AALTO-UNIVERSITY
SCHOOL OF SCIENCE AND TECHNOLOGY

Faculty of Electronics, Telecommunications and Automation
Department of Electronics
Lighting Unit

Mukul M. Joshi

Design of a Graphical User Interface for Home Energy monitoring system

Thesis for the degree of Master of Science
Espoo 13.12.2010

Supervisor: Dr. Liisa Halonen

Instructor: Professor Jouko Pakanen

AALTO-UNIVERSITY
SCHOOL OF SCIENCE AND TECHNOLOGY Abstract of the Master's Thesis

Author: Mukul M. Joshi	
Name of the thesis: Design of a Graphical User Interface for Home Energy monitoring system Date: 13.12.2010	Number of pages: 57
Faculty: Department of Electronics, Lighting unit Professorship: S-118	
Supervisor: Dr. Liisa Halonen Instructor: Prof. Jouko Pakanen	

Abstract:

Excellent graphics are the instruments used for representing quantitative information. Graphics help people to understand complex things easily. Hence the user interfaces developed should be clear, illustrative and designed from the user point of view with respect to their applications. The thesis work deals with the design of a graphical user interface (GUI) developed for a home energy monitoring system. Design methodologies like user centric design and empathic design are followed while creating the user interface and also the effect of various colors on human perception is studied. Hence the final design of user interface provides the end-users a visualization of the energy produced and consumed in a monitored environment.

The monitoring devices are connected to ThereGate system (data logger) via the M-Bus communication protocol. The ThereGate platform uses an Open Source Linux system as the operating language. The other communication platform used over the ThereGate platform is an oBIX bridge, a web based service interface rich in XML support for transferring the data.

The interface has been programmed by using Visual basics 2008 and VB.NET, developed by Microsoft.

The work progresses with an initial explanation on the availability of various home energy monitoring systems on the market and their comparison. The other units discuss the architecture of the ThereGate system, give a brief overview of the M-bus system, and discuss the development of graphical user interface (GUI) from a user centric design perspective using Microsoft's Visual Basics and VB.NET and configuration of M-Bus. The last unit contains a discussion of the goals achieved at the end of the design and the future developments that can be made to have more user interactions with the user interface.

Keywords:

Energy monitoring, M-bus, Visual basics, User interface, ThereGate, PadPuls M2C, Micromaster, OBiX, ZigBee, Z-wave

Preface

I am thankful to our Head of Department Dr. Liisa Halonen for believing in me and giving a chance to work in the field of home automation system. Also I thank Prof. Jouko Pakanen for guiding me during the thesis period and giving me valuable feedback related to the report. I am also thankful to Mr. Andrey Litvinov and Mr. Antti Karjalainen for helping me with oBIX server and M-Bus ThereGate installation. Also I am very much thankful to Mr. William Martin for proofreading the report and suggesting the necessary grammatical modifications.

Last but not least I thank my family members, friends for standing by my side and motivating me during the work.

Espoo, Finland, 13.12.2010

Mukul M.Joshi

Table of Contents

List of symbols and abbreviations

ANSI	American National Standards Institute
APS	Application Support Sub-layer
ASCII	American Standard Code for Information Interchange code
BCD	Binary Coded Decimal
CPU	Central Processing Unit
DFC	Data Flow Counter
DIB	Data Information Block
DIF	Data Information Flow
DIFE	Data Information Flow Extension
DLNA	Digital Living Network Alliance
DRH	Data Record Header
EBCDIC	Extended Binary Coded Decimal Interchange Code
EEPROM	Electrically Erasable Programmable Read Only Memory
EHS	European Home Systems Protocol
EIB	European Bus Installation
E-MAIL	Electronic Mail
FCB	Frame Count Bit
FCV	Frame Count Valid
HES	Home Electronic System
HGI	Home Gateway Initiative
HVAC	Heating, Ventilating and Air Conditioning
IEEE	Institute of Electrical and Electronics Engineers
IGRS	Intelligent Grouping and Resource Sharing
IP	Internet Protocol
ISO	International organization for Standards
ITU	International Telecommunication Union
LAN	Local Area Network
LSB	Lease Significant Bit
MAC	Media Access Control layer
MDH	Manufacturer Data Header
OBiX	Open source Building Information Xchange
OSGI	Open Services Gateway Initiative
OSI	Open System Interconnection
PHY	Physical
RF	Radio Frequency
SPIA	Single Primer Isothermal Amplification
UPnP	Universal Plug n Play
VIB	Value Information Block
VIF	Value Information Field
VIFE	Value Information Field Extension
WAN	Wide Area Network
WPAN	Wireless Personal Area Network

1 Introduction

Conservation of energy is necessary for economic and social prosperity. It is important to optimise the usage of energy in homes and make the users aware of their energy consumption. There are various systems in the house which consume lots of electrical energy. With an increase in the usage of the modern devices on one side and the shortage of the energy supplies on other side, it has become mandatory to use the energy more effectively and if possible conserve it in smarter ways than before.

The consumer should be made aware of both the conservation and efficiency of energy. Awareness can be created by educating the consumer via various mediums like newspapers, seminars, workshops (mainly conducted in poor countries or developing countries due to low per capita income) or by installing advanced energy monitoring and automation devices (mainly in developed countries due to high per capita income) which alert the user of their energy usage and at times take corrective actions to reduce the energy consumption.

The principle behind the installation of these monitoring devices is,

"The m*ore you see what you are using, the more you think how to use it efficiently and the more you use it efficiently, the more you conserve it*".

1.1 Background

<u>Traditional energy meters</u>

In the early days, energy monitoring used to be done with the help of traditional electro mechanical meters. But the main disadvantage of this system was that it could monitor only the consumed energy. Hence the consumer used to know only the combined consumed energy and was unaware of the individual devices or appliances consumption. Also one of the major problems in many countries is 'meter tampering'. At times the consumer used to alter the architecture of the meter (disk and magnets) to show less energy usage which leads to financial loss on the parts of the utility provider.

<u>Smart energy meters</u>

Smart energy meters are those which can store the past energy records of a consumer. These meters can display the current, voltage, power and energy readings over a scale of a period. They have the capability to calculate the energy bill of a consumer instantly and automatically communicate this to the utility service provider. Some of the meters can even actually switch on a device when the supplier informs about the low energy price. [1]

On the basis of communication, the smart meters can be distinguished between two major types [1]:

- *AMR (Automatic reading)*: In this the communication is only from meter to the data-logging system. [1]

- *AMM (Automatic management)*: Here two way communications takes place which has more functionality. [1]

In addition to the above classification, the monitoring systems can be classified on the basis of the medium used to communicate with each other. These can be:

- *Wired/Fixed communication*: By using Power line carriers, telephone lines, fiber optics, M-bus, Modbus, twisted pair cables, etc. [1]

- *Wireless communication*: GSM, Z-Wave, ZigBee, GPRS, etc.[1]

Many large players like Microsoft Corporation with their "Microsoft Hohm" platform and Google with their "PowerMeter" technology are investing in this business. The idea behind both technologies is to collect the energy data from the smart meters owned by the utilities which are compatible with the platforms and provide the user with their energy consumption pattern to create awareness and conserve the energy.

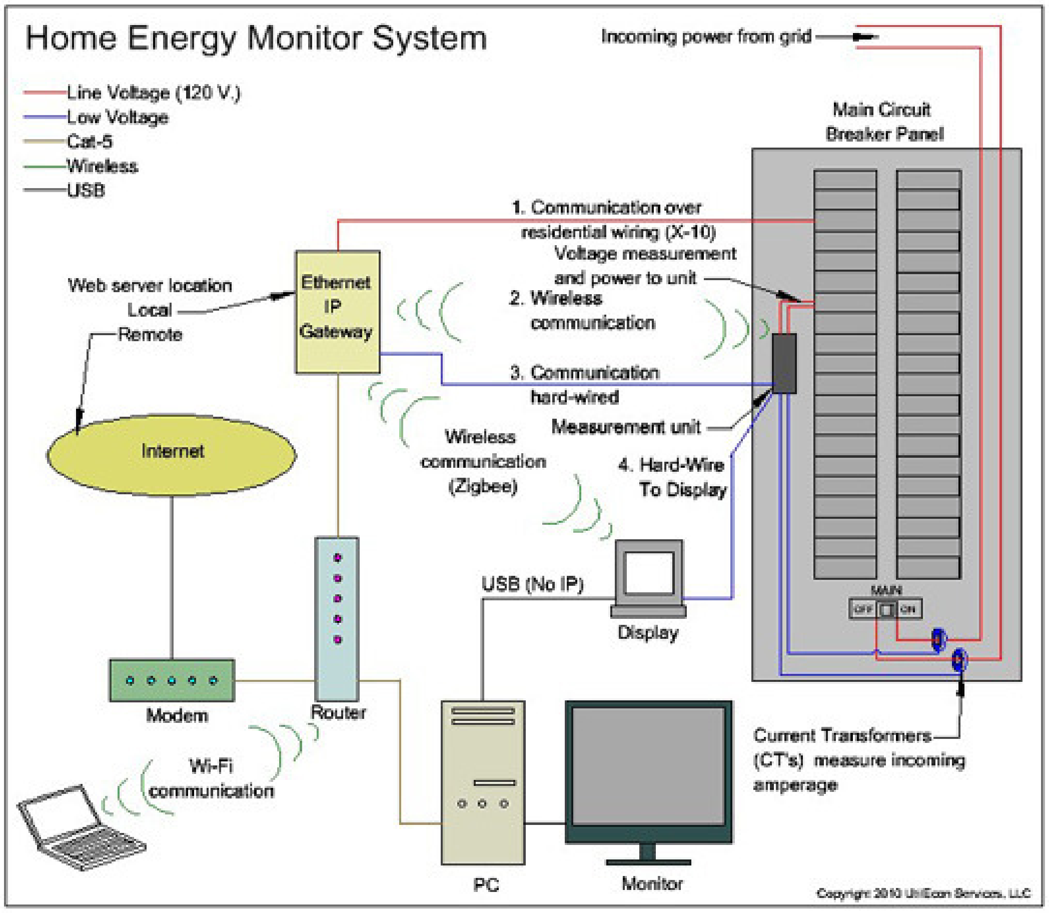

Home Energy Monitor System
Incoming power from grid
Line Voltage (120 V.)
Low Voltage
Cat-5
Wireless
USB
Main Circuit Breaker Panel
1. Communication over residential wiring (X-10)
Voltage measurement and power to unit
2. Wireless communication
3. Communication hard-wired
Measurement unit
4. Hard-Wire To Display
Web server location
Local
Remote
Ethernet IP Gateway
Internet
Wireless communication (Zigbee)
USB (No IP)
Display
MAIN
OFF
ON
Modem
Router
Wi-Fi communication
PC
Monitor
Current Transformers (CT's) measure incoming amperage
Copyright 2010 UtilEcon Services, LLC

Fig 1.1. A typical Home monitoring system architecture [2]

1.2 Motivation and Objective

Visualization or representation of information in an appropriate manner is an important factor that has to be addressed as the systems get more complicated and the demand for representing these complicated parameters increases. This was the driving force behind this thesis work.

The principle objective of the thesis work is to develop a graphical user interface (GUI) that helps the users in understanding their energy consumption in real time and alerts them about their usage pattern with the help of a developed color scheme. However, to attain this objective, three main challenges should be properly addressed.

1. Should the user interface be interactive and customizable according to individual's needs?

2. Shall we follow the rule of single window viewing or multiple window viewing? Having all the entities on a single window helps in viewing all the values at one place and reduces the task of changing the window to view different entities.

3. What kind and pattern of color scheme should be adopted to alert the user? The users should be familiar with the developed pattern, in order to become used to the system as fast as possible.

1.3 Thesis Structure

The thesis is structured in such a way that each chapter provides a background for understanding each subsequent chapter and so that by the concluding chapter the entire work is clear in the reader's mind.

Chapter 2 provides a background about the current technologies available on the market, their advantages and disadvantages. The chapter also introduces the reader to the ThereGate system and its importance in this thesis.

Chapter 3 discusses about the architecture and software used for the ThereGate system. Only a small part of data was allowed to be published due to the confidentiality of the documents and images.

Chapter 4 briefly introduces the reader to the M-Bus protocol used here. It explains about the operating principle, various layers of the M-Bus system, etc.

Chapter 5 explains the oBIX system briefly. The reader becomes acquainted with the oBIX structure, programming format and various other syntaxes used in the oBIX server.

Chapters 6 and 7 discuss about the development of the user interface and configuration of the entire system. Here various aspects of user centric design and how various software tools are used to configure the system are mentioned.

Chapters 8 and 9 finally provide some concluding remarks, discussions and suggestions for future development in the entire project.

2 Technologies on the market

The increases in remote monitoring have driven forward advances in sensor network technologies. The major problem however lies in the *communication* between these networks. Many of the technologies have proprietary communication protocols which at times make it difficult to integrate them together.

UPnP, IGRS, DLNA, OSGI, SPIA, HGI, NMRP, Z-Wave, X-10, CEBus, ZigBee, ThereGate etc. are some of the network technologies available on the market for home and building automation and energy monitoring [5].

Of the developed technologies X-10, Z-Wave, ZigBee, ThereGate and OSGI are some of the attempts made to have a common command language [5]. ThereGate scores over the other technologies in the integration platform with language independence. In this chapter, the above mentioned technologies are mentioned to make the user briefly understand the monitoring systems.

2.1 X-10

The X-10 uses the existing electrical wiring (electricity network) in the house to control the devices and is a standardized home automation protocol. The advantage of this technology over the others is its integration with the installed electrical power distribution network in home for controlling; which reduces the cost [6].

An X-10 system device can be controlled by Radio Frequency; a command is sent to the X10/RF receiver, which is transmitted through the electricity network to an X-10 actuator, which, in turn, operates the device. The radio frequency at which it operates is 433MHz in Europe.

An X-10 protocol has a simple addressing system which consist of 16 home codes (from A to P) and 16 device codes (from 1 to 16) allowing over 256 devices to be addressed without any ambiguity [6].

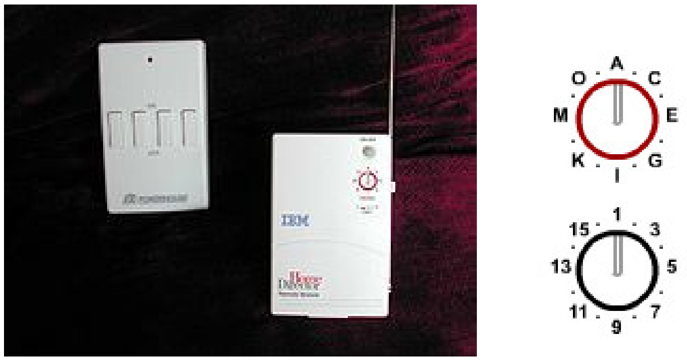


Fig 2.1. A typical 4-channel radio switch and radio-to-power line transponder (left) and X-10 controllers [6]

An X-10 command includes two actions: activate a particular device (message code indicating device), and then send the function to be executed (message with the function code). Once a device is activated, it remains in same condition until another is located. While a device is active multiple commands can be sent to it.

2.2 Z-Wave

Z-wave is a proprietary wireless communication protocol designed for home automation systems. It uses reliable, low-power radio waves. Virtually it can be connected to any appliances like window shades, thermostats and home lighting. The appliances connected through it can be controlled remotely from a PC with Internet connectivity and mobile phones through World Wide Web services [7].

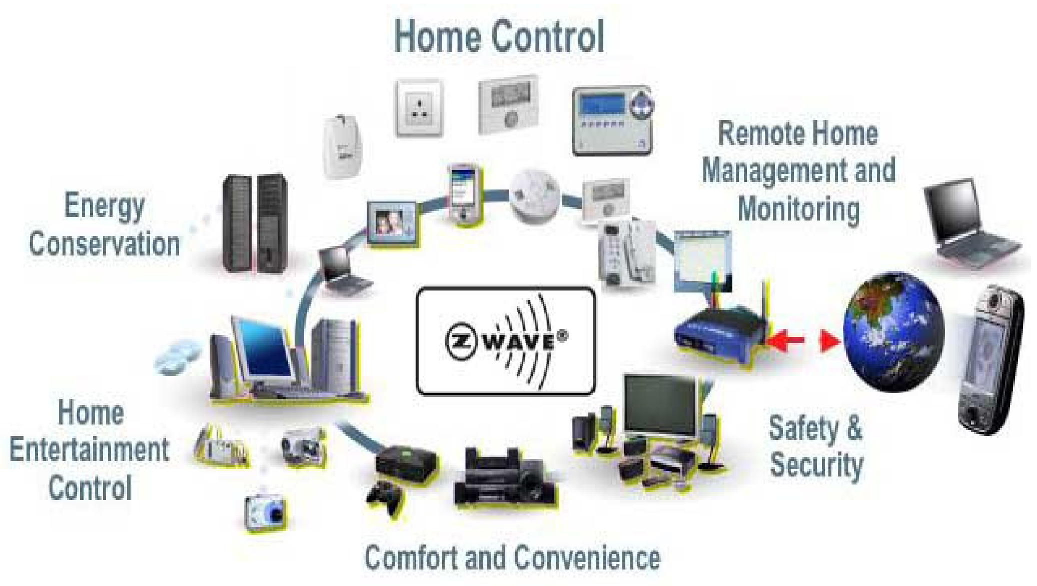


Fig 2.2. Z-Wave Network and its functions [7]

A typical Z-Wave can have a single device to be controlled and a controller. Extra devices with their controllers like traditional hand-held controllers, key-fob controllers, wall-switch controllers and PC applications designed for management and control of the network can be added at any time. Z-Wave has a *source routed mesh network* topology which uses one or multiple master controllers. This helps in communicating the devices efficiently with each other and thus eliminates the radio dead spots that might occur.

