methods of testing RS232 to confirm its functionality is by using RS232 Loopback. It works by testing each serial ends independently by using software such as HyperTerminal and Bus Hound. [8]

2.8.2 DB9 Loopback Pins Configuration

<table>
<thead>
<tr>
<th>DB9 pins no.</th>
<th>Signal Name</th>
<th>Loopback to</th>
<th>Signal Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>RD</td>
<td>3</td>
<td>TD</td>
</tr>
<tr>
<td>3</td>
<td>TD</td>
<td>2</td>
<td>RD</td>
</tr>
<tr>
<td>4</td>
<td>DTR</td>
<td>6,1,9</td>
<td>DSR, DCD, RI</td>
</tr>
<tr>
<td>7</td>
<td>RTS</td>
<td>8</td>
<td>CTS</td>
</tr>
<tr>
<td>5</td>
<td>SGND</td>
<td>5</td>
<td>SGND</td>
</tr>
</tbody>
</table>

Table 2-1: DB9 Loopback Pins Configuration [8]
2.9 Assembly Language

Assembly Language is a family of low-level languages used in programming computers, microprocessors, microcontrollers and others integrated circuits. Assembler (a program) is used to translate assembly language into targeted computer’s machine code and for example, MPASM MPLAB IDE, it translates the assembly coding into HEX CODE. [9]

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOVW f</td>
<td>Move W to f</td>
</tr>
<tr>
<td>MOVF f,d</td>
<td>Move f to the destination</td>
</tr>
<tr>
<td>CLRW</td>
<td>Clear W</td>
</tr>
<tr>
<td>CLRF</td>
<td>Clear F</td>
</tr>
<tr>
<td>SWAPF f,d</td>
<td>Swap nibbles in f</td>
</tr>
</tbody>
</table>

Table 1-2: Assembly Code for Data Transfer [19, 22]

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDDLW c</td>
<td>Add literal value of constant and W</td>
</tr>
<tr>
<td>ADDWF f,d</td>
<td>Add W and f</td>
</tr>
<tr>
<td>SUBLW c</td>
<td>Subtract literal value of constant from W</td>
</tr>
<tr>
<td>SUBWF</td>
<td>Subtract W from F</td>
</tr>
<tr>
<td>ANDLW c</td>
<td>AND literal value of constant with W</td>
</tr>
<tr>
<td>ANDWF f,d</td>
<td>AND W with f and store into the d destination</td>
</tr>
<tr>
<td>INCF f,d</td>
<td>Increment f and store into the d destination</td>
</tr>
<tr>
<td>DECF f,d</td>
<td>Decrement f and store into the d destination</td>
</tr>
</tbody>
</table>

Table 2-3: Assembly Code for Arithmetic & Logic [10, 11, 19, and 22]

2.10 AT Command

AT Command is a set of instructions used to control a modem. It is the abbreviation of ATtention. Each and every command of the set starts with ‘AT’. It is the instruction used to
control wired dial-up modem, but some of the AT Command instructions too, are supported by GSM/GPRS modems and mobile phone. By testing the response of the modem, ‘AT’ is sent to the modem and ‘OK’ will be responded by the modem. Some of the functions of the common AT command code are shown below:

<table>
<thead>
<tr>
<th>AT Command</th>
<th>The Functions of AT Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATD</td>
<td>Dial</td>
</tr>
<tr>
<td>AT+CGMS</td>
<td>Send SMS Message</td>
</tr>
<tr>
<td>AT+CMSS</td>
<td>Send SMS Message from storage</td>
</tr>
<tr>
<td>AT+CMGL</td>
<td>List SMS Messages</td>
</tr>
<tr>
<td>AT+CMGR</td>
<td>Read SMS Messages</td>
</tr>
<tr>
<td>AT+CSCA?</td>
<td>Service Centre Address</td>
</tr>
<tr>
<td>AT+CPMS</td>
<td>To choose storage from ME or SM</td>
</tr>
<tr>
<td>AT+IPR=0</td>
<td>To choose auto baud rate</td>
</tr>
<tr>
<td>AT+CMGF=</td>
<td>To choose PDU Mode or Text Mode</td>
</tr>
</tbody>
</table>

Table 2-4: Common AT Command code [10, 11, 19, and 22]
CHAPTER THREE

SYSTEM DESIGN AND IMPLEMENTATION

Design is directed toward human beings. To design is to solve human problems by identifying them and executing the best solution.

Ivan Chermayeff

3.1 INTRODUCTION

GSM Based Remote Terminal Unit of the Flood Warning and Control System, as the name indicates, it involves Mobile Communications and Control System. But designing this system requires more than that. Electronics Designation, Software Designation, Mechanical Designation disciplines too are required to produce the whole concept.

In this chapter, the conceptual design and the detailed design of both the hardware and software will be discussed. Besides, simulation of the sensors amplification circuits and also setting of modem will be discussed. [19]
3.2 CONCEPTUAL DESIGN

The aim of this project is to develop a transmitting system which can send SMS through the GSM network and develop another receiving system which can continuous read the modem’s message memory storage and display the incoming message on PC.

The first approach of overall is to develop a GSM based platform on PIC, and then building PC GUI interfacing with the GSM Modem, developing and programming the water level control system and finally construct an integrated structure to present them. [19]

Figure 3-1: Overall Approach [19]
The first step of overall is to find a suitable modem in this project. After different ways of approaching GSM Modem Suppliers and price comparing, MOBITEK Sdn Bhd generously loan a Q24 Modem for this project.

Considering the affordability, economical and the commonness of different type’s services provided by GSM services and GSM operators, SMS services are chosen instead of GPRS to continuously monitoring the water level. Serial modem unit is loaned instead of USB modem since it is more convenience and conventional way to interface with PIC and Terminal Equipment.

This GSM Based Remote Terminal Unit of the Flood Warning and Control System needs a powerful, reliable brand, moderate memory space and moderate number of ports microcontroller to control and manage the overall system. It is desired that the microcontroller have the USART compatible and able to be incorporated with LCD which is HD 44780 compatible to monitor the water level. Low power is another factor in choosing microcontroller and hence, PIC16F876A is chosen. RS232 is chosen because of the simplicity, easy to use, moderate data rate and the GSM Modem has a built in RS232 interface. ASC ll protocol has been chosen as the protocol interfacing with modem. [19]
3.3 Detailed Design

The more detailed design on the sub systems of GSM Based Remote Terminal Unit of the Flood Warning and Control System is discussed in this section. The detailed design can be explained by dividing them to 4 wide categories that is Mobile Communication, Electronics Circuit and Simulation, Programming Microcontroller and Software Developing. [19]

![Diagram of Detailed Design of Sub Systems of the Whole System]

**Figure 3-2: Detailed Design of Sub Systems of the Whole System**
3.3.1 Mobile Communication

3.3.1.1 GSM Module

GSM Modem is a wireless modem that is connected to GSM wireless network to sends and receives data through radio waves. It works in this project by providing Machine to Machine (M2M) wireless communication, which means data communications between machines. [19]

Q24 Modem is a Class 10 quad-band modem (850/900/1800/1900 MHz) with 32 Mb of Flash memory and 4Mb of SRAM(32/4). [13]

Q24 Modem can be controlled using a terminal program such as Hyperterminal.

The steps below show how to configure Modem to send SMS from the moment switching on mode.

After executing HyperTerminal.exe, the windows go to HyperTerminal environment where the Connection Description pops up.

