# Integrated Home Automation System

**April 26**

**2011**

*Improving living standards through technology and awareness.*

University of the West of England
BRISTOL

Marios Georgiou
08015381

University of the West of England
BRISTOL

# Integrated Home Automation System

UWE Student number: 08015381

Supervisor
Dr. Sanja Dogramadzi

Module code: UFMEAY-30-3
Academic year: 2010-11
26 April 2011

**Abstract**
An investigation was carried out to produce and integrate an automation system into a domestic household. The need for such a system is enormous especially in the areas of energy efficiency and assistive technologies. The system is solely based onto a wireless infrastructure that consists of low power nodes monitoring and controlling various home appliances. The design of the internetworked devices is kept simplistic yet robust. The collected sensor data are displayed back to the user to reinforce awareness and to study behaviouristic patterns. Biometric data also make the system capable of replacing carers for critical patients or elderly. Remote appliance control and monitoring improves lifestyle, comfort and safety.
The final system was integrated and tested for a certain period of time. It had an impact on energy use, home security and sustainability. In this research paper the complete home automation system is demonstrated and its usefulness discussed.

**Acknowledgements**
I would like to thank my cousin Christina Vasili that recorded all the voice samples for the system. I would also like to thank my supervisor Dr. Sanja Dogramadzi for all the support, ideas and encouragement she has provided me that helped accelerate through the entire process. Also many thanks to Mr. Henning Karlsen for his valuable information on low level LCD writing. I also thank Nahuel Lavino for his automation ideas and support.
Finally I would like to thank my parents for all the support.

# 1 Introduction

Technology advancements are growing exponentially resulting in a plethora of devices emerging on a daily basis. Devices that keep us connected, updated and in-sync with our daily activities. Advancements have also greatly affected the area of home automation, otherwise known as domotics. Integration of technology in buildings and appliances helps provide greater lifestyle quality, comfort, ease of use and safety. Introduction of technology assists households reduce consumption while increasing efficiency. This is achieved by giving real time feedback back to the users and thus creating a consciousness about their actions. Centralised panels allow us to control appliances using revolutionary ways even if we are not physically present in the premises. Interconnected devices can be controlled or used transmit monitoring data to the user regardless to their location.
Active monitoring also enables patients that would normally require a carer, to stay alone without compromising their safety. It offers them back their privacy and their dignity just by allowing assistive technologies into their lifestyle.
Although there are already products for domotics, their designs are ‘closed’ and the expandability is limited to the company’s offerings. The aim of this investigation is to explore multiple ways to create complete automation system using the simplest and most affordable way. The following paper documents the process from concept design to the final product.

# 2 Literature Review

Home automation systems have been around from 1975 when the company Pico Electronics created the X10 standard. This open industry standard used the house's power lines to send encoded data and control various appliances. This protocol is considered the oldest standard used for home automation (Khusvinder et al., 2009).

Progression of technology has enabled many fields to grow and expand rapidly. Home automation is becoming more popular in areas such as medical care, energy efficiency and human interaction. Advancements in wireless digital networks have provided engineers with tools to create low power networks capable of completely interconnecting a domestic household. Connecting appliances together into a network formation allows other technologies to use physical sensing information and activity patterns to provide increased safety and satisfaction (Suh & Ko, 2008). Biometric data can be collected to enable an authorised doctor or a medical advisor to provide telehealth services (Khusvinder et al., 2009). Sensor data also help increase energy efficiency. To achieve that, house consumption is displayed to the user and at the same time the system autonomously regulates appliance's energy consumption (Mouratidis et al., 2009).

However, even though the technology is available, adoption by consumers is considered limited and slow (Khusvinder et al., 2009). The reasons behind limited adoption are, firstly complex and expensive architecture. Existing systems incorporate unnecessary high processing power units that increase the cost of the whole implementation. Secondly, installation of these systems is done in an intrusive manner. Physical wiring is required to some extent, usually because of the wireless equivalent high price. Lack of network interoperability is also an issue for home networks. Many households have a fair amount of wireless networks that are only used for a specific function. For every wireless protocol (WIFI, Bluetooth, GSM, etc) a different device usually exists. This results in many networks, many frequencies and therefore more interference. The networks have to be designed with all these scenarios in mind and allow interoperability (Suh & Ko, 2008).

# 3 Aims of investigation

The aim of the research and experiments is to demonstrate how a home automation system can serve as an assistive mechanism in the modern household. To achieve that, the following important sub-aims need to be realised.

- Background research, investigate previous attempts, problems, solutions
- Research network types & protocols.
- Acquire software and hardware development tools
- Prototype hardware (network, sensors, LCD, etc) & software
- Verify, test circuits in isolation and together
- Improve hardware and software and produce PCBs
- Software development for final hardware
- Add modules if permitted by time frame
- Collect data and determine if system integration was successful
- Test robustness in artificial worst case scenarios

# 4 System Architecture

## 4.1 Introduction

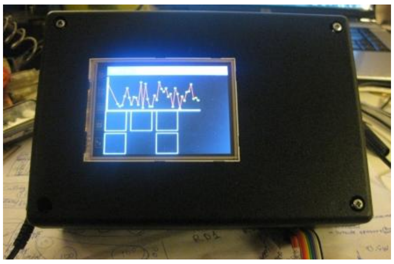


Figure 1 Central system panel plotting consumption data

The system was designed using the bottom-up approach. Every component was tested, bread-boarded and if successful, added as a functional module on the system. Since the system structure consisted of interconnected modules, extra care was taken to ensure that each piece performed as best and fast as possible to avoid unnecessary poling by the processor. Modules that were malfunctioning, were not noise immune or consistent, were removed from the system.

The home automation system is useful when it is tightly integrated into the house. To assist the installation process, it is usually cost and time effective to add wireless capabilities to the individual house components. This added functionality helps add, move or remove nodes without installing any wires. The nodes of the network fall into two categories. The first group are sensing nodes that deal with sensors and their signal conditioning. On the other hand wireless devices are nodes integrated into appliances in order to control them.

The power of each node, charging and energy harvesting, is also an essential feature for low power sensor units. Some of the nodes were fitted with a charging circuit to allow any unregulated voltage from solar cells to even lemon juice to be regulated and charge the Li-ion cell. Having nodes that can recharge from renewable sources means lower maintenance and greater flexibility regarding placement in remote locations.

The system also featured a 16-bit resistive touch screen to display data and to accept commands. The LCD was the main interface of the whole system and was designed to be as responsive and intuitive as possible.
The main board also featured a speaker to notify the user audibly not just visually. The voice commands were recorded and stored on an SD card.

# 5 System Component Functionality

## 5.1 Processing components

The main processing unit is a STM32 ARM Cortex-M3 microcontroller that offers 48 general purpose I/O pins. Having a large amount of I/O helps taking care of all the peripherals needed for the system. The microcontroller was over-clocked to 128MHz in order to achieve a decent LCD refresh rate while communicating with other five peripherals. The STM32 also featured a bootloader to load the software, without the need of any expensive programmers.
On the wireless node side PIC18F4550 microcontrollers were used to handle wireless packets, read sensors and control devices.

