INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Wireless Sensor Based on Bluetooth Technology

Hervé MATHIEU

N° 0289

December 2003

THÈME 3

Rapport technique

ISSN 0249-0803

Wireless Sensor Based on Bluetooth Technology

Hervé MATHIEU (herve.mathieu@inrialpes.fr)

Theme 3: human-computer interaction, image processing, data management, knowledge systems

Technical report n° 0289 – December 2003 - 56 pages

Abstract: This document discusses a project which aimed of building a wireless sensor device. The objective of the project is to master a new sensor approach. Target applications are robotic systems including medical robots.
This project is decomposed in two phases. These two phases are reported in the main parts of this document (*overview* and *proof-of-concept prototype*).

The first phase deals with concept and general architecture. The following actions take place:

- We present target applications, and the kind of sensor that can be used in them.
- Existing similar research projects were investigated: Usually, documents related to these projects are open documents. That means that one has an idea about how work has been done, and what can be done (limits) in terms of technology. The drawbacks are that final products do not exist or that they are not described in any document.
- We investigated also off-the-shelf (commercial) wireless sensors: This class of information is useful to know what is reachable in terms of final products. However, you do not know how company did it: the "Commercial product" term implies usually a secret process.
- We list the electronic modules, which can be candidate for a wireless sensor device. That implies RF (Radio Frequency) modules, micro controller modules, energy technology, and sensor devices.
- We report uncertainties about technology and functionality that should be cleared up within a prototype project.

The second phase is related to a proof-of-concept prototype. The goal of this prototype is to know if the concept, as defined during the first phase, works in real.
We report how we successfully built a wireless sensor. Aspects presented are as follows:

1. About the test bed itself: specification, hardware description, embedded software, purchased list and cost, are given.
2. About the development environment: tools for programming and debugging the test bed, and experiments performed are described.
3. We report also the experiments results and discuss the prototype.

Off-the-shelf modules have been bought when existing. We tested the concept by using a dual axis inclination sensor. We chose this sensor because we had a previous experience with it. When designing the test bed, we tried, as much as possible, to get close to a usable product. Therefore, we took care of the sensor dimensions, the power consumption, etc.

Keywords: sensor, MEMS, wireless, Bluetooth, micro controller

Unité de recherche INRIA Rhône-Alpes
655, Avenue de l'Europe, 38330 Montbonnot-St-Martin (France)
Téléphone : +33 4 76 61 52 00 – Télécopie : +33 4 76 61 52 52

Capteur sans fil basé sur la technologie Bluetooth

Hervé MATHIEU (herve.mathieu@inrialpes.fr)

Thème 3 : interaction homme-machine, images, données, connaissances

Rapport technique n° 0289 – Décembre 2003 - 56 pages

Résumé: Ce document décrit un projet mené au sein du laboratoire d'expérimentation de l'INRIA Rhône-Alpes. Le but de ce projet est de maîtriser la technologie sans fil dans le cadre de capteur pour la robotique, dont la robotique médicale.
Ce projet peut être scindé en deux phases, qui sont reportées dans le document sous deux parties différentes (*Overview* et *proof-of-concept prototype*).
La première partie concerne l'état de l'art, les technologies mises en oeuvre et l'architecture générale. On y retrouve :

- une liste des projets de recherche travaillant sur ce type d'application,
- des produits industriels utilisant les mêmes technologies,
- des modules candidats pour réaliser notre prototype.

Cette partie décrit aussi ce qu'un prototype pourrait valider en terme de technologie et de fonctionnalités.

La seconde partie décrit comment un prototype, conçu sur les bases énoncées dans la première partie, a été réalisé avec succès.
Elle est organisée comme suit :

- la spécification complète (matérielle et logicielle) du prototype,
- les outils utilisés durant cette phase,
- les résultats obtenus.

Un effort tout particulier a été fait pour que ce prototype puisse devenir à moindre coût un outil utilisable dans des applications robotiques réelles.

Mots clés: capteur, MEMS, sans fil, Bluetooth, micro contrôleur

Wireless Sensor Based on Bluetooth Technology

Author:

Hervé MATHIEU

Research Engineer at INRIA Rhône-Alpes
Hardware & Software systems integration

Email: herve.mathieu@inrialpes.fr
Web home page: http://www.inrialpes.fr/sed/people/mathieu/Welcome.html

1 Contents

2 Part one: Overview

2.1 Introduction

One of the major problems with robotic systems comes from cables and connectors. Connecting device is not so reliable, and is expensive too. Everyone with some practical experience on hardware design knows that many hardware dysfunctions are due to connection problems.

Connecting grounds is another challenge when integrating electronic modules together. A bad strategy for connecting different grounds may induce current loops risk. And when it occurs, system might be damaged, and it is always very difficult to fix it. Wireless sensor is a good alternative to this problem.

And what about inserting a centimeter-scale sensor into a human body? It is obvious that a wireless solution is better than a wired solution.

On the other hand, many technology gaps have been bridged during last years:

- RF (Radio Frequency) devices are now ready for industry: wide bandwidth, high reliability and low power consumption. Cell phone application is a perfect example.
- Size and power consumption of sensors are dropping down dramatically, thanks to MEMS, and other CMOS integrated chip technologies.
- Micro controllers are powerful enough for most applications. Power consumption is low. Integrated functions like analog-to-digital converter (ADC), digital input/output, or TIMER are easy to implement.

Because, it seems that wireless sensor technology could improve many robotic and medical applications, we decided to launch a six month project (ref [annex 5]) aiming at mastering this technology.

After a short presentation of the INRIA Rhône-Alpes Experiment laboratory, we will justify why wireless sensors are so interesting, and why it makes sense designing a home-made wireless sensor.
For that we will explain first what target applications are, and then we will report investigation done on existing related research programs and off-the-shelf products.
At the end, we will propose a general architecture for wireless sensor, and give off-the-shelf electronic module candidates.

2.2 INRIA Rhône-Alpes experiment lab

The INRIA Rhône-Alpes experiment lab (ref [web00]) is working on robotic systems, like biped robot, or dedicated medical robotic devices. The team (10 persons) is composed of technicians and engineers with mechanical, electronics and software skills. They are working together with researchers on experiments. They provide high quality robotic systems.

One of the major designs is a biped robot which is discussed below.

2.3 Target applications

2.3.1 Biped Robot

The biped robot is a 15 dof (degrees of freedom) robotic system (ref [1]). This biped robot has been built by the *Laboratoire de Mécanique des Solides de l'Université de Poitiers* (mechanical skill), the *service robotique de l'INRIA Rhône-Alpes* (electronics skill) and the research group *BIP of INRIA Rhône-Alpes* (control law part).
Several sensor devices are plugged onto the biped robot for delivering information to the controller. Force sensor devices are located under feet, potentiometers are fixed on each joint for providing absolute position, and inclination sensors are used for global positioning. The actual version of the biped robot implements sensors in a conventional method: i.e. connecting two wires for power supply and one or two wires (depending on single ended or differential mode) for signal.

In this kind of robotic systems, all the sensor devices are closed to the controller (less than 2 meters). Wires have to go through structure and moving joints. This makes wiring quite hard. Wiring grounds properly is also a big issue. We have to take care of this for avoiding ground loop effects.

Wireless sensor technology would help. It would result in dramatic hardware simplification, and would make the whole system more robust.

2.3.2 Medical applications

Robotic systems are used for medical applications, especially for surgical procedures. Medical robotics aims at improving the capabilities of physicians to perform surgical procedures. Some existing applications are listed here.

1. A typical robot-assisted procedure is a biopsy under CT imager control. The procedure consists in the following steps:
    - Step 1: taking a CT scan where target is supposed to be located,
    - Step 2: target and entry-point identifications are made using CT slices. Target is usually between 20 and 100 millimeters depth inside the body. The skin entry-point represents the hole where the linear tool (needle, catheter or bone drill) will be inserted. This two-point identification gives three parameters in case of a linear tool: two angles for tool orientation and one distance (depth) for insertion.
    - Step 3: manual or robot-assisted placement typically includes the following three decoupled tasks: (1) touch down with the tool tip on the entry-point, (2) orient the tool by pivoting around the skin entry-point, and (3) insert the tool into the body along a straight trajectory.

    Performing this robot-assisted procedure requires registration of robot and imager. Registration process consists in finding a 4x4 matrix. This matrix represents how coordinates change between imager coordinate system and robot coordinate system. Registration requires several points of interest which are paired in both coordinate systems. Pairing is always a critical phase.

    Adding sensors improves the registration process. For instance some of the points of interest described above allow calculating what the vertical direction for both coordinate systems is. An inclination sensor gives directly this information Therefore less points of interest have to be paired. This helps reducing matrix resolution complexity, and so makes the registration process more robust.

    Adding inclinometer sensor on a robot improves also safety. It can be used as redundant encoder, because it gives information about robot orientation too. Both information coherencies can be checked, thus avoiding hazardous robot behavior due to motors encoder dysfunction.

2. During a total knee replacement, procedure includes machining bones. Some robotic systems perform this procedure, like ROBODOC from Integrated Surgical Systems (ref [web01]). A typical problem is doing registration between the robot and the patient's body. A common registration is based on localizers like Optotrack or Polemus from North Digital (ref [web02]). Basically, several tracker devices are attached to the body, and the others to the robot. All tracker devices are seen by the localizer, and so they are represented in a unique coordinate system. This makes possible registration between the robot and the patient's body.

    The drawbacks for using common localizers are the following: (1) tracking failure due to occlusion (tracking devices commonly used cameras), (2) a bunch of wires around patient due to wired tracker devices. These active trackers (versus passive trackers) improve precision, (3) a large area inside surgery room is dedicated to the localizer system.

    Adding wireless inclination sensor improves procedure. It helps on positioning when occlusions occur. Mixing tracking and sensor outputs improves precision. Hence passive (non-wired) tracker could be considered.

In general, adding sensors improves a procedure in terms of safety for the patient and precision for procedure. But adding wires is a big issue. A surgical room is a confined area. All material has to be sterilized…

Having wireless sensors mean fewer wires around the patient and the physician. Therefore it provides a more comfortable and safety working environment.

But to be accepted within medical procedures, wireless sensors must fit following conditions: (1) they must be safe for patient, (2) they must be packaged as small as possible because of confined area, and (3) they must be sterilizable.

We summarize here common features extracted from applications mentioned above. In other terms: what do we need?

- A set of sensors, between 1 and 5 devices.
- All sensor devices are located within a small area (inside a 3 meter diameter sphere).
- Sensor devices must be as small as possible (hand-size).
- It must be sterilizable.

2.4 Existing similar projects

2.4.1 Research projects

The research projects ("Smart Durst" and "Smart its") described below are similar to our project, because they study wireless sensor. The difference is that these projects focus on large-scale wireless sensor devices. They deal with logical connection and data flow complexity problems. In our project, only few sensor devices will be connected. Our connection strategy corresponds to a "master-slave" configuration. Therefore communication is not a big issue.

"Smart Durst" (ref [web31])

This project is located at University of California at Berkeley.
Crossbow (ref [web12]) is an industrial partner in this project.
Intel research is working on a Bluetooth based wireless sensor implementing TinyOS™ (ref [web32]).
The original smart durst goal is building a 2mm x 2mm x 2mm cube including power supply based on solar energy, RF communication and/or Laser based communication, and sensor devices. Technology is based on MEMS. For now, they developed centimeter-scale wireless sensor.

"Smart-Its" (ref [web27])

This is collaboration between Lancaster University, ETH Zurich, University of Karlsruhe, Interactive Institute and VTT (Technical Research Centre of Finland).
The project goal is to develop unobtrusive, deeply interconnected smart devices - called Smart-Its - that can be attached to everyday appliances in order to support new functionality, novel interaction patterns and intelligent collaborative behavior. Smart-Its will be cheap.
Each of the labs involved in this project has developed its own prototype. ETH has the most documented prototype (ref [2] and [3]).

Another project is the **MIT Ultra Low Power Wireless Sensor Project** (ref [web33]), under the direction of Professor Charles Sodini. The project is funded over the next four years by the Advanced Research Projects Agency (ARPA).

2.4.2 Commercial products

Crossbow (ref [web12]) is a USA company delivering products developed at University of California at Berkeley. Products base name is MICA. OEM products are available. Crossbow ships three "Mote Processor Radio" module families – MICA (MPR300), MICA2 (MPR400), and MICA2-DOT (MPR500). These modules are used in end-user and OEM applications (ref [4] and [5]). All of these modules provide a processor that runs TinyOS (developed at Berkeley) based code, a two-way ISM band radio transceiver (not Bluetooth), and a logger memory capable of storing up to 100,000 measurements.

Oceana Sensor (ref [web10]) is a manufacturer of sensing systems for a wide variety of applications including machinery health, seismic monitoring or impact detection. They instrument sensors with Bluetooth interface. Products are clearly end-products, not for OEM.

Millennial Net (ref [web13]) produces low-power wireless sensors. These products are used in industrial automation, building automation, supply-chain management, automatic meter reading, security, and other low-data-rate applications. The company's primary product is the “i-Bean”. It combines analog and digital interfaces, a radio transceiver, and a microcontroller for sensor signal processing, control and networking. Millennial Net uses a portfolio of radio technologies including ultra-low-power narrowband solutions as well as IEEE 802.15.4 wireless personal area network (WPAN) components.
Milennial Net was created in year 2000, and traces its roots to MIT.

Sensicast Systems (ref [web11]) develops and markets low-power, battery-operable, wireless sensor network solutions for the commercial, industrial, agricultural and security industries. Star product is H900 Wireless SensorNet System. It is based on 900 MHz RF band. Sensicast Systems, Inc. was created in 2002.

One can notice that most of the companies were created in the last four years, and are located in USA. They are working together with world class universities. Some products are based on Bluetooth technology, others on 900 MHz or 2.4 GHz RF bands.
Open products (OEM market) are used in network and communication research programs. OEM products usually are based on non Bluetooth technology. Research groups working on this product class claim that the Bluetooth protocol is too complex and too slow during the connecting process. Therefore they choose RF technologies which could provide an easier access.

Products really designed to be a sensor device are end products. Some of them are based on Bluetooth technology. Some companies claim that Bluetooth technology makes products easy to integrate, because of wide use of this technology. “Any PDA or Laptop allows connecting you”.
Bluetooth based products do not really fit the specification described in the “target applications” section: their dimensions are generally too wide. They are designed to be used with a particular sensor, not with the one you would like to use.

At the end, we get the following conclusions:

1. This is the beginning of the wireless sensor history.
2. It makes sense to design a home-made wireless sensor.
3. Among wireless technology used in projects or products reviewed, Bluetooth technology is a good candidate in our context. Bluetooth technology is available on existing computer via dongles, and it is accepted by a lot of industrial companies. Debug phase and experiment integration would benefit.

Two documents help considering Bluetooth technology as a good candidate for a wireless sensor device.
The first document (ref [6]) described how a Bluetooth based module was built. The second document (ref [7]) presents an interesting overview of Bluetooth based sensor architecture.

2.5 Wireless sensor device architecture

We decide to specify the sensor device choosing off-the-shelf electronic modules when possible. Of course, sensor compactness is not optimum, nor power consumption, but time development will be reduced significantly.

The project goal is clearly not developing a 2mm x 2mm x 2mm sensor device as Berkeley final goal within “smart durst” research project. We do not have access to MEMS technology as developer, and people involved in this project are not so many. The project goal is developing a hand-size (50mm x 50mm x 50mm) sensor device.

This wireless sensor device consists on a box containing (1) a sensor based board, (2) a micro controller card, (3) a wireless communication card, (4) and a power supply stage.

The following figure (ref [Figure 1]) presents a device architecture based of functional blocks. Most of the blocks are also materialized with a electronic module.