2.3 ZigBee

The ZigBee protocol is used for wireless personal area networking (WPAN), which means digital radio connection between computer and related devices [8]. This protocol is used by devices having low data transfer rates; it consumes very little power and thus can have a long battery life. Using ZigBee makes it possible to have a single unit control over all the devices in a home network. [8]

The ZigBee network has several layers which can have intrapersonal communication within the network. Usage of this layer architecture has made ZigBee a low cost, easy implementation, reliable data transfer, short-range operations, very low power consumption and adequate security technology [9].

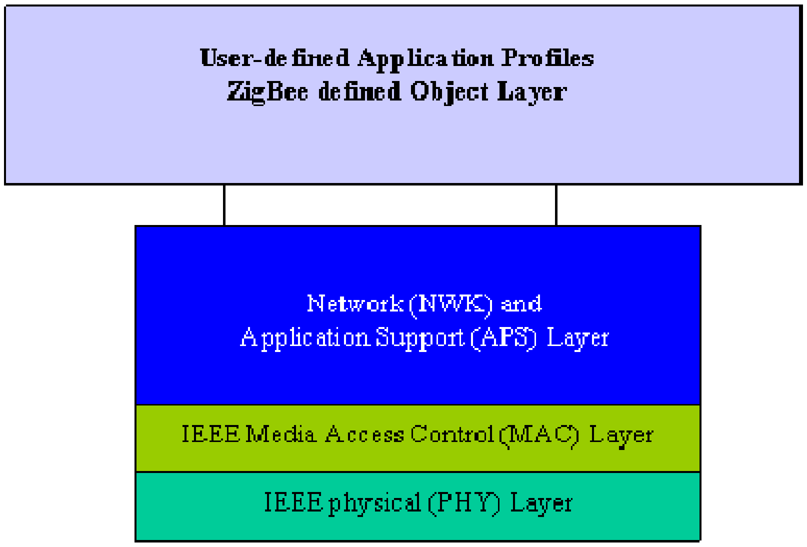

User-defined Application Profiles
ZigBee defined Object Layer
Network (NWK) and
Application Support (APS) Layer
IEEE Media Access Control (MAC) Layer
IEEE physical (PHY) Layer

Fig 2.3. ZigBee stack architecture [9]

The ZigBee devices work in two addressing modes – star or peer-to-peer [10].Also ZigBee uses either of two modes, beacon or non-beacon to enable the to-and-fro data traffic.

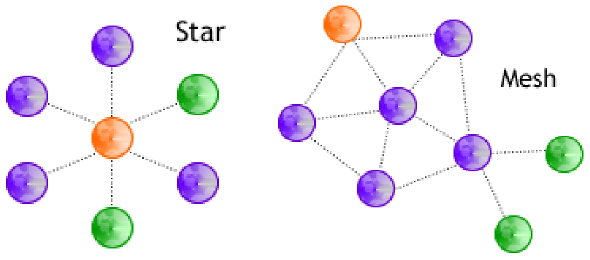


Fig 2.4. ZigBee topologies

2.4 OSGi

The OSGi (open services gateway initiative) platform is based on open standards which deploys software from service providers related to security, remote management and access to the devices which are connected to residential gateways. [11]

The key benefits associated with OSGi are platform independence, application independence, security, multiple services, multiple local network technologies, multiple device access technologies, coexistence with other standards and future proof [11].

The OSGi architecture has the following components [11]:

- Services gateway
- Service provider
- Service aggregator
- Gateway operator
- Wide area network (WAN) and carrier/ ISP
- Local devices and networks

The OSGi framework can be divided into two parts [12]:

- Service platform
- Deployment infrastructure

A service platform is software that supports the service orientation interaction as shown in figure 2.5. The interaction involves three main actors: service providers, service requesters and a service registry. In service orientation interaction, service providers publish service descriptions and service requesters discover services and bind to them. Publication and discovery are based on service description. Service providers and requesters are part of an entity known as a *bundle* that is both a logical and physical entity [11].

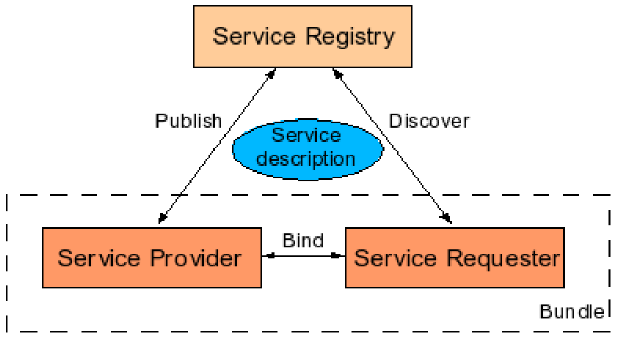


Fig 2.5. Service orientation interaction [12]

The OSGi framework provides mechanisms to support continuous deployment activities. They include installation, removal, update, activation and deactivation of a physical bundle. Once a bundle is installed in the platform, it can be activated if deployment dependencies that are associated to the bundle are fulfilled. Deployment activities are realized according to a well-defined series of states depicted in Figure 2.6. [12]

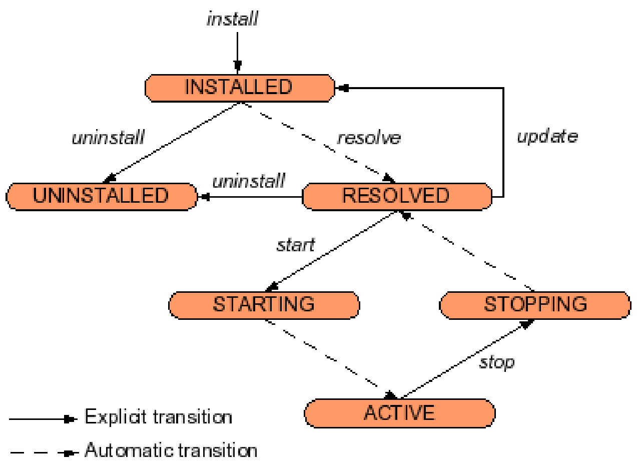


Fig 2.6. Physical bundle life-cycle [12]

2.5 KNX

KNX is a worldwide adopted standard for home and building control applications, like lighting, security systems, HVAC monitoring, alarming, water control, energy management, metering. The technology can be used in new as well as in the existing home and buildings [13].

Interworking is the major asset of KNX technology. It is the successor to three previous standards BatiBus, EIB (European Bus Installation)/ InstaBus and EHS (European Home Systems Protocol) [13].

The KNX standard allows the manufacturer to customize his product according to the needed mode and the market which helps in catering the needs of various appliance and markets. [14]

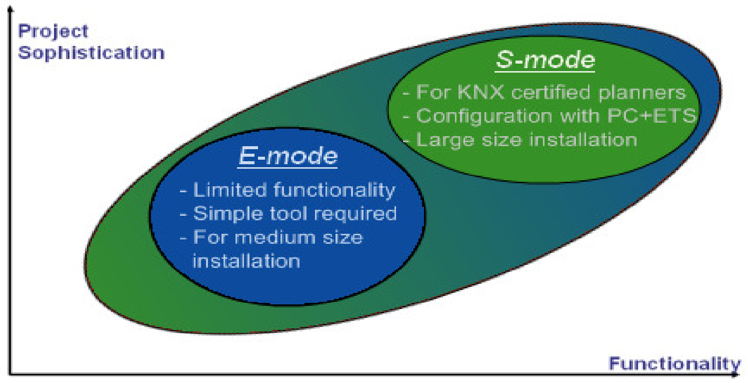

Project Sophistication
S-mode
- For KNX certified planners
- Configuration with PC+ETS
- Large size installation
E-mode
- Limited functionality
- Simple tool required
- For medium size installation
Functionality

Fig 2.7. Configuration modes [14]

The KNX standard includes several communication media. Each communication medium can be used in combination with one or more configuration modes. The available communication means are TP-1 (Twisted Pair), PL110 (Power Line), RF (Radio Frequency), IP (Ethernet) [15].

2.6 Comparison of Technologies

The availability of many technologies can confuse the end-user during the decision making process. Keeping this in mind a small comparison was made between the monitoring technologies related to ease of installation, reliability and price. Various technologies score over each other in different departments as shown in Figure 2.8.

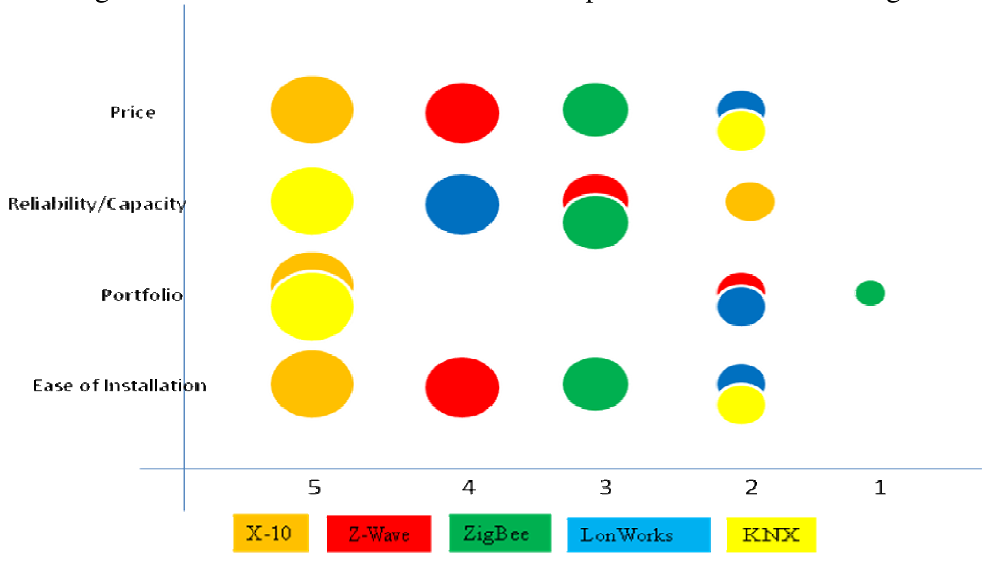


Fig 2.8. Comparison of the above mentioned technologies [16]

The communication protocols mentioned above have advantages over each other in different areas, but the end-user doesnot get the privilege of having all of them on a single platform and ends up paying extra money either for installing new systems which support all the devices or automate only those devices which support the purchased communication technology.

Hence, ThereGate is a solution developed to encounter the above problems. It is a platform consisting of integrated hardware and software in the form of a small computer where all the above mentioned technologies like ZigBee, Z-Wave, X-10, etc. can be integrated together. The current version of the ThereGate system supports the M-bus, Mod-bus, Z-Wave and ZigBee protocols. More details about ThereGate can be studied in the next chapter.

3 ThereGate system

The advantage of a single integration platform is to initiate the communication between the devices having different protocols which results in more sophisticated communication techniques between various devices for monitoring, controlling and taking corrective actions if necessary.

ThereGate is a technology independent open Linux based platform that supports the commonly used smart home technologies. This makes it a universal platform to be used for various applications from different vendors and also for third parties to create new solutions and applications.

Here the goal of using ThereGate system can be identified from its advantages. The most important control nodes of ThereGate are a mobile phone and a web browser. The entire ThereGate system can be integrated into different back-end systems, Internet services and portals for easy and convenient controlling.

Also the back-end servers ensure a seamless and secure link between the mobile phone and ThereGate. Automatic updates of device software keep the entire system up to date without any hassle to the consumers. In addition to this, further value to the end consumer comes from the integration of various devices from 3rd party vendors, all under the control of a single user interface. These devices include various monitoring and controlling systems at times connected to security and safety sensors.

3.1 Basic Architecture

The aim of the thesis work is to fetch the data from the communication bus, transfer it to ThereGate and display the result on the developed Graphical User Interface.

Following figure shows a layered view of the ThereGate system.

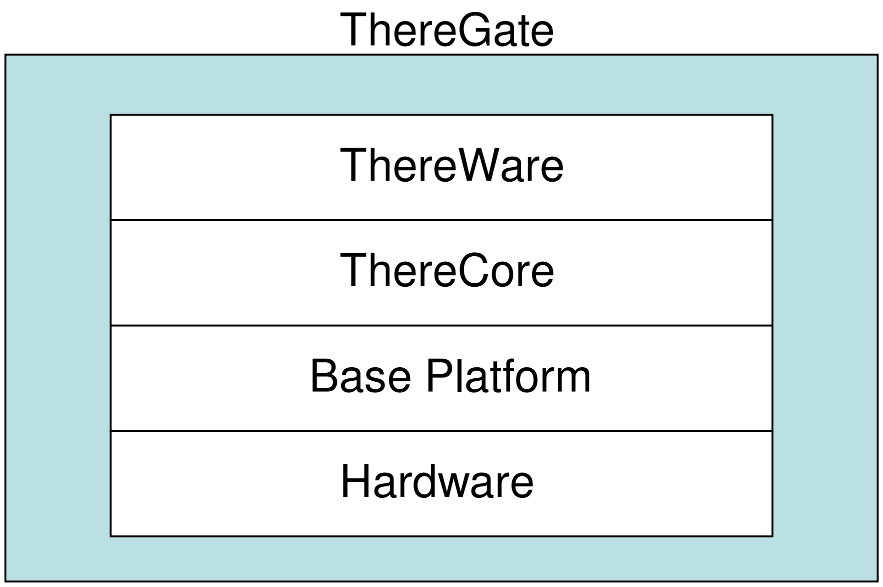


Fig 3.1. ThereGate layered view

3.1.1 Software

The Home Automation Software (HAS) framework is the core of the software platform. It runs on a Linux 2.6 platform which has different software components. New development is possible on the HAS framework which can add extra value to the existing devices. The specialty of the framework lies in making simple devices to act in smarter ways. [5]

The HAS framework has three important components: [5]

1. Technology drivers:
   It helps in translating any communication protocol to the language understood by the framework. This is the core foundation to develop ThereGate.

2. Home Control Logic:
   It is a set of features used for commanding, getting information and grouping the devices. The intelligence for the devices can be developed here which helps in installing more devices to the network and reduces the cost and power consumption.

3. Presentation Layer:
   The presentation layer presents the grouped devices to different control nodes. It can use HTTP in web browsers or UPnP for mobile devices. Also if required a new interface can be developed for the framework (eg: oBIX, Niagara AX).

3.1.2 Benefits of HAS Framework

- Devices and systems from various manufacturers can communicate with each other and take actions according to their needs. [5]

- User interfaces can be easily implemented. [5]

- Peripherals controllers having Bluetooth, USB, WLAN or Ethernet are not required. [5]

- Updating and adding new devices to the present system is easily possible. [5]

The technical details of the ThereGate system are available in the spec sheet in Appendix A.

As our ThereGate box was preloaded with M-Bus communication drivers; none of the systems mentioned in Chapter 4 are in use and are mentioned to make the reader familiar with M-bus.

3.2 Requirements of a BUS

In order to have efficient and flawless communication between the meters and the Master, a metering bus should fulfill the following requirements: [18]

- Interconnection of several devices over long distances- up to several kilometers.
- High degree of transmission integrity for end user billing.
- High speed of transmission due to small amount of data transfer.

- Insensitive to external interferences from capacitive and inductive coupling.
- Slave should be electrically isolated to avoid ground loops.

4 Overview of Meter Bus (M-Bus)

M-Bus is a low cost home electronic system (HES). The Meter-Bus is used for remote reading of utility meters like consumption of gas, water and electricity in the home. The bus can be remotely powered or can be battery driven. When required the meters can deliver the data to a common master, which can be a hand-held computer connected at various levels to read out the data. The applications where M-Bus can be used are alarm systems, flexible illumination installations and heating controls.

It was developed by Prof. Dr. Horst Ziegler of the University of Paderborn in cooperation with Texas Instruments Deutschland GmbH and Techem GmbH.

The advantages of using M-Bus in the thesis is its simple two-wire bus interface for slaves, interchangeable polarity, baud rates which can be varied from 300-38,400 Baud, short circuit proof, a low component cost per slave and an easy availability of components and pre-installed M-Bus drivers on ThereGate moreover there is technical assistance from the company should there be any problems.