![Connection Description of HyperTerminal](image)

Figure 3-3: Connection Description of HyperTerminal
Enter a name and choose an icon for the connection established for your modem in your own favor. This will save the setting of the connection established in the name you have given.

Figure 3-4: Setting the COM Port Number in HyperTerminal
Choose the COM Port that is connected to Q24 modem and click OK.

Figure 3-5: Port Setting at HyperTerminal
This is the setting for the Q24 modem when first switching on. The default baud rate is 115200. The baud rate can be changed after that to interface with PIC later. Click OK.

Figure 3-6: Example of sending AT Command

‘AT’ is sent and the response ‘OK’ shows AT command is supported

‘AT+CPIN?’ is sent and the response ‘+CPIN: READY’ shows PIN number does not needed and modem is ready to be used.

‘AT+CMGF=1’ is sent and the response ‘OK’ shows the mode of SMS has been changed to Text Mode.

‘AT+CPMS=”ME”,”ME”,”ME”’ is sent to change the memory used to read, write, delete, send or receive the SMS. In this case, Mobile Equipment’s memory is used.
The response ‘+CPMS: 0, 99, 0, 99, 0, 99’ and OK shows 0 memory space used and there are total 99 memory spaces to store 99 messages. [22]
3.3.2 ELECTRONICS CIRCUIT & SIMULATION

3.3.2.1 Sensor Circuit

The water will close the circuit of two copper wires placing apart and allowing small current flowing through. However, the current flow through is too small that it could not trigger the PIC pin port which require 5V to enable pin high. Thus, an amplifier circuit is designed using 2N2222A transistor to detect if the water closing the circuit.

3.3.2.2 Sensor Signal Amplifier Circuit Design

By assuming the water resistance is 1000k, the circuit is simulated using CircuitMaker Student Edition and the simulation result is shown as below.

Figure 3-7: Sensor Signal Amplifier Circuit Diagram and Simulation [16]
The current flow through water although is 4.990uA, but the output voltage is 4.999V when connecting in the water.

### 3.3.2.3 Power Supply Design

9V battery has been chosen to be the power supply since it is portable, commonly available and last long. Voltage regulator 7805 is used to regulate the 9V to 5V for which 5V supply is commonly used in LCD, PIC and MAX232 IC. 1uF capacitors here function as to reducing current ripple.

![Circuit Diagram for Voltage Regulator 7805 and its sub-components](image)

**Figure 3-8: Circuit Diagram for Voltage Regulator 7805 and its sub-components**

### 3.3.2.4 Microcontroller Design PIC16F876A

Majority of the program flows rely on the microcontroller and not many analogue systems in entire circuits. Pin 1 of PIC16F876A is the external Reset which is an active low reset. When given Reset pin is grounded, microcontroller will be reset and the instruction of the
program restarts from 0000 location. PIC14 and PIC16 can use high speed crystal up to 20MHz while PIC18 can use up to 10MHz. Port A and some of the port C pins (pin2 – pin7 and pin 11- pin13) are used as inputs of the water level sensor signals while Port B pins are used to configure and show display on LCD. Some of the Port C pins are used as transmitting and receiving data to Terminal Equipment through RS232.

Figure 3-9: Circuit Diagram for PIC16F876A and Its Sub Components [1]
3.3.2.5 LCD Circuit Design

In this LCD Circuit Design, JHD 162A which is HD44780 compatible, 16X2 display is used. The 4 bit data bus is adopted instead of 8 bit data bus to send and receive data from PIC. Besides 3 control lines which are EN, RS and RW are used to control the data in out. Pin3 is the contrast pin which is used to adjust the darkness of the display words. [15]

Figure 2-10: Circuit Diagram for Interfacing LCD with PIC16f876A [1]
3.3.2.6 RS MAX232 Circuit Design

3 DB-9 is connected to Max232 circuit here.

The left hand side one is designed for interfacing PIC with Terminal Equipment software such as HyperTerminal during to send data to PIC and getting respond back from PIC.

The right hand side db-9 connector is designed to plug in GSM Modem.

The middle db-9 connector is to be plugged in to monitor the communication between GSM Modem and PIC during the whole operation is running using HyperTerminal. This can be done by toggling the switch to 3.

To interface PC with modem using HyperTerminal, switch has to be toggled to 1. This is usually used when 1st time GSM Modem is started, the default baud rate is 115200 and baud rate has to be changed to 9600.

To configure USART transmission and receiving at PIC, Transmit Status and Control Register (Address 98h) (TXSTA) and Receive Status and Control Register (Address 18h)(RCSTA) are configured. Transmit Enable Bit (TXEN) and Synchronous (SYNC) mode is enable at the TXSTA while Serial Port Enable bit (SPEN) and Continuous Receive Enable (CREN) bit is enabled at RCSTA. [5]

One of the important calculations being performed during designing Max232 circuit is calculating the baud rate desire. The baud rate formula is as shown below

For BRGH=1 (High Speed), Baud Rate = Fosc/(16(X+1)) [5]

where Fosc is Frequency Oscillator and X is the value is SPBRG

The coding is provided in the Appendix behind.
Figure 3-11: Circuit Diagram Interfacing MAX-232 and DB-9 with PIC16F876A [1]
3.3.3 Programming Microcontroller

PIC 16F876A has been used as the microcontroller in controlling the sensing water level and transmitting and receiving data to RS232 to control the modem. MPASM Assembler and MPLAB IDE v8.36 has been used to compile the assembly coding to build hex file of the program. Programmer software PICKIT2 V2.55 is used to burn the hex file into the PIC16F876A using hardware CYTRON programmer.

3.3.3.1 Program Description

When the water reaches a certain level at the place, the current will be conducted and thus amplified, triggering the corresponding pin of the PIC port. The voltage level at the pin goes high, which is 5 volt. PIC recognize the voltage and will show the voltage level on the display, and send SMS notification to another terminal end.
Figure 3-12 - Flow Chart for Water Level Sensing System
3.3.4 Software Developing

Terminal Equipment on the other side receiving SMS from the Modem, needed to be displayed on the GUI. In this case, Microsoft Visual Basic 6.0 has been used to design the GUI to read the SMS periodically. In this design, timer function and MSCOMM function has been used.

3.3.4.1 Program Description

Before everything starts, the COM Port number assigned to the modem can be checked at Ports of Device Manager (Windows 7). HyperTerminal is opened to send AT+IPR=0 to enable auto baud rate so that modem can suit to the baud rate when interfacing with GUI.

The setting of COM port of the GUI before initializing it is as follows.

- **COM Port Number = 12**
- **Baud rate = 9600**
- **Parity bit = None**
- **Data Bits = 8**
- **Stop bit = 1**
- **Input Length = 0**
- **Handshaking = None**

After COM Port is opened, AT+CMGF=1 is sent to ensure the modem is in Text Mode while AT+CPMS="SM","SM","SM" is sent to ensure the modem stores and read SMS in SIM card. Then the GUI will loop until the 1st SMS sending to modem. The SMS will be stored at the 1st location of SIM card and will be displayed on the GUI. Soon after that the 1st location of SIM memory will be deleted by AT+CGMD=1 command. The program then loop back to continuously monitoring the modem of incoming message. [1]
Figure 3-13- Flow Chart for Receiving SMS Using VB GUI

1. Open Port
2. Loop until next SMS incoming
3. Read SMS from SIM location 1
4. Display on GUI
5. Delete SMS in location 1
Chapter Four

Test and Results

Results! Why, man, I have gotten a lot of results. I know several thousand things that won't work.