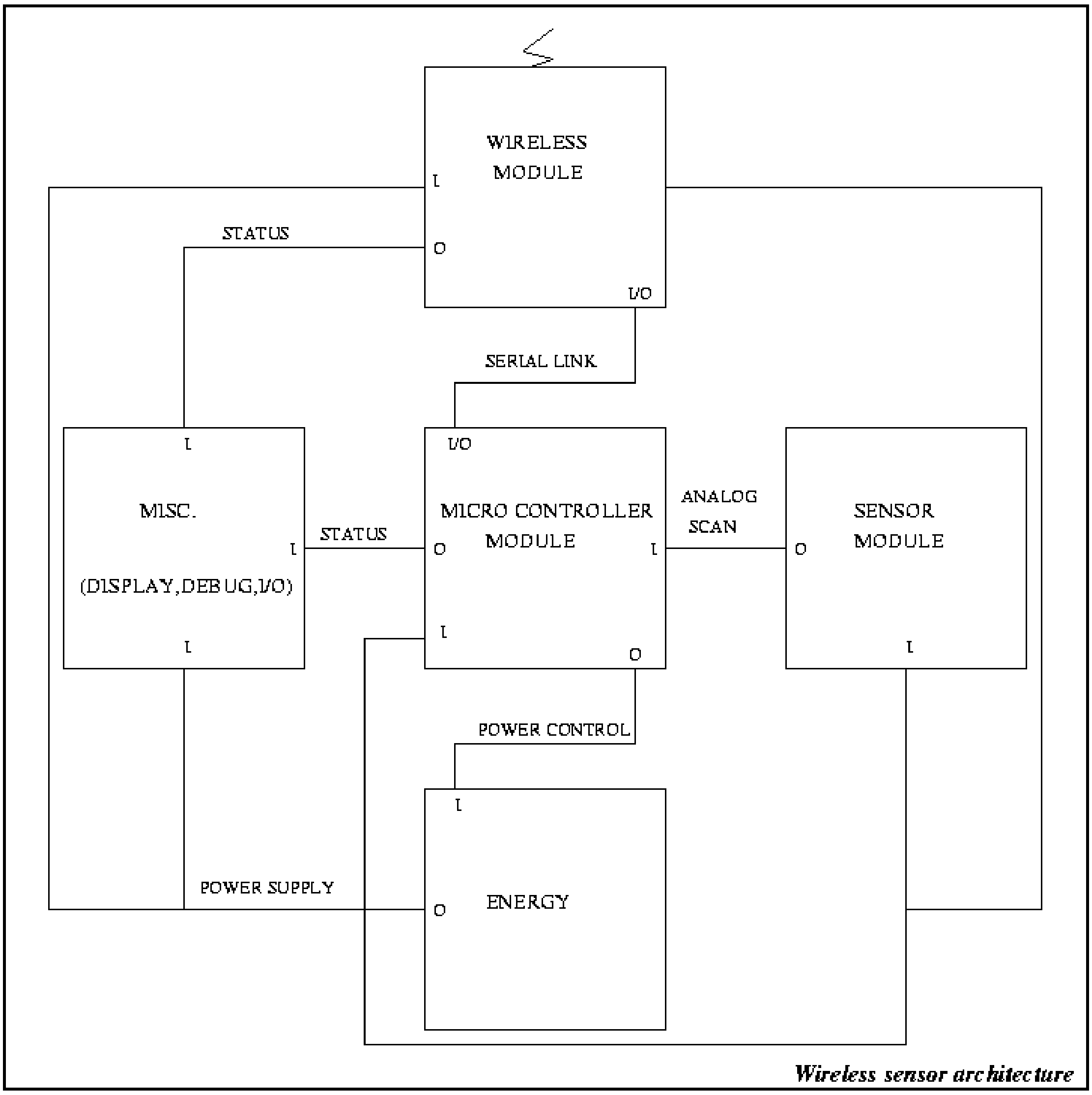


Figure 1: wireless sensor general architecture

Let's see in details each part.

- **Sensor**: MEMS sensors are everywhere today. Magnetic, temperature, humidity, vibration, position, you can measure what you need for few square millimeters and little energy.
- **Micro controller**: tens of micro controller cards are found on the market. What we need is a low power micro controller, with few analog-to-digital converters (ADC). Because some calculation may be done on the micro controller, for instance for data conversion, it is worth to skip 8 bits micro controllers and to focus on 16 or 32 bits versions. Products like Microchip PIC18FXXX family, or Infineon C16x family, Hitachi H8 family, or Texas Instrument MSP430 are fitting.
- **Wireless communication**: Any Bluetooth module on the market has almost all the same hardware design. Some modules are built with antenna, some without. Module control is done with a serial or USB (Universal Serial Bus) communication link. The main difference resides in software layers. The Bluetooth software stack is complex. Modules usually do not implement the whole protocol, but all of them implement Bluetooth stack up to HCI (Host Controller Interface). In this case, a additional micro controller runs Bluetooth stack above the HCI layer. But some modules implement a full profile, for instance the Serial Port Protocol profile (detailed below). This profile emulates a wireless serial communication link. In fact this is exactly what we need for the project.

- **Power supply (energy)**: this part really depends on other modules. During development phase, an external power supply has to be provided first. An exhaustive calculation for power consumption has to be done. Then, a technology shall be chosen.
  Anyway, about technology, Li-ion battery is optimum in terms of power versus weight. Primary battery might be considered also.

The following sections give some details about the technologies listed above. For each of them we list what has to be demonstrated with a prototype (second project phase).

2.6 Sensor

2.6.1 What kind of sensors?

As described in section about target applications, sensors of interest are:

- Accelerometers, which are used as inclination sensor too. Accelerometer sensor is a member of absolute family sensor. It deals with physic laws like gravity field or earth magnetic field. This kind of sensor can be easily integrated into the wireless sensor box. Therefore packaging and sterilizing are much easier.
- Force sensor which are usually based on piezo-electric technology. This kind of sensor has to be attached to some hardware (structure where you need measuring). Packaging is a big issue. It can not be generic, because it depends on structure shape. Also it is much harder to sterilize because of separated pieces.

This kind of sensors has the following common features:

- Frequency is between 10 and 1000 Hertz
- Signal dynamic is between 8 and 16 bits

To be consistent with, we should choose a 16 bits micro controller and a communication link throughput faster than 16 kilo bits per second. In fact most of 8 bits micro controller deal with 16 bits data with no really drop out performance.
We can notice that Bluetooth bandwidth is over 16 kilo bits/sec. In fact throughput is between 128 kilo bits/sec and 1000 kilo bits/sec, depending on Bluetooth configuration.

Other sensors like pressure, temperature or humidity have the same features, but they are not considered for the moment.

Sensors like CMOS video camera are much more complex. They need much higher resources. Micro controllers involved in this class of sensor are usually 32 bits. The communication channel bandwidth required is over one mega byte per second.

MEMS sensors are an emergent technology. They have low power consumption, and are very small. Their precision is now relevant for most applications.

How to define MEMS technology? (Extract from www.memsnet.org (ref [web06]))

Micro-Electro-Mechanical Systems (MEMS) is the integration of mechanical elements, sensors, actuators, and electronics on a common silicon substrate through micro fabrication technology. While the electronics are fabricated using integrated circuit (IC) process sequences (e.g., CMOS, Bipolar, or BICMOS processes), the micromechanical components are fabricated using compatible "micromachining" processes that selectively etch away parts of the silicon wafer or add new structural layers to form the mechanical and electromechanical devices.

2.6.2 Extensions to no-sensor devices

The general architecture shows a communication module plugged with a micro controller module. We may consider using theses bricks to make another king of device, for instance a remote controller device.

Powering a low power laser beam is an example. This kind of device is controlled by a simple digital output. This makes it easy to integrate.

A more ambitious goal could be controlling a motor. In this case a power stage has to be added. The micro controller resources must be consistent, that means for instance the capability for reading encoders. The communication bandwidth detailed above would be enough. Providing the embedded energy is another big issue.

2.6.3 Examples

We listed in table (ref [Table 1]) sensors that could be used.

Trademarks	Type	Reference	Cost (euro)
MEMSIC (ref [web03])	Accelerometer (+inclination)	MXR2999	30
ANALOG DEVICE (ref [web04])	Accelerometer (+inclination)	ADXL202	40
ANALOG DEVICE (ref [web04])	Angular rate sensor (gyroscope)	ADXRS150	40
VECTOR TECHNOLOGY	Laser diode	5001-21	130
CEA-LETI (ref [web05], [8] and [9])	3Accelerometer+3magnetometer	-	Prototype

Table 1: sensors

2.6.4 To be demonstrated with a prototype

- In one hand, the sensor must be properly powered. It must be far away from electronic noise sources. On the other hand, we have a wireless module. Its power consumption is much higher when communication takes place. Therefore it can create power supply noise. A prototype will validate that a sensor is working well in this context.
- How the sensor module should be installed into the wireless sensor box? For instance, an inclination sensor must be positioning with accuracy. Registration between the sensor and the box should certainly be done. Shall we use mechanical references or an experimental calibration phase?

2.7 Micro controller module

2.7.1 Specification

Tens of micro controllers exist. They are based on 8, 16 or 32 bits core.

Implementing a micro controller in a design implies developing an electronic PCB with some circuitry: quartz, resistor, and capacitor. Because we wanted to minimize PCB developments, we look for an off-the-shelf module. The goal was to have rail to rail connections when possible.
In fact, we found very few modules according to the specifications. Modules listed after include modules with dimensions over 50mm x 50mm, but less than 60mm x 60mm.

General requirements for micro controller module are summarized here:

- Small footprint (less than 50mm x 50mm).
- Power consumption less than 120 milli-Amperes.
- UART (any micro controller get at least one). This port is connected to Bluetooth module.
- At least 4 AD inputs (for instance for two dual axis inclinometer sensors), At least 10 bits precision is required. This means 360/1024 equal 0.35 degree angle resolution for a 360 degrees sensor analog signal.
- Over 4 digital inputs.
- Over 4 digital outputs.
- Flash memory.
- Intelligent power management capability for reducing power consumption.
- Single power supply (either 3 or 5 Volts but 3 Volts would be better).
- C language development tool available.

2.7.2 Examples

We report in the next table (ref [Table 2]) modules which fit the specification. MICROCHIP PIC micro controller family is a very interesting candidate for this kind of application. But we did not find any integrated module with mandatory dimensions.

Company	**Model**	**Micro controller**	**Size (mm)**	**Power supply**	**IO**	**Software Dev Tool**	**Cost (euros)**
PHYTEC (ref [web07])	phyCORE-ARM7	Atmel AT91M55800A	60x53	3.3 Volts, 110 mA (typ)	AD:8x11bit Ethernet DA:2x10bit	C compiler (www.keil.com)	295
PHYTEC (ref [web07])	phyCORE-ST10F168	ST10	60x53	5 Volts, 110mA (typ)	AD:10bit	C compiler (www.keil.com)	165
RENESAS (ref [web09])	Module AE3664	H8/3664F	40x27	5 Volts, 50mA (typ)	AD:8x10bit	GNU C compiler www.kpit.com/	60

Table 2: micro controller modules

2.7.3 To be demonstrated with a prototype

- The analog inputs dynamic range has to be consistent with sensor dynamic range.
- The power consumption must be really low. For instance, power management must be easy to implement.
- The flash memory must be large enough for the developed program. This point is crucial, but the problem is that we are not able to know in advance what will be the final compiled code size. For instance concerning the Bluetooth module associated code, its size depends on what kind of Bluetooth functionalities we need. A full control of the Bluetooth stack means hundreds of code lines. But a minimum configuration is done by only five code lines.

2.8 Bluetooth technology

Note: a special award for the http://www.palowireless.com/infotooth/tutorial website (ref [web30]). This is clearly the best website we found about Bluetooth technology.

The following websites (ref [web15], [web16], [web18], [web28] and [web29]) have been also read for this part.

2.8.1 Technology Overview

Bluetooth is of course not the only wireless technology available. Other technologies like WLAN, HYPERLAN, DECT exist. The main features of the Bluetooth technology are: short range RF (<10 meters for 1 milli-Watt version), low cost, up to 8 connected nodes together (*piconet*). It fits projects working on local sensor devices networks.

A large majority of the project reviewed uses either Bluetooth technology or wireless technologies based on the same RF specification (2.4 GHz, spread spectrum, frequency hopping).

Bluetooth™ is the codename for a technology specification for low-cost, short-range radio links between mobile PCs, mobile phones and other portable devices, and connectivity to the Internet.
Unlike many other wireless standards, the Bluetooth wireless specification includes both link layer and application layer. Radios that comply with the Bluetooth wireless specification operate in the unlicensed, 2.4 GHz radio spectrum ensuring communication compatibility worldwide. These radios use a spread spectrum, frequency hopping, full-duplex signal at up to 1600 hops/sec. The signal hops among 79 frequencies at 1 MHz intervals to give a high degree of interference immunity. Up to seven simultaneous connections can be established and maintained.
The Bluetooth specification contains the information necessary to ensure that diverse devices supporting the Bluetooth wireless technology can communicate with each other worldwide. The document is divided into two sections: a Core Specification (Volume I) and Profile Definitions (Volume II).

Note about power classes: Each device is classified into 3 power classes, Power Class 1, 2 & 3.

- Power Class 1: is designed for long range (~100m) devices, with a max output power of 20 dBm,
- Power Class 2: for ordinary range devices (~10m) devices, with a max output power of 4 dBm,
- Power Class 3: for short range devices (~10cm) devices, with a max output power of 0 dBm.

What is a *piconet*: a collection of devices connected via Bluetooth technology in an ad hoc fashion. A *piconet* starts with two connected devices, such as a portable PC and cellular phone, and may grow to eight connected devices. All Bluetooth devices are peer units and have identical implementations. However, when establishing a *piconet*, one unit will act as a master and the other(s) as slave(s) for the duration of the *piconet* connection. They all share the same physical channel defined by the master parameters (clock and BD_ADDR).

2.8.2 The Bluetooth Name

The Bluetooth name itself comes from the 10th century Danish Viking King, Harald Blåtand (Bluetooth in English), who united and controlled Denmark and Norway, hence the inspiration of the name, by uniting products through Bluetooth.
Harald apparently enjoyed eating Blueberries, to such an extent that his teeth were stained blue!

2.8.3 Available Bluetooth hardware modules

When integrating Bluetooth technology into an application, you should consider purchasing a Bluetooth module. It is clearly not reasonable developing a home-made electronic module.

A Bluetooth module is a small card (about 30mm x 20mm x 4mm), it contains few components, including a Bluetooth chip.

Only few Bluetooth chipset exist: Ericsson/Infineon (ref [web17]), CSR (ref [web14]), Philips, ST (ref [web19])…. but a large number of modules are on the market.
Common features are:

- Inputs control modes available are USB, UART and PCM for voice transmission,
- A chipset with flash memory,
- An antenna, which may be an option, for enabling more powerful antenna.

The module power consumption is between 50 and 80 milli-Amperes when transmitting, less during other modes. The power supply voltage is usually 3 Volts.

Some existing Bluetooth modules are listed in next table (ref [table 3]). Common features of the listed modules are: (1) French distributor available, (2) class-II Bluetooth, (3) on-module antenna, (4) GAP&SPP profiles, and (5) UART port control.

Company	**Product name**	**Size (mm)**	**Misc.**	**Power supply**	**Cost (euros)**
ConnectBlue (ref [web20])	OEM Serial Port Adapter 13i	23x36x5	Multi connections	3-6 Volts, 40 mA@5V max	158
Mitsumi (ref [web22])	WML-CL10	12x18x2	Point to point only	1.8&3.3 Volts, 60mA	240

Table 3: Bluetooth modules

Euridis (ref [web21]) is a French distributor for ConnectBlue products.
Micropuissance (ref [web23]) is a French distributor for Mitsumi products.

2.8.4 Bluetooth specification protocol stack

The Bluetooth specification deals with hardware and software. The software specification is quite complex. It includes several stacks (horizontal slices) and deals with several protocols (vertical slices) (ref [Figure 2]). 13 different protocols have been defined during the very first specification. More profiles have been added during last years.

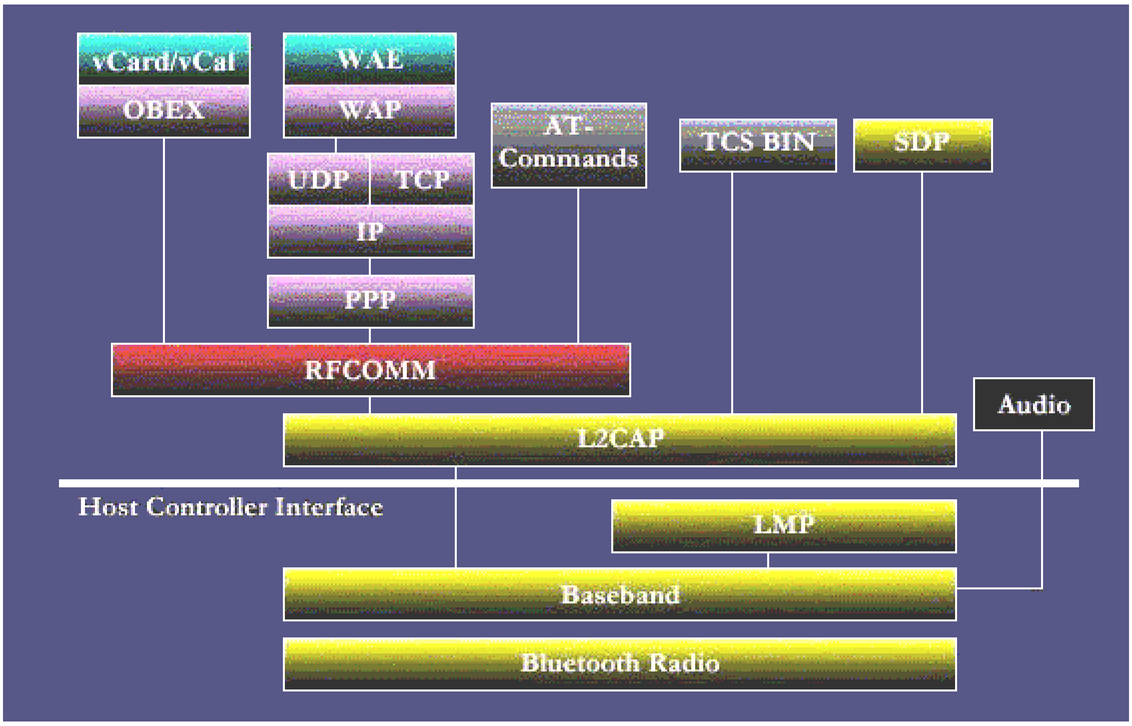


Figure 2: Bluetooth specification protocol stack

1. **Radio** The **Radio** layer defines the requirements for a Bluetooth transceiver operating in the 2.4 GHz ISM band.
2. **Baseband** The **Baseband** layer describes the specification of the Bluetooth Link Controller (LC) which carries out the baseband protocols and other low-level link routines.
3. **LMP** The **Link Manager Protocol (LMP)** is used by the Link Managers (on either side) for link set-up and control.
4. **HCI** The **Host Controller Interface (HCI)** provides a command interface to the Baseband Link Controller and Link Manager, and access to hardware status and control registers.
5. **L2CAP Logical Link Control and Adaptation Protocol (L2CAP)** supports higher level protocol multiplexing, packet segmentation and reassembly, and the conveying of quality of service information.
6. **RFCOMM** The **RFCOMM** protocol provides emulation of serial ports over the L2CAP protocol. The protocol is based on the ETSI standard TS 07.10.
7. **SDP** The **Service Discovery Protocol (SDP)** provides a means for applications to discover which services are provided by or available through a Bluetooth device. It also allows applications to determine the characteristics of those available services.