## 5.2 Wireless infrastructure

### 5.2.1 Introduction

Implementing a home network requires all the nodes to be robust, cheap, easy to use, require as minimum poling as possible and low power.
As with WIFI & Bluetooth technologies, each frequency spectrum has many composing channels. For example WIFI's channel one is 2412MHz and channel fourteen is 2484MHz. The very same channel structure applies to the MRF24J40ma module detailed below.
In contrast, the RF link (434/315Mhz) network consists of nodes sending and receiving data on the same frequency channel spectrum. Operating on the same channel means more packages will be dropped. When two nodes "talk" at the same time the packages collide.
When packets are lost or a node restarts, the system should be able to adapt and recover as fast as possible.
Research about different transceivers, frequencies and protocols is discussed below. All the different wireless components and software stacks were tested for their performance, range and redundancy.

### 5.2.2 Microchip MRF24J40ma transceiver

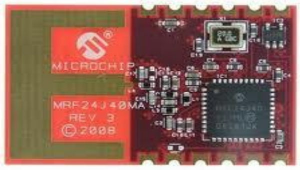



Figure 2 MRF24J40ma transceiver module

The initial experiments used the MRF24J40ma 2.4GHz transceivers from Microchip. These modules had a built-in antenna and they were operated using a host microcontroller via 4- wire SPI. The modules were the physical layer (PHY) of the system and the packet handling and coordination was done by the stack, on the microcontroller. The stack could be Zigbee™, MIWI™ or a custom stack. All have their advantages and disadvantages which are listed below. Microchip offers Zigbee™ and MIWI™ on their website. The code is compatible only for their development boards and it is very processing intense for an 8-bit micro.

The approach taken was to clone the development boards enabling us to use the software provided by the vendor. Three PCBs were etched that featured a PIC18F4620. This allowed us to use exactly the same code as Microchip's PICDEM-Z (schematic & layout in appendix B) development board.

The reason for the specific PIC was that it had SPI and UART pins separated so that both peripherals could be used at the same time. The main concern was the transceiver and code complexity and how well it could be modified for this application.

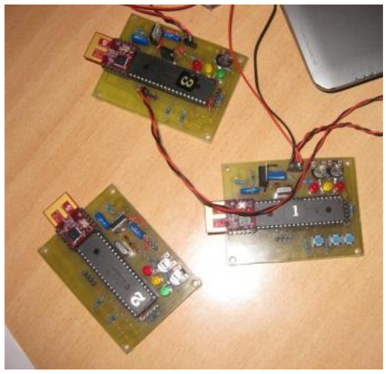


Figure 3 Two End devices and a Coordinator forming a star network

The MRF transceiver was first tested with a demo program from Microchip to test the cloned PCBs compatibility. Once that was done each node was loaded with the protocol stack and assigned a MAC address.

#### ZIGBEE™ - Mesh network topology

The Zigbee™ is a full featured high level protocol for use in low power, low data rate networks. The Zigbee™ devices form a mesh network topology. It is considered a simpler alternative to WIFI and Bluetooth. However while Zigbee™ is already used in home automation systems (Khusvinder et al., 2009) the level of complexity is still considered reasonably high. Many of the features add unnecessary complexity and this usually leads to the need for more powerful host microcontrollers. The protocol has three types of nodes.

Zigbee™ coordinator (ZC): The coordinator is capable of forming a network and performs security tasks such as adding or removing nodes and storing sensitive information.

Zigbee™ Router (ZR): The router devices are used to relay packet data and also take readings or control devices.

Zigbee™ End Device (ZED): The end device is just a slave that performs a task and goes to sleep until next sampling.

The code provided for the PIC microcontroller has a size of (40K-100K) and barely performs optimally on an 8-bit PIC. Apart from the fact that the code size is enormous, the stack doesn't feature any fail-safe mechanisms. Adding code to the Zigbee$^{TM}$ core is very difficult due to size. The Zigbee$^{TM}$ stack was not used for the system due to the non-optimised code provided for the PIC18F series.

**MIWI$^{TM}$ - Star network topology**

The MIWI$^{TM}$ protocol is considerably smaller than the Zigbee$^{TM}$ (3K-17K) and more efficient. It forms a star network and has only two types of devices, a coordinator and an end device. The MIWI$^{TM}$ stack performed better than Zigbee$^{TM}$ but when a node was out of range or disconnected, the coordinator was not aware of the current status.
MIWI$^{TM}$ protocol was abandoned since it did not have any fail-safe mechanisms and the time required to add this kind of functionality on unknown code, a complete stack could be written from scratch. The MIWI$^{TM}$ protocol is an excellent alternative for Zigbee$^{TM}$ stack but not mature enough.

### 5.2.3 Holy Stone Enterprise RF links

Figure 4 RF link receiver

Figure 5 RF link transmitter

The other option was to go for the simpler RF links. These links mostly consist of passive components, a wire antenna, a SAW oscillator and a modulation IC. The principle of their operation is whatever is currently at the input on the transmitter side will be transferred and come out at the output of the receiver. It is important to note that these RF links use the same channel. One transmitter can talk to many receivers but if two transmitters are ON then the data are mixed. A checksum is sent along with the package to verify that the packet came intact. The operating voltages for the modules are 5V for the receiver and 3-12V for the transmitter (more voltage, more range). The transmitter circuit is simpler since it doesn't need any regulation to operate. The receiver module required a step-up circuit in order to use a 3.7V Li-ion cell. The chosen wireless operation frequency of the network was 315 MHz with the exception of one node that was at 434 MHz (CurrentCost power clamp). The range of the network depends on many variables such as obstacles (walls, wall wiring and material) and also from the transmitter operating voltage. Other factors such as channel energy (amount of transmitters on same band) affect the range significantly.
Another feature these modules have is antenna auto-gain. This affects the receiver primarily. When the receiver does not receive any data for a short period of time (5ms) it increases the antenna gain until it satisfies that condition, receiving data. The result is just noise that is picked up by the high gain antenna. One of the biggest problems was that once a receiver went into high gain mode, it couldn't listen to data anymore since the ambient noise levels were high enough and mixed with the useful data. The receiver essentially drifts away and to resolve this issue, a square wave was transmitted to bring the antenna gain back to normal levels. In wireless systems this technique is called pre-ample header. Pre-ample is some dummy data in front of each packet stream, to "get" the receivers attention and normalise the antenna gain.

The network stack, discussed below, is responsible for node identification, prioritisation, data encoding, checksum generation etc. The stack makes bidirectional communication on the same channel possible.