The other reasons to choose the M-Bus system are its support for 250 slaves, arbitrary bus topology, a maximum distance for a single device which can be stretched till 10km, meeting all EMC, ESD and EMI requirements and European standards; there is a dedicated user group for application development and trouble shooting and tap proof transmission of data via current modulation.

4.1 M-Bus in OSI Model

The concept was based on ISO based OSI (Open Source integration) model.

In the M-bus system, the transport, session and presentation layers are missing. The levels four to six of the OSI model are empty. Hence only the physical layer, data link layer, network and application layer are provided with functions.

As in OSI model changing of baud rates or address by higher layers is not allowed, there is a **Management Layer** beside and above the seven OSI-layers as shown in Table 3.2. [18]

Hence the address 253 is reserved for the network layer and addresses 254 and 255 are reserved for managing the physical layer as mentioned in Table 2. [18]

Table 3.1. M-Bus layers in OSI model [18]

Layer	Functions	Standard
Application	Data structures, data types, actions	EN1434-3
Presentation	empty	
Session	empty	

Transport	empty	
Network	extended addressing (optional)	-
Data Link	transmission parameters, telegram formats, addressing, data integrity	IEC 870
Physical	Cable, bit representation, bus extensions, topology, electrical specifications.	M-Bus

Table 3.2. Management layer in OSI model [18]

MANAGEMENT LAYER	
Application Layer	
Presentation Layer	
Session Layer	
Transport Layer	
Network Layer (if address = 253)	Address 253 / Enable Disable CI=$52/$56
Data Link Layer	
Physical Layer	Address 254 (255)

4.2 Operating Principle

In M-bus system communication is controlled by a master. A typical M-Bus system has a master, a number of slaves and a two-wire connecting cable as shown in figure 4.1 below:

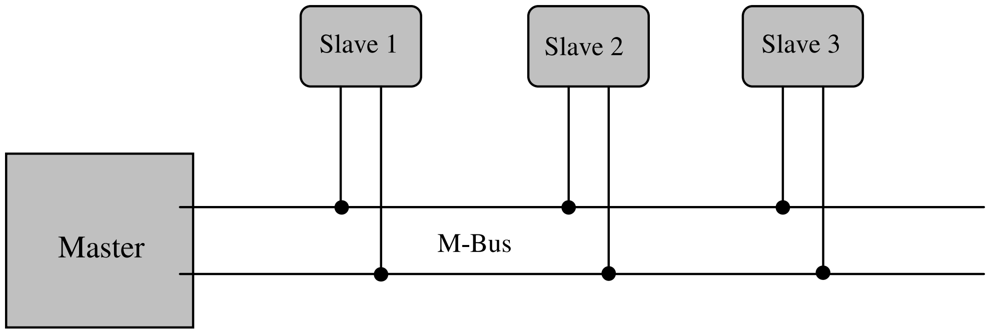


Fig 4.1. Block diagram showing M-Bus system principle [18]

The slaves are always connected in parallel to the transmission medium. The serial data transfer technology is used.

In order to allow remote powering of the slaves, the bits are represented as follows: [18]

Bits transferred from **master to slave** are accomplished by means of **voltage level shifts**. A logical "1" (Mark) corresponds to a nominal voltage of +36 V at the output of the bus driver (repeater), which is a part of the master; when a logical "0" (Space) is sent, the repeater reduces the bus voltage by 12 V to a nominal +24 V at its output.

Bits sent in the reply direction from **slave to master** are accomplished by **modulating the current consumption of the slave**. A logical "1" is represented by a constant current (versus voltage, temperature and time) of up to 1.5 mA, and a logical "0" (Space) by an increased current drain requirement by the slave of an additional 11-20 mA. The mark state current can be used to power the interface and possibly the meter or sensor itself. The transmission of a space by a slave results in a slight reduction in the bus voltage at the repeater due to output impedance, as can be seen in the figure below:

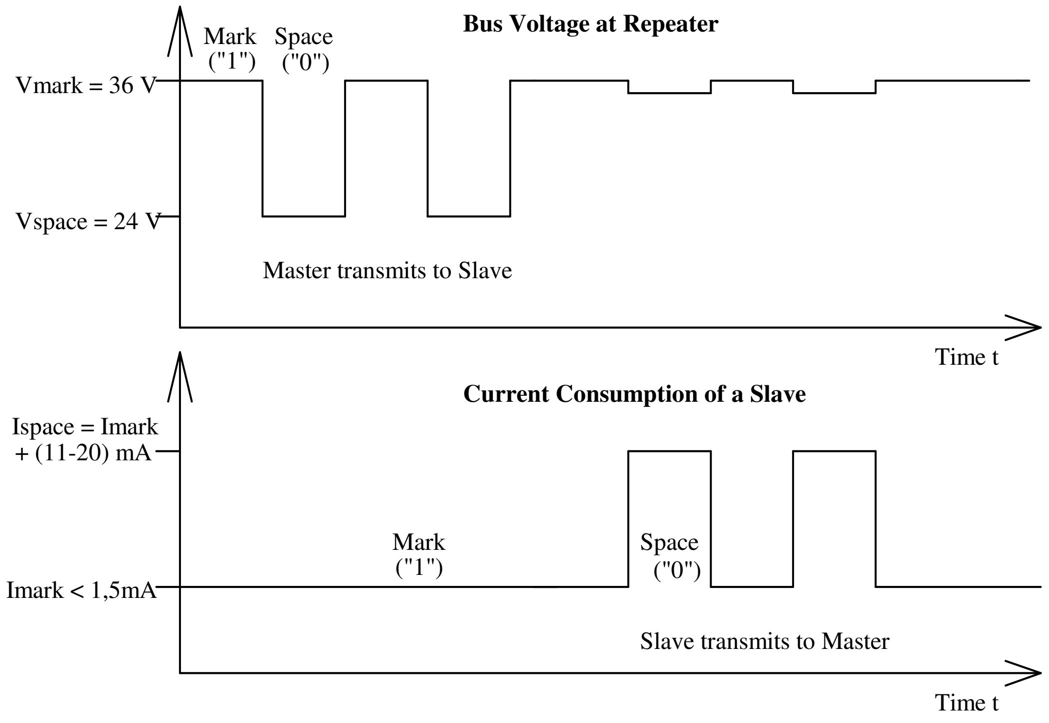


Fig 4.2. Representation of bits on the M-Bus [18]

The quiescent state on the bus is a logical "1" (Mark), i.e. the bus voltage is 36 V at the repeater, and the slaves require a maximum constant quiescent current of 1.5 mA each.

When no slave is sending a space, a constant current is been drained from the repeater which drives the bus. As a result of this, the Mark voltage at the slave is less than +36 V. The slave must therefore not detect absolute voltage levels, but instead for a space detect a voltage reduction of 12 V. The repeater must adjust itself to the quiescent current level (Mark), and interpret an increase of the bus current of 11-20 mA as representing a space. This can be realized with acceptable complexity only when the mark state is defined as 36 V. This means that at any instant, *transmission is possible in only one direction - either from master to slave, or slave to master* (Half Duplex). [18]

Due to transmission in the master-slave direction with a voltage change of 12 V, and in the answering direction with at least 11 mA, besides remote powering of slaves a *high degree of insensitivity to external interference* is achieved.

4.2.1 Bus Specifications

Segmentation
An M-Bus system can have many zones; each one of them has its own group address, and is connected to each other by zone controllers. Each zone consists of segments, which are connected by remote repeaters.

Normally, the M-Bus system has only one segment, which is connected to a Personal Computer (PC) via a local repeater which is called a *master*. [18]

Cable
A two-wire standard cable is used as the transmission medium for the M-Bus. The *maximum distance between a slave and the repeater should be 350m*; this length corresponds to a cable resistance of up to 29Ω. This distance applies for the standard configuration having Baud rates between 300 and 9600 Baud, and a maximum of 250 slaves. The maximum distance can be increased by limiting the Baud rate and using fewer slaves, but the bus voltage in the Space state must at no point in a segment fall below 12 V, because of the remote powering of the slaves. In the standard configuration the total cable length should not exceed 1000m, in order to meet the requirement of a maximum cable capacitance of 180 nF. [18]

4.3 Slave Design

Remote Powering
The interface between the slave and the bus should take the current it needs from the bus system. If possible, the complete slave should be fed from the bus.In case the bus fails, the slave should switch over to battery operation automatically in order to save the collected data.If the slaves are operated only from batteries, then care must be taken to install the batteries which have longer life (maybe of several years). [18]

Protective measures
The bus interfaces of the slaves are polarity independent. In order to operate the bus in case of a short circuit of one of the slaves, there must be a protection resistor with a nominal value of (430±10)Ω in the bus lines. Reverse polarity protection, reception, transmission and powering of the procesor are some of the charactersitsics which should be taken into consideration while designing the slaves for the M-Bus systems. [18]

4.4 Data Link Layer

The protocol of the data link layer is based on IEC 870-5. The signal quality requirements between master and slave are defined by IEC 7480 in order to have a Master-Slave structure, since the slaves cannot communicate with each other.

4.4.1 Transmission Parameters

The transmission here is carried out by an asynchronous serial bit and the synchronization by start and stop bits for each character. Pauses are not allowed within a telegram. The start bit must be a *Space* and the stop bit a *Mark*. In between them, the eight data bits and an even parity bit are transmitted, ensuring that at least every eleventh bit is a Mark. The bit of data having the lowest value (LSB) is sent first. The transmission is half duplex has a data rate of at least 300 Baud. In the figure below, transmission of a byte in *calling* and *replying* directions are represented. [18]

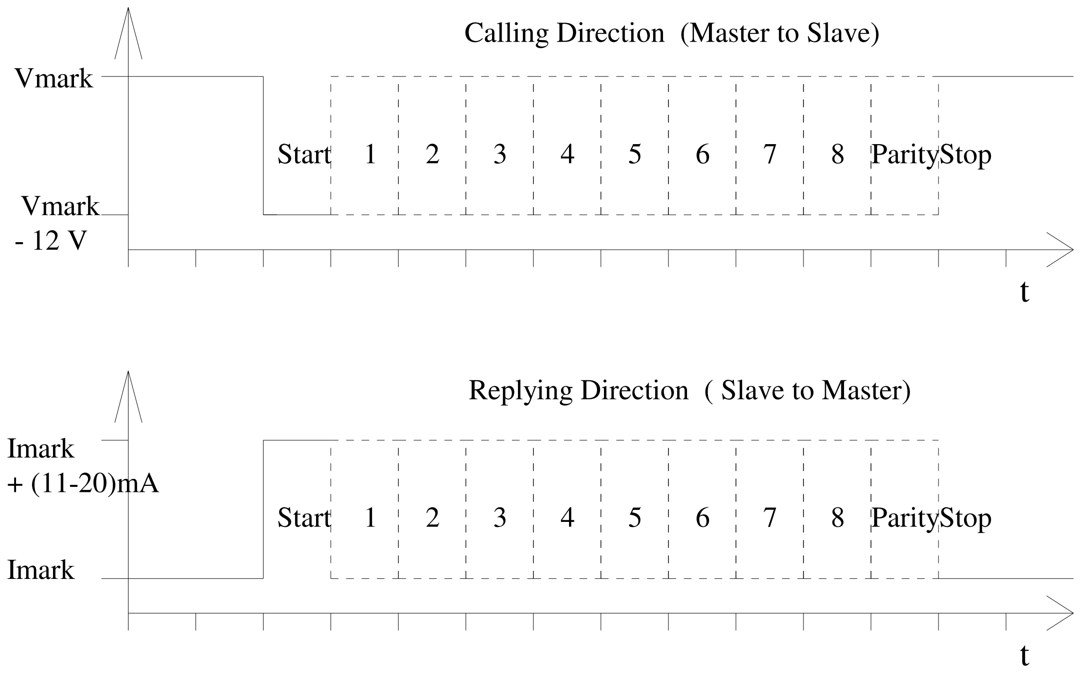


Fig 4.3. Transmission of a Character in Calling and Replying Direction [18]

4.4.2 Telegram Format

As per IEC 870-5, transmission of remote data can be carried out by three different data integrity classes I1, I2 and I3. For the M-Bus protocol, the format class FT 1.2 is used, which is in the data integrity class I2, having a Hamming Distance of four. The format class FT 1.2 has different telegram formats, which can be identified by means of special start characters. [18]

Table 3.3. Telegram formats of M-Bus protocol as per FT 1.2 [18]

Single Character	Short Frame	Control Frame	Long Frame
E5h	Start 10h	Start 68h	Start 68h
	C Field	L Field = 3	L Field
	A Field	L Field = 3	L Field
	Check Sum	Start 68h	Start 68h
	Stop 16h	C Field	C Field
		A Field	A Field
		CI Field	CI Field
		Check Sum	User Data (0-252 Byte)
		Stop 16h	Check Sum
			Stop 16h

5 oBIX

oBIX, an acronym for *Open Building Information Exchange* is a standard for web based interfaces for building control models. The M2M web interfaces can be developed using enterprise friendly technologies like XML, HTTP, and URIs. It was developed by OASIS in December 2006.

As oBIX has a web service interface, it is not necessary to interact with the underlying control system, thus eliminating the complexity of data accessibility. This web service interface can be used to secure data from various entities in a simple format. Besides this it enables the mechanical and electrical systems to provide continuous visibility of performance and operational status. [19]

Hence due to these advantages, it was decided to use oBIX over the ThereGate platform to access the data from the monitoring devices and embed them in the interface easily. Also the XML support by Visual Basics is easy to program. This helps the end user with brief knowledge about XML and VB/.NET to design their own user interfaces according to the requirements.

The following design points can be solved by using oBIX: [19]

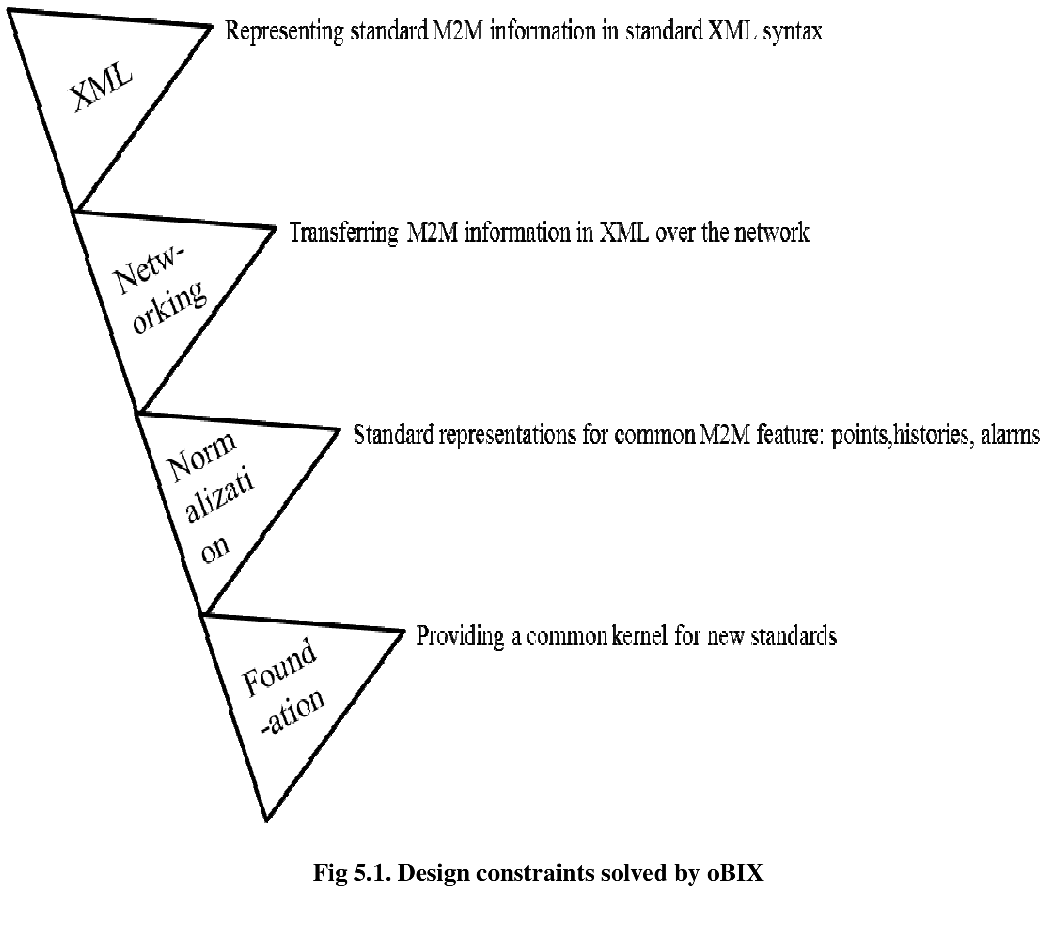
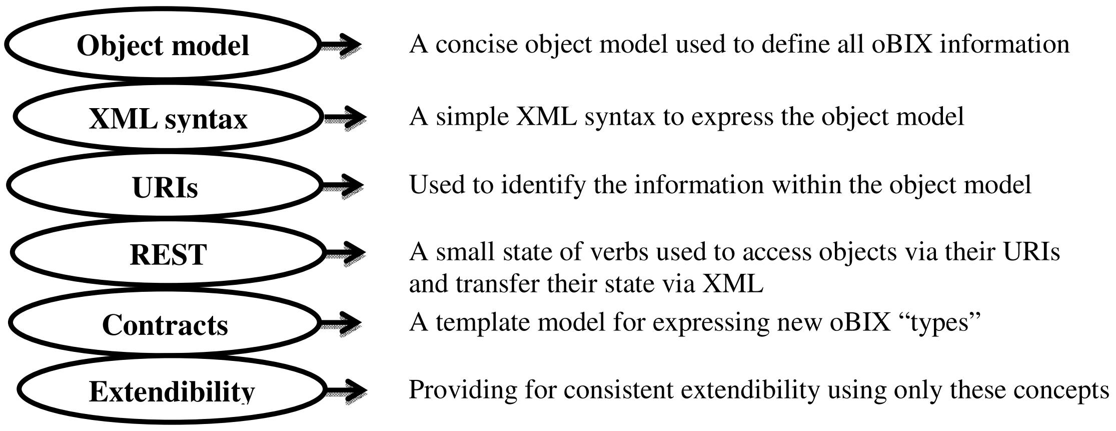

XML
Representing standard M2M information in standard XML syntax
Netw-
orking
Transferring M2M information in XML over the network
Norm
alizati
on
Standard representations for common M2M feature: points,histories, alarms
Found
-ation
Providing a common kernel for new standards

Fig 5.1. Design constraints solved by oBIX

Developing a common XML syntax for representing information is the principle requirement of oBIX. [19]

5.1 Architecture

The architecture of oBIX has [19]:



Object model	A concise object model used to define all oBIX information
XML syntax	A simple XML syntax to express the object model
URIs	Used to identify the information within the object model
REST	A small state of verbs used to access objects via their URIs and transfer their state via XML
Contracts	A template model for expressing new oBIX "types"
Extendibility	Providing for consistent extendibility using only these concepts

Fig 5.2. oBIX architecture [19]

The object model has small, fixed set of object types. These object types map one to one to an XML element type as seen in figure 5.3 below.