First profiles defined are:

1. GAP, generic access profile
2. SDAP, service discovery application profile
3. CTP, cordless telephony profile
4. IP, intercom profile
5. SPP, serial port profile
6. HP, headset profile
7. DNP, dial-up networking profile
8. FP, fax profile
9. LAP, LAN access profile
10. GOEP, generic object exchange profile
11. OPP, object push profile
12. FTP, file transfer profile
13. SP, synchronization profile

Off-the-shelf modules are usually delivered with a partial stack, up to HCI. Three solutions are on the market for completing the Bluetooth stack.

- Some companies (for instance Widcomm ref [web26]) have written a certified Bluetooth stack. You get the whole source code for a big amount of money. One or two compiled profiles can be delivered for less money.
- Download a free but not certified Bluetooth stack. OpenBT (ref [web25]) and Bluez (ref [web24]) are based on the Linux distribution.
- Write your own Bluetooth stack upon the HCI, which means a six months software project.

Due to the Bluetooth chips limitation, too many profiles loaded on module reduce dramatically RF performance. A second host (for instance a micro controller) usually runs part of the protocol.

We looked for purchasing a module operating a profile like the serial port, or the LAN profile. This kind of module is not so common, it is expensive too. But it can reduce dramatically the development cycle.

We consider in a future project working with a basic module. Therefore, it could reduce the module cost and we could purchase any commercial module. The drawback is that the Bluetooth protocol stack will end at the HCI protocol stack. Missing Bluetooth stacks will have to be developed.

2.8.5 To be demonstrated with a prototype

- How complex is a development using a Bluetooth module? In one hand, very few experiences have been reported using Bluetooth within research project (ref [2], [3 and [7]]). On the other hand, Bluetooth companies announce day to day new modules available.
- The power consumption and the power management are not clearly described in technical documents. Some reports concerning research project rise up problems concerning the power consumption.
- Does it works within a robotic environment? Bluetooth primary market includes cell phone, computers, PDA. Robotic or medical environments include a lot of metallic parts, noisy machine like X-Ray scanner.

2.9 Energy

2.9.1 Technologies

Technologies used in such systems are usually based on solar power or battery power. We do not here address self powered technology based on RF transmission. Solar power is a valuable choice for outdoor experiments and the described applications sets typically indoor. Thus, we give up with solar power. At the end we looked for battery technology.

The Li-ion battery technology is today the most efficient. Li-ion voltage element is 3.6 Volts. Cell power is between 400 milli-Amperes per hour and 1500 milli-Amperes per hour for AA battery type. The typical discharge current value is between 1 milli-Ampere and 400 milli-Amperes. The charge strategy is to provide a constant voltage (4.2 Volts). The charge process ends when the load current reaches zero. Li-ion battery is intensively used for video recorder, laptop, and digital camera. It becomes also a growing market for Radio Control Modeler and embedded systems in general.

A drawback of this technology is its relative weakness. Charging a Li-ion cell over 4.2 Volts, or discharging a Li-ion cell when voltage is under 2.7 Volts, will definitively damage your cell. Most of Li-ion batteries are packaged in a hermetic box containing a safety electronic circuitry. This circuitry plays as a safety switch to prevent damaging the cell. We did not find any off-the-shelf Li-ion cells in a raw format.

NiMH (Nickel Metal Hybrid) battery is more affordable than Li-ion battery. The technology is less efficient in terms of power versus weight, but is more suitable for high current load. NiMH element voltage is 1.2 Volts. AAA or AA (cylindrical) format is the common packaging format.

The document (ref [10]) gives an interesting overview of battery technology. The following websites (ref [web34], [web35], [web36] and [web37]) describe existing products.

We consider using a lab power supply for developing the test bed. In a second step, primary battery will be used in our first experiment. But the specification will take into account secondary battery as power supply. In fact we do not consider embedding a battery loader. So the design does not depend on technology used. Both primary and secondary battery are considered as voltage generator.
The challenge for the sensor is to have the capability for rising up an alert when energy is low. Then the strategy is different according to the battery technology. Li-ion battery output voltage is almost linear with respect to discharge. We do not know how to deal with the primary battery technology.

2.9.2 Power saved strategies

A major challenge for a sensor device is being awake as many hours as possible. Power saved strategies might be activated when possible for extending autonomous period.

First power saved strategy consists on activating module power saved. Most of micro controller chips do have this feature. Bluetooth modules seems having this feature too. But, because of module complexity, it is not clear what kind of power saved feature may be activated. It might also depend on Bluetooth configuration. This has to be investigated.
Sensor modules usually are "non intelligent" chip with no power saved feature.

Then, an action to switch off modules may be considered. The Micro controller might cut power of unused modules.

At the end, we shall consider the capability to switch off the whole system. This implies: (1) to get an external action for switching on the system, for instance a push button, (2) the capability to switch off the module by using a button or by using a programmed watchdog process.
This strategy step is important for applications meaning. The wireless sensor device would not be an autonomous sensor device anymore. It is clearly a limitation for some applications. It has to be considered as last chance for saving energy.

2.9.3 To be demonstrated with a prototype

- An exhaustive power consumption calculation must be done. Depending on the battery technology used, we will estimate how long a wireless sensor device can be powered (life cycle). Different power saved strategies might be tested.
- The capability for avoiding power supply failure when the battery charge is empty.
- Batteries are usually packaged in a metallic box. It could disturb wireless transmission. This has to be evaluated.

2.10 Packaging

The packaging is a big issue, but it might be finalized only when all internal parts will be defined.
The general requirements for packaging are:

- Hand-size container: box sizes must be less than 50mm x 50mm x 50mm
- Sterilizable
- Radio Frequency free way compatible. It implies avoiding metallic parts.

Most of plastic boxes are suitable.

2.11 Conclusion (overview part)

We present target applications where wireless sensor could improve dramatically the process. In other words we explain why wireless sensors were so interesting.

Then we listed existing related projects or products. We mentioned also that none of existing devices fitted the needs, thus justifying making a prototype.

We presented a complete and detailed review of technology candidates, and a global architecture. In other words, we suggested that technologies to build such a device were ready.

But we also list some device features which have to be demonstrated in a prototype: the power management facilities, the wireless link robustness,… Building a prototype is an opportunity to makes things clear.

Next part gives the proof-of-concept prototype specification and talks about experiments and results performed.

3 Part two: proof-of-concept prototype

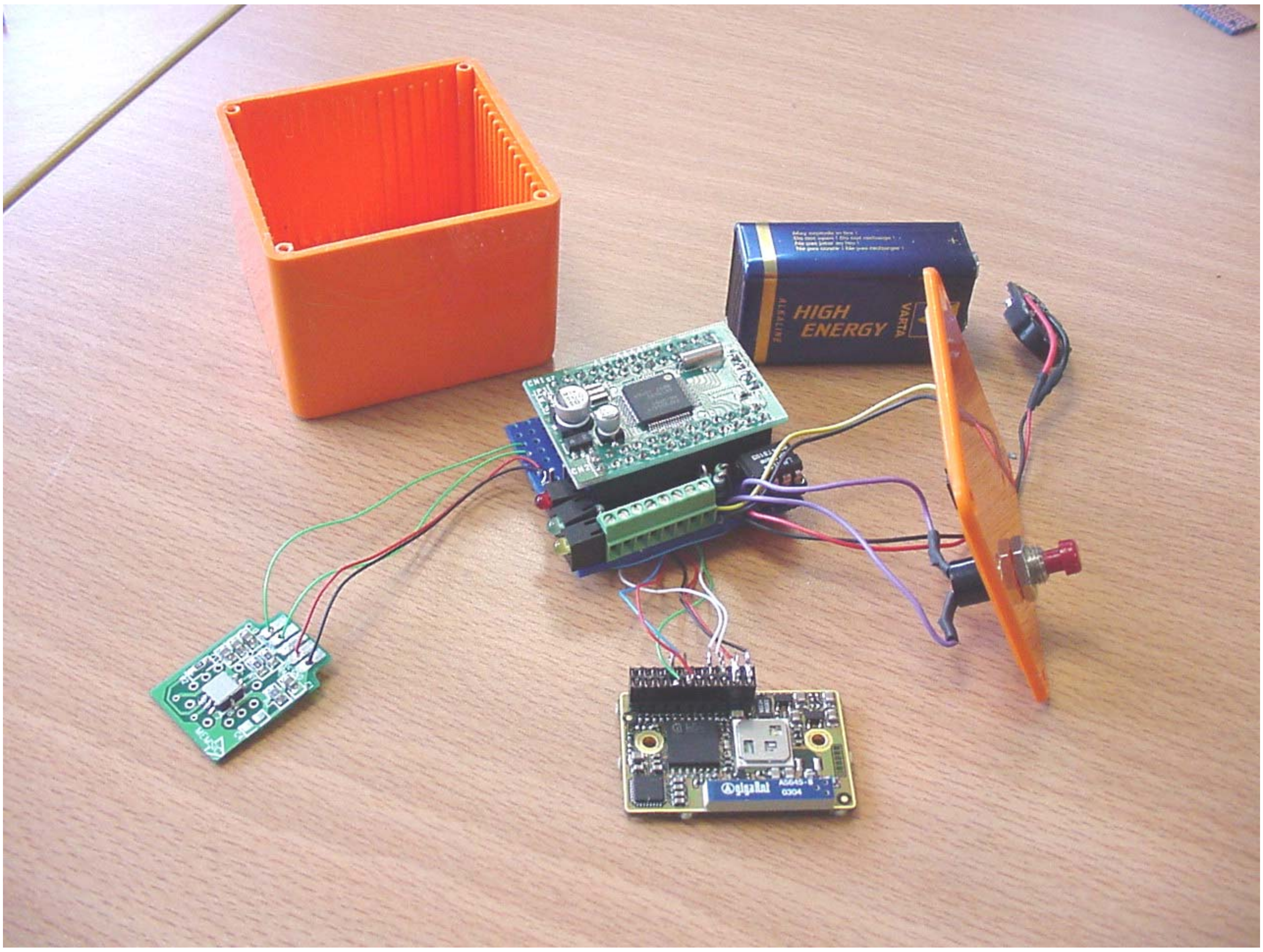


Figure 3: the prototype

3.1 Introduction

The objective for the test bed experiment consists on validating wireless sensor concept, as defined in the first part (overview). Roughly, it consists on packaging inside a small box, a sensor, in order to evaluate its communication performance over a wireless channel, and its power consumption.

In practice, test bed experiment consists on a remote dual axis inclination sensor (ref [Figure 3]).

This document is composed of following parts:

- First it gives how the sensor device works.
- Then the off-the-shelf modules used in test bed are detailed.
- It reports also the interface module specification. This module has been developed for inter-connecting the off-the-shelf modules.
- The embedded software and the development environment are described.
- Energy and packaging specifications are presented.
- At the end, the results concerning power consumption and communication experiments are reported.

This document may be considered as:

- An electric and mechanical manual for building a sensor device.

- A user manual for working with the sensor device.

3.2 How does the sensor device work?

The sensor box is referenced to the sensor itself. Therefore, the remote sensor device gives the box orientations.

A typical wireless sensor device session is described with the following states:

- **Power on state**: when powering on the device (pushing button), the micro controller and the Bluetooth module execute an initialization phase. Then they both switch to a power saved mode.
- **Connect state** is initialized by a master, typically a PC instrumented with a Bluetooth dongle. It establishes a Bluetooth channel between the master and the remote sensor. Bluetooth wakes up first, and then micro controller wakes up too.
- **Communication state:** the master requests X and Y axis values from the remote sensor. Another similar function requests a bunch of data, thus avoiding communication delays.
- **Disconnect state:** the master closes Bluetooth channel. Bluetooth switches first to power saved mode, and then micro controller switches to power saved mode too.
- **Power off state:** All is off!

A led located on the top panel of the box indicates the device status as follows:
- LED OFF, the device is off.
- LED ON, the device is connected.
- LED BLINKING, the device is on but not connected.

After a connection has been established with the wireless sensor device, commands are sent using ASCII characters format.
A very simple protocol is implemented:
1. The host sends a command (= a character),
2. The remote device sends back a character string with an ending delimiter. End delimiter consists on ASCII new line (0x0A), then ASCII carrier return (0x0D). Answer begins with command echoing. It may include figures or strings which are delimited with '<' and '>'.

The following table (ref [Table 4]) gives the available commands:

Cmd	Answer	Meaning
'i'	<i><module id><ok>\n\r	Return module id name (inria ra – ws 001,…)
'a'	<a><a0><a1><a2><a3><ok>\n\r	Analog scan (channels 0,1,2,3) (value 0…1024)
'A'	<A><a4><a5><a6><a7><ok>\n\r	Analog scan (channels 4,5,6,7) (value 0…1024)
'l'	<l><a0>…<a3>…<a0>…<a3><ok>\n\r	Analog scan (channels 0,1,2,3) x 100
'L'	<L><a3>…<a7><a3>…<a7><ok>\n\r	Analog scan (channels 4,5,6,7) x 100
'f'	<f><ok>\n\r	Request synchronous analog scans (20 ms)
	<[><a0>..<a7><ok>\n\r	Sending synchronous analog scans
'F'	<F><ok>\n\r	Stop sending synchronous analog scans
'd'	<d><status><ok>\n\r	Request for digital input status (0,1)
'o'	<o><ok>\n\r	Turn on (1) digital output
'O'	<O><ok>\n\r	Turn off (0) digital output
'w'	<w><value><ok>\n\r	Request for power off watchdog time out value (second)
'W'	<W><current state><ok>\n\r	Enabled(1)/Disabled(0) auto power off watchdog (toggle flag)
's'	<s><ok>\n\r	Switch off module
'b'	<b><ok>\n\r	Bluetooth AT command configuration for stop mode
't'	<t><time><ok>\n\r	Current local time 0…65535 (1 second = 1024)
'T'	<T><current state><ok>\n\r	<local time>(1) replaces <ok>(0) within messages (toggle flag)
'v'	<v><version><ok>\n\r	Return monitor version
'm'	<m><…><ok>\n\r	Command list menu
Other	<.><…><ok>\n\r	Command list (should not be used)

Table 4: commands tags

General comments about monitor are:

- <error> may replace <ok> or <local time> when error occurs on micro controller module program.
- During the analog scan stream mode (‘f’, ‘F’), the timer frequency is changed from 4 Hertz to 122 Hertz. Therefore checking local time before-during-after the analog scan stream mode may induce time discontinuities.
- Except for the stream mode, the communication is initiated by host computer. So a very simple protocol based on “send a command, wait for answer” may be used. The stream mode makes the host communication state machine a bit harder. Multi-threaded processes might be programmed for avoiding conflicting returned messages.

When the watchdog is enabled (default mode), the module turns off its power if no command is received during a programmed period.
In details, a timer interrupt handler decrements a counter. When an incoming command occurs; the counter is updated with a predefined value. When the counter reaches zero, the program turns off a digital output, thus cutting the power supply.
Remark: The power supply will be turned off even if a client is connected. Criteria used deals only with incoming commands.

3.3 Off-the-shelf devices description

3.3.1 Sensor module: MEMSIC mxr2999m

The MXR2999 (ref [Figure 4] and [11]) is a low cost, dual axis accelerometer fabricated on a standard, submicron CMOS process. The MXR2999 measures acceleration with a full-scale range of ±1 g and a sensitivity of 1000mV/g at 25°C. It can measure both dynamic acceleration (e.g. vibration) and static acceleration (e.g. gravity). The MXR2999 design is based on heat convection and requires no solid proof mass.