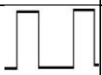
#### 5.2.3.1 Wireless custom stack - Star network topology

The 2.4GHz transceivers offered a great solution for a house automation system but most of the code from microchip was relatively new at the time and without any good documentation. Apart from that, the stack didn't have any fail-safe mechanisms. In the event of an end device going offline the network would continue sending data to it without realising what happened. To reconnect the coordinator had to do a rescan and re-add the end device. Modifying the stack was a gigantic effort since the code had assembly pieces in it and was heavily dependent upon interrupts. To use a PWM pin along with the stack took days since the interrupts were too many and used up all the resources of the 8-bit micro. The debugging also was very hard since an expensive packet sniffer tool was needed to be purchased to view the packet traffic.
The RF links were chosen along with a stack made from scratch. This helped us grasp the fundamentals of wireless since we had an in-depth understanding of all the low level packet handling. The structure of the stack followed a simplistic model and focused on being lightweight for use in microcontrollers. In order for the network to operate only one channel a half-duplex model needed to be followed. Each node was assigned an ID similar to a MAC address and only responded to that. The master panel calls the slave by sending a string that matches the ID and structure the slave is listening for. Other slave devices discard the packet. The packet format is very simple.

| **Header** | | **BYTE 1** | | **BYTE 2** | | **BYTE 3** | | **Checksum** | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Pre-ample decimal 15 | Packet Description BYTE | Data Descriptor | Data | Data Descriptor | Data | Data Descriptor | Data | Init BYTE 'C' | Part 1 | Part 2 |

The pre-ample byte is the decimal number 15. The reason for this is because the binary value of 15 is 00001111 (ASCII shift-in). This creates a square wave when sent through the serial interface. Values like 'U' (binary 01010101) also create a square wave but the most efficient wavelength was with decimal 15. The pre-ample value is sent around ten times, enough to bring the antenna gains back to normal.
The bytes are sent as ASCII characters to help identify the packet. A packet with a descriptor 'L' in the beginning is destined for the RGB lamp device, a 'K' is for the key pinger and an 'S' is a sensor data packet etc.
If the packet has an 'L' as a packet descriptor the lamp device checks the other data descriptors (byte 1-3) as well, verifies the checksum and then the values are considered valid and stored into the memory.
The checksum is a very important add-on to the packets. It helps distinguish between the corrupted and intact ones. Since ambient noise is an issue with the antenna auto-gain feature, the checksum helps filter all the noise out. The two bytes carrying the checksum data are located at the last two slots of the packet and are named part one and two.

To calculate the checksum the software gets the data, adds them together, splits the result to hundreds, tens and units. It later converts the hundreds to ASCII format and adds it to part one of the checksum. The tens and units are added together, converted as well to ASCII and stored in part two of the checksum. The numbers are split using modulus arithmetic.
The header and description bytes are already in ASCII format. The numbers however are shifted to start from number 48 (ASCII zero ‘0’) to avoid earlier characters in the ASCII table such as ‘delete’, ‘carriage return’, ‘null’.
All these conversions are done so that a packet is more readable when debugging in a terminal window.
The following is a sample packet to turn the lamp colour purple. The lamp intensity range is from 0-99. To get purple we send Red=99, Green=0, Blue=45.

| Description | Pre-ample | Head | Discr 1 | Data 1 | Discr 2 | Data 2 | Discr 3 | Data 3 | CHECKSUM ‘C’ + part + part2 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Data start from ASCII zero (value+’0’) | 15 | L | R | 99+’0’ | G | 0+’0’ | B | 45+’0’ | C | 1+’0’ | 44+’0’ |
| Decimal result | 15 | L | R | 147 | G | 48 | B | 93 | C | 49 | 92 |
| Final packet in ASCII format |  | L | R | ô | G | 0 | B | ] | C | 1 | \ |

**Principle of operation**
When all the devices are booted, the master panel addresses the specific node it wants to read or control. The master panel then turns its transmitter OFF so that the nodes transmitter can be switched ON. The node carries out the task and responds back with the data requested or with a status code. The node turns OFF its transmitter so that the master panel can resume talking to other nodes.
Apart from bidirectional nodes some are only transmitting short streams of information in specific intervals. Removing the receiver from the circuit reduces the component count, size and consumption. For a remote sensor it is more effective to send data between long intervals rather than having the receiver always ON waiting for a request to send (RTS).
This helps reduce power consumption around 200 times and makes it very attractive for battery operated nodes.

**Manchester encoding**

Wireless modules that transmit 'ones' using high energy levels and 'zeros' by remaining OFF require a balanced amount of ones and zeros to function properly. If many ones are sent the range of the transmission will be higher than if we were transmitting a lot of zeros. To balance our data to an equal amount of zeros and ones we need to use encoding techniques such as Manchester encoding. This type of encoding allows us to blend the data with the clock signal. The result is similar to the following.

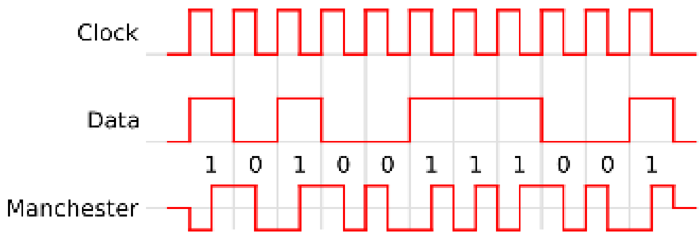



Figure 6 Leroy Davis(2003) Available from: http://www.interfacebus.com/Manchester-encoding.png [Accessed 28 April, 2011].

To decode the data back we use exclusive OR arithmetic.

### 5.2.3.2 Antenna design

The RF links have no internal antenna. Their datasheet states that adding a specific length of wire is sufficient. The general rule for a wireless wire antenna length is that it has to be the 1/4 of the wavelength. In order for the transmitter to have optimal performance there has to be also an impedance match. If the antenna is shorter than the 1/4λ, then a loading coil needs to be added. A loading coil is an inductor that is placed between the antenna and the groundplane to counteract the capacitive reactance in the circuit caused by the impedance imbalance.

The second important part of a monopole antenna is the groundplane. The groundplane is a conductive area that has to be as solid as possible and connected to ground. The wire antenna needs to extend away from the groundplane. Between the groundplane and the wire, current flows and emits electromagnetic energy.

Two antenna types were tested. One was a simple wire(1/4λ length) and the other one was a coil wrapped wire.

### 5.2.4 GSM Network

To enable the system to communicate with devices outside the home network, a GSM modem was used. This enables the user to request information about the house while away and also control devices. Furthermore in case of an emergency the system also calls the next of kin.
The GSM module is a Telit GM862 modem and its controlled through the ARM serial port using AT commands.

Figure 7 Telit GM862 GSM module

### 5.2.5 Bluetooth ad-hoc

To help data logging more efficiently a serial Bluetooth module was used. At any time, any data required to be read or logged are sent out through the serial port of the STM32 and via Bluetooth end up into a terminal on the PC. A terminal can save the data to a file. The data can be organised and plotted using software like Microsoft Excel. The Bluetooth module is the GP-GC021 class 2 by Sure Electronics Co., which is a stripped down model that registers only as a virtual com port. The module pairs to the host computer like every other Bluetooth device does and provides a reliable and flexible gateway for the information. The Bluetooth module has a range of around 15 metres.