Thomas A. Edison (1847 - 1931)

4.1 Introduction

This chapter presents the output of the each individual hardware and software part designs and testing and discusses the results of the combination of the whole system. One of the most anticipating results is none other than the result of the integrating the whole system. This is because it will reveal the weakness of the project and a lot of critical issues surface then. Discussion will be on how to overcome those problems in this chapter too.
4.2 Design Outputs

In this section, the outlook of the project will be shown and each testing part of the hardware will be discussed.

The Controller board for Alerter

The Controller board performs the function of sending ASCII words to Modem, replying to the modem and performing loop operation to continuously check the water level.
Figure 4-1: Main Board of the “Alerter”
4.3 Sensor System Board

The sensor system board is as shown. It is responsible to amplify to the signals from the water sensor and amplify them to 5V.

Figure 4-2: Outlook of Sensor System
4.4 Integrated System

The outlook of the whole system after being integrated is as shown below.

Figure 4-3: Outlook of Whole System
4.5 Phases of experiments

4.5.1 First Phase - Interfacing GSM Modem with PC

The figure above shows how the experimental setup is done at the first phase. GSM Modem is connected to PC to interface with different software such as HyperTerminal, Bus Hound and others. Different AT Commands are tested at that phase.

4.5.2 Second Phase - Testing of sending AT Command using microcontroller in HyperTerminal Environment

AT Command demonstration below shows how microcontroller sends AT command and how modem respond to the command.
Figure 4-5: Response Sending AT Command Using Microcontroller
The AT command above shows how to read all the SMS from SIM.

The command above shows how to change to auto baud rate mode and acquiring the service centre number.

Figure 4-6: Another Response Sending AT Command Using Microcontroller
4.5.3 Third Phase- Testing of AT Command in the programme created in Visual Basic 6.0 environment

AT+CMEE=1
OK
AT+CMGF=1
OK
AT+CMGL="ALL"
+CMGL: 2,"REC READ"."+60165213793"."09/11/17.01:13:31-32"
Test2
+CMGL: 3,"REC READ"."+60165213793"."10/01/17.19:16:02-32"
Hello
Hello
+CMGL: 5,"REC READ"."66777"."10/02/28.19.12:58+32"
Pmbl- Serta barul Aktikan 3G+ sekarang! Sms ON CLUB ke 36787. Ekseuil utk DiGi. Pdtrm
PECUMA, Ringtone Ungu, Krispatih, Rosa & bvl. Ig. CS.012345678
+CMGL: 6,"REC READ"."+60165213793"."10/03/02.13:08:00-32"
Hello
+CMGL: 11,"REC READ"."+60165213793"."10/03/02.20:37:28+32"
Hello

Figure 4-7: Sending AT Command in Visual Basic 6 Environment

AT+CMEE command is used to change the error messages display mode.

AT+CMGF is used to select whether to operate in PDU mode which is 0 or operate in text mode which is 1.
Since we cannot monitor what Visual Basic program send to modem since only one port can be used by one program at one time, I have used Bus Hound Trial Edition to monitor what has been sent through the comm. port.

Figure 4-8: Monitoring Comm. Port through Bus Hound
4.6 Integrated Test

After assembling every parts of the project together, PC is powered up and HyperTerminal is executed in Windows environment. GSM Modem is power up and plug in to the DB9- to USB converter and thus plugged into USB port. At the Device Manager, check the port number uses. GSM Modem driver and USB to RS-232 driver are ensured properly installed. GSM Modem’s led light is ensured blinking and SIM card is properly inserted.

In HyperTerminal environment, baud rate, port setting, memory stored location are properly configured. Switch the switch at the board to change the mode to interface GSM Modem with Microcontroller. Plug out and connect the modem into the Controller Board of Alerter. Switch on the power switch of the controller board to continuously monitor the water sensors.

On the other hand, another GSM Modem is plugged into the PC and the setting above is configured as above. The Visual Basic Programme developed is executed and Initialise Button is pressed following by Read Continuously. This will monitor the SMS received every continuously.

Whenever the water level reaches a level, it will trigger the sensor and the signal is amplified at the amplifier system. Microcontroller responding to the signal will send the info to MAX-232 IC to amplify the data and thus send to the transmitter (modem). What receives at the other ends is then monitored by the GUI.
4.7 Results

What shows below is the LCD display when sensors sense the water at area 1 is at level 1 at the transmitting part and the GUI display when Modem receiving the SMS.

Figure 4-9: Results Showing Water Level at Different Area through LCD
Figure 4-10: Results Showing Water Level at Different Area in Visual Basic 6.0 Environments
4.8 Results Discussion

The Microcontroller successfully continuously sent the SMS at different water level to another Terminal Equipment and the SMS sent has been successfully continuously read. The use of SMS however, shows some delay at certain busy hour time, due to the unavailability and traffic busy of GSM network. This is avoidable since nothing is perfect in this world. We can only update the program to suit the changes.
Chapter 5

Conclusion and Recommendations

A conclusion is the place where you got tired of thinking.
Harold Fricklestein

5.1 Introduction
Designing a water level monitoring system using GSM technology and executing the system is the main objective of this project. The discussion of the designation on the system is presented in chapter 3 and execution of the combination system was discussed in chapter 4. In chapter 5, conclusion and recommendations will be drawn here.

5.2 Design Conclusions
A lot of different approaches has been undertaken to monitor water level through GPRS. However, in the current GSM-GPRS infrastructures, GPRS technology can only be implemented through IP protocol and there should be a server responding to the data sent. Terminal Equipment cannot directly connect to another Terminal Equipment via GPRS network. SMS is irreplaceable because of its low cost and effectiveness in sending information.

5.3 Problems Encountered and Solutions

5.3.1 Controller board
While designing the PCB boards, care has been taken to ensure the copper circuit line as wide as possible. However, during the test of the PCB board, data couldn’t be sent to
MAX232 IC and DB-9 connector because of the impedance matching. A few single core wires are connected so that the data can be sent over.

5.3.2 LCD

The holes designed sometimes not perfectly match to the LCD board and the LCD might display some characters error. The solution is to place the LCD on bread board and link the pins with single core wire.

5.3.3 DB-9 RS-232 connector

It is not possible to use PC to send and receive ASCII characters to modem while microcontroller interfacing with modem at the same time.

Hence, I have created a design with a switch which able to change from microcontroller interface modem with PC monitoring mode to PC interface with modem mode.

5.4 Recommendations and Future Work

5.4.1 PCB board

PCB board can be used double layer to save the spaces and it would be another good experience to design in double layer.

5.4.2 LCD

Four lines LCD will be another good try to allow more words display on LCD.

5.4.3 Modem

USB modem can be used instead of DB-9 connector modem.

5.4.4 Water Level sensor

Water Level Sensors can be made of capacitors circuits or using the pressure to measure the water levels since the deeper the water, the higher the pressure.
5.4.5 VB6 software

The GUI can be beautified and be refined to make it more user-friendly. Other software such as Microsoft VB.NET can be used as well.