The MXR2999 provides a ratio metric analog output that is proportional to 50% of the supply voltage at zero g acceleration. The typical noise floor is 0.2 $mg/\sqrt{Hz}$ allowing signals below 1 milli-g to be resolved at 1 Hz bandwidth.

The inclination measurements (X and Y) are given for +-60 degrees range and at 30 Hertz frequency update.

The MXEB-002_L (ref [Figure 4] and [12]) is a 26 mm x 19 mm board. It is developed by MEMSIC to make easier the integration of the MXR2999 sensor chip. The MXEB-002_L pin out is reported in table (ref [Table 5]).

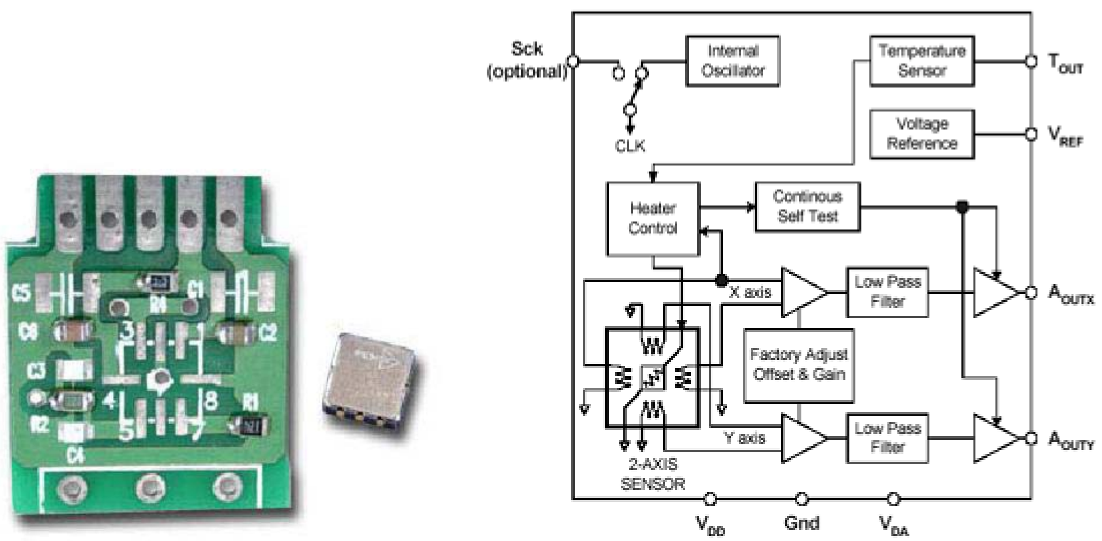


Figure 4: MXEB-002 L Basic Board (with MEMSIC IC on the side) &
MXR2999GL/ML Functional Block Diagram

Pin Number	Pin Name	Description
1	VSS	ground
2	VCC(5 VOLTS)	4mA
3 (component)	VREF OUT	2.5 Volts
3 (solder)	TOUT (temperature)	Analog 0-5 Volts
4	YOUT	Analog 0-5 Volts
5	XOUT	Analog 0-5 Volts

Table 5: MXEB-002 L pin out

Please refer to MEMSIC “MXR2999GL/ML datasheet” (ref [10]) and “Evaluation Board MXEB-2” (ref [11]) for a complete hardware description.

How to measure the angles?

We consider here using the sensor as an inclination sensor (not an accelerometer sensor).
Please refer to the application note “Inclination Sensing with Thermal Accelerometers” (ref [13]) for a complete description.

XOUT and YOUT are acceleration ratio metric outputs.
The related formulas are the following: XOUT = gravity . sin(ANGLE_X) and YOUT = gravity . sin(ANGLE_Y).

The formula calculation is performed on the Host PC side.

On the wireless sensor module side, two offsets are applied on analog scan values. These offsets represent angle offsets (X and Y direction) between the sensor itself and the box container.

The offsets calculation protocol is as follows:

1. The offsets are set to zero.
2. The wireless sensor device relays on a horizontal table. An analog scan is performed.
3. The analog scan results are written in the embedded code, as desired offsets.

In fact we suppose that X and Y directions are parallel with the box main directions. This is achieved by gluing sensor card below top panel box with accuracy. This method is consistent with sensor precision which is more or less 0.5 degree arc.

For higher precision, another registration process must be done. We have to know how values obtain in the sensor coordinate system can be represented in the box coordinate system. This is achieved by finding a 2x2 matrix. The registration protocol could be as follows: (1) we dump X and Y sensors outputs when moving the box around the box X axis, then (2) we dump X and Y sensors outputs when moving box around the box Y axis, then (3) we calculate by a mean square algorithm the sensor axes represented in the box coordinate system.

3.3.2 Micro controller H8/3664F

Hardware

The Renesas (formerly Hitachi) H8/3664F is a 16 bits micro controller.

The table (ref [Table 6]) and the figure (ref [Figure 5]) give the device overview:

• 5V operation	• 2 x 8-bit timer
• 16MHz H8/300H 16-bit CPU core	• 1 x watchdog timer
• 32 kHz sub-system oscillator	• 1 x USART
• Wide range of low power operating modes	• 1 x I²C
• 8 - 32kBytes ROM, 512 - 1kBytes SRAM	• 8 channels 10-bit A/D
• 32kBytes single supply Flash, 2kByte SRAM (Flash version only)	• 37 I/O pins (8 x 20mA ports)
• 1 x 16-bit timer with 4 capture/compare	• 11 External Interrupts
	• On-chip debug port

Table 6: H8/3664F main features

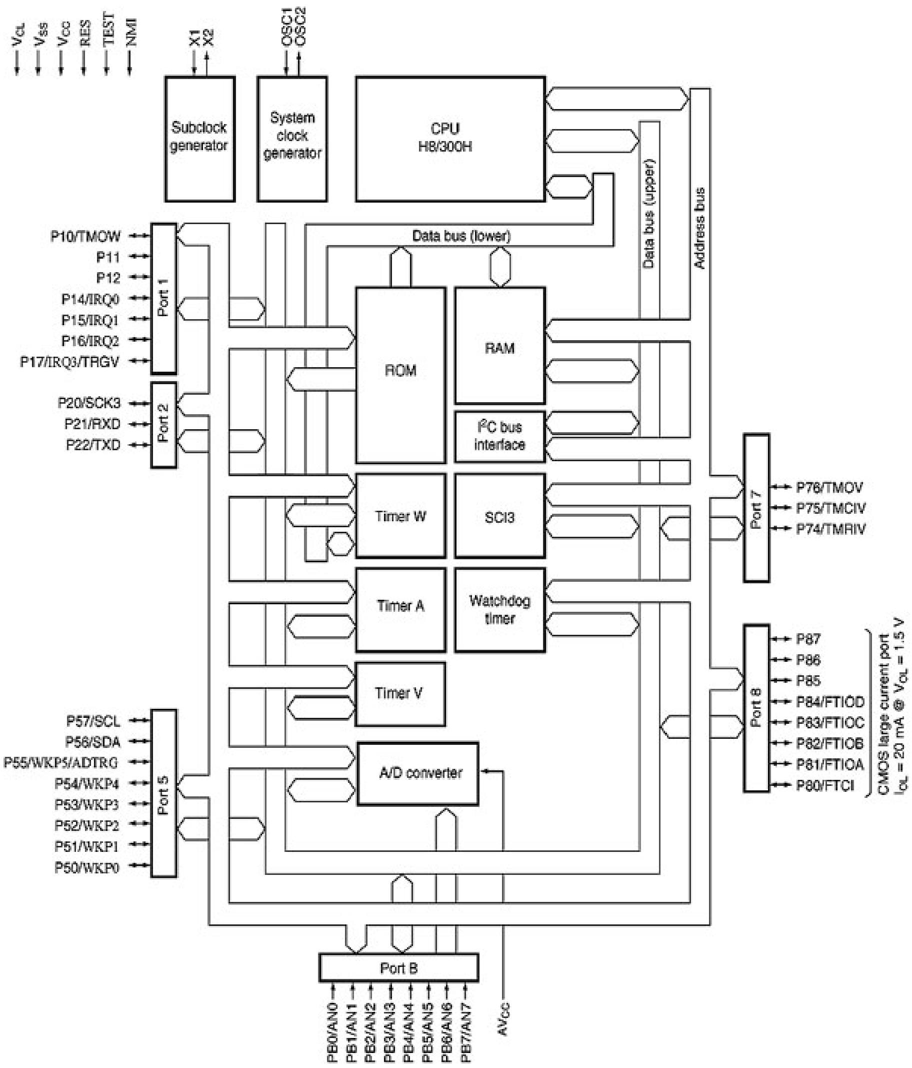


Figure 5: H8/3664F general architecture

The **AE-3664FP** board (ref [Figure 6]) is a 40mm x 27mm module, based on H8/3664F Renesas micro controller.

Figure 6: AE-3664FP module

H8 tiny I/O board (TERA2) is a development board for programming AE-3664FP module.

The AE-3664FP IO signals are reported to two (CN1 & CN2) 26 pins (dual 2.54 millimeters pitch) connectors (ref [Table 7]).

CN1	Name	3664 pin out	CN2	Name	3664 pin out
1	VSS	9	1	VSS	9
2	VSS	9	2	VSS	9
3	PB4/AN4	55	3	P56/SDA	26
4	PB5/AN5	56	4	P57/SCL	27
5	PB6/AN6	57	5	P74/TMRIV	28
6	PB7/AN7	58	6	P75/TMCIV	29
7	PB3/AN3	59	7	P76/TMOV	30
8	PB2/AN2	60	8	NMI(JP3)	35
9	PB1/AN1	61	9	P80/FTCI	36
10	PB0/AN0	62	10	P81/FTIOA	37
11	VC1	6	11	P82/FTIOB	38
12	RESET	7	12	P83/FTIOC	39
13	TEST	8	13	P84/FTIOD	40
14	P50/WKP0	13	14	P85(JP2)	41
15	P51/WKP1	14	15	P86	42
16	P52/WKP2	19	16	P87	43
17	P53/WKP3	20	17	P20/SCK3	44
18	P54/WKP4	21	18	P21/RXD(uart)	45
19	P55/WKP5/ADTRG	22	19	P22/TXD(uart)	46
20	P10	23	20	P14/IOQ0	51
21	P11	24	21	P15/IOQ1	52
22	P12	25	22	P16/IOQ2	53
23	VCC (5 Volts)	12	23	P17/IOQ3/TRGV	54
24	PS-IN		24	RXD (RS232)	
25	VSS	9	25	TXD (RS232)	
26	VSS	9	26	VSS	9

Table 7: AE-3664FP module pin out

Please refer to Renesas "Hitachi Single-Chip Microcomputer H8/3664F Hardware Manual" (ref [14]) for a complete hardware description.

Software development tools

HEW 2.2 (High-performance Embedded Workshop) from Renesas is a flexible code development and debugging environment for applications target at Renesas microcontrollers. It provides an up-to-date "look and feel" with all of the features you would expect from a modern development environment.

The Flash Development Toolkit (FDT 2.2) is used for flashing programs on the micro controller. HEW GUI allows FDT access.

KPIT GNU compiler/linker: KPIT Cummins proposes an installable GNUH8 cross-compiler tool chain based on freely available source code from the Free Software Foundation (www.gnu.org) to help those who wish to build embedded system applications for Hitachi's H8 series of targets. GNUH8 is fully integrated inside HEW.

3.3.3 Bluetooth module: ConnectBlue OEMSPA13i

The OEMSPA13i (ref [Figure 7] and [16]) (cB-0701-01 technical module id) is a small size Bluetooth module based on the Infineon (former Ericsson Microelectronics) PBM 990 80 baseband controller and the PBA 313 05 (0dBm) radio.
The PBM 990 80 has on chip SRAM and FLASH stacked in the same package.

The main features are: short range (0 dBm), logic-level UART and RS232, internal antenna, 3.3-6 VDC power supply.

The module state modes are either the "Configuration mode", or the "Data mode".
The "Configuration mode" is used for configuring the Bluetooth parameters. This configuration may be achieved using AT commands (ref [17]) or ECI protocol (ref [16]). A Microsoft Windows wizard based on AT commands allows configuring part of the Bluetooth parameters. Other parameters must be modified using AT Commands or ECI protocols directly.
The "Data mode" is a totally transparent serial channel mode. Each character sent or received goes through the Bluetooth module.

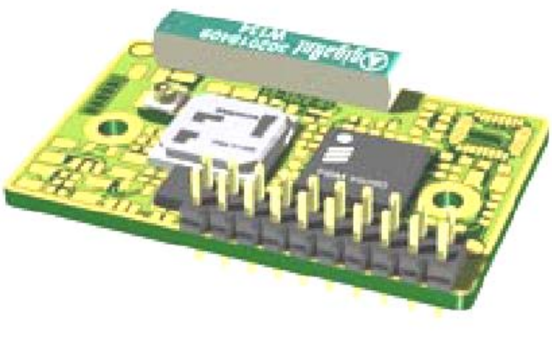


Figure 7: OEMSPA13i ConnectBlue Bluetooth module

All OEMSPA13i IO pins (except RESET) are located on a dual 2 millimeters connector (ref [Table 8]). Hardware RESET is located on J3 connector (pad on PCB).

J1 Pin Number	Pin Name	Signal Name	Signal Level	Type
1,2	VSS	GROUND	GROUND	
3,4	VCC	POWER	3.3V-6V	
5	RS232-CTS	CLEAR TO SEND	RS232	INPUT
6	RS232-TXD	TRANSMIT DATA	RS232	OUTPUT
7	RS232-RTS	REQUEST TO SEND	RS232	OUTPUT
8	RS232-RXD	RECEIVE DATA	RS232	INPUT
9	RS232-DTR	DATA TERMINAL READY	RS232	OUTPUT
10	RS232-DSR	DATA SET READY	RS232	INPUT
11	RED/MODE	RED LED/SERIAL MODE	CMOS	IN/OUT
12	SWITCH-0	FUNCTION SWITCH	CMOS	INPUT
13	GREEN/SWITCH-1	GREEN LED/RESTORE	CMOS	IN/OUT
14	BLUE	BLUE LED	CMOS	OUTPUT
15	UART-CTS	CLEAR TO SEND	CMOS	INPUT
16	UART-TXD	TRANSMIT DATA	CMOS	OUTPUT
17	UART-RTS	REQUEST TO SEND	CMOS	OUTPUT
18	UART-RXD	RECEIVE DATA	CMOS	INPUT
19	UART-DTR	DATA TERMINAL READY	CMOS	OUTPUT
20	UART-DSR	DATA SET READY	CMOS	INPUT
J3 Pin Number				
19	RESET	HARDWARE RESET	CMOS	INPUT

Table 8: Bluetooth module pin out

Please refer to ConnectBlue “OEM Serial Port Adapter: Electrical & Mechanical Datasheet” (ref [15]) for a complete hardware description.

An OEM Adapter (ref [Figure 8]) allows configuring the OEMSPA13i Bluetooth module. This board is actually a mother board including a DB9 RS232 connector, an external power supply, a RGB led on board for status purpose. The mother board is connected to PC via a RS232 cable. A Windows wizard program is used for modifying and saving parameters over the RS232 channel.

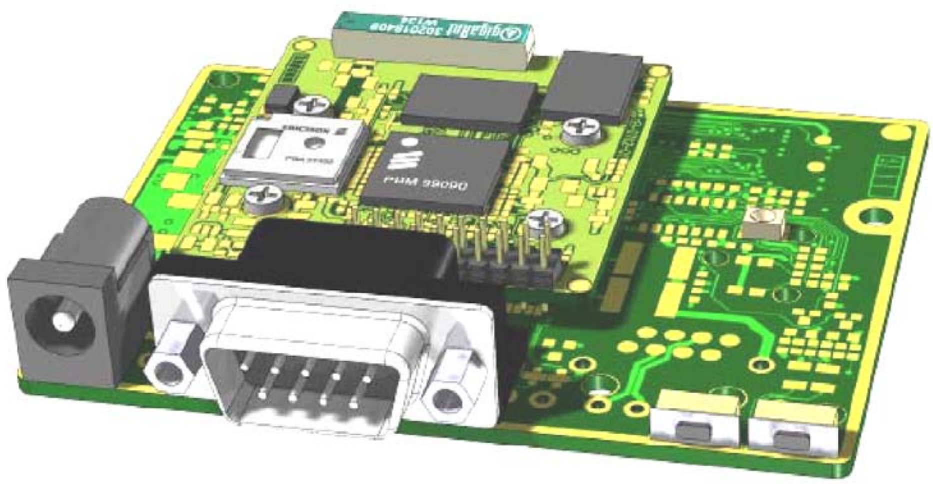


Figure 8: OEM adapter for the OEMSPA13i Bluetooth module

Example of parameters saved on the Bluetooth module:

- Act as server
- Security disabled
- Accepting pairing
- Serial configuration is 19200 bauds, 8 bits, 1 bit stop, no flow control

- Device name: "inria ra - ws 001"
- "Always connected" feature disabled

Software layers

AT commands are a set of ASCII commands recognized by the Bluetooth module. An escape sequence sent over serial line to Bluetooth module changes its internal mode. It switches from "data mode" to "AT commands mode". Then the host configures the Bluetooth module using AT commands. "Data mode" is switched back using another specific AT command.