Figure 8 GP-GC021 Bluetooth Serial Link

## 5.3 Sensors

To make the most of the capabilities of the network sensing nodes is an essential piece. Gathering data and essentially logging those reveals information about usage, consumption and the relation between them. Comparing these data we can see the behaviouristic outcome at a specific time, at a specific condition such as consumption versus outside temperature. The sensor data collection is done by the small wireless microcontroller boards that take care all the signal conditioning, process all the data and send the results to the main board which displays them to the user. The system also uses the data for internal use. If for any reason any of the sensor combinations reach critical levels the system alerts the user.

### 5.3.1 Liquefied Petroleum Gas (LPG) sensor – MQ-6

This gas sensor is able to detect any concentrations of mainly butane, methane, hydrogen and propane in the atmosphere. The sensor active range is from 200 to 10000 parts per million. Specifically the sensor is composed from a micro $AL_2O_3$ ceramic tube, Tin Dioxide ($SnO_2$) sensitive layer, measuring electrode and heater element. The ceramic heating element needs to be hot in order for the sensor to pickup any particles. The output of the sensor is analogue and pulled down using a resistor for output stability. The sensor response is linear and the steepness of it depends on the gas type.

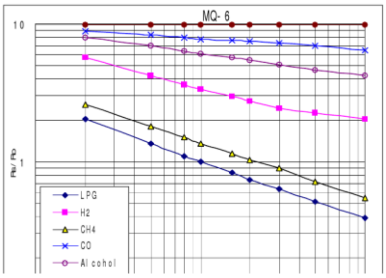



Figure 9 HwSensor Co.(2003) Available from: [www.sparkfun.com/datasheets/Sensors/Biometric/MQ-6.pdf](www.sparkfun.com/datasheets/Sensors/Biometric/MQ-6.pdf) [Accessed 29 April, 2011]

Figure 10 MQ-6 Liquefied Petroleum Gas (LPG) sensor

### 5.3.2 Carbon Monoxide Sensor - MQ-7

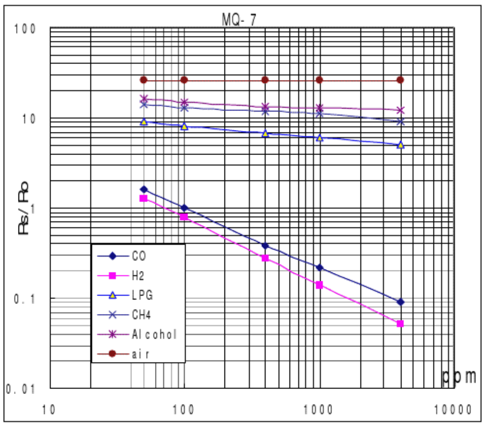



Figure 12 HwSensor Co.(2003) Available from: [www.sparkfun.com/datasheets/Sensors/Biometric/MQ-7.pdf](www.sparkfun.com/datasheets/Sensors/Biometric/MQ-7.pdf) [Accessed 29 April, 2011]

This CO sensor also uses a heater element to detect particles (ranging from 20-2000ppm of CO) and has a proportional analogue output. The output is pulled-down by a 10kΩ resistor and read by the PIC's analogue input. The datasheet states that this sensor gets extremely hot due to the heater inside it. The solution provided was to pulse the heater with a 60 seconds high pulse and a 90 seconds low at 1.4V to maintain the sensor at an optimal temperature.

Although the solution was proven to work, the pulsing created a noisy output that required software filtering. To fix this the operating voltage was dropped from 5V to 3.6V using two diodes in series. This kept the sensor cool enough and provided a stable voltage.


Figure 11 MQ-7 Carbon Monoxide Sensor

### 5.3.3 Alcohol Sensor - MQ-3

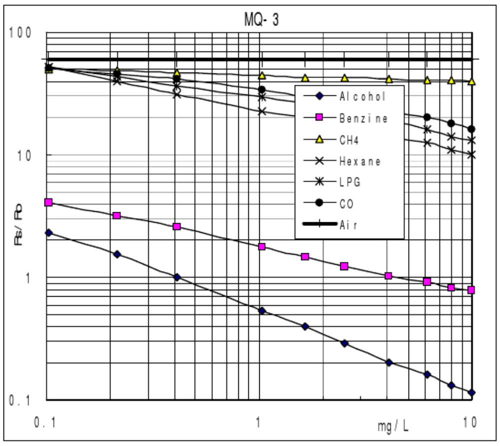



Figure 14 HwSensor Co.(2003) Available from: [www.sparkfun.com/datasheets/Sensors/Biometric/MQ-3.pdf](www.sparkfun.com/datasheets/Sensors/Biometric/MQ-3.pdf) [Accessed 29 April, 2011]

Figure 13 MQ-3 Alcohol sensor

The Alcohol sensor measures the alcohol particles in the atmosphere and it's completely immune to other flammable gases except benzene. The sensor was added to the system as a part of an experiment to measure the alcohol percentage in the atmosphere between a Saturday and a Sunday. The sensor did show different readings between the two days. It is worthy to note that the sensor was added without any specific reason at first, but it later became apparent that data from any sensor might end up useful comparing to others. (See appendix A)

### 5.3.4 HIH-4030 humidity sensor


Figure 15 Relative humidity sensor

This sensor consumes 200µA and outputs a linear voltage proportional to the relative humidity percentage (%RH).
The reason relative humidity is considered a much accurate and useful than absolute humidity is because RH represents a humidity percentage of between 0% (no water molecules in air) and 100% (maximum water molecules, right before dew point). Combining RH with temperature data we can derive useful information about preservation metrics versus comfort. Preservation metrics can give us information about natural aging (preservation index) of different materials, mechanical damage or corrosion (% Equilibrium Moisture Content) and mould risk. Although these thresholds take many parameters, such as dimensional change, they are important when we need to keep a room in the comfort zone.

### 5.3.5 CurrentCost consumption and temperature sensor

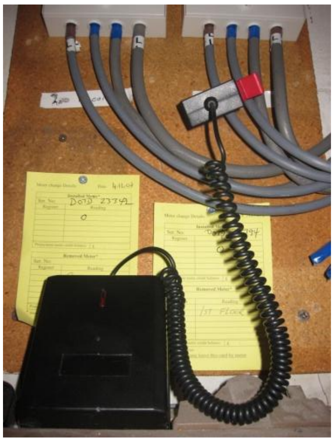


Figure 16 Clamp sensor in main electric meter box

The CurrentCost device is a commercial device that features a current clamp connected to the main electric meter box.
The clamp itself consists of a current transformer (CT) that measures the magnetic field intensity. The clamp part around the mains wire is considered to be the primary coil of the CT. The density of the magnetic flux is proportional to the current passing through the mains wiring. Unfortunately the current clamp cannot measure current through normal appliances wiring since both live and neutral wires would cancel each other out. To perform a reading the cable of the appliance should be split into two wires with the clamp attached to one. The clamp in figure 14 is the current clamp sensor along with the transmitter circuit and batteries. It is notable to mention that the transmitter is streaming data every six seconds and is capable of running for more than 7 years before a battery change.