The objects can be identified either by names or hrefs. The names are tagged by label *displayName*. Href is used to attach an URI to a particular object. The URIs can be divided as base URIs required for the root object and relative URIs which always require normalization against the base URI. Each service consists of state variable and operations. An example of a typical XML format is shown below.

```
<objname="TestDevice"href="http://testbed.tml.hut.fi/obix/test/TestDevice/"displayName=
"Device for tests"

<intname="temperature"href="int/"displayName="Temperature"val="25"writable="true"/
>
```

Base URI: http://testbed.tml.hut.fi/obix/

Relative URI: http://testbed.tml.hut.fi/obix/test/TestDevice/

Object Model:

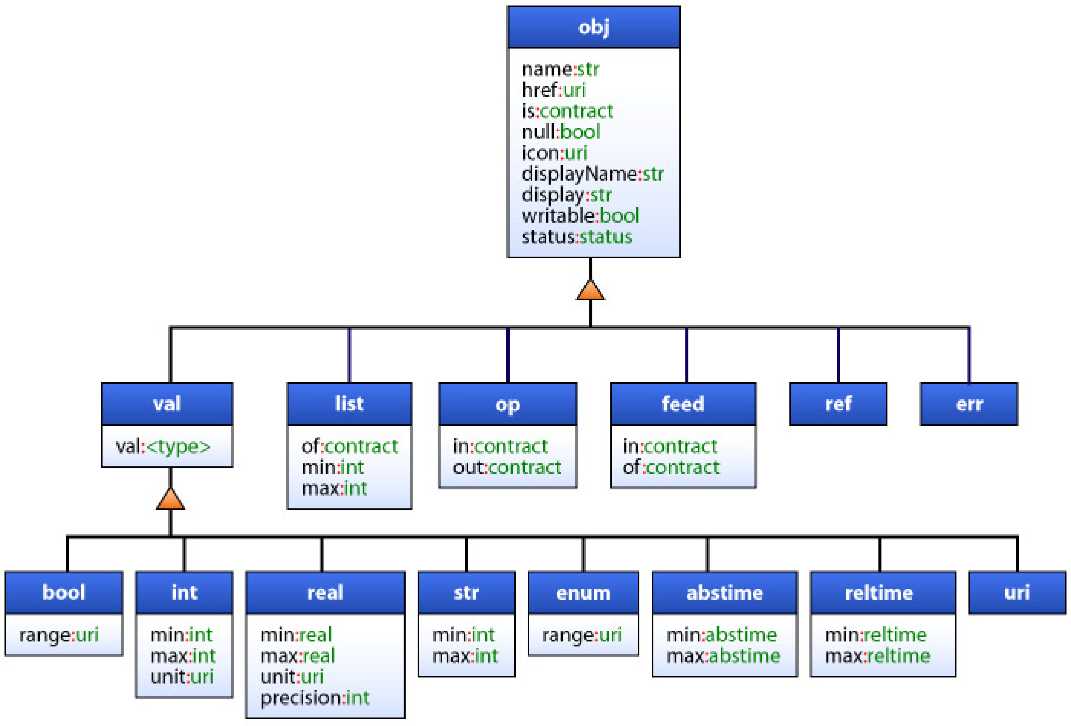


Fig 5.3. Object model of oBIX [19]

The information in oBIX is represented by a small, fixed set of primitives. The base abstraction of these primitives is called an object. An object has a URI and can have other objects in it. There are eight special kinds of value objects in which information can be stored. They are as follows with their contract definition:

- *bool*: stores a boolean value - true or false
  `<bool href="obix:bool" is="obix:obj" val="false" null="false"/>`

- *int*: stores an integer value
  `<int href="obix:int" is="obix:obj" val"0" null="false"/>`

- *real*: stores a floating point value
  `<real href="obix:real" is="obix:obj" val="0" null="false"/>`

- *str*: stores a UNICODE string
  `<str href="obix:str" is="obix:obj" val="" null="false"/>`

- *enum*: stores an enumerated value within a fixed range
  `<enum href="obix:enum" is="obix:obj" val="" null="true"/>`

- *abstime*: stores an absolute time value (timestamp)
  `<abstime href="obix:abstime" is="obix:obj" val="1970-01-01T00:00" null="true"/>`

- *reltime*: stores a relative time value (duration or time span)

```
<reltime    href="obix:reltime"    is="obix:obj"    val="PT0S"
null="false"/>
```

- *uri*: stores a Universal Resource Identifier
```
<uri href="obix:uri" is="obix:obj" val="" null="false"/>
```

The other object types are

- *list*: a special type of object which stores lists of other objects
```
<list href="obix:list" is="obix:obj" of="obix:obj"/>
```

- *op*: used to define a function
```
<op href="obix:op" is="obix:obj" in="obix:Nil" out="obix:Nil"/>
```

- *feed*: used to define a topic for a feed of events
```
<feed     href="obix:feed"     is="obix:obj"     in="obix:Nil"
of="obix:obj"/>
```

- *ref*: used to create out of document reference to another obix document
```
<ref href="obix:ref " is="obix:obj"/>
```

- *err*: a kind of a special object used to indicate an error
```
<err href="obix:err" is="obix:obj"/>
```

- *displayname*: It provides a localized human readable name of the object stored as a `xs:string`
```
<obj name="spaceTemp" displayName="Space Temperature"/>
```

- *icon*: provides a URI reference to a graphical icon which may be used to represent the object in an user agent.
```
<object icon="/icons/equipment.png"/>
```

- *writeable*: it specifies if an object can be written by the client. If false, then its just read-only.
```
<str name="userName" val="jsmith" writable="false"/>
<str name="fullName" val="John Smith" writable="true"/>
```

6 User Interface Development

A user interface is a bridge between a *user* and a *program* [20].To understand the "human visual system", it is necessary for a developer to present the information to the users in an interactive fashion. The designs and the presentation should always be understandable and pleasing to the eyes.

6.1 Factors Considered Before Layout Design

A design process should have a balance between technical functionality and visual elements. A good design is the one which can finish a task at hand without requiring unnecessary attention to it.

A design process should have functionality and user analysis aspects into consideration while developing an appropriate user interface. The definition of functionality here is assembling a list of the requirements of the users to be fulfilled by a system.

User analysis can be done by asking the following questions to the user:

- What he/she wants the system to do?
- How technical savvy he/she is?
- How the system should fit with their normal workflow?

6.1.1 Elements of User Centric Design

User centric design is a concept used during design stages keeping *user* in mind. It is used to enhance the experience of the user. The essential elements are:

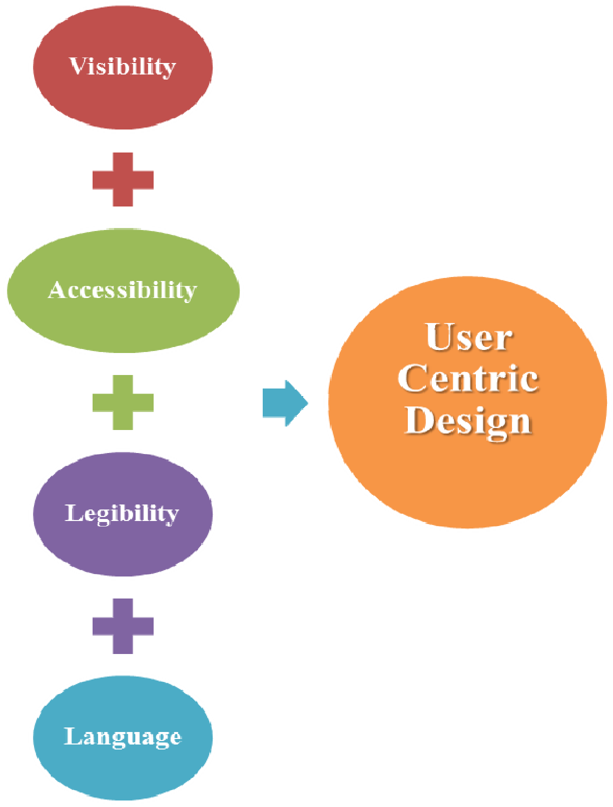


Fig 6.1. Elements of user centric design

In *accessibility*, the design should be able to help the user to find information quickly and easily. With respect to this, the data related to generation is placed on the left hand side in GUI and the data related to consumption is placed on the right hand side. Also the important aspect of "chunking" is considered here. As a result of which the logged data shows the level of consumption (eg: from low to high (green to red)). The *legibility* element helps in designing the text of the user interface. The text on display should be easily readable; a mixed combination of bold and italic letters is used to enhance the user reading experience. Capital letters are purposefully not used in the UI as they make reading difficult.

6.2 Selection of Colors and Shapes

Colors play an important part in designing the color scheme of a user interface. Different colors have different meaning and at times can convey a direct message to the user instead of the text, for example they can make us happy, sad, angry, nervous, comfortable, etc. Various colors have different meanings in different countries. Hence having a proper color combination (customizing country wise) along with the design layout is an important part of designing aspect.

Generally, *White* is considered best for all types of backgrounds due to its refreshing and superlative nature [21]. *Red* color represents action, passion, tension, etc. [21] [22]. *Blue* represents hope, stability and wisdom hence it is the second most popular color after black [22]. Websites having combination of blue and white are a instant hit. Viewing blue and red color together can have strain on the eyes as the lens has to adjust itself to focus on them [22]. *Yellow* is hardest of all on the eyes. If used in small amounts it is an attention seeker, cheerful and also increases the concentration. *Green* is highly compatible color with eyes [21]. It has a healing power in it and symbolizes prosperity and harmony [21] [22]. *Black* is the most famous among the designers. It determines the speed, excitement and is an instant attention grabber [21]. It is not appropriate, however to use it as a background color.

6.2.1 Use of Colors Related to UI Design

In this thesis work, a combination of all the above mentioned colors was made to represent the useful information to the user about his/ her energy consumption.

The Black color is used as the background (though not recommended). As this UI will be in operation continuously for a long period of time, those many pixels on the screen will be lighten for that particular time. This increases the power consumption of the display system to light those pixels. Hence from energy conservation point of view it is used as the background color.

Blue is used as background in combination with white for textboxes which display the values of generated and consumed energy. This helps in figuring out the text values easily.

Green, Yellow and Red colors are used in combination to inform the user about the region where his energy consumption is heading. Green signals that consumption is way below the limit and is in optimum level. Yellow signals that the consumption is increasing but still below the danger level. This will attract the user's attention and will help in taking the corrective measures. Red signals that the usage of energy has crossed the danger level and requires an immediate action from the user to reduce it. Now here the flashing of red color on the screen and reading of the value from the textbox simultaneously can be a difficult task. But a thought goes; if red color is flashing, it's better to see the region of danger rather than reading the values.

6.2.2 Use of Shapes Related to UI Design

The shape of the entire UI design is a *rectangle*. This is considered to be a world-wide accepted shape. The region which displays the colors green, yellow and red is of arc type. This is considered keeping in mind that an average user is familiar with a SPEEDOMETER of an automobile. It is the best tool to inform the users about their own actions. Also the color scheme embedded into it is selected considering that a user has knowledge of the SIGNAL system used on roads.

6.2.3 Components of UI design

The design consists of various *labels* used to display the energy readings (generated/consumed). Generally *labels* are used to display static text, titles, etc. The properties assigned with *labels* are *caption, backcolor/ forecolor, backstyle, font, alignment and multiline*.

The *timer* is used to refresh the consumed energy label to update the display so the user gets real time updates.

The programs done for testing of working of the UI color scheme and extraction of XML are attached in Appendix B.

The graphical user interface design developed after considering all the aspects mentioned above is as follows in Figure 6.2.

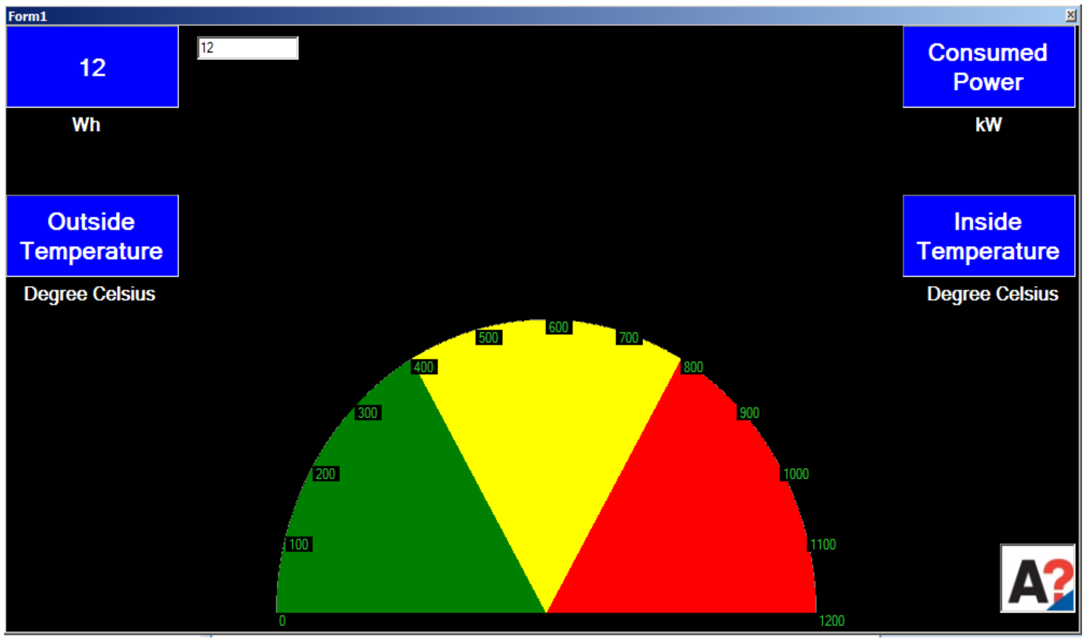


Fig 6.2. Graphical user interface layout

6.3 Use of Visual Basics and VB.Net

The developing tool used for the UI was Visual Basics 2008 due to its ease for programming and availability of various other tools for designing and considering the given time limit. The main advantage of using Visual Basics over other programming languages is that *what could be created in minutes using VB could take days to create in other languages*. Hence "Visual Basic is all about personal programming".