The ECI protocol (ref [Figure 9]) is delivered as a C library. It runs on the host with an interface towards all the Bluetooth functionality embedded in the Serial Port Adapter. The ECI protocol is a lightweight protocol. It provides access to the profiles GAP, SPP, DUN, and LAN as well as some other miscellaneous functions. It supports true multi-point. ECI protocol is delivered with source code, and so can be compiled on different target.

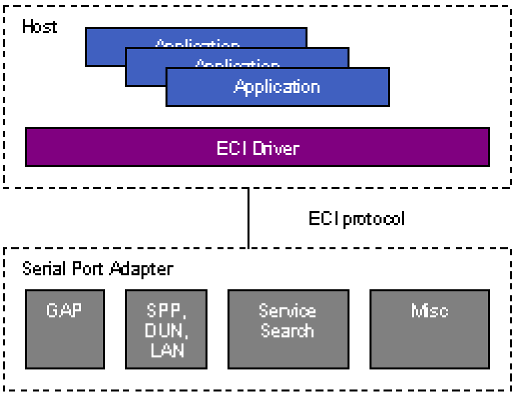


Figure 9: ECI protocol architecture

3.4 Energy

The original idea was to design an "always on" wireless sensor. In other words, the wireless sensor is powered when battery is connected, and stops working when battery is discharged.

3.4.1 Power consumption calculation based on datasheets

The H8/3664F micro controller needs a 5Volts/50mA power supply. A 7805 (5Volts/150mA) regulator is located on this module. Either a regulated 5Volts/50mA or 7-20Volts is used for powering the module. When 7-20Volts power supply is used, the regulated 5Volts may be used as output for supplying other modules.

The OEMSPA13i Bluetooth module needs a 3-6Volts/100mA power supply. Intra module voltage is 3.0 Volts.

The MXR2999 sensor module needs a 5Volts/4mA power supply.

At the end, the power consumption calculation gives 5 Volts / 154 milli-Amperes for the three off-the-shelf modules. The on board 7805 is not strong enough for supplying all modules. For the very first experiment, we decided to add a 5 Volts regulator (7805) on the interface module.
In fact, experiments show us that power consumption never get over 90 milli-Amperes. So we give up using an additional regulator component.

3.4.2 Power saved mode

The micro controller and the Bluetooth module have such a power saved mode. The strategy for getting this mode is presented in the "software" section.
The sensor module does not have any power saved mode, but the power consumption is low (less than 4 milli-Amperes).

The power consumption results are reported in “Experiments & Results” section. Anyway, the minimum current consumption is obtained when both micro controller and Bluetooth module are in power saved mode. The result value is 18 milli-Amperes.
Considering a 700 milli-Amperes hour battery, a power supply failure could occur after about 36 hours. Of course, it is reduced when using the device in transmitting mode for a while.
This is clearly a limitation for using the wireless sensor.

3.4.3 Auto power off

We decide to implement an ultimate power management procedure; the power off sensor feature.
This is a two steps procedure. First of all, a push button is added for powering on module. Then a software watchdog implements an “auto power off” mode.
The final sensor device behavior is as follows:

- A push button (pushed then released phase) is the only way to switch on sensor device,
- The micro controller switch off sensor module when not used for a while,
- A software command generated by host allows switching off module.

From the software point of view, two asynchronous processes take place, and one semaphore memory cell is used. Semaphore cell is either TRUE or FALSE.
The first process is a watchdog function activated every xx minutes (for instance 10 minutes). The function core does a test on semaphore. If the semaphore is TRUE, then a digital output state is changed, thus cutting power supply, else the semaphore is changed to TRUE state.
The second process consists on changing the semaphore to FALSE when an incoming command occurs. Therefore, module will switch off 10 minutes after last command occurs.

3.4.4 Battery specification

The prototype is energized by a primary battery.
A PP3 (rectangular) format is used because of the following features:

- It is a commonly used format.
- The voltage is 9 Volts. It is compliant with 7-12 Volts range for the regulator located on the micro controller module.
- The compactness (25mm x 17mm x 46mm) is better than the AA or AAA cylindrical formats, especially for a cubic box container.

3.5 Interface module

3.5.1 Why do we need a home-made interface module?

In fact, it is obvious that connecting modules coming from different module makers requires an interface module.

The reasons are as follows:

- Due to incompatible electric signals, rail-to-rail connections can not always be achieved. For instance, a 3.3 Volts CMOS Bluetooth output should not be directly connected to a 5 Volts CMOS H8/3664 input. In this case, circuitry must be added.
- The off-the-shelf modules do not have same dimensions, and input/output connectors do not have the same shape. So modules can not be stack up directly.
- Some features, not supported on the modules, can be added. Using LEDs for visualizing the Bluetooth module state is an example.
- From the mechanical point of view, an interface module provides a reference for fixing other modules, thus avoiding moving parts inside the container.

At the end, the interface module implements the following features:

- A mechanical interface for screwing Bluetooth module (two 2.5 millimeters diameter through holes).

- A 7x1 connector for the Bluetooth signals: two pins for the Bluetooth module power supply, three pins connector for the Bluetooth module LEDs output, and two pins for the RS232 communication.
- An interface to the micro controller module: two (2.54 millimeters pad, two rows, 13 columns).
- A power stage: two pins connector for the battery (+ and -), a static relay for the software control of the power, two resistors coming with the static relay.
- A connector for the dual axis sensor: Power, Ground, Xout and Yout signals.
- Four LEDs coming with four resistors. Three of them are dedicated to the Bluetooth module LEDs. The fourth LED is used as a micro controller status display (Diagnostic LED). A jumper allows powering or not the Bluetooth LEDs. This jumper may be installed for debug purpose. Removing the jumper provides energy saving.
- A 74HCT08 (4 AND logic chip) for adapting the 3 Volts CMOS outputs (Bluetooth side) to the 5 Volts CMOS inputs (micro controller side).
- Two extension connectors for the future uses: analog inputs, digital inputs and outputs, power and ground.

3.5.2 Interface module design

The next figure (ref [Figure 10]) represents the interface module schematic. The extensions mentioned in the above section are not represented. Pin out is given except for VCC and GND signals.

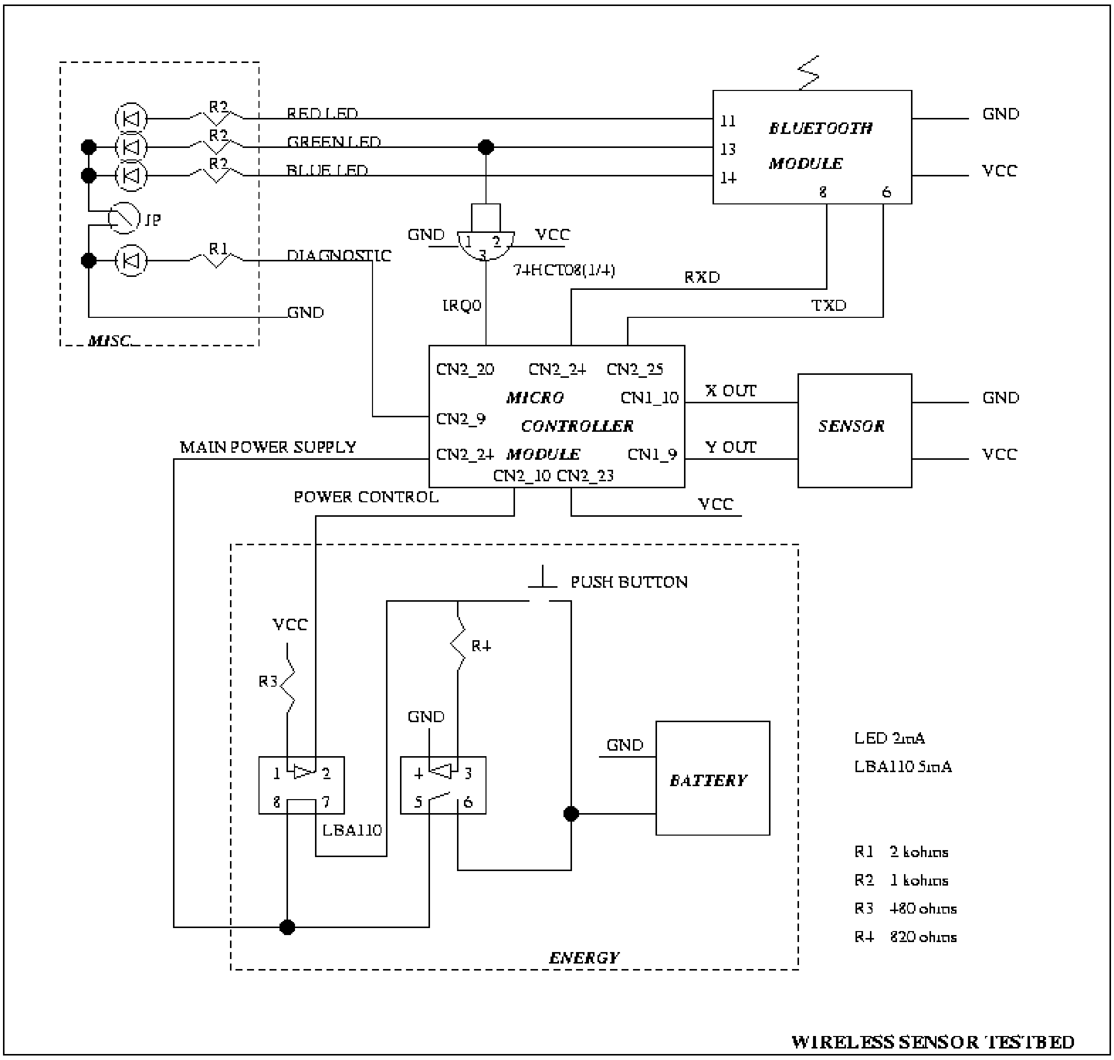


Figure 10: schematic layout

The following figure (ref [Figure 11]) represents the interface module PCB top view.

Remarks about the PCB:

- LED_DIAG (diagnostic led), PUSH_BUTTON, and Sensor device are fixed to front end of containing box.
- The Bluetooth module is located behind the interface board, and is screwed on the interface board using two through holes.

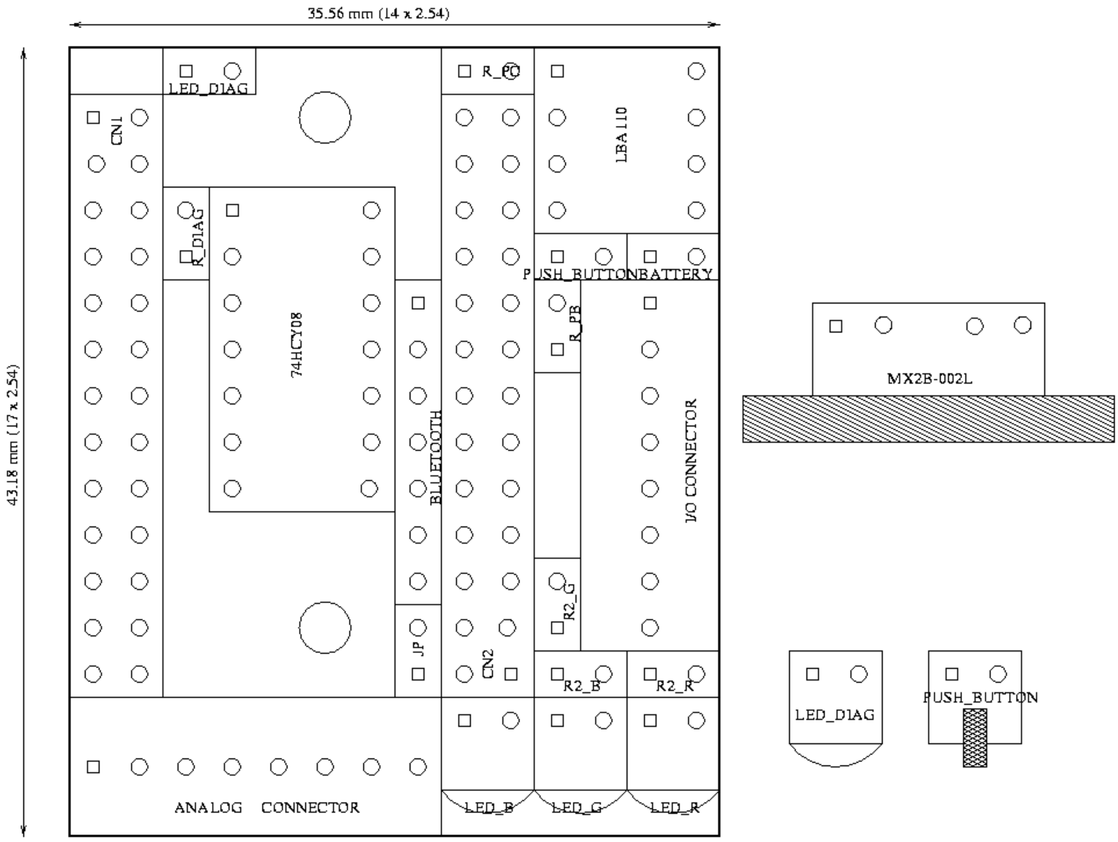


Figure 11: PCB layout

The full module schematic is given in the document annex (ref [annex 1]), and the net list is reported in the document annex (ref [annex 2]).

3.6 Container & Packaging

We purchase a 55mm x 55mm x 42mm box made in ABS plastic.

A rectangular (50mm x 36mm) plate is inserted between the battery and the electronics parts as insulator material.

Two through holes are machined on the top panel box. One is a 2 millimeters diameter for diagnostic LED. The other one is a 7.5 millimeters diameter for the push button. Both components are glued.

The sensor module is glued on back side of the top panel box. The other modules take place as shown in next figure (ref [Figure 12]).

Note: the battery consists on metallic material, which could affect the radio frequency (RF) transmission. Therefore, the Bluetooth module antenna is located at the opposite side from the battery, thus limiting RF problems.

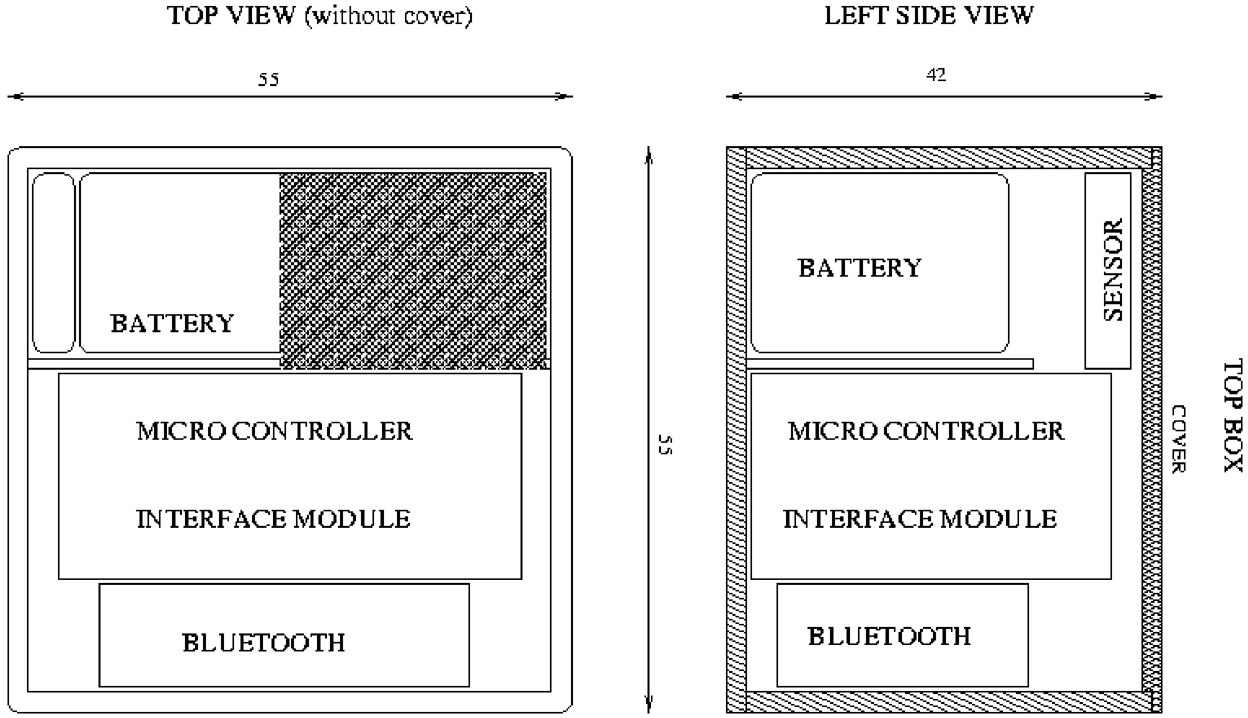


Figure 12: module packaging

The modules mounting instructions are given in the document annex (ref [3]).