5.5 Conclusion

GSM Based Remote Terminal Unit of the Flood Warning and Control System can be modified to control others home applications and as home security tool. The works on this project will not just stop here and will be further explored to have more functions on it.
Reference


Appendix

1. Circuit Board Schematics [1]
2. Sensor Circuit [16]

ROM.asm

;-------------------------------------------------------------
; ROM data storage
; file: ROM.asm
;-------------------------------------------------------------

Text00 EQU D'00'

Text00St
DE "Checking Water Level10" ;"X" will be overwritten by variable data

Text00End

MsgTab
:---- Message 0
DE HIGH Text00St
DE LOW Text00St
DE Text00End - Text00St
WATER_LEVEL.asm

; FILE NAME: WATER_LEVEL.asm

TITLE "GSM Based Remote Terminal Unit of the Flood Warning and Control System"
; set name of the project
processor 16f876 ; set type of chip
#include <p16f876.inc> ; include PIC16F876
ERRORLEVEL -302 ;
ERRORLEVEL -305 ;
ERRORLEVEL -206 ;

; CONFIG _CP_OFF & _PWRTE_ON & _BODEN_OFF & _WDT_OFF & _DEBUG_OFF & _HS_OSC &
; _LVP_OFF & _CPD_OFF & _WRT_ENABLE_OFF
#include MACDEF.asm ; macros definition

; constants for the microcontroller hardware

; constants for the memory
RAMPG0 EQU 0x20 ; start address of the RAM page 0
ERAMPG0 EQU 0x7F ; end address of the RAM page 0
CSRAM1 EQU 0x0A0 ; start address of the RAM page 1
CSRAM2 EQU 0x120 ; start address of the RAM page 2
CSRAM3 EQU 0x1A0 ; start address of the RAM page 3
CSRAME EQU 0x1EF ; end address of the RAM page 3
ROMPg3StartAdr EQU 0x1800 ; start address of ROM page 3

; Port Pins Declaration

IOA EQU PORTA
PORTA0 EQU 0
PORTA1 EQU 1
PORTA2 EQU 2
PORTA3 EQU 3
PORTA4 EQU 4
IOA_1 EQU 5
#define SENSE0 IOA, PORTA0
#define SENSE1 IOA, PORTA1
#define SENSE2 IOA, PORTA2
#define SENSE3 IOA, PORTA3
#define SENSE4 IOA, PORTA4
#define LED IOA, IOA_1

IOB EQU PORTB
DB4 EQU 0
DB5 EQU 1
DB6 EQU 2
DB7 EQU 3
RS EQU 4
RW EQU 5
E EQU 6
PORTB7 EQU 7
#define DiD4 IOB, DB4
#define DiD5 IOB, DB5
#define DiD6 IOB, DB6
#define DiD7 IOB, DB7
#define D1RS IOB, RS
#define D1RW IOB, RW
#define D1DE IOB, E

IOC EQU PORTC
PORTC0 EQU 0
PORTC1 EQU 1
PORTC2 EQU 2
PORTC3 EQU 3
IOC_SDA EQU 4
IOC_DTR EQU 5
IOC_TXD EQU 6
IOC_RXD EQU 7

#define SENSOR0 IOC, PORTC0
#define SENSOR1 IOC, PORTC1
#define SENSOR2 IOC, PORTC2
#define SENSOR3 IOC, PORTC3
#define SDA IOC, IOC_SDA
#define DTR IOC, IOC_DTR
#define TXD IOC, IOC_TXD
#define RXD IOC, IOC_RXD

; RAM Register Reservation

cblock ramp60 ; RAM PAGE 0

Reg1 ; Register 1
R_Area1 ; events of SMST, bit wise organized
R_Area2
R_Sensor1 ; Read of Register at Area 1
R_Sensor2 ; Read of Register at Area 2
R_LSensor1 ; Last Read of Register Sensors at Area 1
R_LSensor2 ; Last Read of Register Sensors at Area 2
R_Time ; Counter
R_TimeLow ; counter for low byte
R_TimeHigh ; counter for high byte
R_TimeInt ; Counter
R_WorkReg ; Working Register
R_WorkReg1 ; Working Register 1
R_WorkReg2 ; Working Register 2
R_WorkReg3 ; Working Register 3
R_WorkReg4 ; Working Register 4
R_Bytes ; number of bytes to be receive
R_RegLen ; universal length register
R_SrcPtr ; pointer to source
R_DestPtr ; pointer to destination
LCWord ; LCD Word Register

endc

TRBuff EQU CCSRAMP2&0XFF
BufferLast EQU CERAM2

#define Dest R_WorkReg, 1 ; input bits for CopyRAMPgX2PgY
#define NoDisplay R_WorkReg, 2 ; input bit for
ShowMessageOnDisplay/GetMessageForDisplay
#define _Event R_Area1, 0 ; 0: no sensor event occurred; 1: a sensor
event had occured
#define _Event2 R_Area2, 0

LEVEL0 EQU 0 ; 1: test button pressed
LEVEL1 EQU 1 ; 1: door sensor recognized an open door
LOCK
LEVEL2 EQU 2 ; 1: door lock sensor recognized an open door
LEVEL3 EQU 3 ; 1: power available recognized
LEVEL4 EQU 4 ; 1: water sensor recognized water

; flag declarations for the flag register
#define Bufferoverflow Regl, 0 ; 1: overflow of receive buffer had occured
C_RS232Timeout EQU D'10'; timeout for RS232 receive routine is ca. 1s
ORG 0x000; reset vector of the CPU
GOTO Main_Init; goto main routine
Main_Init
GOTO ClrRAM; set the whole RAM to 0
Main_RAMcleared
CALL CfgMuCtrl; set number of LED flashes
CALL FlashLED
CALL CfgDisplay
CALL ShowSwitchOnMessage
Mainloop
PAGESEL ReadSensors
CALL ReadSensors

;-------- an event had occurred and must be processed
CALL WLChek
CALL ShowLevelNArea