Figure 18 CurrentCost stock look

Figure 17 CurrentCost useful PCB

The second part of the current cost device is the receiver that displays the data on an LCD display. Apart from printing the data on the screen, the receiver also has an RJ45 output port for use with a PC connection. The RJ45 port outputs all the LCD data in an XML format using serial protocol. The device was taken apart, the LCD was removed and the PCB cut in half leaving only the receiver board connected to the main processor. All the data from this device end up on the main panel display. The board in figure 16 features a real time clock IC, temperature sensor and the consumption readings from the clamp.

### 5.3.6 Reed Switches

Reed switches are used to check the status of doors and windows. The main panel will alert the user if something is not locked before leaving the property. Since the system knows the window and door status at any time, these states can be transmitted to a mobile device via GSM. Magnetic switches can be easily implemented due to their tiny size. The switch is connected to a wireless node and reports back to the main panel. Since reed switches have only two states, they can be easily connected to the microcontroller's input pins.
An alternative way to read reed switches when we don't have enough GPIO is to use one analogue pin and a network of resistors. When a switch is activated, the voltage output changes relatively to the position of the switch. The principle is the same with a digital to analogue converter (DAC). Many bits have different significance and therefore their combination produces a unique analogue voltage. When combined with low tolerance resistors many switches can be connected to only one pin.

### 5.3.7 Sensor calibration

Sensor calibration is very important especially when the data are used for critical applications such as monitoring patients. The MQ sensors use a ceramic heating element to detect gas particles. As stated in the datasheet the heater needs to stay ON for one day to get more accurate readings. This is an important step that improved the readings accuracy. In some cases the sensor even has to stay ON for more than 48 hours to stabilize completely.

## 5.4 Wireless Nodes

### 5.4.1 Wireless Sensors

The wireless nodes, used for sampling data or controlling appliances, have the same layout. The board features a host microcontroller (PIC18F4550) an RF transmitter, a receiver slot (depending on device) and the necessary power electronics. The board has its pin outs broken out so that sensors can be connected easily. The board supports analogue output, SPI and serial interface sensors.

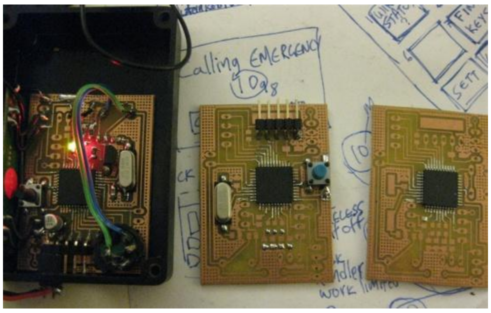


Figure 19 Population of the wireless nodes SMD PCBs

#### 5.4.1.1 CO, Gas, alcohol, humidity and light sensors

Since a wireless node can interface with more than one sensor, this board has five sensors connected to it. The sensors are a carbon monoxide, a gas, an alcohol, a light and a humidity sensor. The scaling and linearization are done on the PIC microcontroller and later sent to the main system panel. All of the sensors have linear response in this case and therefore only need a small offset value.

The specific sensor box was used indoors and outdoors for two days.

Figure 20 Sensor box outdoors data logging

#### 5.4.1.2 Patient strapped accelerometer monitor

The accelerometer device is strapped onto the patient and monitors any rapid movements. This helps alert the patient's doctor or relative when the patient faints or has not moved for a period of time. The module features a rechargeable battery and has a small form factor. In the event of a sudden move the module reports to the main console and a countdown starts. If the patient does not cancel the ten second countdown, a call is placed to the close relative or doctor. The call is placed by the GSM module.

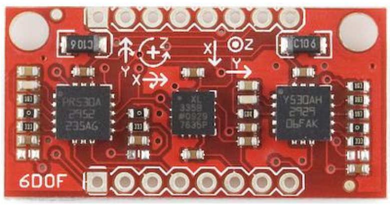



Figure 21 Accelerometer breakout board

#### 5.4.1.3 CurrentCost

Figure 22 Central panel internals

The current cost device is connected to the STM32's serial port.
Every six seconds the device outputs an XML stream.
The stream has the following format.

<msg><date><dsb>00043</dsb> DSB is Days Since Birth
<hr>00</hr><min>10</min><sec>43</sec></date> Time data
<src><name>CC02</name><id>02538</id><type>1</type></src> ID data
<ch1><watts>00484</watts></ch1> CH1 consumption
<ch2><watts>00000</watts></ch2> The other two channels
<ch3><watts>00000</watts></ch3> are disabled
<tmpr>22.2</tmpr></msg> Temperature data

The XML data are stored in a buffer and the important data are extracted.

### 5.4.2 Wireless devices

Figure 23 Lamp ON/OFF states

#### 5.4.2.1 RGB Lamp

The wireless lamp was built as a single model to demonstrate functionality that would then be applied to several different house lights. Since the wireless network protocol allows us to send a lot of information per packet we can have more than just an ON/OFF function. This prototype was built from an Ikea desk lamp due to the low cost, simple design and the semi-transparent built material. The 240V bulb was removed and low voltage electronics were fitted.

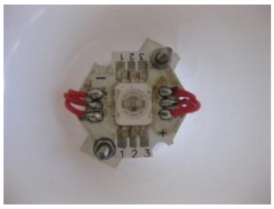


Figure 24 High power RGB LED

This device consists of a high power RGB LED, the control circuit for each channel and the wireless microcontroller circuit which is connected to.
The high power LED draws around 400mA per channel and therefore requires a transistor drive circuit. The circuit consists of three NPN transistors that are connected to the three PWM pins. The Lamp has a unique ID in the network that is used to access the device, any other calls are ignored.

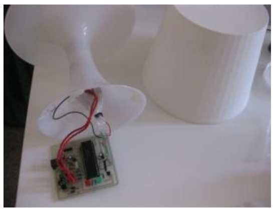


Figure 25 Lamp wireless circuit

The packet structure, which the lamp is configured to respond to, is shown below. The light blue shaded parts of the packet are colour dependant.

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Pre-Ample | L | R | Red Value | G | Green Value | B | Blue value | C | Check | sum |

The yellow packet parts are always the same for the lamp device.

### 5.4.2.2 High voltage applications

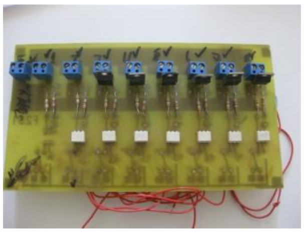


Figure 26 PCB layout for the high voltage circuit

Controlling low voltage appliances is a safer option, however the majority of them is operated using mains 240V AC. For the purpose of exploring high voltage circuits to control light bulbs, televisions, toasters etc, the opto-isolator TRIAC configuration proved to be the best option considering cost and simplicity.