The sterilization process is quite easy with a hermetic plastic box. Only the push button and the diagnostic led could induce problems. Fortunately, theses components are commonly used in medical applications. For the prototype we did not take into account sterilization process, and we choose standard components.

3.7 Embedded programs

Before talking about the final version of embedded program, let describe briefly the test procedures we performed. These programs implement incrementally features.

Hello World – learning development tools

From the hardware point of view, the micro controller is plugged on the development board (H8 Tiny Board TERA2). A RS232 cable connects development board to a PC running Microsoft Windows XP.
The developed program uses the H8/3664F RS232 hardware resource. The complete software development toolkit was validated: compiler, linker, flash toolkit. On the PC side, we use a HyperTerminal application. We validate also the serial cable.

Timer & Analog reading – using micro controller features

From the hardware point of view, the micro controller is plugged on the development board (H8 Tiny Board TERA2). A RS232 cable connects development board to a PC running Microsoft Windows XP. MEMSIC dual axis inclination sensor is wired on the development board (H8 Tiny Board TERA2) using the mirrored micro controller connectors.
With this program, we master two of the major micro controller features: timer and analog converter.

Bluetooth module test – configuring and testing module

We test here the Bluetooth module. The module is configured using the mother board and the wizard Microsoft Windows program described above. After the configuration ends, we run a HyperTerminal program for connecting the module over the COM1 channel.
On the second hand we use a Bluetooth dongle connected on the same PC over an USB port. We use a wizard application delivered with dongle for discovering Bluetooth entities present in the vicinity. Hopefully, the module is recognized. The wizard allows pairing a virtual port (COM5 in our case) with the entity discovered. A second HyperTerminal is opened over COM5.
First we select the first HyperTerminal window. Any character typed in it, appears on the second HyperTerminal window. Then we select the second HyperTerminal window. Any character typed, appears on the first HyperTerminal window.
We get a transparent RS232 channel over a Bluetooth link.

Bluetooth+H8/3664F+MEMSIC – first test program using complete hardware configuration

We use the program written for "Timer & Analog reading" case. MEMSIC, the micro controller and the Bluetooth module are connected via a home-made interface module. On the other hand we run a HyperTerminal on the PC connected over COM5 (Bluetooth dongle). Program runs successfully.
For the very first time we are able to measure the whole module power consumption. The measurements are reported in the "Experiments and results" section.

Bluetooth+H8/3664F+MEMSIC – Console program, access time measurement

We use the program written for "Timer & Analog reading" case. MEMSIC, the micro controller and the Bluetooth module are connected via the home-made interface module. On the other hand we run a Console program developed with Visual C++. Program sequence is: (1) opening a COM5 communication channel, then (2) requesting a hundred analog scans while measuring the elapsed time, then (3) closing the communication link, then (4) displaying on screen the elapsed time (total and average) measured.
In fact there are two methods for requesting analog scans. The first method consists on doing a loop which does a bunch of analog scan requests. The second method consists on doing only one request for a bunch of analog scans. This makes a big difference in terms of timing, because of the transmission delays.
All measurements are reported in the "Experiments and results" section.

Power saved mode – Bluetooth side

The goal of this program part is switching Bluetooth module to a power saved mode in order to reduce the power consumption. It consists on sending over a RS232 channel several AT commands (ref [Table 9]). Within embedded code running on micro controller, this program part takes place just after the micro controller initialization phase.

Added lines	Comments
WaitOneSecond	Wait one second for Bluetooth module to wake up after power on
PutString("///")	Escape sequence for switching to AT commands mode
WaitOneSecond	Wait one second for Bluetooth module to switch mode
PutString("AT*AMPM=3,1")	Configure module for enabling stop mode
WaitForResponse	Wait until module response
PutString("AT*ADDM")	Switch back to data mode
WaitForResponse	Wait until module response
WaitOneSecond	Wait one second for Bluetooth module to switch mode

Table 9: Bluetooth power saved mode commands

The Bluetooth mode is now entering the stop mode (power saved mode) at power on. It leaves the stop mode when a Bluetooth client connection occurs. It goes back to stop mode as soon as client closes its connection. Power consumption was measured. Results are reported in "Experiments and results" section.

Power saved mode – micro controller side

The goal of this program part is switching the micro controller module to a power saved mode in order to reduce the power consumption.
The strategy is as follows:

- A new electronic signal has been connected between the Bluetooth module GREEN LED pin and the micro controller IRQ0 pin. In fact this signal goes through a 74HCT08 for electric compatibility. The Bluetooth outputs are 3 Volts CMOS, the micro controller inputs are 5 Volts CMOS.
- The GREEN LED signal goes to active mode when a connection occurs. It leaves active mode only when the client is disconnected. Note that the BLUE LED behavior is almost the same except that the BLUE LED signal flicks during communication.
- The micro controller goes to standby mode when the function “sleep” is called. It wakes up for different reasons. One of them is an incoming interrupt.
- An interrupt handler attached to an IRQ0 input pin sets (1) or resets (0) a flip-flop semaphore depending on the interrupt edge is rising or falling. The main C function is based on an infinite loop, waiting for an incoming command over RS232. A test on a semaphore state has been added into this loop. The followings actions happen within a session:
    1. At power on, the micro controller does its initialization, then calls the “sleep” function and enter “standby” mode.
    2. When a client connects, the Bluetooth module wakes up and enters the connecting mode. Then GREEN LED signal output goes to high state and the micro controller interrupt rises up due to a rising edge input. This makes the micro controller leaving “standby” mode. And during the next loop program, the “sleep” function will be bypassed, thanks to the semaphore state.
    3. When the client disconnects, the Bluetooth module switches GREEN LED signal output to low state, then it enters “stop” mode. Micro controller interrupt rises up due to a falling edge input. The semaphore changes its state, and during the next loop program, the “sleep” function will be called. Thus makes entering the micro controller to “standby” mode.

Final program

The final program consists on a C program with the following parts:

- Timer interrupt handler: it manages local time, watchdog, and synchronous analog scans requests.
- External interrupt handler: It is connected to the Bluetooth module for going or leaving the power saved mode (micro controller “standby” mode).
- UART interrupt handler: It receives incoming characters.
- Main program: First of all, it initializes the internal registers. Then an infinite loop waits for an incoming character (a command). The command is treated, and an answer is sent within the loop.

The C code is about 1000 lines. The file produced is 7 kilo bytes Motorola S-Format type, and 2.75 kilo bytes on the micro controller flash.

3.8 Program running on host

A test program running on Microsoft Windows station provides an access to all features available on the module.

This test program is developed using Microsoft Visual C environment on Microsoft Windows XP. Because this is a Console type program, it can be ported easily on other operating systems like Linux.

The program accesses a Bluetooth dongle connected over an USB port. CreateFile/CloseHandle/ReadFile/WriteFile basic Microsoft Windows functions are used for the USB communication. The communication channel number (COMx) is given by a Bluetooth utility program. This utility program provides several Bluetooth features (discovery mode, connecting device) which are not implemented inside the program. The utility program provides also Microsoft Windows systems features like mounting Bluetooth connection channel on a Microsoft Windows COM channel.

The program consists on two sub-programs. An argument given when program is called activates one of the sub-programs.

The first sub-program is used for measuring communication time. It is described within the “embedded programs” section.

The second sub-program consists on a menu providing functionalities described within the "How does the sensor device work?" section.
The program state machine runs as follows:

1. Wait for a command (a character) on standard input.
2. Send the character over communication channel.
3. Wait for an answer.
4. Split incoming message and fill a semantic structure containing the "returned command", the "module status", and eventually the "returned value(s)" fields.
5. Analyze the common fields.
6. Analyze the fields depending on the command sent.
7. Back to state 1.

Stream mode is an exception for the above state machine. Messages due to stream mode (incoming messages with no associated commands) are treated within state 6 (Analyze fields depending on command sent).

In details, when a "stream start" command is sent, the associated answer is received, then a predefined number of "stream messages" are received, then a "stream stop" command is sent (automatically). It supposed that no commands are sent within this program phase.

A more elaborate protocol, including multi-threaded processes, would allow managing the whole protocol (including stream mode) in a more efficient way.

3.9 Experiments and Results

The following picture (ref [Figure 13]) shows how experiments take place. A power supply replaces the battery for the development phase. Current measurements are done on it. Voltages are measured with a multimeter. A breadboard provides a smart tool for testing new hardware component. The prototype parts relay on the table. The PC and the Bluetooth dongle are not represented.

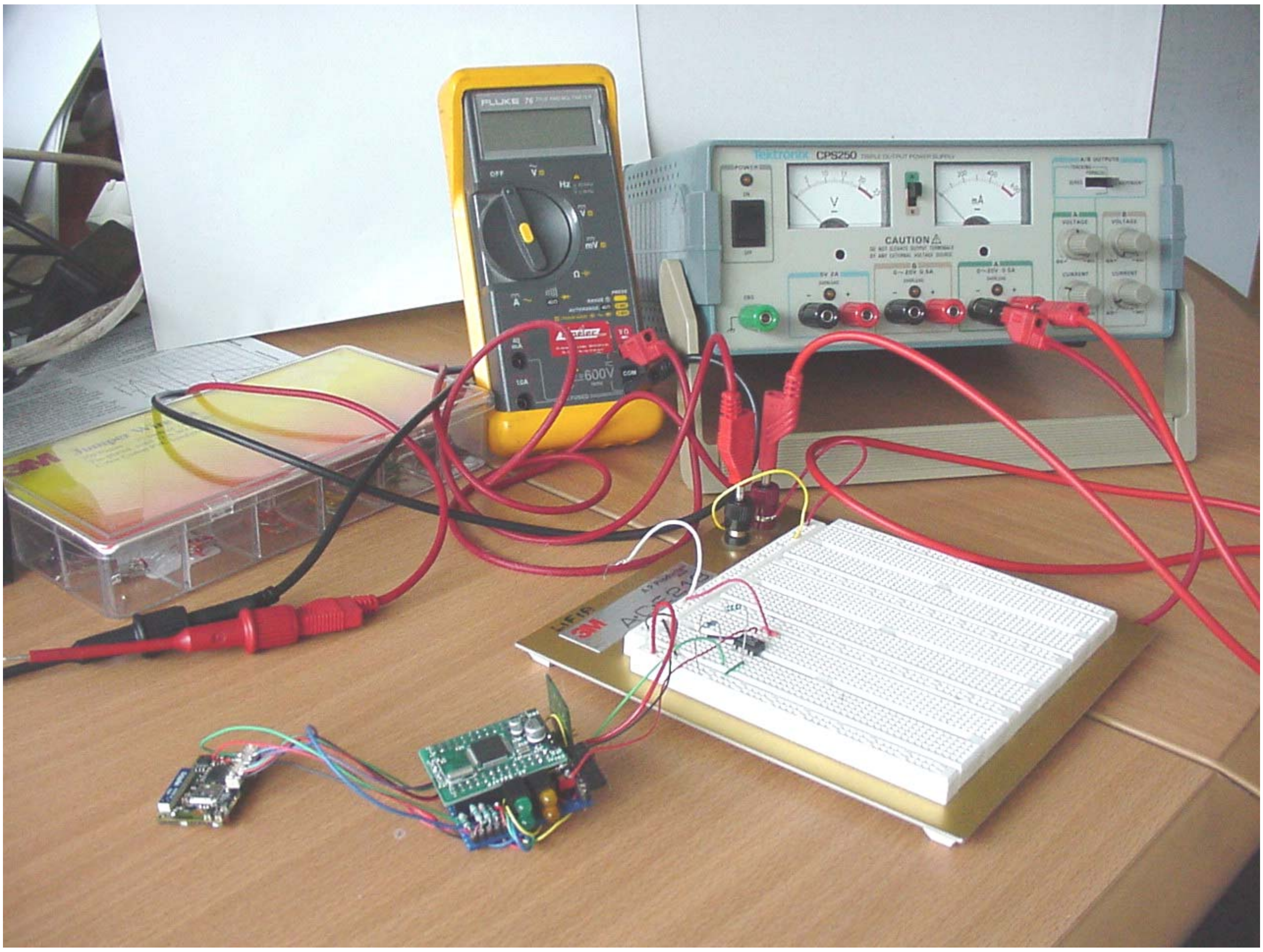


Figure 13: experimental table

The first objective was to contain the whole system in a "small" box. We succeed with a 55mm x 55mm x 42mm box. The total weight is 120 grams. The weight distribution is reported in the next table (ref [Table 10]).

Name	**Weight (gram)**
Battery	48
Micro controller module	9
Bluetooth module	5
Box	35
Sensor module	3
Interface module	15
Misc. (wires, insulator,…)	5
Total	120

Table 10: weights

The second objective concerned the power consumption. The following table (ref [Table 11]) reports the power supply measurements performed in the different device configurations. The lines match the incrementally building steps. First the micro controller is alone, then the sensor is added, then the Bluetooth module is connected. At the end, the whole system including the LEDs and the "auto power off" feature is tested. At the same time we tested the power saved modes. It is reported as "Idle" or "Stand by" for the micro controller, and "Not Connected", "Connected, or "Stop Mode" for the Bluetooth module.

Hardware	Status	Power (9 Volts) (mA)
H8/3664F	Idle	28
H8/3664F+MEMSIC	Idle	30
H8/3664F+MEMSIC+BT	Idle, Not connected	60
H8/3664F+MEMSIC+BT	Idle, Connected	80
H8/3664F+MEMSIC+BT	Idle, Stop Mode	35
H8/3664F+MEMSIC+BT	Stand by, Stop mode	15
H8/3664F+MEMSIC+BT+LED+POWEROFF	Idle, Connected	90
H8/3664F+MEMSIC+BT+LED+POWEROFF	Stand by, Stop mode	19

Table 11: power consumption in different operating modes

The two last lines show what the nominal power consumption for using the device is. In fact the case “Idle,Connected” with the whole device can be considered during a transmission phase. Indeed, we did not notice any additional current when transmitting. We can suppose that the “Connected Mode” is already a “transmission mode” for the Bluetooth technology. This has to be investigated in the future.
As mentioned within the “Energy” section, an “auto power off” feature was added during the project period. At the end the module should be usable without changing the battery for several weeks. Of course it depends on how long it is connected (highest power consumption configuration).

The transmission measurement was the third objective. During the experiment, a program developed on PC (Visual C++ Console application) runs as follows:

1. Open a USB channel to the Bluetooth dongle; it activates a connection between the PC and the wireless sensor.
2. (Test 1) Run a 100 loops for requesting an analog scan
3. (Test 2) Then send a request for 100 analog scans
4. Close the USB channel.

The analog scan answer message includes the duration for scanning.

The test was done at 19200 bauds and 57600 bauds (micro controller UART configuration). On the PC side, a USB is used, and we do not get any information about baud rate.

The next table (ref [Table 12]) reports the results.

Algorithms	Time (milli-second)
(Test 1) 100 analog requests @19200bauds	32
(Test 1) 100 analog requests @57600bauds	29
(Test 2) one request for 100 analog scan @57600bauds	2.2
Analog scan duration	2

Table 12: communication throughput

As conclusion:

- The baud rate is not a major issue. The transmission is only 10 percent faster when the rate is multiplied by three. 57600 bauds is nominal baud rate for final test bed version.
- The results show that a delay occurs during transmission. Further tests must be done for full understanding the transmission model. We can suppose that the same delay occurs from PC to sensor device and from sensor device to PC. Then the delay calculation gives about 13 milli-seconds.
  This is clearly a limitation for the bandwidth communication.

At the end, two communication protocols (synchronous and asynchronous) are implemented.

The synchronous protocol is as follows:

1. The PC requests for an analog scan.
2. The micro controller receives the command.
3. The micro controller performs an analog scan.
4. The micro controller sends back the analog scan result.
5. The PC receives the analog scan result.

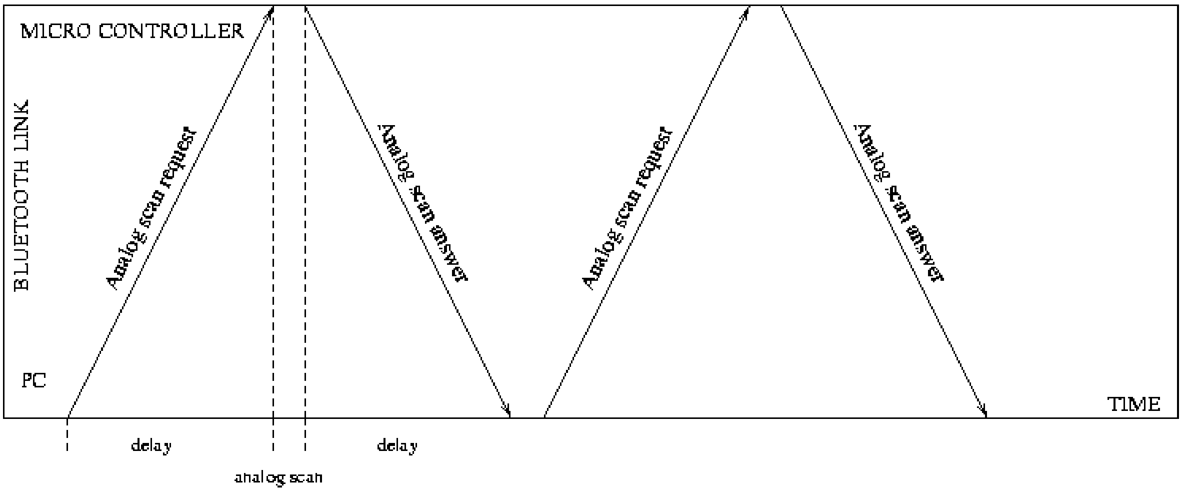
The resulting bandwidth is about 30 Hertz.