T_SendAT
RAMBank0
TLEDOff
PAGESEL D2s
CALL D2s
RAMBank2
CLRF TRBuff+0
CLRF TRBuff+1
CLRF TRBuff+2
CLRF TRBuff+3
RAMBank2
Move "A", TRBuff+0
Move "B", TRBuff+1
Move "C", TRBuff+2
Move "D", TRBuff+3
Move "E", TRBuff+4
Move "F", TRBuff+5
Move "G", TRBuff+6
Move "H", TRBuff+7
RAMBank0
Move D'8', R_Biyes
RAMBank0
PAGESEL SendStr
CALL SendStr
RAMBank2
Move 0x22, TRBuff+0
Move 0'0", TRBuff+1
Move 0'1", TRBuff+2
Move 0'6", TRBuff+3
Move 0'5", TRBuff+4
Move 0'2", TRBuff+5
Move 0'1", TRBuff+6
Move 0'3", TRBuff+7
RAMBank0
Move D'8', R_Biyes
RAMBank0
PAGESEL SendStr
CALL SendStr
RAMBank2
Move "7", TRBuff+0
Move "8", TRBuff+1
Move "9", TRBuff+2
Move 0x22, TRBuff+3
Move 0x0D, TRBuff+4
RAMBank0
Move D'5', R_Biyes
RAMBank0
PAGESEL SendStr
CALL SendStr
LEDOff
Move D'2', R_Biyes
PAGESEL ReceiveStr
CALL ReceiveStr
RAMBank2
Move "W", TRBuff+0
Move ":", TRBuff+1
Move ":", TRBuff+2
RAMBank0
Move D'3', R_Bytes
RAMBank0
PAGESEL SendStr
CALL SendStr
BTFSC R_Sensor1, LEVEL3
GOTO MSGLEVEL_3
BTFSC R_Sensor1, LEVEL2
GOTO MSGLEVEL_2
BTFSC R_Sensor1, LEVEL1
GOTO MSGLEVEL_1
BTFSC R_Sensor1, LEVEL0
GOTO MSGLEVEL_0
GOTO Continue
MSGLEVEL_3
RAMBank2
Move "3", TRBuff+0
Move 0x1A, TRBuff+1
RAMBank0
Move D'2', R_Bytes
RAMBank0
PAGESEL SendStr
CALL SendStr
PAGESEL R_LSensor1
CLRF R_LSensor1
GOTO Continue
MSGLEVEL_2
RAMBank2
Move "2", TRBuff+0
Move 0x1A, TRBuff+1
RAMBank0
Move D'2', R_Bytes
RAMBank0
PAGESEL SendStr
CALL SendStr
PAGESEL R_LSensor1
CLRF R_LSensor1
GOTO Continue
MSGLEVEL_1
RAMBank2
Move "1", TRBuff+0
Move 0x1A, TRBuff+1
RAMBank0
Move D'2', R_Bytes
RAMBank0
PAGESEL SendStr
CALL SendStr
PAGESEL R_LSensor1
CLRF R_LSensor1
GOTO Continue
MSGLEVEL_0
RAMBank2
Move "0", TRBuff+0
Move 0x1A, TRBuff+1
RAMBank0
Move D'2', R_Bytes
RAMBank0
PAGESEL SendStr
CALL SendStr
PAGESEL R_LSensor1
CLRF R_LSensor1
GOTO Continue
Continue
RAMBank2
Move "A", TRBuff+0
Move "R", TRBuff+1
Move ":", TRBuff+2
Move "1", TRBuff+3
RAMBank0
  Move D'4', R_Bytes
RAMBank0
  PAGESEL SendStr
  CALL SendStr
RAMBank2
  Move ":", TRBuff+0
RAMBank0
  Move D'1', R_Bytes
RAMBank0
  PAGESEL SendStr
  CALL SendStr
RAMBank2
  Move "W", TRBuff+0
  Move "L", TRBuff+1
  Move ":", TRBuff+2
RAMBank0
  Move D'3', R_Bytes
RAMBank0
  PAGESEL SendStr
  CALL SendStr
  BTFSC R_Sensor2, LEVEL3
  GOTO MSGLEVEL2_3
  BTFSC R_Sensor2, LEVEL2
  GOTO MSGLEVEL2_2
  BTFSC R_Sensor2, LEVEL1
  GOTO MSGLEVEL2_1
  BTFSC R_Sensor2, LEVEL0
  GOTO MSGLEVEL2_0
  GOTO FINISH
MSGLEVEL2_3
RAMBank2
  Move ":", TRBuff+0
  Move 0x1A, TRBuff+1
RAMBank0
  Move D'2', R_Bytes
RAMBank0
  PAGESEL SendStr
  CALL SendStr
  PAGESEL R_LSensor2
  CLRF R_LSensor2
  GOTO FINISH
MSGLEVEL2_2
RAMBank2
  Move ":", TRBuff+0
  Move 0x1A, TRBuff+1
RAMBank0
  Move D'2', R_Bytes
RAMBank0
  PAGESEL SendStr
  CALL SendStr
  PAGESEL R_LSensor2
  CLRF R_LSensor2
  GOTO FINISH
MSGLEVEL2_1
RAMBank2
  Move ":", TRBuff+0
  Move 0x1A, TRBuff+1
RAMBank0
  Move D'2', R_Bytes
RAMBank0
  PAGESEL SendStr
  CALL SendStr
  PAGESEL R_LSensor2
  CLRF R_LSensor2
  GOTO FINISH
MSGLEVEL2_0
RAMBank2
  Move "0", TRBuff+0
  Move 0x1A, TRBuff+1
RAMBank0
  Move D'2', R_Bytes
RAMBank0
  PAGESEL SendStr
  CALL SendStr
  PAGESEL R_LSensor2
  CLRF R_LSensor2
  GOTO FINISH
FINISH
RAMBank2
  Move "A", TRBuff+0
  Move "R", TRBuff+1
  Move ":", TRBuff+2
  Move "2", TRBuff+3
RAMBank0
  Move D'4', R_Bytes
RAMBank0
  PAGESEL SendStr
  CALL SendStr
  PAGESEL Mainloop
  GOTO Mainloop

Initialise
  ORG ROMPglStartAdr
  ; start address for the included program code
  ; locate the following code/data to page 3 of
  ; the ROM
  #include ROM.asm
  ; configuration of the microcontroller,
  #include Initialise
  ; data tables stored in the ROM
  #include CONFIG.asm
  ; delay routines
  #include DELAY.asm
  ; utilities for receiving and transmitting
  #include RS232.asm
  ; read ans evaluate the sensors
  #include READSENS.asm
  ; useful utilities
  #include LEDRAM.asm
  ; useful copy utilities
  #include CopyROMtoRAM.asm
  ; visualisation by use of the display
  #include DIS_VIS.asm
END
CONFIG.asm

;---------------------------------------------------------------
; Configure input/output pins for PIC16F876A
; file: CONFIG.asm
;---------------------------------------------------------------

; configuration of the microcontroller

CfgMuCtrl
;----- set input/output pin for port A
RAMBank1
BSF TRISA, PORTA0 ;input
BSF TRISA, PORTA1 ;input
BSF TRISA, PORTA2 ;input
BCF TRISA, IOA_LED ;output
BSF TRISA, PORTA3 ;input
BSF TRISA, PORTA4 ;input
RAMBank0

;----- set input/output pin for port A
RAMBank1
BCF TRISB, DB4 ;output, connect to display DB4
BCF TRISB, DB5 ;output, connect to display DB5
BCF TRISB, DB6 ;output, connect to display DB6
BCF TRISB, RS ;output, connect to display RS
BCF TRISB, RW ;output, connect to display R/W
BCF TRISB, E ;output, connect to display E
BSF TRISB, PORTB7 ;input
BCF OPTION_REG, NOT_RBPU

;----- set input/output pin for port A
RAMBank1
BSF TRISC, PORTC0 ;input
BSF TRISC, PORTC1 ;input
BSF TRISC, PORTC2 ;input
BSF TRISC, PORTC3 ;input
BSF TRISC, IOC_SDA ;input,
BCF TRISC, IOC_DTR ;output
BCF TRISC, IOC_RXD ;output for RS-232
BSF TRISC, IOC_TXD ;input for RS-232
RAMBank0

;----- configure USART
RAMBank1
Move 0x20, TXSTA ;configure transmission register
RAMBank0
Move 0x90, RCSTA ;configure receive register
RAMBank1

Move D'31', SPBRG ;baud rate 9600 bit/s at 20 MHz
BCF TXSTA, BRGH ;set baud rate generator to high baudrate

; Move D'64', SPBRG ;baud rate 19200 bit/s at 20 MHz
; BSF TXSTA, BRGH ;set baud rate generator to high baudrate
; Move D'20', SPBRG ;baud rate 57600 bit/s at 20 MHz
; BSF TXSTA, BRGH ;set baud rate generator to high baudrate

;----- configure timer 1
RAMBank0
BCF T1CON, TMR1CS ;internal clock
BSF T1CON, TCKPS0 ;prescaler to 8
BSF TICON, TCKPS1
BCF TICON, TSYNC ; synchronized to PIC clock
BCF TICON, TMRI0N ; deactivate timer
CLRF TMRIH
CLRF TMRI1
RETURN