VB has a 3-step approach for creating programs:

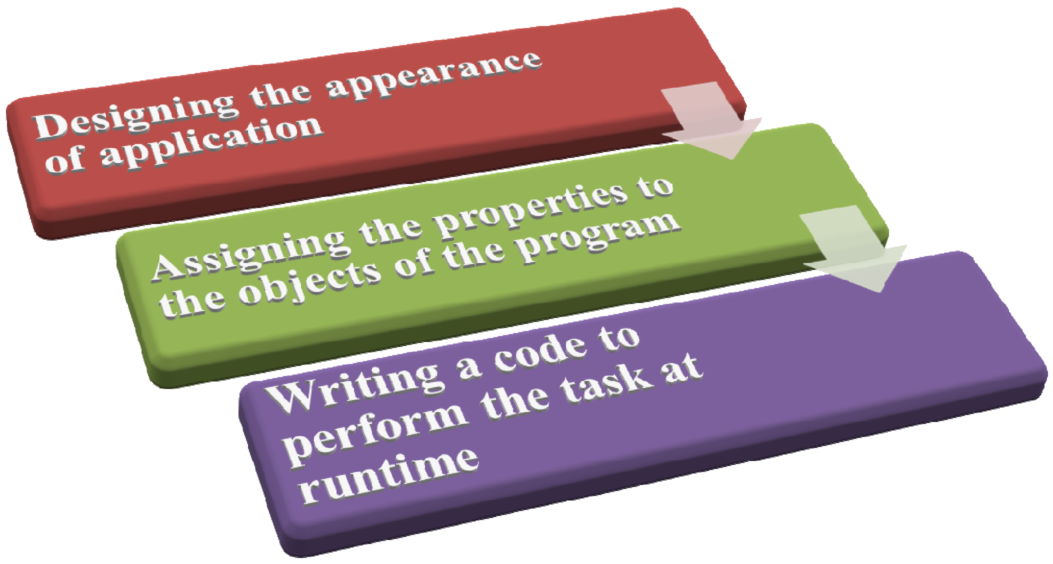

Designing the appearance of application
Assigning the properties to the objects of the program
Writing a code to perform the task at runtime

Fig 6.3. Creating program in Visual Basics

In this thesis work the UI design layout displays information related to energy generated and consumed in a house. This is displayed by connecting to a device (dummy device created using pulse generator to show the energy consumption).

The programming was done with the help of Visual Basics and VB.NET. The other languages used for Object Oriented Programming (OOP) are Action Script (Adobe), Ada, C++, Curl, Eiffel, JAVA, etc.

Here a small comparison between JAVA and VB.NET is made (used for Object Oriented Programming (OOP)) so that the reader can understand the choice of programming language.

Java and .NET have rich standard libraries and high-level frameworks for graphical user interfaces, database access, XML, and web development [23].

JAVA is considered to be more mature language with the ability of having cross-platform server side development. It is also used for mobiles and small devices application development, whereas .NET platform is more developer friendly and restricted only to the Win32 environment for developing rich client applications (but the Open Source project Mono is developing multi-platform runtime for .NET) [23]. But

.NET scores over JAVA in the department of multi-language support in the form of Visual Basic and C#. VB.NET has a more integrated development environment, as the IDE, runtime and server all come from Microsoft as a standard package, while Java is based on third-party tool and server providers [24]. Functionality wise, there is not much difference between the two languages when used for web applications development, but for desktop applications .NET has an advantage with Windows integration and other MS offerings with a set of ready-made components [24]. For clear understanding of the text; a small comparison table was made as follows:

Table 6.1. JAVA and .NET comparison [25]

Criteria	**JAVA**	**.NET**	**Comments**
Ease Of Use (Development Environment)	**	****	VB.net and C# are easier to use than J2EE
Scalability	***	**	Execute Java Code on Mainframe
Single Language Multiple Platforms	****	*	Java Can run on many platforms through the JVM
Multiple Languages Single Platform	*	****	VB,C#,J# all run in the same run-time environment
Reliability	**	****	VB/Com development in 1993
Performance	***	***	Equal Performance
Speed of development	*	***	VB code easiar to learn
Reuse	****	**	Deploy same code on multiple platforms and multiple projects
Open Standards	*****	*	Java, JVM are open standards

6.4 User Experience

To test the user experience, the GUI was shown in the below pattern as in Figure 6.4 to three-four users. Out of the selected users, two had spectacles required for reading. The other two users had normal eye sight. Their experiences were as follows:

User 1 experience: Sorting of generation and consumption data on two sides gave the user a clear idea of what he is looking at. The idea of a speedo indicator conveyed him the clear message about his consumption.

User 2 experience: The user had problems in reading the text on the UI and preferred the speedo indicator over the text to know about his consumption.

User 3 experience: The user demanded more information from the UI. In this case the user wanted to have the speedo indicator for individual devices and suggested to combine them on the home screen indicating the overall consumption indication.

User 4 experience: The user suggested an alternate solution of eliminating the speedo indicator by a larger font size text displaying the consumption data and having the color scheme embedded inside it.

Many of the above users were concerned about the user interface's multi-platform operations. One of them suggested developing the user interface in Open Source platform JAVA due to its extra features and cross platform compatibility and preferred developing the application in JAVA rather than in Microsoft Visual studio.

One of the user's demanded a more enhanced graphical experience and suggested us to use Adobe Flash coding which has more graphics oriented architecture.

Hence, our conclusions from the user experiences were:

- The UI provided the required information just when it is needed.
- The indicator scheme had a better understanding of the information and an alternative tool for the persons less interested in the text.
- Selection of black color for background grabbed the user's attention instantaneously.
- One user demanded more information from the UI, while others were satisfied with the information provided to them.

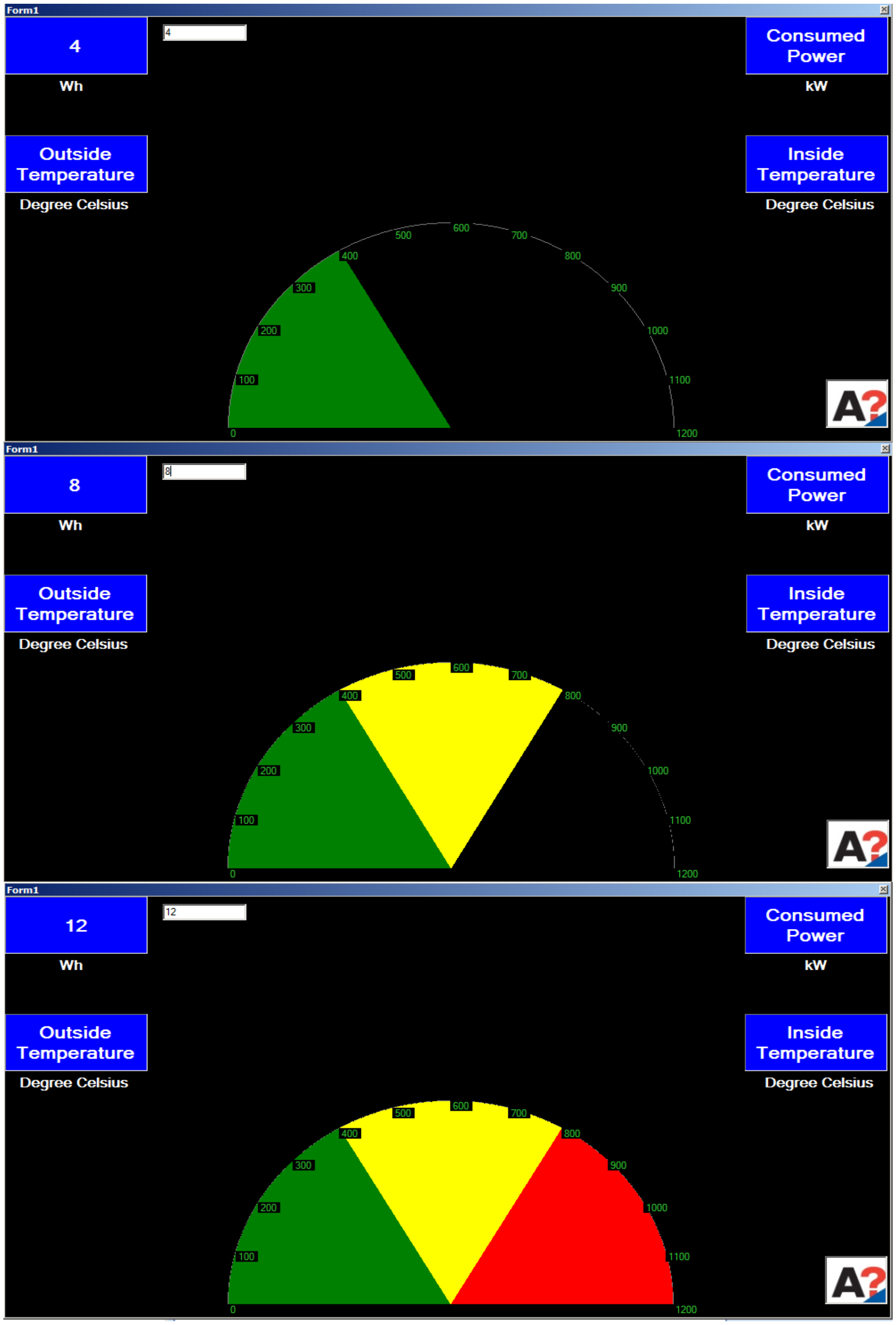


Fig 6.4. Display pattern of UI to the users

7 Configuring Measurement system

7.1 Configuring M-BUS

The M-Bus can be configured via MBCONF software. The master is connected to the USB port of the PC and the slaves are connected to the master through two-wire cable as shown in the figure below:

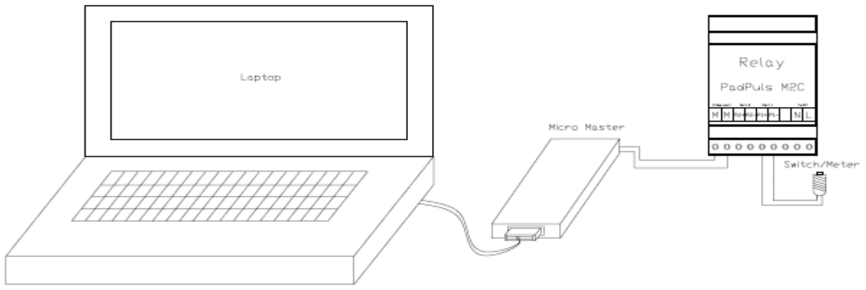
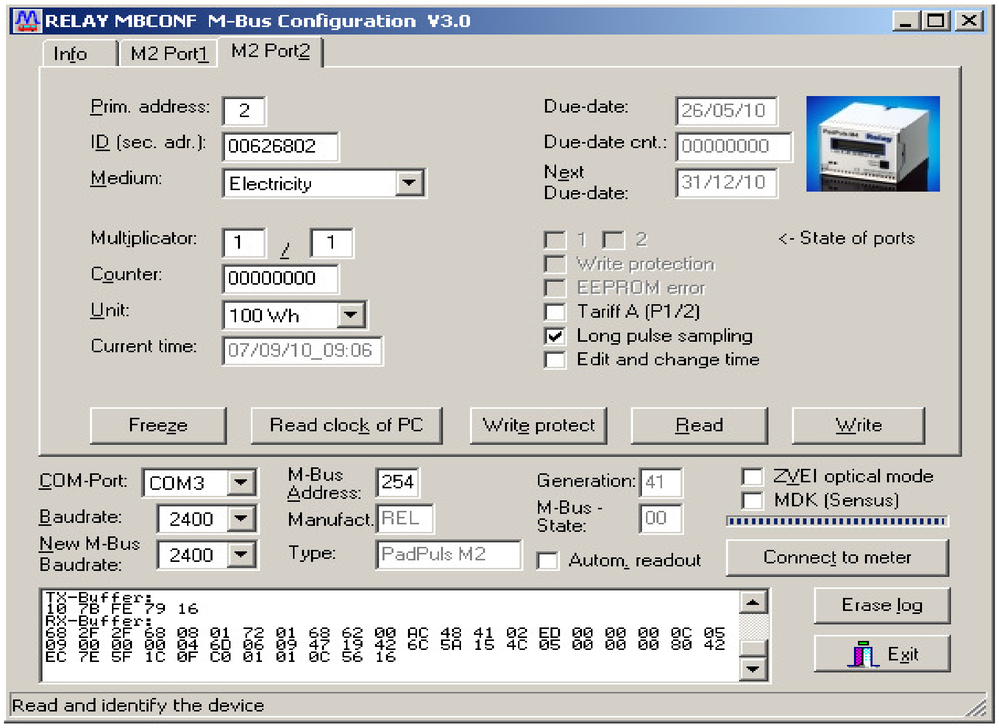


Fig 7.1. Connection of Micro-Master and PadPuls M2C to PC [26] [27]

The MBCONF software (as shown in Figure 7.2) is used to configure the Slave. It can program the slave for various monitoring values required to monitor. It is used to assign the primary address to the slave and also to assign the scaling factor of the monitored quantity. The M-Bus level converter is the M-Bus Micro Master that handles the task of converting the serial USB communication into UART levels and then into the M-Bus signals level.

In the following figure, it is observed that port 2 of the slave is being configured for *Medium: Electricity and has been assigned Prim. Address: 2, Multiplicator: 1:1 and the Unit: Wh*



Prim. address: 2
ID (sec. adr.): 00626802
Medium: Electricity
Multiplicator: 1 / 1
Counter: 00000000
Unit: 100 Wh
Current time: 07/09/10 09:06

Due-date: 26/05/10
Due-date cnt.: 00000000
Next Due-date: 31/12/10

[ ] 1 [ ] 2 <- State of ports
[ ] Write protection
[ ] EEPROM error
[ ] Tariff A (P1/2)
[✓] Long pulse sampling
[ ] Edit and change time

COM-Port: COM3
Baudrate: 2400
New M-Bus Baudrate: 2400
M-Bus Address: 254
Manufact. REL
Type: PadPuls M2
Generation: 41
M-Bus - State: 00
[ ] Autom. readout
[ ] ZVEI optical mode
[ ] MDK (Sensus)

```
TX-Buffer:
10 7B FE 79 16
RX-Buffer:
68 2F 2F 68 08 01 72 01 68 62 00 AC 48 41 02 ED 00 00 00 0C 05
09 00 00 00 04 6D 06 09 47 19 42 6C 5A 15 4C 05 00 00 00 80 42
EC 7E 5F 1C 0F C0 01 01 0C 56 16
```

Read and identify the device

Fig 7.2. MBCONF software [28]

7.2 Data Logging

The data logging is done at the heart of There Corporation's ThereGate system. After configuring the slave, the master is connected to ThereGate system. When the M-Bus Micro Master is attached into the ThereGate, special M-Bus driver software is launched. This driver knows how to communicate with the M-Bus bus and also knows how to decode M-Bus messages.

7.3 Data Extraction

The logged data can both be extracted and linked to the user interface in the two following ways:

- Direct XML linkage to the UI (this can be done with the help of oBIX system)
- Direct HTTP linkage using cURL

For using the XML format, a third party application called oBIX (as explained earlier) had been installed over the ThereGate platform and is not supported by There Corporation.

7.4 System Operation

The operation of the entire system has been represented in the following Figure 7.3. The process is explained in detail through the flowchart and its explanation.

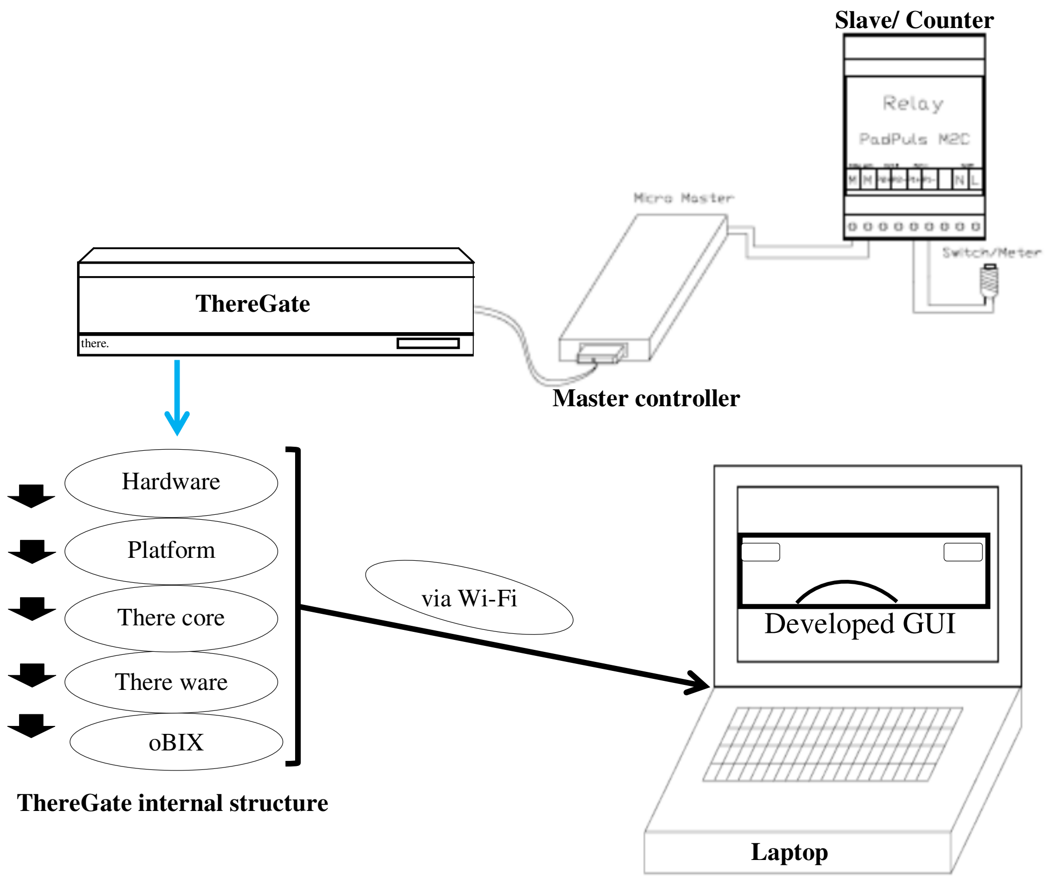


Fig 7.3. Setup for thesis

The switch is used as a replica of an energy meter which logs the change in the energy consumption of an electrical device and generates an equivalent amount of electrical pulses. This switch is connected to the counter (PadPuls M2C) which counts the pulses coming from the switch (energy meter) and transfers them to the master controller which converts them to the language compatible with ThereGate system. This connection from Master to ThereGate is done via the M-Bus protocol. The data is analyzed in ThereGate and then sent to oBIX server for displaying. Simultaneously, this data is displayed on ThereGate's own webpage. A secured connection between ThereGate and a computer is established via a Wi-Fi signal to access this data through a web browser installed on the computer.