The asynchronous protocol is as follows:

1. The PC requests for an analog scans.
2. The micro controller receives the command.
3. The micro controller performs analog scans and sends back the result. Scans are synchronized using the micro controller timer, thus allowing different output frequencies. At the same time, the micro controller carries on receiving incoming commands.
4. The PC receives the incoming analog scan results.
5. When the PC decides to stop, it requests the micro controller for stopping sending analog scan results.

The second protocol uses more micro controller and PC resources, because it needs several processes running in concurrent mode. It has been successfully implemented on the micro controller and the PC.
And, it should be preferred for applications when the bandwidth is critical.

The next two figures (ref [Figure 14]) summarize both protocol timings.



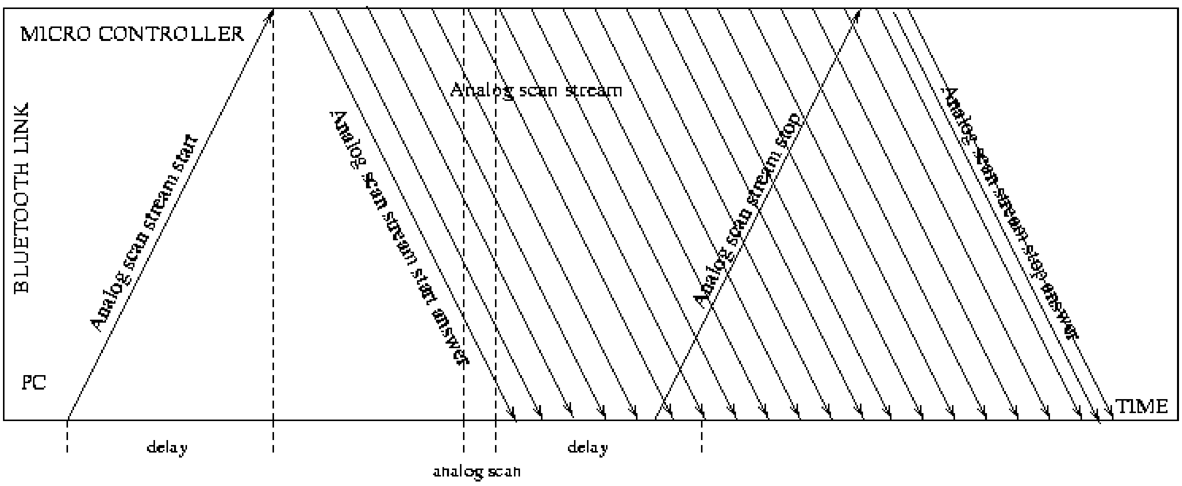


Figure 14: communication protocols (top: synchronous, bottom: asynchronous)

3.10 Prototype limitations

The sensor device main limitation concerns its accuracy.

The sensor sensitivity is 1000 milli-Volts per gravity acceleration (“g”). Because the sensor is used as an inclination sensor, only one “g” is measured, therefore output signal is between -1 Volt and 1 Volt (2 Volts range) for a range of -90 +90 degrees arc.
On the micro controller side, the analog inputs range is 10 bits for a 0-5 Volts input signal.
As conclusion, the measurement resolution is 0.44 degrees arc; 180 deg divided by (2/5) range of 1024 points.
The sensor resolution is better than 1 milli-G as announced in the sensor datasheet. Therefore a better analog converter would increase the final accuracy. This analog converter could be interfaced with the micro controller using the I2C channel link.

Concerning the micro controller module, the limitations are as follows:

- The H8/3664F implements only one UART. This UART is connected to the Bluetooth module. A second UART would have been great for connecting another device. During the development phase the micro controller module was move back and forth between the development board and the interface board. A second UART would have made it easier.
- The second limitation coming from the UART is due to a lack of flow control. Only TXD and RXD signals exist on the micro controller side. The Bluetooth module provides all necessary signals for flow control, but it can not be used here. Therefore, the embedded program never knows if the Bluetooth module receives all the characters sent. Fortunately, the Bluetooth module implements a 128 bytes FIFO (First In First Out) memory for the incoming RS232 characters.
- The ADC is a 10 bits converter. The accuracy may be an issue for other sensor devices than the MEMSIC dual axis inclination sensor we tested. Other sensor dynamic range may exceed 10 bits.
- Another limitation of the ADC concerns the input impedance that is required. The input impedance must be less than 5 kilo-Ohms. In other terms, the sensor device must provide more than 1 milli-Ampere for a 5 Volts supply. MEMS sensor are designed for low power applications. This can be a limitation for connecting rail-to-rail sensor devices. An operational amplifier might be used to overcome this problem.
  The MEMSIC sink current is 0.1 milli-Ampere which should not be enough. Nevertheless we did not notice any problem during tests.

The main Bluetooth module limitation is the transmission delay. Experiments show a 13 milli-seconds delay when transmitting over the Bluetooth channel.

4 Conclusion

We successfully build a wireless sensor. Main features are reported in the document annex (ref [annex 6]).

Our contributions are as follows: (1) we have reviewed existing research or commercial wireless sensor devices; (2) a prototype has been built and successfully tested.

Therefore, research groups have now the opportunity to use the wireless sensor technology within their applications. The tests performed are relevant and the limitations are clearly identified, thus making the integration process clear.

The prototype developed is not a final or a commercial product. The making process was very intuitive. Also components and modules placement is clearly not efficient.

But the final step for getting a usable product is not so high. All modules have been validated. The desired functionalities have been implemented with success. The package must be optimized in order to fit the application. The interface board has to be cleaned up, and a new version might use integrated technology like CMS components.

The wireless sensor cost is about 300 euros. The purchase list is given in the document annex (ref [annex 4]). A 40 euros Bluetooth dongle allows working with the device from any computer. Half of the device price is due to the Bluetooth module. This module includes a useful software stack that provides an easy access to all Bluetooth functionalities. But the cost can be reduced significantly by using cheaper Bluetooth modules and developing the missing software stack. At the end we would get an affordable sensor device.

Further works will concern: (1) getting a “more industrial” wireless sensor device version, then (2) including this new sensor class within applications.

During this project we mastered; a micro controller module based on the H8/3664F RENESAS chip, and a Bluetooth module. Research teams working in virtual reality and in robotics are now interested for using these bricks in their applications.

5 Bibliography

- [1] RT-0243 - L'armoire de commande du robot bipède bip2000
  INRIA Technical Report
  Baille Gérard, Di Giacomo Pascal, Mathieu Hervé, Pissard-Gibollet Roger
  July 2000
- [2] A Minimal Bluetooth-Based Computing and Communication Platform.
  ETH, Technical Note.
  Jan Beutel, Oliver Kasten.
  May 2001
- [3] First experiences with bluetooth in the smart-its distributed sensor network.
  Swiss Federal Institute of Technology
  Oliver Kasten and Marc Langheinrich.
  2001
- [4] A wireless-embedded architecture for System Level Optimization.
  UC Berkeley, Technical Report.
  Jason Hill and David Culler.
  2002
- [5] Wireless Sensor Networks for Habitat Monitoring
  Intel Corporation, University of California at Berkeley, College of the atlantic
  AlanMainwaring, JosephPolastre, RobertSzewczyk, DavidCuller, JohnAnderson
  2002
- [6] Bluetooth Prototype Tag.
  Telenor, R&D Note.
  Tore Syversen
  2001
- [7] Bluetooth Based Wireless Sensor Networks –Implementation Issues and Solutions
  Applied Research Lab, Ericsson Ireland
  Srdjan Krco
  2002
- [8] Sourceless Human Body Motion Capture
  LETI, CEA Grenoble France
  David Fontaine, Dominique David, Yanis Caritu
  2003
- [9] ‘Pen-like’ natural graphic gesture capture disposal, based on a micro-system
  Department of Systems for Information and Health - CEA-Léti Grenoble
  Cédric Joguet , Yanis Caritu , Dominique David
  2003
- [10] Batteries et chargeurs (Technologies, Conseils, Réalisatons)
  Club de robotique de l’E.S.E.O.
  Olivier Cochelin, Emilie Sosna
  2002
- [11] MXR2999GL/ML datasheet
  MEMSIC
  2003
- [12] Evaluation Board MXEB-2
  MEMSIC
  2003
- [13] #AN-00MX-007 Application Note Rev A 05/02
  Inclination Sensing with Thermal Accelerometers
  MEMSIC
  2003
- [14] Hitachi Single-Chip Microcomputer H8/3664F Hardware Manual
  Renesas
  2002
- [15] OEM Serial Port Adapter: 2nd Generation: Electrical & Mechanical Datasheet
  ConnectBlue

2003

- [16] Serial Port Adapter: 2nd Generation: User Manual 1.3
ConnectBlue
2003
- [17] Serial Port Adapter: 2nd Generation: AT Commands
ConnectBlue
2003

6 Related web sites

[web00] http://www.inrialpes.fr/sed
[web01] http://www.robodoc.com
[web02] http://ndigital.com

6.1 Sensor providers

- [web03] http://www.memsic.com
- [web04] http://www.analogdevice.com
- [web05] http://www-leti.cea.fr
- [web06] http://www.memsnet.org

6.2 Micro controller module providers

- [web07] http://www.phytecfrance.com
- [web08] http://www.microtronique.com
- [web09] http://www.eu.renesas.com

6.3 Wireless sensor providers (including Bluetooth & other RF technologies)

- [web10] http://www.oceanasensor.com
- [web11] http://www.sensicast.com
- [web12] http://www.xbow.com/Products/Wireless_Sensor_Networks.htm
- [web13] http://www.millenial.net

6.4 Bluetooth chip or module providers

- [web14] http://www.csr.com
- [web15] http://www.btdesigner.com
- [web16] http://www.codebluecommunications.com
- [web17] http://www.infineon.com
- [web18] http://www.brightcom.com
- [web19] http://eu.st.com/bluetooth
- [web20] http://www.connectblue.se
- [web21] http://www.euridis.com
- [web22] http://www.mitsumi-components.com/Catalog/index_e.html
- [web23] http://www.micropuissance.fr

6.5 Bluetooth stack providers

- [web24] http://bluez.sourceforge.net
- [web25] http://sourceforge.net/projects/openbt
- [web26] http://www.widcomm.com

6.6 Other Bluetooth related companies

- [web27] http://www.smart-its.org
- [web28] http://www.bluetooth.com
- [web29] http://www.nohau.co.uk
- [web30] http://www.palowireless.com

6.7 Other RF solutions

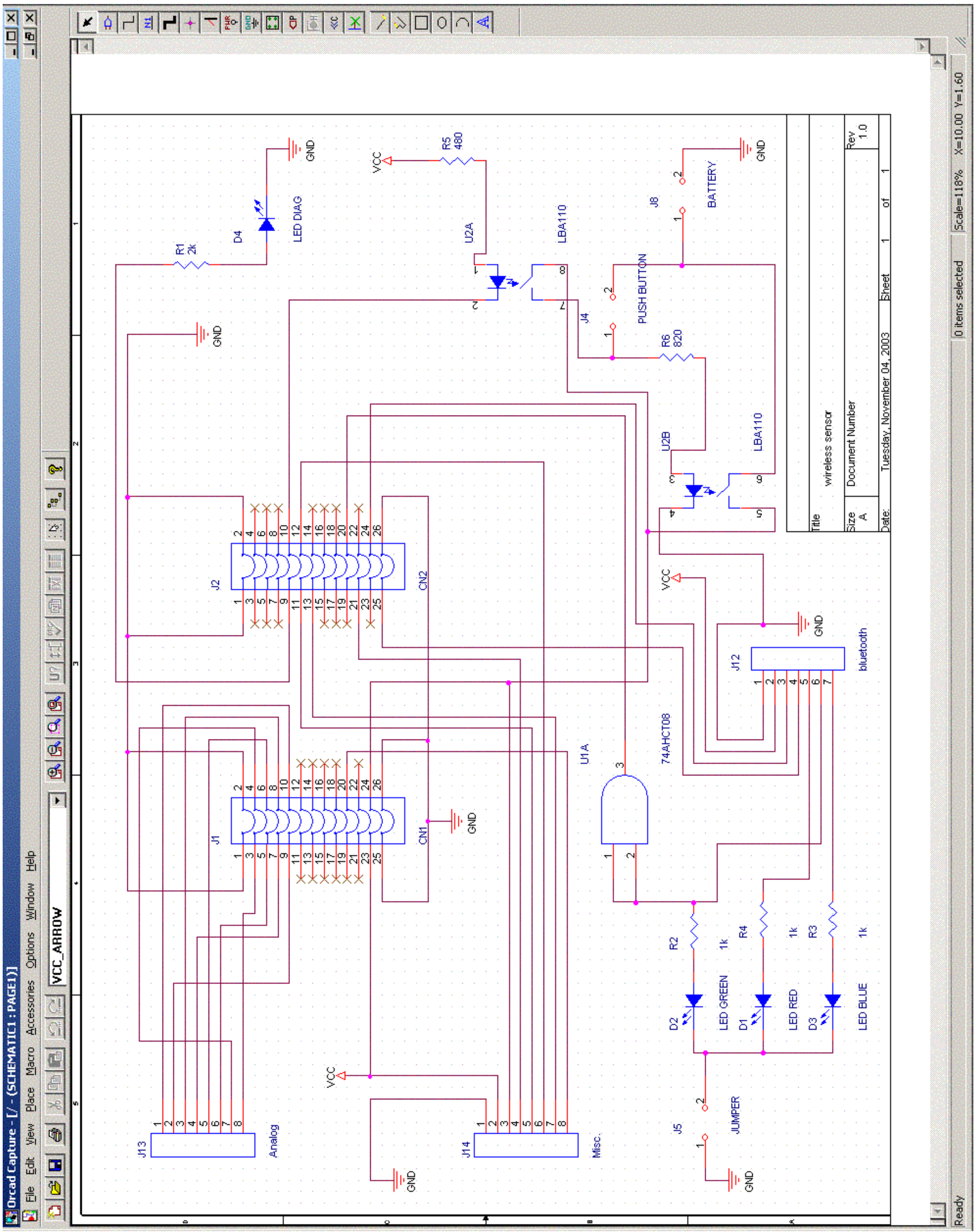
- [web31] http://robotics.eecs.berkeley.edu/~pister/SmartDust/
- [web32] http://webs.cs.berkeley.edu/tos
- [web33] http://www-mtl.mit.edu/~jimg/project_top.html

6.8 Battery related web sites

- [web34] http://www.bktsi.com/powerflite/powerflite.htm
- [web35] http://industrial.panasonic.com/ww/products_e/battery_e/battery_e.html
- [web36] http://www.sanyo-energy-europe.com/
- [web37] http://www.buchmann.ca/faq.asp/

7 Annexes

7.1 Interface module schematic



7.2 Interface Module PCB component and net lists

	Component Name	Abbreviation
1	BATTERY	BAT
2	PUSH_BUTTON	PB
3	LBA110	LBA
4	R4	R4
5	R3	R3
6	LED_R	LR
7	R2_R	R2R
8	LED_G	LG
9	R2_G	R2G
10	LED_B	LB
11	R2_B	R2B
12	LED_DIAG	LD
13	R1	R1
14	SENSOR	S
15	BLUETOOTH	BT
16	74HCT08	HCT
17	MC-CN1	CN1
18	MC-CN2	CN2
19	ANALOG CONNECTOR	ANA
20	JUMPER	JP
21	IO CONNECTOR	IO

Table: component list

Net	Connections
1=GND	BAT(1) LBA(4) LD(2) S(1) BT(1) HCT(7) CN1(1,2,25,26) CN2(1,2,26) JP(2) IO(1)
2=VCC(5Volts)	R3(1) S(2) BT(2) HCT(14) CN1(23) IO(2)
3=BAT+	BAT(2) PB(1) LBA(6)
4	PB(2) R4(1) LBA(7)
5=POWER	LBA(8,5) CN1(24) IO(3)
6	R4(2) LBA(3)
7	LBA(1) R3(2)
8=PWR_CTL	LBA(2) CN2(10)
9=LED_DIAG	CN2(9) R1(1)
10	R1(2) LD(1)
11	JP(1) LR(2) LG(2) LB(2)
12	R2R(2) LR(1)
13	R2G(2) LG(1)
14	R2B(2) LB(1)
15=LED_R	R2R(1) BT(5)
16=LED_G	R2G(1) BT(6) HCT(1,2)
17=LED_B	R2B(1) BT(7)
18=RXD	CN2(24) BT(3) (TXD for the Bluetooth module side)
19=TXD	CN2(25) BT(4) (RXD for the Bluetooth module side)
20=IRQ0	HCT(3) CN2(20)
21=AN0	ANA(1) CN1(10)
22=AN1	ANA(2) CN1(9)
23=AN2	ANA(3) CN1(8)
24=AN3	ANA(4) CN1(7)
25=AN4	ANA(5) CN1(3)
26=AN5	ANA(6) CN1(4)
27=AN6	ANA(7) CN1(5)
28=AN7	ANA(8) CN1(6)
29=IRQ2	IO(4) CN2(21)
30=IO0	IO(5) CN2(11)
31=IO1	IO(6) CN2(12)
32=IO2	IO(7) CN2(13)
33=TMOW	IO(8) CN1(20)