; configuration of the JHD 162A

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

CfgDisplay
Move D'20', R_Time ; wait 20 ms
CALL D1Bms

MOVLW B'00000011' ; 8 bit interface
CALL Transmit8BitData

Move D'S', R_Time ; wait 5 ms
CALL D1Bms

MOVLW B'00000011' ; 8 bit interface
CALL Transmit8BitData

MOVLW B'00000011' ; 8 bit interface
CALL Transmit8BitData

MOVLW B'10000010' ; 4 bit interface, 2 line display, 5x7
dot display
CALL TransmitControlData

MOVLW B'01100000' ; set cursor move right, disable
display shift
CALL TransmitControlData

MOVLW B'00010000' ; clear display, set cursor at home
position
CALL TransmitControlData

MOVLW B'11000000' ; display on, cursor off
CALL TransmitControlData

RETURN

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;}
DELAY.asm

; Routines for different time
; File Name: DELAY.asm

D3s
  Move2 0x0B, 0xB8, R_TimeLow
  GOTO D2Bms
  ; wait 3 s

D2s
  Move2 0x07, 0xD0, R_TimeLow
  GOTO D2Bms
  ; wait 2 s

D1s
  Move2 0x03, 0xE8, R_TimeLow
  GOTO D2Bms
  ; wait 1 s

D500ms
  Move2 0x01, 0xF4, R_TimeLow
  GOTO D2Bms
  ; wait 500 ms

D250ms
  Move2 0x00, 0xFA, R_TimeLow
  GOTO D2Bms
  ; wait 250 ms

D100ms
  Move2 0x00, 0x64, R_TimeLow
  GOTO D2Bms
  ; wait 100 ms

D50ms
  Move2 0x00, 0x32, R_TimeLow
  GOTO D2Bms
  ; wait 50 ms

D2Bms
  DECF R_TimeLow, 1
  Move 0x7D, R_TimeInt
  ; timing adjustment

DmsLoop
  NOP
  CALL D2BmsRet
  CALL D2BmsRet
  CALL D2BmsRet
  CALL D2BmsRet
  CALL D2BmsRet
  CALL D2BmsRet
  CALL D2BmsRet
  CALL D2BmsRet
  DECFsz R_TimeInt, 1
  GOTO DmsLoop
  TSTF R_TimeLow
  BNZ D2BmsLoop
  TSTF R_TimeHigh
  BNZ D2BmsDecHigh
  RETURN

D2BmsRet
  RETURN

D2BmsDecHigh
  DECF R_TimeHigh, 1
  GOTO D2BmsLoop

D1ms
  Move 0x01, R_Time
  GOTO D1Bms
  ; wait 1 ms

D1Bms
  MOVF R_Time, 0
  ; check ISR 10 ms prescaler
DELAY.asm

BZ    D18msEx ;branch if ISR 10 ms prescaler is 0
DECF  R_Time, 1 ;decrement ISR 10 ms prescaler
Move  D'125', R_TimeInt ;timing adjustment

D1msLoop
CALL  D18msEx
CALL  D18msEx
CALL  D18msEx
CALL  D18msEx
CALL  D18msEx
CALL  D18msEx
CALL  D18msEx
CALL  D18msEx
DECFSZ R_TimeInt, 1
GOTO  D1msLoop
BNZ   D18ms ;branch if ISR 10 ms prescaler is 0
RETURN

D18msEx
RETURN

D150us
Move  0x7C, R_Time
GOTO  D1B

D200us
Move  0xA4, R_Time
GOTO  D1B

D300us
Move  0xF8, R_Time
GOTO  D1B

D1B
D1BLoop
TSTF  R_Time
BZ    D1BRet
DECF  R_Time, 1
GOTO  D1BLoop

D1BRet
RETURN
RS232.asm

; Routines for receiving and transmitting of data over RS232 serial line with
; UART support
; file: RS232.asm

; Sending a string via RS232

SendStr
  RAMBank0
  Move TRBuff&0xFF, FSR ; set the FSR pointer to the first byte of
  the tx buffer
  BSF STATUS, IRP ; set RAM bank 2/3

SendLoop
  MOVFW INDF ; INDF=>W
  MOVWF TXREG ; W=>TXREG

RAMBank1
  ; RAM bank 1

SendRegNotEmp
  BTFSS TXSTA, TRMT ; bit test file TXSTA, Skip the next
  command if TRMT is set
  GOTO SendRegNotEmp ; direct to SendRegNotEmp

RAMBank0 ; RAM bank 0
  ; Move 0x28, R_Time
  Stopbit = 50us
  Move 0x20, R_Time
  Stopbit = 50us

Call DELB ; call delay

INCF FSR, F ; increment tx buffer pointer
DECF R_Bytes ; decrease R_Bytes by 1
BNZ SendLoop ; Go back to SendLoop if the minus

RETURN

; Receiving string using RS232

ReceiveStr
  BCF Bufferoverflow ; reset bit for rx buffer overflow

SetRecPtr
  Move TRBuff&0xFF, FSR ; set FSR pointer to the first byte of the
  rx buffer
  BSF STATUS, IRP ; RAM bank 2 or 3

RCLoop
  ReceiveLoop
  BTFSS PIR1, RCIF ; bit test PIR1, RCIF file, skip the
  command if the file is set
  GOTO ReceiveLoop ; go back to Receiveloop

MOVFW RCREG ; RCREG=>W
MOVWF INDF ; W=>INDF
DECF R_Bytes ; R_Bytes - 1=> R_Bytes
BTFSC STATUS, Z ; Bit test the minus operation above, skip
the next command if it results zero
GOTO AllBytesReceived ; go to AllBytesReceived

INCF FSR, 1 ; FSR+1=> FSR
MOVF FSR, 0 ; FSR=>W
SUBLW (BufferLast+1) & 0xFF ; BufferLast+1-W
BTFSC STATUS, Z ; Skip Next command if the minus operation
above is doesn't give zero
GOTO BufferOverflow
GOTO RCLoop
BufOverflow
BSF Bufferoverflow
GOTO SetRecPtr
AllBytesReceived
RETURN
; read all sensors and store the sensor values in RAM
; file: READSENS.asm

READSENS.asm

;----------------------------------

; ReadSensors

BCF R_Sensor1, LEVEL0 ; set prophylactic test button is not pressed
BTFSC SENSE00 ; skip next command if test button is not
pressed (=0)
BSF R_Sensor1, LEVEL0 ; set test button is pressed (=1)

BCF R_Sensor1, LEVEL1 ; set prophylactic door is closed
BTFSC SENSE01 ; skip next command if door 1 is
closed (=1)
BSF R_Sensor1, LEVEL1 ; set door is open (=0)

BCF R_Sensor1, LEVEL2 ; set prophylactic door lock is closed
BTFSC SENSE02 ; skip next command if door lock is
closed (=1)
BSF R_Sensor1, LEVEL2 ; set door lock is open (=0)

BCF R_Sensor1, LEVEL3 ; set prophylactic water sensor recognise
BTFSC SENSE03 ; skip next command if water sensor
recognise no water (=0)
BSF R_Sensor1, LEVEL3 ; set water sensor recognise water (=1)

; BSF R_SenseActRead, C_Power ; set prophylactic that power is
present ; BTFSS POWER ; skip next command if power is
present (=1)
; BCF R_SenseActRead, C_Power ; set power is present (=0)

BCF B_Event ; set event flag to no event
MOVF R_LSensor1, 0 ; get last sensor values
SUBWF R_Sensor1, 0 ; compare last sensor values with actual
sensor values
BZ RS Exit ; branch if no event happened
BSF B_Event ; set event flag to an event happened

; Rs232 event are separated from this event handler

RS Exit
MOVF R_Sensor1, 0 ; set actual read to last read
MOVWF R_LSensor1 ;...