An opto-isolator is used to isolate live from low voltage side. The Triode for Alternating Current (TRIAC) is essentially two inverted thyristors. A thyristor is like a transistor but after activation, the collector-emitter path stays active even if the base voltage is removed. To deactivate a thyristor the collector-emitter voltage current needs to drop down under a certain threshold. The TRIAC has the property of staying ON but since we apply alternating current, the two thyristors are ON only when the base is high. TRIACs have a high voltage and current tolerances at a low cost which makes them suitable for controlling almost all appliances. They are also suitable for inductive loads other than light bulbs such as refrigerators, microwaves, etc.

Figure 27 Light bulb control using a microcontroller

Figure 28 Controlling 240V light bulbs

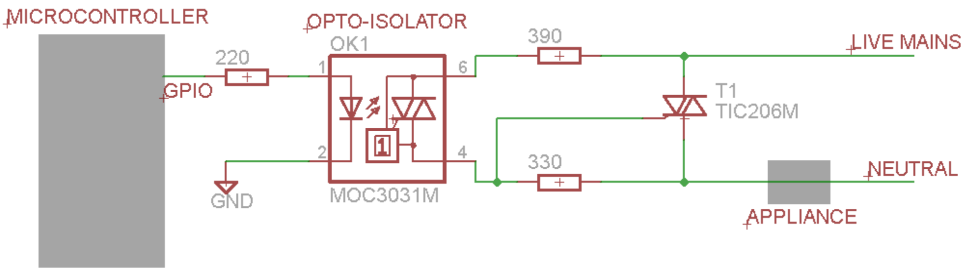



Figure 29 Triac and opto-isolator circuit to control high voltage devices

#### 5.4.2.3 Keys locator

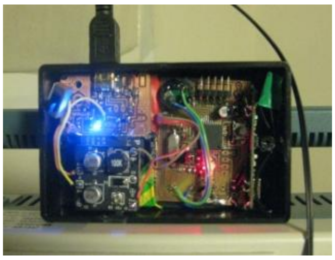


Figure 30 Keys locator charging

The key pinger helps locate your keys by making them beep. The device consists of a charging circuit, a step-up regulator and the receiver circuit. The pinger uses a high decibel buzzer to alert the user and also some LEDs. By selecting the "ping keys" option from the touch screen menu, the main panel sends the following command packet.

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Pre-ample | K | M | Mode Select | V | Volume Value | D | Dummy Value | C | Check | sum |

The 'M' prefix is attached at the mode select data. The mode select data specifies the type of the buzzing. It can be a pattern or a frequency. The 'V' prefix goes with the volume data. This allows us to track the keys without waking up the neighbours. The PCBs are surface mount to have as little footprint as possible.

## 5.5 LCD screen and touch panel

A 3.2" LCD with an integrated resistive touch panel was used as the main interface. The LCD protocol was a 16-bit parallel interface and on the other side the touch panel interface was 4-wire SPI. The complete setup of the touch screen interface used 30 GPIO in total. The LCD resolution was 320x240 and had 65,000 colours. The screen also has the option of 8-bit mode. This mode is two times slower but helps save eight GPIO. Since a fast refresh rate was one of the priorities, the 16-bit wide bus option was chosen.

All the display letters, shapes, lines were coded pixel by pixel using low level algorithms.
In order to write to a pixel we issue a command, send the address of the pixel and the colour data. In order to do something more complicated such as writing a character on the display, we need to define a pixel array with the character front and background pixel locations.
To define the font arrays we need to create or choose the font we need.

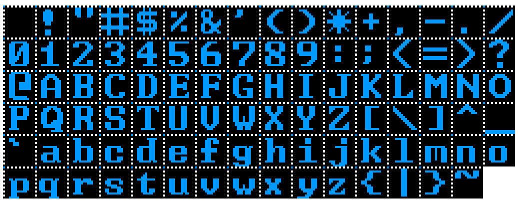



Figure 31 Font choice

First an image of a font was created that includes all the commonly used ASCII characters. Each character was then split into 16x16 pixel areas.

The letter is converted to ones and zeros depending on the pixel colour.

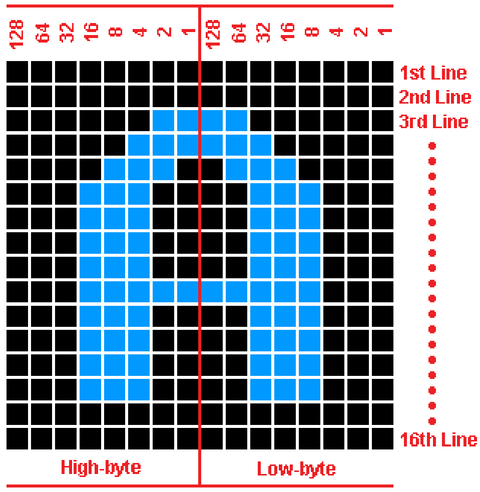



Figure 32 Letter 'A' 16x16 pixels

```
00000000 00000000
00000000 00000000
00000011 11000000
00000111 11100000
00001110 01110000
00011100 00111000
00011100 00111000
00011100 00111000
00011100 00111000
00011111 11111000
00011100 00111000
00011100 00111000
00011100 00111000
00011100 00111000
00000000 00000000
00000000 00000000
```

Figure 33 Binary version of the letter 'A'

Each horizontal line consists of two 8bit numbers. The numbers in binary are:

```
B00000000,B00000000,B00000000,B00000000,B00000011,B11000000,B00000111,B11100000,
B00001110,B01110000,B00011100,B00111000,B00011100,B00111000,B00011100,B00111000,
B00011100,B00111000,B00011111,B11111000,B00011100,B00111000,B00011100,B00111000,
B00011100,B00111000,B00011100,B00111000,B00000000,B00000000,B00000000,B00000000,
```

To conserve space we convert the numbers into hexadecimal.

```
0x00,0x00,0x00,0x00,0x03,0xC0,0x07,0xE0,0x0E,0x70,0x1C,0x38,0x1C,0x38,0x1C,0x38,
0x1C,0x38,0x1F,0xF8,0x1C,0x38,0x1C,0x38,0x1C,0x38,0x1C,0x38,0x00,0x00,0x00,0x00,
```

The complete ASCII table was defined as an array and used to produce low level functions like “print character” on screen.
Using the print character function as a foundation, the print number and string function was created.

For shapes the first functions created were horizontal, vertical and angled lines. The angled line algorithm used was created by Jack E. Bresenham. He developed this simple algorithm for IBM in 1962. The algorithm approximates the best line path given the pixel sizes and angle (figure 32).

Figure 34 Straight line approximation using Bresenham algorithm

Apart from mathematical shapes images and icons had to be displayed. To achieve that an image converter was used to convert jpeg images to a pixel array. The array was then copied as a library onto the microcontroller.

## 5.6 Voice technical

### 5.6.1 Playback

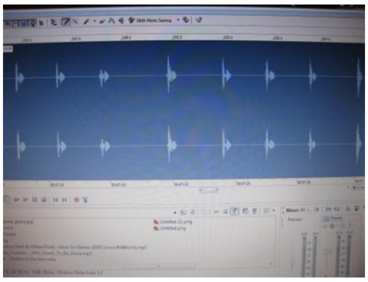


Figure 35 Voice data editing

The voice feedback is an essential piece of the system. It gives audible feedback which is as important as the visuals.
The playback module used was the SOMO-14D from 4D systems. Using SPI the main processor selects the track to be played. The audio tracks are loaded onto a 2GB SD card and provide a basic sound bank to establish simple communication. The module sampling speed is 32 KHz and the voice tracks are stored in the 4-bit ADPCM file format. A 1W speaker is driven by two differential PWM lines to produce sound.