Table: net list

7.3 Mounting instructions

	Part (PCB)	Qt			Part (other)	Qt
	PCB	1			Box	1
U1	DIP socket 14 pts	1			Insulator	1
U2	DIP socket 8 pts	1		PB1	Push button	1
U3 U4	Breakaway header 2x13	2		D4	Led	1
JP1	Breakaway header 1x2 (for shunt)	1		BAT	Battery clip type PP3	1
T1 T2	Terminal blocks 8x2.54	2			2.5mm x 10mm screws	2
D1	Led Red	1			2.5mm nut	2
D2	Led Green	1			3.0mm washer	2
D3	Led Blue (or Yellow)	1		U1	74HCT08	1
R1 R2 R3	Resistor 1 kilo-Ohms	3		U2	LBA110	1
R4	Resistor 2 kilo-Ohms	1			Wire awg24 30mm long	11
R5	Resistor 820 Ohms	1			Wire awg24 70mm long	6
R6	Resistor 460 Ohms	1			Shunt	1
					Bluetooth connector 2x10	1
				BtM	Bluetooth module	1
				SM	Sensor module	1
				MC	Micro controller module	1

Table: components list

To machine:

- The top panel thickness may be reduced for improving push button and led accessibility (ref [Annex 3 - Figure 1]).
- The battery+clip length is 1mm too wide for the box. So the inner box width must be enlarged to fit (a rape tool can be used for removing several board guides) (No figure)
- Drill one through hole on top panel box for the push button (ref [Annex 3 - Figure 1]).
- Drill one through hole on top panel box for the led (ref [Annex 3 - Figure 1]).
- Drill one through hole in PCB for screwing Bluetooth module. (No figure)
- Cut a 50mm x 35mm insulator plate. Cut a 40 mm x 15 mm insulator plate. Glue both plates with a 90 degrees angle. Resulting system must bind the battery inside the box (ref [Annex 3 - Figure 4]).

Mounting instructions for the interface PCB: (ref [Annex 3 – Figure 2]).

- Solder 7 wires (30mm) for connecting the PCB and the Bluetooth connector (2mm, 2x10) (ref [Annex 3 – Figure 2]).
- Solder 2 wires (70mm) for connecting the push button to the PCB (no polarity).
- Solder 2 wires (70mm) for connecting the led to the PCB (caution to polarity).
- Solder 2 wires (70mm) for connecting the battery clip to the PCB (caution to polarity).
- Solder the components (located in table on the left column) on the PCB.

About the sensor:

- Solder 4 wires (70mm) on the sensor card (VCC, GND, XOUT, YOUT) (ref [Annex 3 – Figure 2]).

To glue: (ref [Annex 3 – Figure 1]).

- Glue push button on top panel box.
- Glue led on top panel box
- Glue sensor card on top panel box.

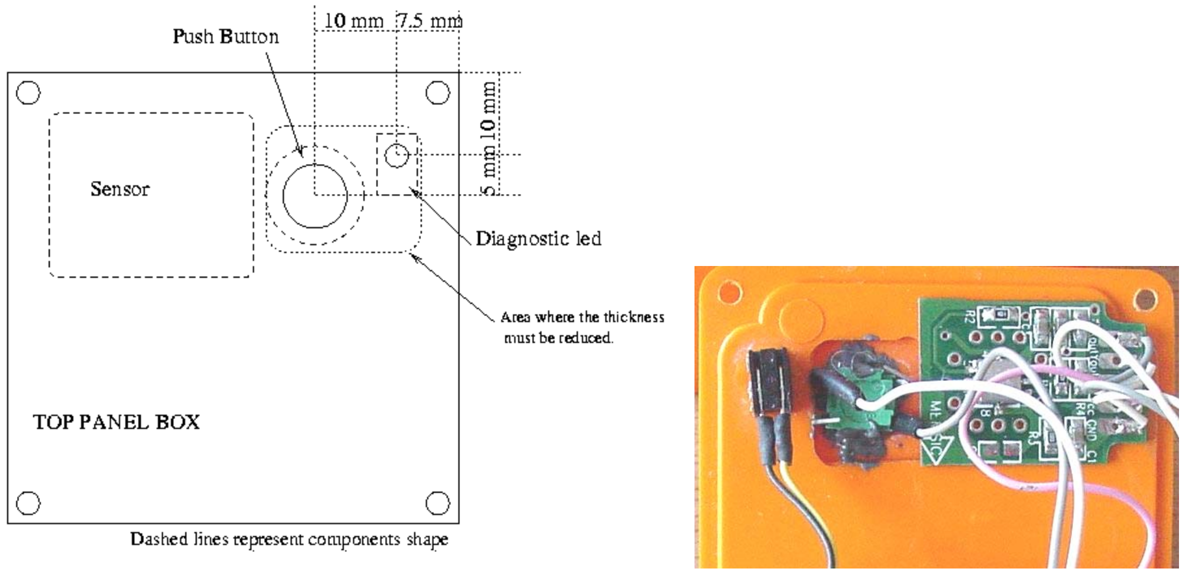
To plug: (ref [Annex 3 – Figure 2]).

- Plug the shunt if needed.
- Plug the 74HCT08 chip.
- Plug the LBA110 chip.
- Plug the micro controller card.

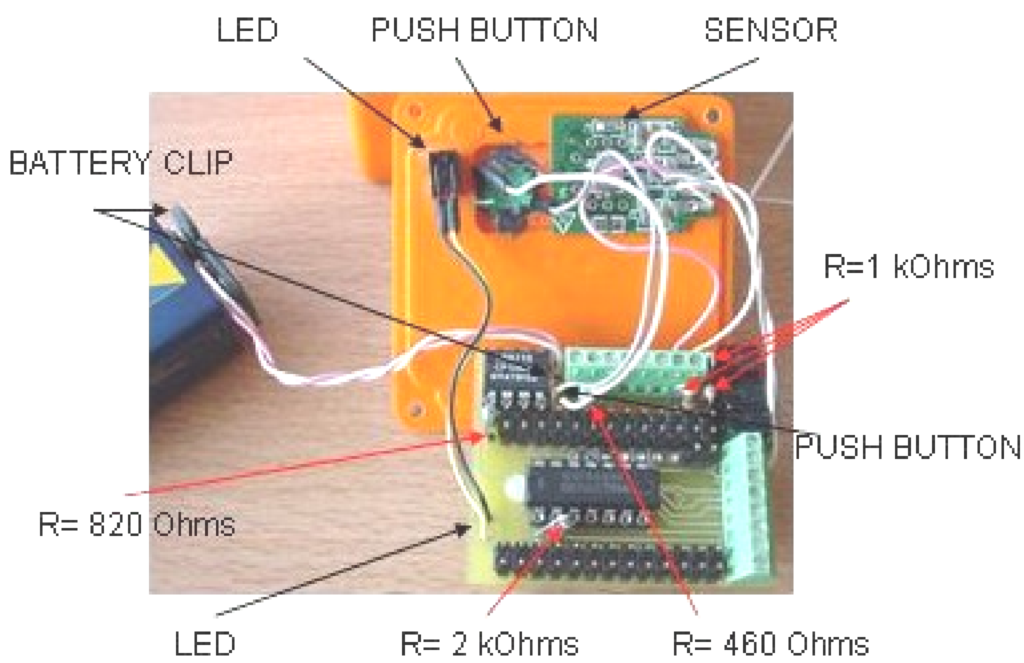
Screwing:

- Screw the Bluetooth module on the interface PCB, then the Bluetooth connector can be plugged on the Bluetooth module (ref [Annex 3 – Figure 3]).
- Screw wires coming from the sensor in the PCB connector. (ref [Annex 3 – Figure 2]).

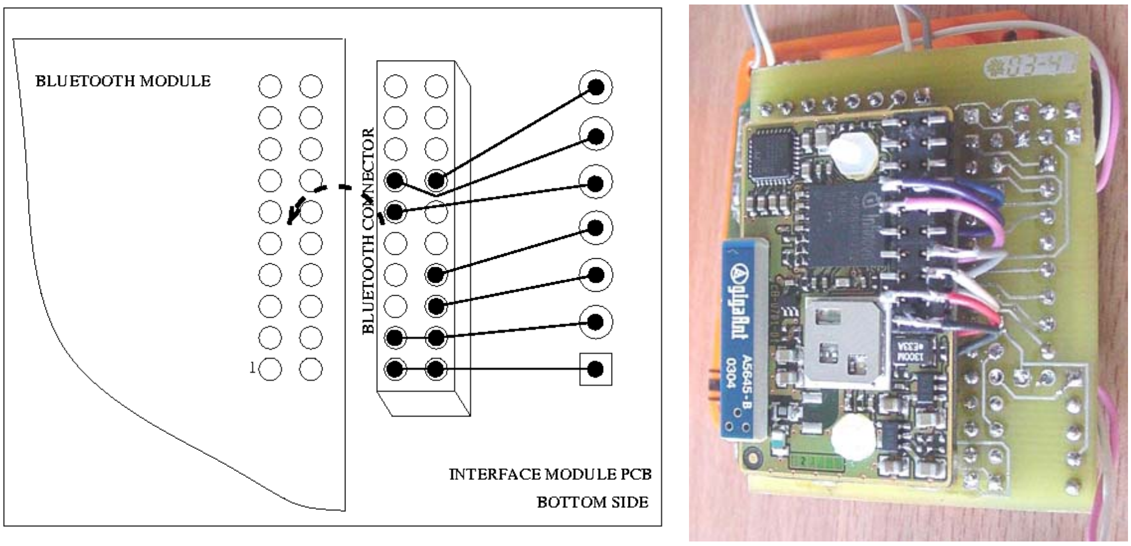
Final mount: (ref [Annex 3 – Figure 5]).
- Plug battery using the battery clip.
- Insert the battery, then the insulator plate.
- Insert the devices (Bluetooth module + Interface module + Micro controller module).
- Screw the top panel on the box using the four screws.



Annex 3 – Figure 1: top panel (left: outside, right: inside)

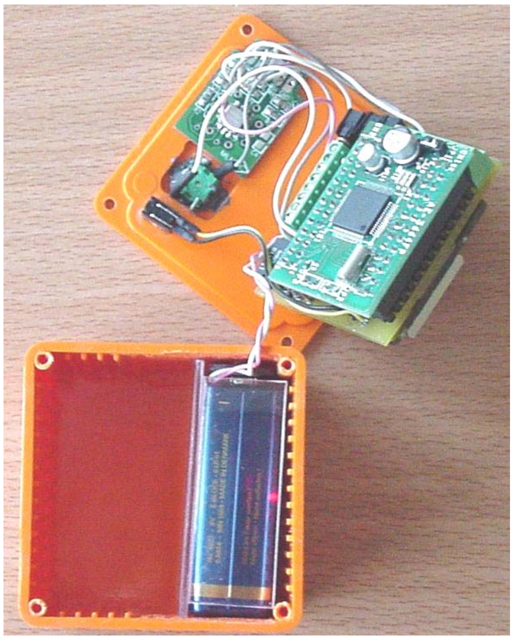


Annex 3 – Figure 2: interface module PCB placement



Annex 3 – Figure 3: Bluetooth connector

Annex 3 – Figure 4: insulator plates



Annex 3 – Figure 5: the whole device opened

7.4 Component list and cost

We list in the following table any software and hardware items used for building the wireless sensor.

Name	**Company**	**Reference**	**Distributor**	**Reference**	**Price (euros)**
Micro controller module	Renesas	AE-3664FP	Microtronique (ref [web08])	AE-3664FP	39
H8 tiny board	-		Microtronique		
H8 development board	-		Microtronique		30
Software H8	Renesas	HEW 2.2			0
Flash Toolkit	Renesas	FDT 2.2			0
GNU H8 C	KPIT	wingnuh8v0303coff			0
BlueTooth Module	ConnectBlue	OEMSPA13i	Eurodis (ref [web21])		158
Software ECI API	ConnectBlue	CB BTEDK01	Eurodis		305
Sensor	MEMSIC	mxr2999m	Eurodis		20
Sensor PCB	MEMSIC	MXEB-002 L	Eurodis		
Interface module PCB	INRIA	Inria ws	Teleph	Inria ws	7
Static relay	CLARE	LBA110	Radiospares		8
Logic And	Texas Instrument	745(A)HCT08	Radiospares	221-0883	0.7
LED 2 mm (x4)	Dialight	551-1(123)07	Radiospares	197-126	0.61x3
Clips for battery			Radiospares	172-5996	0.5
Connector I/O (x2)	Phoenix Contact	MPT 0,5/8-2,54	Radiospares	220-4327	2x2
Connector Bluetooth (2 mm pitch)			Radiospares	180-6578	5
Push button					
Cube box			Radiospares	281-6936	3
USB Bluetooth dongle					40

Table: purchase list

The prototype price is about 300 euros.
The only device you need to work with the sensor device is a Bluetooth dongle connected to a PC. A dongle can be purchased in any computer accessories store. Its cost is about 40 euros.

7.5 Project milestones

Date	**Action done**
March 2003	Idea that technologies are ready for efficient wireless sensor devices.
April-may 2003	Investigate technologies using World Wide Web (existing wireless sensors, research projects).
May 2003	Choose the Bluetooth technology for wireless communication.
May 2003	Choose a dual axis inclination sensor as a test bed sensor.
May 2003	Investigate battery power supply technologies (Li-ion, Nimh,…).
June 2003	First document draft: Wireless Sensor Module Based on Bluetooth technology: overview.
June 2003	Choose the micro controller (H8/3664F).
July 2003	Choose the Bluetooth module (ConnectBlue OEMSPA13i).
July 2003	Buy the micro controller module and the Bluetooth module.
August 2003	First programs on the micro controller (no sensor, no Bluetooth module).
September 2003	Run a program on the micro controller using the dual axis inclination sensor.
September 2003	Test bed interface module first design.
September 2003	Run a program on the micro controller using the Bluetooth module. The Bluetooth module configuration is made by a Windows wizard & AT commands.
September 2003	Power saved mode on the micro controller and on the Bluetooth module mastered.
October 2003	Tests about the communication performance performed on the whole module.
October 2003	Measurement of the power consumption performed on the whole module.
October 2003	Write documentations (overview & prototype).
October 2003	Bluetooth firmware upgraded.
October 2003	Purchase additional micro controller modules.
October 2003	Interface module design updated (CAD).
November 2003	Auto power off feature added.
November 2003	First packaging.
November 2003	Final tests.
November 2003	Final document.

Table: project milestones

7.6 Wireless Sensor datasheet

The “Wireless sensor” device provides remote sensing in an efficient way. No physical connection is necessary. Any off-the-shelf Bluetooth dongle connected to a PC allows interfacing this device.

Dimensions: 55mm x 55mm x 42mm (cube box made in ABS plastic).

Weight: 120 grams (including 48 grams for battery).

Energy:
- Primary battery type PP3 9 Volts,
- Autonomy better than one month (depending on day-by-day use),
- Auto power off feature.

Technologies used:
- Bluetooth module based on Infineon chips (Power Class II),
- 16-bits micro controller from Renesas (formerly Hitachi); H8/3664F,
- MEMS (Micro Electro Mechanical Systems) dual axis inclination sensor.

Performance:
- +/-60 degrees dual axis inclination measurement,
- Resolution better than 0.5 degree arc,
- Bandwidth better than 100 Hertz,
- Distance between PC and device up to 10 meters (Bluetooth class II).

Software development (C language):
- Micro controller is ready for working. Source codes are available. Renesas software development tool-kit is free. Therefore embedded program can be modified very easily for fitting application.
- A Windows Console type program with full device functionalities support is available as source code.

Medical application:
- Sterilizable package,
- Read in the news: a Bluetooth based device received a FDA approval.

A technical report written in English includes a detailed hardware and software description.

Other packaging can be designed easily.

Extensions to other sensors are possible, thanks to the eight analog inputs available on the micro controller.

Extensions to other devices, including LASER or motor remote controllers, are possible, thanks to digital inputs and outputs available on the micro controller.

Unité de recherche INRIA Rhône-Alpes
655, Avenue de l’Europe, 38330 Montbonnot-St-Martin (France)
Téléphone : +33 4 76 61 52 00 – Télécopie : +33 4 76 61 52 52