7.4.1 System Flowchart

The earlier sections explained the development of the user interface and the configuration of M-Bus system. The following flowchart explains the working of the entire system when the user interface, ThereGate and M-bus are integrated together.

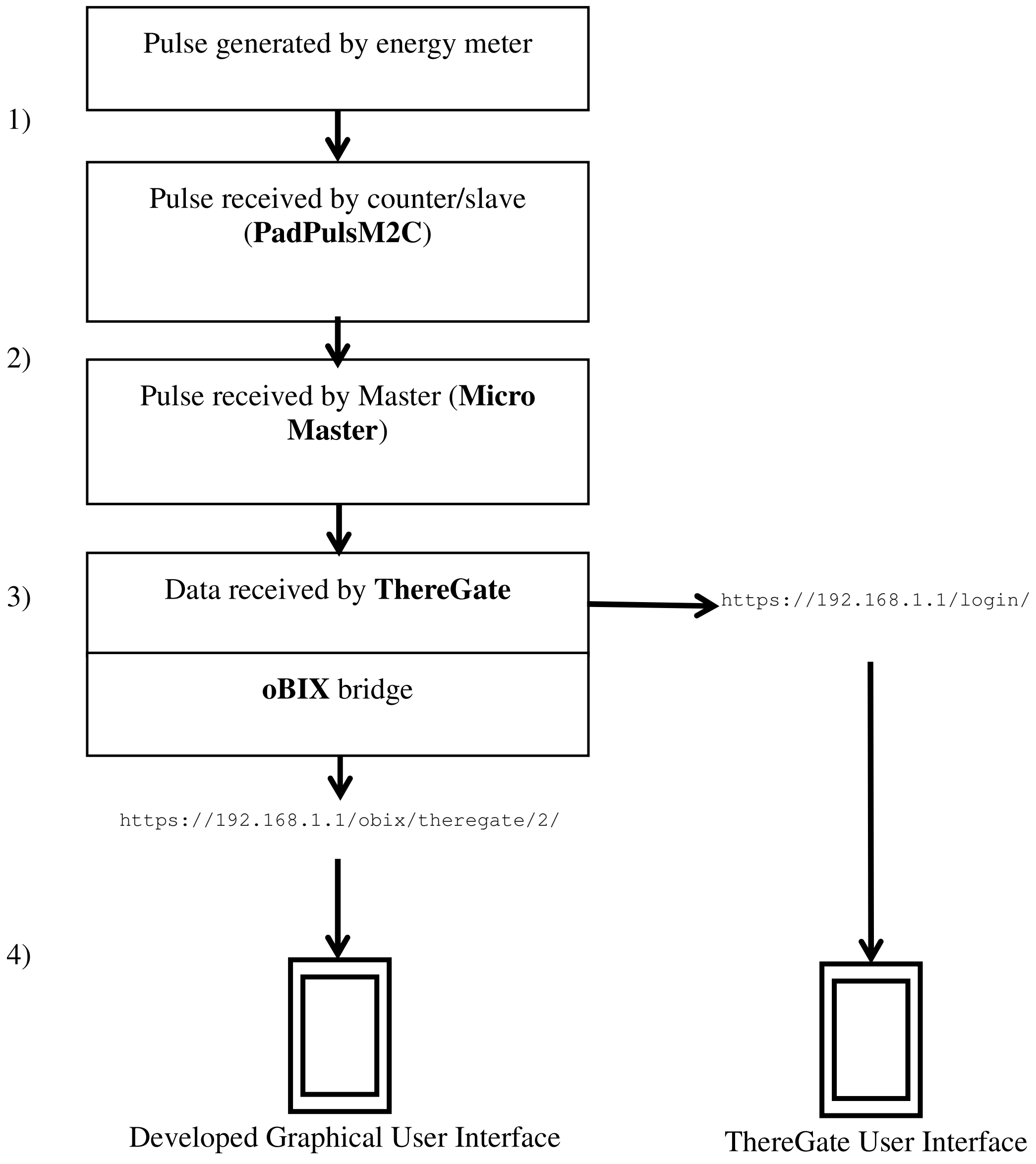

Pulse generated by energy meter
1)
Pulse received by counter/slave (PadPulsM2C)
2)
Pulse received by Master (Micro Master)
3)
Data received by ThereGate
oBIX bridge
https://192.168.1.1/login/
https://192.168.1.1/obix/theregate/2/
4)
Developed Graphical User Interface
ThereGate User Interface

Fig 7.4. System's functional flowchart

Operations

1. When a change in energy is sensed by the energy meter, it generates the electrical pulses equivalent to the change in energy and sends them to the pulse counter (PadPuls M2C/ slave).

2. The slave transfers the pulses towards the Micro-Master (master) which converts the M-bus signal to UART level and then to serial USB communication level. This signal is now compatible for ThereGate to read and is presented via the ThereCore software and can be accessed through an HTTP API or through the oBIX server.

3. Every ThereGate has its own web based user interface which can be accessed through the assigned URI.

4. The data is being displayed on the developed GUI via oBIX server as explained earlier.

The following picture shows the practical setup of the entire system.

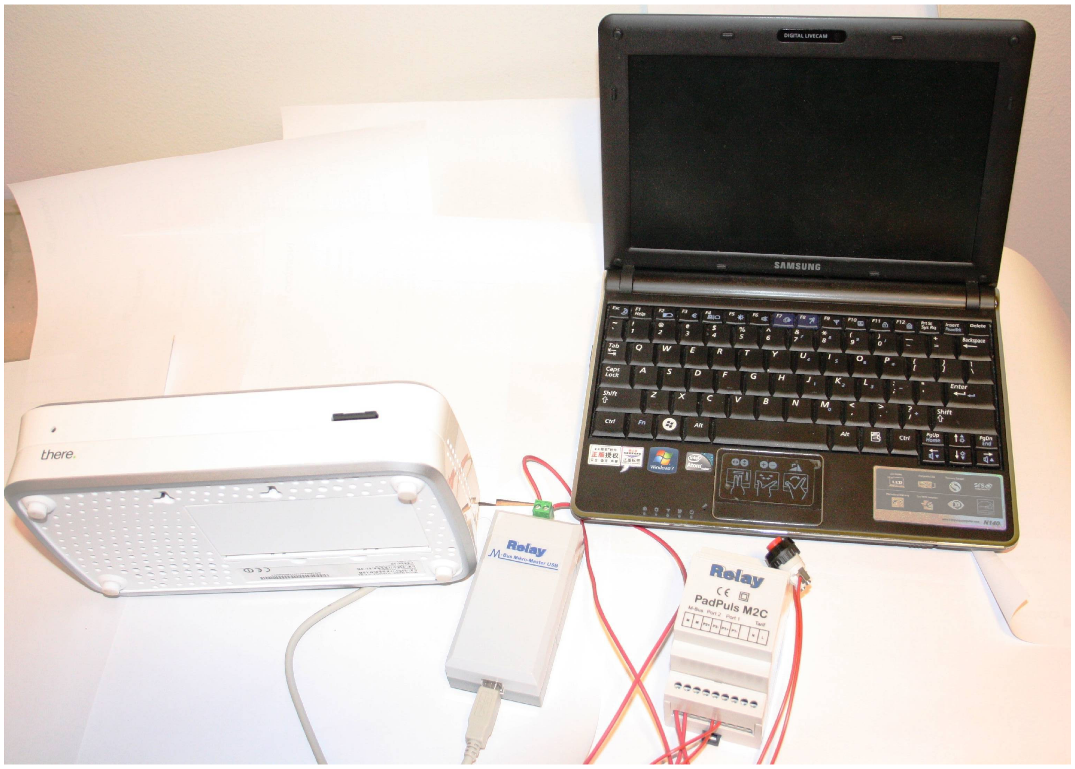


Fig 7.5. Practical setup

7.4.2 oBIX Programs

Hence, in order to communicate with the devices (read/write/execute a value of a service (object)) that are connected to ThereGate, we have to first find out the URI of that particular object/ service.

In this thesis work, our main task was to display the logged value and transfer it to the developed GUI. Therefore, only the following URI addresses are to be used in this thesis.

1) https://192.168.1.1/obix/

```
<objhref="https://192.168.1.1/obix/"is="obix:Lobby/obix/def/Sign
UpLobby/obix/def/DevicesLobby"xsi:schemaLocation="http://obix.or
g/ns/schema/1.0">
<ref name="devices" href="devices/" display="Device Info"/>
```

2) https://192.168.1.1/obix/devices/

```
<listhref="https://192.168.1.1/obix/devices/"displayName="Device
List"of="obix:ref"xsi:schemaLocation="http://obix.org/ns/schema/
1.0">
<ref href="/obix/theregate/2/" displayName="Electricity 1"/>
```

3) https://192.168.1.1/obix/theregate/2/

```
<objhref="https://192.168.1.1/obix/theregate/3/"displayName="Ele
ctricity 1" xsi:schemaLocation="http://obix.org/ns/schema/1.0">
<strname="TotalActiveImportEnergy"displayName="TotalActiveImport
Energy"href="/obix/theregate/2/AdvancedElectricityMeterService/T
otalActiveImportEnergy/"null="true"writable="true"val="15.000"/>
```

4) https://192.168.1.1/obix/theregate/2/AdvancedElectricityMeterService/TotalActiveImportEnergy/

```
<strname="TotalActiveImportEnergy"displayName="TotalActiveImport
Energy"href="https://192.168.1.1/obix/theregate/2/AdvancedElectr
icityMeterService/TotalActiveImportEnergy/"null="true"writable="
true"val="15.000"xsi:schemaLocation="http://obix.org/ns/schema/1
.0"/>
```

Here, the data from 4th URI (highlighted by Yellow) is to be extracted and displayed on the developed GUI. It is "Total Active Import Energy" object and its value is denoted by *val* is to be displayed. The rest of the URIs are the base URIs to access the 4th URI.

The following oBIX programs explain the detailed back end programming work done on the oBIX server for fetching the monitoring data. The section first shows the program related to fetching followed by its explanation.

Program 1:

```
<objhref="https://192.168.1.1/obix/"is="obix:Lobby/obix/def/SignUpLobb
y/obix/def/DevicesLobby"xsi:schemaLocation="http://obix.org/ns/schema/
1.0">

 <ref name="watchService" href="watchService/"is="obix:WatchService"/>

 <ref name="about" href="about/" is="obix:About"/>

 <op name="batch" href="batch/" in="obix:BatchIn"out="obix:BatchOut"/>
 <ref name="alarms" href="alarms/" is="/obix/def/alarm:AlarmService
obix:AlarmSubject" display="Alarm Service" displayName="Alarm
Service"/>

 <ref name="devices" href="devices/" display="Device Info"/>

 <op name="signUp" href="signUp/" in="obix:obj" out="obix:obj"/>
</obj>
```

Explanation:

The program is called an *obix: lobby* where vendors place vendor specific objects used for service and data discovery. The other objects like; *obix: watchService* which provides the functionality to user to add new watch to the server to get the data according to the time interval specified by the user, *Obix: batch* is an aggregation of read, write and invoke requests as a part of standard operation, *obix: devices* display the names of the devices connected to the server and *obix: about* is a standardized list of summary information about an oBIX server, for example:

```
<obj href="obix:About">

      <str name="obixVersion"/>

      <str name="serverName"/>
      <abstime name="serverTime"/>
      <abstime name="serverBootTime"/>

      <str name="vendorName"/>
      <uri name="vendorUrl"/>

      <str name="productName"/>
      <str name="productVersion"/>
      <uri name="productUrl"/>

</obj>
```

Program 2:

```
<list href="https://192.168.1.1/obix/devices/" displayName="Device
List" of="obix:ref"xsi:schemaLocation="http://obix.org/ns/schema/1.0">
 <refhref="/obix/test/TestDevice/"name="TestDevice"displayName="Device
 for tests"/>

  <ref href="/obix/theregate/1/" displayName="HomeControlCenter 1"/>
  <ref href="/obix/theregate/2/" displayName="Electricity 1"/>

  <ref href="/obix/theregate/3/" displayName="Electricity 2"/>

</list>
```

Explanation:

The URI lists down the devices that are connected to the oBIX server. "/theregate/2 or /theregate/3/" are identified as the devices connected to the oBIX server (on PadPuls M2C, they are represented as Port 1 and Port 2 respectively) with their functional names Electricity1 and Electricity 2 configured via MBCONF software. In the next program we shall have a look at the URI for device "/theregate/2"

Program 3:

```
<objhref="https://192.168.1.1/obix/theregate/2/"displayName="Electrici
ty 1" xsi:schemaLocation="http://obix.org/ns/schema/1.0">

 <strname="type"displayName="DeviceType"href="/obix/theregate/3/type/"
val="EnergyMeter:1"/>

<strname="manufacturer"displayName="Manufacturer"href="/obix/theregate
/3/manufacturer/" val="Relay GmbH"/>

 <str name="model" displayName="Model" href="/obix/theregate/3/model/"
val="PadPuls_M2C"/>

<strname="creation_time"displayName="CreationTime"href="/obix/theregat
e/3/creation_time/" val="2010-09-25T14:41:49Z"/>

<objname="service"displayName="BasicDeviceService"href="/obix/theregat
e/3/BasicDeviceService/">
</obj>

<objname="service"displayName="AdvancedElectricityMeterService"href="/
obix/theregate/3/AdvancedElectricityMeterService/">
</obj>
```

Explanation:

"/theregate/2/" URI represents Port1 which gives detailed information related to its manufacturer name, model, type (configured) and services it provides. The device provides two basic services to the user. They are:

Basic Device Service, providing information related to version, name, service availability, battery indication, etc. and **Advanced Electricity Meter Service**, providing information on energy usage, power consumption and various operations performed to get these values.