### 5.6.2 Recognition Vrbot module

The Vrbot module from Veear offers voice recognition capabilities in a small package. The use of voice recognition is important for users that are visually impaired or have mobility problems. The Vrbot processor does the heavy processing and analysis of the voice commands and sends the results back to the main processor. The module offers a range of speaker independent commands (SI). These are pre-programmed to respond to various users and not just the owner. The second type of commands are speaker dependent (SD). These are recorded and trained by a unique user. There are strictness levels that can be altered to allow more room for error. The module also provides voice passwords and four different languages. One of the issues with these modules was the requirement to have a quiet environment. Any ambient noise would deem the voice command unrecognised.

## 5.7 Graphical and audible user interface

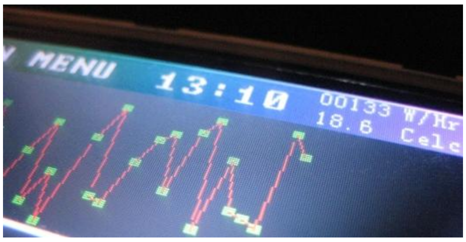



Figure 36 Consumption data plotted on display

The graphical user interface was designed to be intuitive, simple to use and effective. The main functions of the system are presented in the main screen. Whether the user is looking for the car keys or wants to check if the house windows are open, all the information are updated every few seconds and displayed on the main screen. The voice bank recorded is capable of pronouncing numbers, and features 123 different words. These individual words were recorded, edited and converted to allow playback from the STM32.
See Appendix C for voice bank listing.

## 5.8 Power management

Power management is very important in wireless systems since without it the device will be tethered thus providing none of the mobility that was initially designed for. Since most of the system's circuits required stabilised 5V there were issues when using a Li-ion cell.
A Li-ion cell has a nominal voltage of 3.7V and a voltage range of 3.1-4.2V (Panasonic Co., 2007). In order to reach 5V, a step-up circuit had to be built which produces 5V no matter how charged the battery is.
Another circuit that is needed is a charger for the Lithium cell. These two circuits work together to provide a power pack that can charge a Lithium cell, produce stable 5V and have a flexible voltage range while following charger standards.
The charger circuit for the battery uses the Maxim MAX1555 charger IC. The IC accepts USB charging (limited current to 100mA) or other DC sources up to 8V (drawing up to 280mA). No input-blocking diodes are required to keep the battery from draining. The IC is also able to charge batteries below 3.0V using trickle charging (0.1C).
Stepping-up to 5V means that switching noise is introduced and has to be dealt using filter capacitors.

### 5.8.1 Energy harvesting

To take advantage of the mobility and flexibility a wireless network offers the nodes have to be battery powered. Although a built-in charger circuit exists, the energy still has to come from a wall outlet that again, tethers the device. The solution is to use energy harvesting from different sources to create a truly mobile sensor device that you can place somewhere and don't have to worry about it.

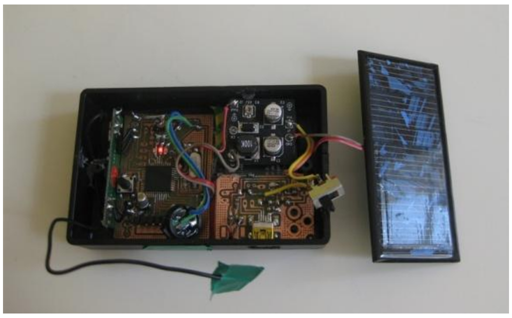


Figure 37 Wireless node charging using solar power

The first step taken into consideration was the varying voltage input. Any renewable source never outputs a stable voltage. The voltage varies depending on the wind speed, sun position etc. To be able to use the varying voltage a charger IC was chosen that allows voltages up to 8V. This allows us to connect a single solar cell onto it without worrying about the current regulation for the Lithium batteries (Li-Ion & Li-Po).

# 6 Concept Designs

The system went through many different concept ideas. The aims of the project were to build a working system and depending on the time frame, incorporate as many features as possible. Other removed functionality was replaced by a more efficient version. Many extra features were not included in the final version; however the prototypes are listed below.

**SPI connected processors**

The initial design concept was to use multiple PIC microcontrollers interconnected using the SPI protocol. This would give enough serial and SPI ports to accommodate all the sensors and modules. A total of 5 PICs was to be used just for the main system panel. The master PIC would provide the bus clock and also poll all the PIC boards one by one to check for new data. All the data and power would be distributed along with the SPI lines. Each board that was connected on the SPI bus had a command table. The table was a “dictionary” for the bus commands. If the command 0xAA was received the slave returned the status. The 0xBB forced the slave to do a read etc.

**Results**

The initial tests were with three PIC microcontrollers.
The downside of this design is that it is too complicated, slow, error prone and especially really hard to debug. If a variable was to be shared between them they had to be reprogrammed with the updated firmware. This would cause delays during development. Managing five different workspaces in MPLAB is time consuming especially when only one can be opened at a time. The development of this architecture was abandoned due to the extreme modularity and complexity. At the time it was realised that a higher specification processor was needed.

**Linux board and Android tablet**

In order to create a complete user interface we had to design and create our own PCBs, program low level functions to print characters to the LCD display, record audio, edit it, convert it, create menus and use the wireless modules to control or read devices. Although we have a complete understanding of our hardware and software, a second alternative was also tested. This option did not require us to have a complete understanding of our hardware as before but rather knowledge in higher level languages such as Java. The hardware was an ARM Cortex-A8 1GHz processor powering Android OS. The GUI was an Android application and the RF links were connected to the serial port of the device.

Figure 38 Android 2.2 MID816 tablet connected to the key locator node

**Results**

Even though Ubuntu is a more flexible OS and was the one preferred, ARM devices with pre-loaded Android OS were cheaper. Since the tablet doesn't have a hardware serial port, a USB to serial converter was used.

The available source code was for the Profilic PL2303 IC. The source was compiled and the driver was loaded as a kernel module. The way the module was loaded was with the **insmod /modules/pl2303.ko** command. The converter was recognised and the serial port was registered. Basic write, read operations were performed but unfortunately there wasn't enough time for an android application.

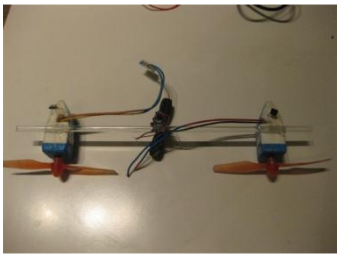


Figure 39 Propellers prototype

Figure 40 Blimp envelop next to tiny DC motors with propellers

**Blimp**

The concept behind the blimp was to add a robot that would be able to move easily between rooms, without having to deal with obstacles frequently.