;----------------------------------

;ReadSenso

BCF R_Sensor2, LEVEL0 ; set prophylactic test button is not pressed
BTFSC SENSE00 ; skip next command if test button is not
pressed (=0)
BSF R_Sensor2, LEVEL0 ; set test button is pressed (=1)

BCF R_Sensor2, LEVEL1 ; set prophylactic door is closed
BTFSC SENSE01 ; skip next command if door 1 is
closed (=1)
BSF R_Sensor2, LEVEL1 ; set door is open (=0)

BCF R_Sensor2, LEVEL2 ; set prophylactic door lock is closed
BTFSC SENSE02 ; skip next command if door lock is
closed (=1)
BSF R_Sensor2, LEVEL2 ; set door lock is open (=0)
READSENS.asm

BCF R_Sensor2, LEVEL3 ;set prophylactic water sensor recognise
no water
BTFSC SENS03 ;skip next command if water sensor
recognise no water (=0)
BSF R_Sensor2, LEVEL3 ;set water sensor recognise water (=1)
no water
BCF R_Sensor2, LEVEL4 ;set prophylactic water sensor recognise
BTFSC SENS04 ;skip next command if water sensor
recognise no water (=0)
BSF R_Sensor2, LEVEL4 ;set water sensor recognise water (=1)

BCF B_Event2 ;set event flag to no event
MOVF R_LSensor2, 0 ;get last sensor values
SUBWF R_Sensor2, 0 ;compare last sensor values with actual
sensor values
BZ RS_Exits ;branch if no event happened
BSF B_Event2 ;set event flag to an event happened

;RS232 event are separated from this event handler

RS_Exits
MOVF R_Sensor2, 0 ;set actual read to last read
MOVF R_LSensor2
RETURN
; LED Flashing Routines

FlashLED
MOVF R_WorkReg1

FL_loop
LEDoFF CALL D50ms
LEDOn CALL D50ms DECF R_WorkReg1, F BNZ FL_loop
LEDoFF RETURN

; Clear the 4 page of the RAM

ClrRAM
BCF STATUS, IRP

;----- Clear RAM page 0
Move RAMPG0&0x0FF, FSR
RAMG0
CLRF INDF
INCF FSR, F
MOVLW ERAMG0+1
SUBWF FSR, W
BNZ RAMG0

;----- Clear RAM page 1
Move CSRAM1&0x0FF, FSR
RAMG1
CLRF INDF
INCF FSR, F
MOVLW (CERAM1+1)&0x0FF
SUBWF FSR, W
BNZ RAMG1

;----- Clear RAM page 2
Move CSRAM2&0x0FF, FSR
RAMG2
BSF STATUS, IRP
CLRF INDF
BCF STATUS, IRP
INCF FSR, F
MOVLW (CERAM2+1)&0x0FF
SUBWF FSR, W
BNZ RAMG2

;----- Clear RAM page 3
Move CSRAM3&0x0FF, FSR
RAMG3
BSF STATUS, IRP
CLRF INDF
BCF STATUS, IRP
INCF FSR, F
MOVLW (CERAM3+1)&0x0FF
SUBWF FSR, W
BNZ RAMPg3

GOTO Main_RAMCleared

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
CopyROMtoRAM

RAMBank3
BSF EECON1, EEPGD
BSF EECON1, RD
NOP
NOP
RAMBank2
MOVF EEDATA, W
RAMBank0
MOVWF _WorkReg1
MOVF _DestPtr, W
MOVWF FSR
MOVWF _WorkReg1
BCF STATUS, IRP
BTFSC Dest
BSF STATUS, IRP
MOVWF INDF
RAMBank2
INC EEDR, 1
BNZ Nooverflow
INC EEDRH, 1
Nooverflow
RAMBank0
INC _DestPtr, 1
DEC _RegLen, 1
BNZ CopyROMtoRAM

RETURN
DIS_VIS.asm

: display and visual routines
: file: DIS_VIS.asm

GetMsgDisplay
  BSF NoDisplay
  GOTO GMFD_Entry

GMFD_Entry
  MOVWF _R_WorkReg1
  RAMBank2
  Move HIGH MsgTab, EEADR
  Move LOW MsgTab, EEADR
  RAMBank0

TableEntryCalcLp
  TSTF _R_WorkReg1
  BZ TableEntryCalcFin
  MOVWF D'03'
  RAMBank2
  ADDWF EEADR, F
  BNC SMOD_NoOverflow
  INCF EEADR, F
  SMOD_NoOverflow
  RAMBank0
  DECF _R_WorkReg1
  GOTO TableEntryCalcLp

TableEntryCalcFin
  Move _R_WorkReg2, R_DestPtr
  BCF Dest
  Move D'03', R_RegLen
  CALL CopyROMtoRAM

  MOVF _R_WorkReg2, W
  RAMBank2
  MOVWF EEADR
  RAMBank0
  MOVF _R_WorkReg3, W
  RAMBank2
  MOVWF EEADR
  RAMBank0
  BCF Dest
  MOVF _R_WorkReg4, W
  MOVWF R_RegLen
  Move LCDWord, R_DestPtr
  CALL CopyROMtoRAM

  BTFSC NoDisplay
  GOTO SMOD_Exit

Move0ff _R_WorkReg4, R_RegLen
  Move LCDWord, FSR
  CALL ShowTxtDisplay
  CALL D2s

SMOD_Exit
  RETURN

: Routines for showing the switch on message
: -------------------------------

ShowSwitchOnMessage

  MOVFLW Text00
  CALL GetMsgDisplay
DIS_VIS.asm

Moveff R_WorkReg4, R_RegLen ; get and set message length
Move LCDword, FSR ; set FSR pointer to the first byte of the
display data
CALL ShowTxtDisplay

CALL D2s

RETURN

; Display on the LCD, the water level and area

WLCheck
Move "W", LCDword+D'00'
Move "L", LCDword+D'01'
Move ",", LCDword+D'02'
Move ",", LCDword+D'03'
Move "A", LCDword+D'04'
Move "R", LCDword+D'05'
Move ",", LCDword+D'06'
Move ",", LCDword+D'07'
Move "W", LCDword+D'08'
Move "L", LCDword+D'09'
Move ",", LCDword+D'10'
Move ",", LCDword+D'11'
Move "A", LCDword+D'12'
Move "R", LCDword+D'13'
Move ",", LCDword+D'14'
Move "2", LCDword+D'15'
Move D'16', R_RegLen
BTFSC R_Sensor1, LEVEL3
GOTO LEVEL_3
BTFSC R_Sensor1, LEVEL2
GOTO LEVEL_2
BTFSC R_Sensor1, LEVEL1
GOTO LEVEL_1
BTFSC R_Sensor1, LEVEL0
GOTO LEVEL_0
GOTO GER_Exit