Program 4:

```
<objname="service"displayName="AdvancedElectricityMeterService"href="h
ttps://192.168.1.1/obix/theregate/2/AdvancedElectricityMeterService/"x
si:schemaLocation="http://obix.org/ns/schema/1.0">

<intname="version"displayName="ServiceVersion"href="/obix/theregate/2/
AdvancedElectricityMeterService/version/" val="1"/>

<strname="TotalActiveImportEnergy"displayName="TotalActiveImportEnergy
"href="/obix/theregate/2/AdvancedElectricityMeterService/TotalActiveIm
portEnergy/" null="true" writable="true" val="15.000"/>

<strname="ActiveImportPower"displayName="ActiveImportPower"href="/obix
/theregate/2/AdvancedElectricityMeterService/ActiveImportPower/"null="
true" writable="true" val=""/>

<opname="GetAttributeApproximation"displayName="GetAttributeApproximat
ion"href="/obix/theregate/2/AdvancedElectricityMeterService/GetAttribu
teApproximation/"in="/obix/theregate/2/AdvancedElectricityMeterService
/GetAttributeApproximation/inContract/" out="obix:str">
```

```
<opname=”SM_GetAttributeHistoryXml”displayName=”SM_GetAttributeHistory
Xml”href=”/obix/theregate/2/AdvancedElectricityMeterService/SM_GetAttr
ibuteHistoryXml/” in=”obix:str” out=”obix:str”/>

<op name=”SM_SetAttributeLogging” displayName=”SM_SetAttributeLogging”
href=”/obix/theregate/2/AdvancedElectricityMeterService/SM_SetAttribut
eLogging/”in=”/obix/theregate/2/AdvancedElectricityMeterService/SM_Set
AttributeLogging/inContract/”>

<op name=”SM_GetAttributeLogging” displayName=”SM_GetAttributeLogging”
href=”/obix/theregate/2/AdvancedElectricityMeterService/SM_GetAttribut
eLogging/” in=”obix:str” out=”obix:str”/>

<op name=”SM_GetAttribute” displayName=”SM_GetAttribute”
href=”/obix/theregate/2/AdvancedElectricityMeterService/SM_GetAttribut
e/”in=”/obix/theregate/2/AdvancedElectricityMeterService/SM_GetAttribu
te/inContract/”out=”/obix/theregate/2/AdvancedElectricityMeterService/
SM_GetAttribute/outContract/”>

<opname=”SM_SetAttribute”displayName=”SM_SetAttribute”href=”/obix/ther
egate/2/AdvancedElectricityMeterService/SM_SetAttribute/”in=”/obix/the
regate/2/AdvancedElectricityMeterService/SM_SetAttribute/inContract/”>

<opname=”SM_PV_GetXmlFile”displayName=”SM_PV_GetXmlFile”href=”/obix/th
eregate/2/AdvancedElectricityMeterService/SM_PV_GetXmlFile/”in=”/obix/
theregate/3/AdvancedElectricityMeterService/SM_PV_GetXmlFile/inContrac
t/” out=”obix:str”>

<opname=”SM_PV_UploadXmlFile”displayName=”SM_PV_UploadXmlFile”href=”/o
bix/theregate/2/AdvancedElectricityMeterService/SM_PV_UploadXmlFile/”
in=”/obix/theregate/2/AdvancedElectricityMeterService/SM_PV_UploadXmlF
ile/inContract/”>

<opname=”SM_PV_Start”displayName=”SM_PV_Start”href=”/obix/theregate/2/
AdvancedElectricityMeterService/SM_PV_Start/”
</obj>

<strname=”TotalActiveImportEnergy”displayName=”TotalActiveImportEnergy
”href=”https://192.168.1.1/obix/theregate/2/AdvancedElectricityMeterSe
rvice/TotalActiveImportEnergy/” null=”true”
writable=”true”val=”15.000”xsi:schemaLocation=”http://obix.org/ns/sche
ma/1.0”/>
```

Explanation:

The URI displays a string name “Total Active Import Energy” which resembles to the amount of energy used by a device connected to the oBIX server via PadPuls M2C. The attributes null, writable and val give information about the status of the device and the recorded value. The `null=”true”` represents that the connected object has some value to display while `writable = “true”` states the value that can be edited by the user and `val= “15.000”` represents the recorded value (in this case the pulses logged from the pulse generator switch). It is this value which will be displayed on the developed graphical user interface.

The URIs mentioned with “`op`” are the operations which actually perform the process of getting the data related to energy consumption. The operations have an input and output contract which specifies some terms or objectives that the program should perform while fetching the data. They are shown as follows:

Program 5:

```
<objname="inContract"href="https://192.168.1.1/obix/theregate/2/Advanc
edElectricityMeterService/GetAttributeApproximation/inContract/"xsi:sc
hemaLocation="http://obix.org/ns/schema/1.0">
      <str name="StartTime"/>
      <str name="EndTime"/>
      <int name="Interval"/>
      <str name="AttributeName"/>
      <str name="Approximation"/>
      <int name="MaxDeviation"/>
      <int name="MaxSamples"/>
</obj>
```

Explanation for program 5:

As mentioned above, the op GetAttributeApproximation has an ‘in contract’ which specifies the data like Start time, End time, Attribute time, etc. to be fetched while extracting data.

7.4.3 UI Program

Program 6:

```
If (i_powerValue >= 1) And (i_powerValue < 5)

ElseIf (i_powerValue >= 5) And (i_powerValue < 9)

ElseIf (i_powerValue >= 9) And (i_powerValue <= 12)
```

Explanation for program 6:

The UI picks the data and scans through the above conditions. Here for the testing basis, we restricted the input data to 12 Wh, but can be expanded to further level by modifying the formulae used to show the speedo indicator scheme. The rest of the program can be found in Appendix B.

8 Discussion

The objectives set in the initial part of the report were reviewed carefully while designing the new UI. Also the comparison of two widely used programming languages helped in selecting the best and most simple language for programming. The questions mentioned in the user centric designs were set as benchmarks to filter the available information. After filtering, various UI sketches were made and the final UI design was selected keeping in mind that our end user will be a common man who just needs the information about the energy consumption. Also the components and schemes were designed keeping in mind that the user knows about basic simple things with regard to the system.

The data transfer system, ThereGate was used keeping in mind that the user has to purchase the existing market available protocol devices and install them in their home. The platform itself embeds the devices into it and starts recording the data. The M-Bus communication protocol was used due to the fact that there was technical assistance available from There Corporation's engineers.

The other systems like X-10 or ZigBee or KNX were not used due to the following reasons, X-10 is around for a long time and requires no extra wiring for data transfer, but has relatively low data transfer speed (20b/s) thus making it operational only for on/off commands given by push buttons and has security and reliability issues. KNX has high installation cost and requires dedicated twisted pair cables to be laid which increases the complexity of the system. It has however, on other hand the highest security related to channel and authentication of the system. ZigBee being a wireless communication protocol is easy to install, but suffers from the drawback of requiring its own infrastructure which ultimately increases the cost as compared to ThereGate system.

The other Open Source platforms like the Niagara framework and Sedona framework were not considered before using the oBIX platform. The main reason behind this was the support for oBIX from the Media technology Lab at Aalto University and no support for the other technologies. It is not sure whether the other platforms are superior to oBIX or not and needs further studies to explore their use over the ThereGate platform. Also these platforms including oBIX are not supported by the There Corporation.

Evaluation of the developed design from a user perspective by having the user experiences helped in opening the doors for new ideas to flow in and made us think of how to make the user interface more user friendly and how complex things can be represented in simple formats.

The use of Visual Basics 2008 as a tool for designing and programming the user interface was a helpful task as the package had various inbuilt tools which were easy to understand and use. As compared to JAVA, where you have to write a small code for developing a form, label and color them, VB provided these tools integrated in it. Hence VB was an obvious choice considering the thesis timeline.

A SWOT analysis was done to evaluate the project work. In the SWOT analysis, various strengths, drawbacks, future opportunities and threats related to both the GUI and the protocol are discussed.

The entire setup has strengths in the area of the M-bus protocol which has a simple configuration, low component and installation cost, has a dedicated user group for development and is a standardized protocol in Europe. The user interface design has advantages due to a single window viewing concept which helps in viewing the data on a single form in a user friendly format. Also the use of the oBIX platform saved us the tedious task of programming due to dedicated URIs to access the recorded data.

The drawbacks of using M-Bus are its slow data transfer rates as compared to the new developed protocols. At times the single window viewing concept can display data related to only one utility and may need to have other windows to display the data from other utilities. Compared to VB and VB.Net, there are other programming languages like JAVA and Adobe Flash which can be used for better cross platform compatibility and enhanced graphical experience.

In future the changes can be made to the user interface to make it more interactive and more user friendly. The users can arrange the components in the UI layout according to their needs and convenience. Also the user interface can have multiple windows embedded into one to display different entities. More advanced protocols can be used for remote monitoring. There are many manufacturers who are building mobile applications to control and monitor these devices from mobile phone via SMS, GPRS, 3G, Wi-Fi, etc.

The threat to this model can be seen if the user is installing the monitoring devices for the first time. Other protocols like Z-wave or ZigBee have devices which cover entire monitoring range and can be installed at once for the first time users.

9 Summary

In the future home monitoring systems are going to have more complex architectures. This means there will be more information available to the user. Representing this information graphically in an understandable and interactive manner to the user is an important task for a graphic design engineer as the graphical user interface is the face of any system. It helps in understanding the user; operations, important information displayed by a system. Hence designing, customizing and presenting the data in an appropriate way is necessary. The other aspects like color schemes or shapes and sizes play an important role in conveying the right message to the user.

The oBIX framework and data logger used here (ThereGate) being a multi-language integration platform (supports M-bus, ModBus (now), ZigBee and Z-Wave) encourages more developers to built third party applications and integrate with them to benefit the end-user. Much information was not allowed to be published due to the confidentiality of the documents.

Due to continuous developments in various communication protocols and initiative of various companies to integrate these protocols together on a common platform, the market share for home monitoring systems is going to increase compared to industrial monitoring systems.

References

[1] Tobias Hochwallner, Lukas Lang, *Approaches for monitoring and reduction of energy consumption in the home*, Faculty of Informatics, Vienna University of technology, June 2009, pp. 13-15.

[2] Home Energy Metering.COM, *Home Energy Monitoring Systems* [Online], Accessed on: 2010-09-02, Available: http://www.home-energy-metering.com/home-energy-monitor.html

[3] Topbits, tech community [Online], Accessed on 2010-07-17, Available: http://www.tech-faq.com/osi-model.html

[4] OSI model- Computer Dictionary Definition, Your Dictionary.com [Online], Accessed on 2010-07-17, Available: http://www.yourdictionary.com/computer/osi-model

[5] Nokia, *Nokia Home Control Centre*, December 2008, pp. 3, 8-10.

[6] Euro X10, X10 info [Online], Accessed on 2010-07-15, Available: http://www.tech-faq.com/osi-model.html

[7] Z-Wave, Z-Wave: The new standard in wireless remote control [Online], Accessed on 2010-07-15, Available: http://www.z-wave.com/modules/AboutZ-Wave/

[8] ZigBee tutorials, Tutorials-Reports.com [Online], Accessed on 2010-07-15, Available: http://www.tutorial-reports.com/wireless/zigbee/tutorial.php

[9] ZigBee tutorials, Tutorials-Reports.com [Online], Accessed on 2010-07-15, Available: http://www.tutorial-reports.com/wireless/zigbee/zigbee-architecture.php

[10] ZigBee tutorials, Tutorials-Rreports.com [Online], Accessed on 2010-07-15, Available: http://www.tutorial-reports.com/wireless/zigbee/zigbee-device-types.php

[11] Gerard O' Driscoll, *The essential guide to home networking technologies*, Prentice Hall PTR, Upper Saddle River, NJ 07458 (2001), pp. 76-78, 80-81, 302-304, 314-315.

[12] OSGi, *OSGi in nutshell* [Online], Accessed on 2010-07-16, Available: http://gravity.sourceforge.net/servicebinder/osginutshell.html

[13] KNX, *Introduction* [Online], Accessed on 2010-07-17, Available: http://www.knx.org/knx-standard/introduction/

[14] KNX, *Configuration modes* [Online], Accessed on 2010-07-17, Available: http://www.knx.org/knx-standard/configuration-modes/

[15] KNX, *Communication media* [Online], Accessed on 2010-07-17, Available: http://www.knx.org/knx-standard/communication-media/

[16] Conclusion, *X10 info* [Online], Accessed on 2010-07-15, Available: http://www.eurox10.com/Content/x10conclusion.htm

[17] Stream Measurement Limited, *30 Excellent Reasons why to choose MBus*, www.stream-measurement.com

[18] M-Bus, *the M-Bus: A Documentation rev. 4.8* [Online], Accessed on 2010-07-18, Available: http://www.m-bus.com/mbusdoc/default.php

[19] OASIS, *oBIX 1.0*, Committee specifications 01, December 2009.

[20] Webopedia, *user interface* [Online], Accessed on 2010-08-03, Available: http://www.webopedia.com/TERM/U/user_interface.html

[21] Web Design Library, *Importance of colors in website design* [Online], Accessed on 2010-08-03, Available: http://www.webdesign.org/web-design-basics/color-theory/importance-of-colors-in-web-site-design.14448.html

[22] Cyberindian.com, *Importance of colors in web design, colors and their meaning* [Online], Accessed on 2010-08-04, Available: http://www.cyberindian.com/web-designing/importance-of-color-in-web-design.php

[23] Marc Evelyn, *.NET versus JAVA, and the future of C++* [Online], Accessed on 2010-10-08, Available: http://www.vex.net/~marc/resources/dotnetvsjava.shtml

[24] Answers.com, *What is the difference between .Net and Java* [Online], Accessed on 2010-10-08, Available:
http://wiki.answers.com/Q/What_is_the_difference_between_.Net_and_Java

[25] Suresh Ramchandran, *Microsoft .Net vs. J2EE*, Duke Energy, slide no. 8

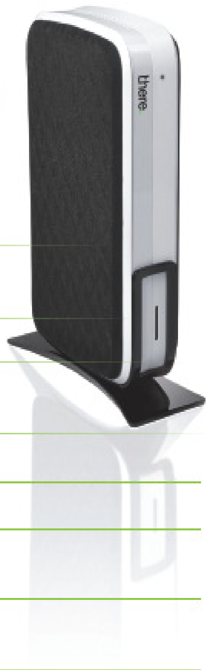
[26] Relay Flyer, *M-Bus Micro-Master* [Online], Accessed on 2010-07-26, Available: http://relay.de/pub/Anleitungen/MR003USB_DE.PDF

[27] Relay Flyer, *PadPuls M2* [Online], Accessed on 2010-07-26, Available: http://relay.de/pub/Anleitungen/IM003GC_E.PDF

[28] Relay software, *MBCONF*.

Appendix A

there.gate™ TG800G & TG800GZ specifications



Ethernet Ports	4 x Gbps LAN Ports. 1 x Gbps WAN Port.
VLAN	802.1 Q.
Wireless network	802.11 b/g/n, 300 Mbps transfer rate. 2 internal antennas.
Sensitivity	-70dBm (11g @54Mbps).
Security	TPM (Trusted Platform Module), v. 1.2 specified by Trusted Computing Grou
HAN Radio	Integrated Z-Wave controller Additional radio interfaces may be connected to USB expansion ports.
USB Ports	4 x USB 2.0 full speed host. 1 of the USB ports is powered running ThereGate on battery power.
Memory	2 GB internal storage standard, may be ordered with up to 16 Gb. End user accessible SD card reader for extension. 256 MB DDR2 RAM.
CPU	533 Mhz Broadcom 4718 SOC with integrated MIPS74k CPU.
Power Adapter	Wall Mount. AC input: 90V .. 264 V, 47 ~ 63 Hz. DC output: 5 V, 4 A max current.
Battery	8 x AA batteries, no recharging in ThereGate. Battery backup possibilyty for ThereGate operation during power failures.
RTC (Real Time Clock)	72 hours minimum run time without mains power or batteries.
Cooling	Passive cooling (no fan).
Indicators	14 LED:s.
Size	250mm x 139mm x 50mm (without canvas panels).
Weight	~500 g.
Operating Environment	0 ~ +40 ºC 10 ~ 90 % Relative Humidity, Non Condensing.
Storage Environment	-20 ~ +60 ºC 10 ~ 90 % Relative Humidity, Non Condensing.
Network security	Kernel firewall, Network Address Translation (NAT), Port forwarding, MAC address filtering
Operating system	Linux, Kernel 2.6
Antenna	Internal multi band antenna.
UMTS Module (G-model only)	BandRich M250V. SIM holder under battery cage.
WCDMA (G-model only)	HSUPA 2Mbps uplink modem operation (SW upgradable to 5.7Mbps). HSDPA 7.2Mbps downlink modem operation
E-GPRS (G-model only)	850/900/1800/1900 MHz.