The research and experiments showed that the blimp is a good way to navigate and collect data without any complex mapping that terrestrial robots need.

To keep it simple the function of the blimp was to follow the owner and wait for any commands. The blimp idea couldn't proceed with further implementation due to time limitations.

**Ethernet board**

An Ethernet module was tested using Arduino. The Ethernet PCB was responsible for the TCP/IP stack and the Arduino IC for the HTTP protocol handling. A server was implemented on the microcontroller. The server allowed us to control the microcontroller's peripherals by login-in to the server with PC client. However the device was very limited. There was not DHCP support and the network settings setup would deem it very difficult to demonstrate especially with the university's security regulations.

# 7 Final Design

**Main panel architecture**
The final design uses the simplest structure and keeps all the peripherals under one processor. The main processor is the STM32 ARM microcontroller. It features 48 GPIO has 512KB of storage (useful for image storing) and runs at 128MHz. The reason for not exploring this option first, was the uncertainty of transitioning to a new architecture. Moving from an 8-bit to a 32-bit architecture was not only a steep learning curve but a risky choice given the time needed to learn it.
The 90% of the processor peripherals are connected to external devices.
**Pin assignment**

- 30 pins dedicated to the LCD display and touch panel
- 4 pins for voice playback
- 1 pin for buzzer
- 1 LED
- 2 for Bluetooth (voice recognition module was replaced)
- 3 wireless(RX, TX, enable bit)
- 2 CurrentCost / shared with programming port

Since everything runs under the same processor, sharing and processing data between peripherals is easier.

**Software Architecture**
The software approach was keeping it simple but robust.
Having to maintain a number of peripherals is a tedious task since external asynchronous tasks can slow down internal processes.
The obvious solution was to use a Real-time operating system such as FreeRTOS. However a similar architecture was used, very similar to RTOS. The two main components that allowed our system to respond to many peripherals and keep processes running as fast as possible were the task handler and the interrupt tied peripherals.

**Task Handler**
Task handler is a function that does housekeeping such as refreshing specific portions of the screen (notification bar only), measures system speed and processor ticks and schedules when new values are shifted out of the system. The task handler function call is placed into every software function and the faster is called the better for the whole system. If the screen takes longer to refresh the wireless might be late but incoming data will never be lost due to the interrupts.

**Interrupt tied peripherals**
The other main component is interrupts. They allow incoming data to be saved without wasting processor cycles polling the peripherals. The way this is done is really simple. The processor serial ports fire an interrupt upon incoming data that takes us to the interrupt vector where our code is executed. The code checks the data, and places them into a buffer. Several buffers exist for all the serial ports. At the same instant a flag is raised that new data arrived and need to be processed.

Wherever the program counter is, the task handler is called after a few cycles, reads the flag, clears it and prints the data.

## 7.1 Results –Analysis

Almost all of the system modules had been extensively tested. The wireless nodes were connected to a DC supply and stayed ON for months.
The results of the data logging are located in the appendix A.

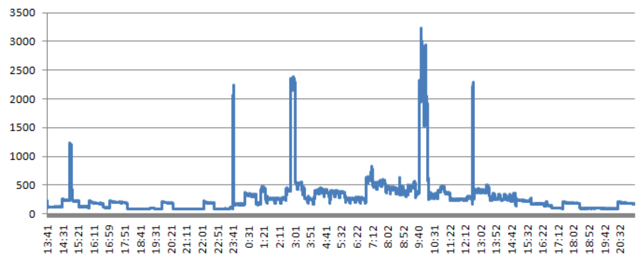



Figure 41 Winter consumption data for one day

# 8 Future thoughts - Discussion

One of the biggest aesthetic downfalls was the screen refresh rate. Although the processor was clocked fast enough the LCD interface mode created a bottleneck on the system. A whole screen refresh took around one second restricting the use of any graphics that could make the system much user friendly.
Apart from the screen refresh rate, the circuit for driving the LCD and touch panel required 30 general purpose I/O. This used two thirds of the processor peripherals leaving space for 18 GPIO. These were used for wireless, voice playback, Bluetooth, GSM module and other minor functionality such as LEDs and buzzers.
The ARM microcontroller on-board resources were all used for the system while keeping performance. However this system came up to a cost of 300GBP and triple that calculating the man-hours put into it. If someone was to sacrifice the ability to understand every component and go for a ready-made solution a commercial device such as a mobile phone can do a similar task with a lower cost. An android device mobile or tablet usually offers the same and more features. The Android switch will be further examined for future versions.

# 9 Conclusion

The emergent system came out to be fully functional and versatile. The wireless nodes can be exchanged and modified in a matter of minutes. Using this system for the past few months improved house consumption and gave us a taste of how a complete implementation would affect us. A broader understanding of action-consequence model indeed drove us to use energy more efficiently. All the goals have been met but there is always room for more functionality.

# 10 References

Khusvinder, G., Fang, Y., Lu, X. & Shuang, Y.H., 2009. A ZigBee-Based Home Automation System. *IEEE Transactions on Consumer Electronics*, 55(2), p.9.

Mouratidis, N. et al., 2009. Enabling Applicability of Energy Saving Applications on the Appliances of the Home Environment. *IEEE Network: The Magazine of Global Internetworking*, 23(6), p.9.

Panasonic Co., 2007. *Lithium Ion Batteries Technical Handbook*. [Online] Available at: http://industrial.panasonic.com/www-data/pdf/ACI4000/ACI4000PE5.pdf [Accessed 25 April 2011].

Smith, K., 2009. *ANTENNAS FOR LOW POWER APPLICATIONS*. [Online] Available at: http://www.rfm.com/corp/appdata/antenna.pdf [Accessed 25 April 2011].

Suh, C. & Ko, Y.B., 2008. Design and Implementation of Intelligent Home Control Systems based on Active Sensor Networks. *IEEE Transactions on Consumer Electronics*, 54(3), p.8.

# 11 Bibliography

Khusvinder, G., Fang, Y., Lu, X. & Shuang, Y.H., 2009. A ZigBee-Based Home Automation System. *IEEE Transactions on Consumer Electronics*, 55(2), p.9.

Mouratidis, N. et al., 2009. Enabling Applicability of Energy Saving Applications on the Appliances of the Home Environment. *IEEE Network: The Magazine of Global Internetworking*, 23(6), p.9.

Panasonic Co., 2007. *Lithium Ion Batteries Technical Handbook*. [Online] Available at: http://industrial.panasonic.com/www-data/pdf/ACI4000/ACI4000PE5.pdf [Accessed 25 April 2011].

Smith, K., 2009. *ANTENNAS FOR LOW POWER APPLICATIONS*. [Online] Available at: http://www.rfm.com/corp/appdata/antenna.pdf [Accessed 25 April 2011].

Suh, C. & Ko, Y.B., 2008. Design and Implementation of Intelligent Home Control Systems based on Active Sensor Networks. *IEEE Transactions on Consumer Electronics*, 54(3), p.8.

**Table of figures**