LEVEL_3
Move "1", LCDword+D'03'
GOTO GER_Exit

LEVEL_2
Move "2", LCDword+D'03'
GOTO GER_Exit

LEVEL_1
Move "3", LCDword+D'03'
GOTO GER_Exit

LEVEL_0
Move "4", LCDword+D'03'
GOTO GER_Exit

GER_Exit
BTFSC R_Sensor2, LEVEL4
GOTO LEVEL_4
BTFSC R_Sensor2, LEVEL3
GOTO LEVEL_3
BTFSC R_Sensor2, LEVEL2
GOTO LEVEL_2
BTFSC R_Sensor2, LEVEL1
GOTO LEVEL_1
BTFSC R_Sensor2, LEVEL0
GOTO LEVEL_0
RETURN

LEVEL_4
Move "5", LCDWord+D'11'
GOTO GER_Exit2

LEVEL_3
Move "6", LCDWord+D'11'
GOTO GER_Exit2

LEVEL_2
Move "7", LCDWord+D'11'
GOTO GER_Exit2

LEVEL_1
Move "8", LCDWord+D'11'
GOTO GER_Exit2

LEVEL_0
Move "9", LCDWord+D'11'
GOTO GER_Exit2

GER_Exit2
RETURN

ShowLevelNArea
Move LCDWord, FSR
CALL ShowTxtDisplay
CALL Dis
RETURN

ShowTxtDisplay
CALL ClrDisplay
BCF STATUS, IRP
Move D'17', R_WorkReg3

SOD_Loop
DECF R_WorkReg3, 1
BZ SOD_addLF
SOD_backFromaddLF
MOVFW INDF
CALL TransmitDisplayData
INCF FSR, 1
DECF R_RegLen
BNZ SOD_Loop
RETURN

SOD_addLF
CALL SetCursor2ndLine
GOTO SOD_backFromaddLF

; clear display, set cursor at home position
ClrDisplay
MOVWL B'00010000'
; clear display, set cursor at home position
CALL TransmitControlData
RETURN
DIS_VIS.asm

; set cursor to the 1st position of 2nd line
SetCursor2ndLine
    MOVLW B'000001100'
    ; set cursor to the 1st position
    CALL TransmitControlData
    RETURN

; send 1 byte (= 8 bit) control data to the display
Transmit8BitData
    MOVWF PORTB
    BSF DiDE
    NOP
    BCF DiDE
    ; data on bus will be captured
    now by display
    Move D'1', R_Time
    ; wait 1 ms
    CALL D18ms
    ...;
    RETURN

; send 1 byte control data which are divided in 2 nibbles to the display
TransmitControlData
    MOVWF R_WorkReg1
    ; save W register
    CALL WaitAsBusy
    MOVF R_WorkReg1,0
    ; restore W register
    ANDLW H'0F'
    MOVWF PORTB
    BSF DiDE
    NOP
    BCF DiDE
    ; data on bus will be captured
    now by display
    SWAPF R_WorkReg1, W
    ; get low nibble
    ANDLW H'0F'
    MOVWF PORTB
    BSF DiDE
    NOP
    BCF DiDE
    ; data on bus will be captured
    now by display
    RETURN

; send 1 byte display data which are divided in 2 nibbles to the display
TransmitDisplayData
    MOVWF R_WorkReg1
    ; save W register
    CALL WaitAsBusy
    SWAPF R_WorkReg1, 1
    ; restore W register
    MOVF R_WorkReg1, 0
    ; put data high nibble on bus
    ANDLW H'0F'
    MOVWF PORTB
    BSF DiDE
   NOP
    BCF DiDE
    ; data on bus will be captured
    display data are on the bus
    BSF DiDE
    NOP
    BCF DiDE
    ; enable bus
    now by display
    SWAPF R_WorkReg1, W
    ; get low nibble
    ; inform display that display data are on the bus
    BSF DiDE
    NOP
    BCF DiDE
    ; data on bus will be captured
DIS_VIS.asm

ANDLW 00H
MOVF PORTB

; put data low nibble on bus
; inform display that
BFS D1RS

; data are on the bus
NOP

; enable bus
BCF D1DE

; data on bus will be captured
BCF D1RS

; set to default state
RETURN

; wait as long as the display is busy

WaitAsBusy

Move D3', R_Time

; wait 3 ms, value have to be adjusted to
the used display
Call DIBms

RETURN
4. Design at UV board
Private Sub Initialise_Click()
Dim sms As String
Dim buffer$ = buffer$ & MSComm1.Input
Dim MsgNumber(1000) As Integer
Dim msglist(1000) As String
With MSComm1
  .CommPort = 12
  .Settings = "9600,N,8,1"
  .InputLen = 0
  .Handshaking = commNone
  .PortOpen = True
End With
MSComm1.Output = "AT+CMEE=1 & Chr$(13)
Do
DoEvents
buffer$ = buffer$ & MSComm1.Input
Loop Until InStr(buffer$, "OK")
Text1.Text = Text1.Text & vbCr & buffer$
buffer$ = ""
MSComm1.Output = "AT+CMGF=1 & Chr$(13)
Do
DoEvents
buffer$ = buffer$ & MSComm1.Input
Loop Until InStr(buffer$, "OK")
Text1.Text = Text1.Text & vbCr & buffer$
buffer$ = ""
MSComm1.Output = "AT+CMGL=" & Chr$(34) & "ALL" & Chr$(34) & vbCrLf
Do
DoEvents
buffer$ = MSComm1.Input
If buffer$ <> "" Then
  Text1.Text = Text1.Text & vbCr & buffer$
  'List2.AddItem Replace(buffer$, Chr$(13) & Chr$(10), "") 'remove all the carriage returns before adding to the listbox
End If
Loop Until InStr(buffer$, "OK")
buffer$ = ""
J = 0
'For I = 0 To List2.ListCount - 1
  'If InStr(List2.List(I), "REC READ") Then 'This is from the Inbox
  'MsgNumber(I) = Val(Mid(List2.List(I), 9, InStr(List2.List(I), ",","(2), Chr$(34), ",")) 'EXTRACT THE MOBILE NUMBER
  'MsgBox (Mid(List1.List(I), 9, InStr(List1.List(I), ",","(9)) // FINDS THE MSG NUMBER
  'J = J + 1
  'End If
  'Next
  'Now getting the highest number which will hopefully be the newest SMS
  SMSNumber = 0
  For I = 0 To J - 1
    If MsgNumber(I) > SMSNumber Then
      SMSNumber = MsgNumber(I)
    End If
  Next
  'Received read message
  MSComm1.Output = "AT+CMGR=" & SMSNumber & Chr$(13)
Do
vbcodetxt.txt

DoEvents
buffer$ = buffer$ & MSComm1.Input
Loop Until InStr(buffer$, "OK")

Text1.Text = Text1.Text & vbCrLf & buffer$
Text2.Text = buffer$
'buffer$ = ""

MSComm1.PortOpen = False

End Sub

Private Sub Command2_Click()
Dim y As Variant

If MSComm1.PortOpen = False Then MSComm1.PortOpen = True
MSComm1.Output = "AT+CMGR=1" & Chr$(13)
Do
DoEvents
buffer$ = buffer$ & MSComm1.Input
Loop Until InStr(buffer$, "OK")

y = Split(buffer$, ",")
MsgBox Mid(Replace(Mid(x, 13, Len(x) - 13), vbCrLf, ""), 1, Len(Replace(Mid(x, 13, Len(x) - 13), vbCrLf, ""))) - 3)
Text2.Text = Mid(Replace(Mid(y, 13, Len(y) - 13), vbCrLf, ""), 1, Len(Replace(Mid(y, 13, Len(y) - 13), vbCrLf, ""))) - 3)

End Sub