Appendix B

1) UI working code:

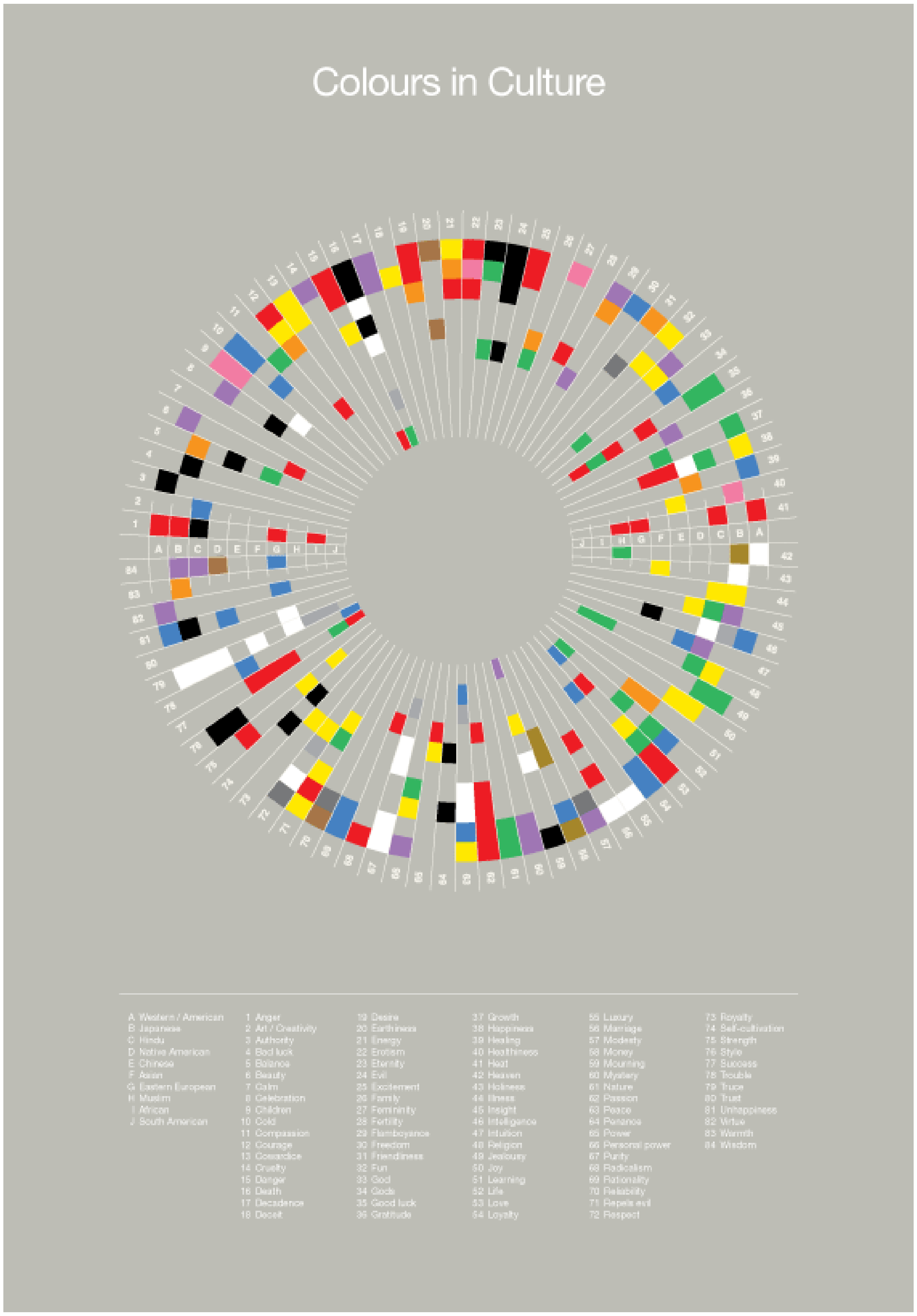
```
 Public Class Form1

    Private Sub Form1_Paint(ByVal sender As Object, ByVal e As
System.Windows.Forms.PaintEventArgs) Handles Me.Paint

        Dim gf As Graphics
        gf = Me.CreateGraphics
        gf.DrawArc(Pens.Gray, 250, 250, 500, 500, 0, -180)

    End Sub

Dim FILE_NAME As String = "C:\Users\mukul\Desktop\numbers.txt"
Dim objReader As System.IO.StreamReader = New
System.IO.StreamReader(FILE_NAME)

Private Sub Label21_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Label20.Click

        Dim gf As Graphics
        gf = Me.CreateGraphics

If objReader.Peek() >= 0 Then

            Label21.Text = objReader.ReadLine

End If

        Dim i_powerValue As Integer = CInt(Label21.Text)

If (i_powerValue >= 1) And (i_powerValue < 5) Then

            Dim i_anglevalue As Integer = CInt(Label20.Text) * 15

            Dim brushcolor1 As New SolidBrush(Color.Black)
            gf.FillPie(brushcolor1, 250, 250, 500, 500, -180, 180)

            Dim brushcolor2 As New SolidBrush(Color.Green)
            gf.FillPie(brushcolor2, 250, 250, 500, 500, -180,
i_anglevalue)


ElseIf (i_powerValue >= 5) And (i_powerValue < 9) Then

            Dim i_anglevalue As Integer = CInt(Label21.Text) * 15 - 60

            Dim brushcolor1 As New SolidBrush(Color.Black)
            gf.FillPie(brushcolor1, 250, 250, 500, 500, -180, 180)

            Dim brushcolor2 As New SolidBrush(Color.Green)
            gf.FillPie(brushcolor2, 250, 250, 500, 500, -180, 60)

            Dim brushcolor3 As New SolidBrush(Color.Yellow)
            gf.FillPie(brushcolor3, 250, 250, 500, 500, -120,
i_anglevalue)

ElseIf (i_powerValue >= 9) And (i_powerValue <= 12) Then

           Dim i_anglevalue As Integer = CInt(Label21.Text) * 15 - 120
```

```
            Dim brushcolor1 As New SolidBrush(Color.Black)
             gf.FillPie(brushcolor1, 250, 250, 500, 500, -180, 180)

            Dim brushcolor2 As New SolidBrush(Color.Green)
             gf.FillPie(brushcolor2, 250, 250, 500, 500, -180, 60)

            Dim brushcolor3 As New SolidBrush(Color.Yellow)
             gf.FillPie(brushcolor3, 250, 250, 500, 500, -120, 60)

            Dim brushcolor4 As New SolidBrush(Color.Red)
             gf.FillPie(brushcolor4, 250, 250, 500, 500, -60,
i_anglevalue)

End If

      End Sub

             End Class
```

2) XML extraction code:

```
Imports System
Imports System.Net
Imports System.IO

Module Module1
    Sub Main()

        Dim URL As String
        URL =
"https://192.168.1.1/obix/theregate/2/AdvancedElectricityMeterService/
TotalActiveImportEnergy/"

        Dim request As WebRequest
        request = WebRequest.Create(URL)

        request.Credentials = CredentialCache.DefaultCredentials

        Dim response As HttpWebResponse = CType(request.GetResponse(),
HttpWebResponse)

        Console.WriteLine(response.StatusDescription)

        Dim dataStream As Stream = response.GetResponseStream()

        Dim reader As New StreamReader(dataStream)

        Dim responseFromServer As String = reader.ReadToEnd()

        Console.WriteLine(responseFromServer)

        reader.Close()

        dataStream.Close()

        response.Close()

        Console.ReadLine()

End Sub

       End Module
```

Appendix C


Colours in Culture
A Western / American
B Japanese
C Hindu
D Native American
E Chinese
F Asian
G Eastern European
H Muslim
I African
J South American
1 Anger
2 Art / Creativity
3 Authority
4 Bad luck
5 Balance
6 Beauty
7 Calm
8 Celebration
9 Children
10 Cold
11 Compassion
12 Courage
13 Cowardice
14 Cruelty
15 Danger
16 Death
17 Decadence
18 Deceit
19 Desire
20 Earthiness
21 Energy
22 Eroticism
23 Eternity
24 Evil
25 Excitement
26 Family
27 Femininity
28 Fertility
29 Flamboyance
30 Freedom
31 Friendliness
32 Fun
33 God
34 Gods
35 Good luck
36 Gratitude
37 Growth
38 Happiness
39 Healing
40 Healthiness
41 Heat
42 Heaven
43 Holiness
44 Illness
45 Insight
46 Intelligence
47 Intuition
48 Religion
49 Jealousy
50 Joy
51 Learning
52 Life
53 Love
54 Loyalty
55 Luxury
56 Marriage
57 Modesty
58 Money
59 Mourning
60 Mystery
61 Nature
62 Passion
63 Peace
64 Penance
65 Power
66 Personal power
67 Purity
68 Radicalism
69 Rationality
70 Reliability
71 Repels evil
72 Respect
73 Royalty
74 Self-cultivation
75 Strength
76 Style
77 Success
78 Trouble
79 Truce
80 Trust
81 Unhappiness
82 Virtue
83 Warmth
84 Wisdom

Appendix D

M-Bus Micro-Master

Portable M-Bus master

Operates up to 10 M-Bus slaves

Power supply by laptop

Transmission speed max 9600 baud

Short circuit protection at the M-Bus

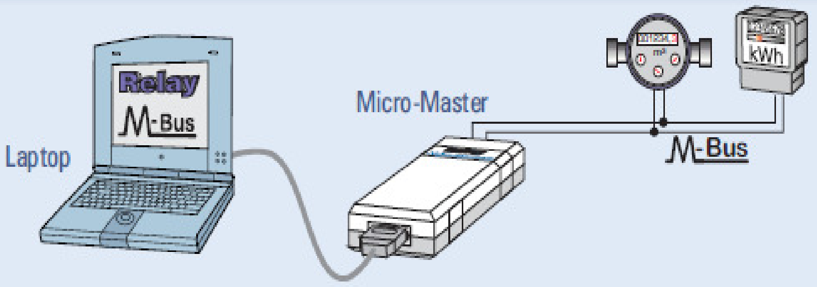
Energy-saving mode

The M-Bus Micro-Master has specially been designed for mobile service. In combination with a laptop it is suitalbe for setup of individual M-Bus slaves during installation and also for fast and easy reading of small M-Bus systems by external service personnel.
Up to 10 M-Bus slaves can directly be operated by the Micro-Master. With its compact and light construction it is ideally suited for service purposes.
Two different versions of the Micro-Master are available.
The serial model is connected to the laptop by a special RS232/PS2-keyboard cable, while the second version is a real USB device.

Relay PadMess

For mobile operation:

M-Bus Micro-Master



Function of the Micro-Master

The Micro-Master is a small mains-independent M-Bus master with a RS232 serial or an USB port. It is supplied by the external PS2-keyboard connection or the USB port of a laptop, with a special cable contained in the scope of supply. This offers an easy setup of M-Bus slaves in the field installation.
The Micro-Master is able to operate small M-Bus networks with up to 10 slaves and offers a built-in energy-saving mode controlled by software.

Advantages of the M-Bus system

- ✓ Cost-saving field bus system
- ✓ Two-wire bus supplying power to the bus users
- ✓ Large range (several kilometres)
- ✓ European standard (EN 1434)
- ✓ Good availability of system components
- ✓ Suited for applications at home and in industry
- ✓ Remote reading of consumption (water, heat, gas, electricity, ...)
- ✓ Total energy monitoring
- ✓ Data logging by mouse click
- ✓ Transmission rates up to 38 400 baud

Technical data

Operating voltage:	5V DC ± 5%	Transmission speed:	300 .. 9600 baud with echo suppression in the USB version
Power input:	max. 1.5W		
Temperature range:	0 .. 55 °C		
M-Bus voltage:	28V (mark) 16V (space)	Housing:	light-grey, plastic protection type IP40
M-Bus quiescent current:	max. 15mA (10 unit loads)		H x W x D: 30 x 54 x 110 mm
Overcurrent threshold:	60mA	Connectors:	D-SUB 9-pin socket for RS232 or USB socket type B
Bus resistance:	approx. 100 Ω		
Bus extension:	max. 500m (J-Y(St)-Y n×2×0.8) (10 devices at bus end)		M-Bus plug-in screw terminal

Order information

M-Bus Micro-Master RS232	Art.-No. MR003
M-Bus Micro-Master USB	Art.-No. MR003USB

Delivery contains:
MR003: special RS232 / PS2 cable to laptop
MR003USB: USB cable A-B and device driver for Windows

Accessories

M-Bus readout software:

Look@M-Bus for Windows95/98/NT	Art.-No. SW006
LocalService@M-Bus (time module)	Art.-No. SW006Z
M-Bus OLE server for Windows95/98/NT	Art.-No. SW005

Relay
Reinecke Elektronikentwicklung und Layout GmbH
Stettiner Str. 38 *Tel.: 05251 / 1767-0*
D-33106 Paderborn *Fax.: 05251 / 1767-20*
www.relay.de *EMail: info@relay.de*

PadMess
Meß- und Kommunikationstechnik GmbH
Stettiner Str. 38 *Tel.: 05251 / 1769-0*
D-33106 Paderborn *Fax.: 05251 / 1769-20*
www.padmess.de *EMail: info@padmess.de*

PadPuls M2

2-channel M-Bus pulse collector

Input for tariff switch signal

Due-date function

Power supply by the M-Bus

All pulse inputs free adjustable

Fully operable in case of M-Bus failure !

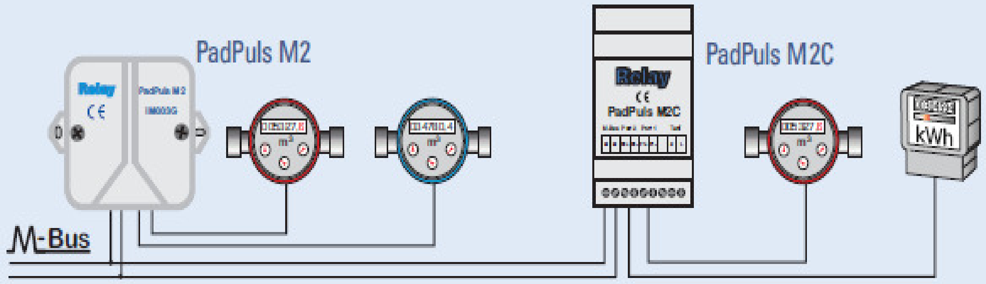
Wall or rail mounted

The devices of the PadPuls M2 series adapt up to two conventional meters with pulse output to a M-Bus system.
The two counter inputs are nearly free parameterizable and fit in already existing installations.
Optionally the user can activate a tariff function, by which energy or volume pulses can be accumulated in separate meter readings for primary and secondary tariffs. Only one pulse signal and one changeover signal for the tariff are required, for example out of a ripple control receiver from the electricity supplier.
The integrated battery ensures that the impulse adaptor is fully operable, even if the M-Bus network fails. Additional security is provided by the periodic saving of the meter readings in non-volatile memory.

Relay PadMess

Adaptable:

PadPuls M2



Function of the PadPuls M2(C)

The PadPuls M2 adapts up to two impulse meters with floating output (for example gas, water, electricity meter,...) to the M-Bus system or, in many cases, electronic So outputs as well. Both pulse inputs can individually be configured. PadPuls M2 thus acts as two stand-alone M-Bus slaves.
The supply for the impulse counting function is taken from the M-Bus. In case of a bus failure an integrated battery ensures data integrity and count operation.

An optional available battery with greater capacity allows M-Bus independent metering for several years.

Another feature of the PadPuls M2 is the due-date function. Meter data are saved separately at the preset due-date by the implemented clock with calendar function. By this, for example, annual consumption data can be obtained without additional calculation software.

The PadPuls M2C version has the same functional features as the M2, however in addition it has a tariff input for 230VAC signal (for example from a ripple-control receiver).
The PadPuls M2 is delivered in a housing for wall mounting while the PadPuls M2C housing is mounted on a DIN rail (3TE).

Technical data

Power supply:	by M-Bus, switches automatically to battery in case of bus failure
Bus operation:	max. 1.5mA (1 unit load),
Battery operation:	current taking max 50µA
Battery expectancy 0.23Ah:	10 years at max. 18 failure days p.a. (changeable coin-type battery)
Battery expectancy 1.35Ah:	10 years at max.110 failure days p.a.
Temperature range:	0 .. 55 ºC
Pulse inputs:	2, individual adjustable
Contact voltage:	2.5V .. 3.6V
Contact current:	30µA
Debouncing time:	5ms
Cable pulse generator:	max. 10m

<u>Requirements to the contacts of the impulse generators:</u>

Potential:	floating
Resistance:	open > 1MΩ, closed < 2kΩ
Contact duration, pause:	min. 30ms
Pulse frequency:	max. 14 Hz
Tariff signal:	floating (data see above)
	Or: 100..250VAC (PadPuls M2C)
M-Bus protocol:	according to EN1434-3
Transmission rate:	300, 2400 baud (autom. detection)
Case mounting:	rail (DIN-EN 50022) or wall mounting
Protection type:	IP40
Dimensions (M2):	W x H x D: 80 x 80 x 52 mm
Dimensions (M2C):	W x H x D: 93 x 51 x 58 mm

Order information

PadPuls M2 (0,23Ah-battery, wall)	Art.-No. IM003G
PadPuls M2 (1,35Ah-battery, wall)	Art.-No. IM003GB
PadPuls M2C (0,23Ah-battery, rail mount.)	Art.-No. IM003GC
PadPuls M2C (1,35Ah-battery, rail mount.)	Art.-No. IM003GCB

Accessories

<u>Delivery contains:</u>
PC-software for 32Bit-Windows for configuration

<u>M-Bus readout-software:</u>
Look@M-Bus for Windows95/98/NT Art.-No. SW006

Relay
Reinecke Elektronikentwicklung und Layout GmbH
Stettiner Str. 38 *Tel.: 05251 / 1767-0*
D-33106 Paderborn *Fax.: 05251 / 1767-20*
www.relay.de *EMail: info@relay.de*

PadMess
Meß- und Kommunikationstechnik GmbH
Stettiner Str. 38 *Tel.: 05251 / 1769-0*
D-33106 Paderborn *Fax.: 05251 / 1769-20*
www.padmess.de *EMail: info@padmess.de*