Smart-Its – Communication and Sensing Technology for UbiComp Environments

Michael Beigl, Tobias Zimmer, Albert Krohn, Christian Decker and Philip Robinson
Telecooperation Office (TecO), University of Karlsruhe
{michael, zimmer, krohn, cdecker, philip}@ teco.edu

Abstract

More and more Ubiquitous computing scenarios arise where computers are embedded into everyday objects as secondary ariefacts. The need for a generic platform to enhance such artifacts is addressed by the Smart-Its project. TecO's Smart-Its are small-scale embedded computer devices that can be easily attached to everyday objects and simply programmed for a special task. Just like paper Post-Its are able to add some information to an object a Smart-Its could add some computing, perception and communication capability to an object. All this capability was integrated into the tiny Smart-Its board itself that is able to operate without any additional infrastructure. This paper introduces the architecture of the Smart-Its hardware, the design decisions for the hardware itself and presents experiences we made while constructing and operating these devices; some of these experience are also based on the deployment and use of the first 160 Smart-Its devices.

1. Introduction

TecO's Smart-Its platform for communication and sensing in ubiquitous computing environments was developed at the University of Karlsruhe over the last two years based on prior experiences with similar devices, e.g. the MediaCup [1] or MemoClip project [2]. Smart-Its are small computing devices intended for attachment to every-day objects, thereby augmenting them with sensing, computing and communication capabilities. Smart-Its follow a post hoc approach to the enhancement of every day objects with small computers. Smart-Its are the enabling technology for developers and researchers to rapidly prototype and test new applications and application scenarios for ubiquitous computing environments.

The Smart-Its platform consists of two independent boards, a core board that consists mainly of processing and communication hardware and basic output components and a sensor board containing a separate processing unit, various sensors and actuators. This sensor board is intended to run the user application for context gathering.

Based on the special requirements in ubiquitous computing environments, this document gives an insight into the technical details of the implementation of the Smart-Its hardware. It describes critical design decisions and criteria for the selection of used hardware components with respect to these requirements like e.g. low power consumption, high mobility and mechanical robustness.

In the following section we introduce the general concept behind Smart-Its and then explain the Smart-Its architecture starting with the core board in chapter 3. The sensor board, its sensors and some experience are described and compared in chapter 4. A summary of our experience with development and application and an outlook of the Smart-Its is found in chapter 5.

1.1. Background and Related Work

Prototyping has a long tradition in Ubiquitous computing, from the first Ubicomp system, XeroxParcs ParcTab [3], up to several systems today. Many ubiquitous computing platforms also deal with the construction of a small device platform to be adapted to the tasks of ubiquitous computing. Such applications are important to collect experiences to evaluate the benefits of the new technology introduced into everyday settings. A prominent example of this is the Aware Home Project at GeorgiaTech [4]

But building such environments requires enormous effort for all of these projects and sometimes leads to awkward workarounds. For example in GeorgiaTechs CyberGuide a location system was built utilizing TV remote controls as infrared location beacons [5]. In other settings, such as the SpotOn environment [6], devices were specially adopted for the task of location detection. More general small devices are the Motes from the DARPA funded Smart Dust project [7]; actually these devices follow in their hardware design a quite similar approach to Smart-Its. The mentioned devices especially share the same network philosophy: A rather small bandwidth network provides the communication channel for exchanging data and still takes care of the energy consumption. One of the first systems of this kind was the Prototype Embedded Networking (PEN) [8] of the AT&T Labs in Cambridge. Other projects where devices are directly attached to an object are more concerned with retrieving information from persons or objects via a bunch of sensors. E.g. in the implementation of the Affective computing project at the MIT several devices are build to collect sensor information directly at the optimal place (e.g. the ear) to retrieve a highly reliable set of context information [9].

Most of these devices have one thing in common: they are developed for one special purpose and are not intended for reuse in other applications. But there is a need for a general device and architecture to support tasks in Ubiquitous Computing environment. With Smart-Its we would like to introduce a general architecture, device and application support following the idea of facilitating a wide range of applications in Ubiquitous Computing. Rather than being designed as a new type of objects, Smart-Its are intended to extend everyday objects to make them aware of the surrounding environment and situation and to let them join other objects. Smart-Its are therefore an enabling technology for augmenting objects in the real world with additional functionality that include hardware, software and tools for developers.

2. Requirements and Architecutre

2.1. Requirement analysis

Requirements for Smart-Its were collected from experiences with previous scenarios and Ubicomp settings, among them MediaCup [1], MemoClip [2], TEA [10], AIDE [11] that have been implemented at TecO.

All settings required a very flexible use of the hard- and software. Changes in the configuration where needed in all phases of the setting, the design, the implementation and the evaluation phase. Thus a modular concept with exchangeable and expandable modules and attachable additional hardware and the use of existing software libraries are important here. Flexible use is also a matter of physical dimensions: Most of these attachable devices are typically rather small, and have been tailored especially for their application domain with respect to their physical outline. By way of examples, in the MediaCup setting a small round circuit board was developed to fit the base of the cup, while the board developed for the TEA project was made to fit on a mobile's riser. A special physical consideration in AIDE was the facilitation of one-handed input such that a device that fits in one hand was developed.

The robustness of these artifacts is also significant. The hardware must resist mechanical stress, while the software should be administration free. Furthermore, low power consumption was found to be a major issue in all settings. As the user of such small devices and objects is often not aware of the state such devices are in, energy becomes an important subject regarding robustness of the application.

2.2. Architecture: Separation of concerns

As Smart-Its were designed for use in laboratory environments, the architecture of the Smart-Its was mainly driven by the needs of the designers and developers. We identified two major functional components that were often developed independently: Firstly, for facilitating communication between devices including high-level communication, processing and filtering of information and, secondly, for accessing, processing and storing of sensor and actuator related information. As a consequence, each functional part was separated into autonomous functional units that were developed independently from the other. Consequently, both units supply full, independent processing and storage functionality.

To allow for a clear separation of concerns we decided to split the functionality not only by software but also to develop two self-contained, independently operational hardware boards, each containing their own processing, memory and system software. One of these boards is responsible for communication tasks, while the other supports sensing and actuating functionality. Additionally, splitting the overall functionality of a Smart-It allows for an increased flexibility for the developer in altering parts of the system. Developers may choose from a variety of upcoming boards and may choose to replace sensor boards with more sophisticated versions of the boards, or may even choose to attach more specialized boards - e.g. high speed communication boards - that better fit their needs. This sort of component interchange is indeed possible

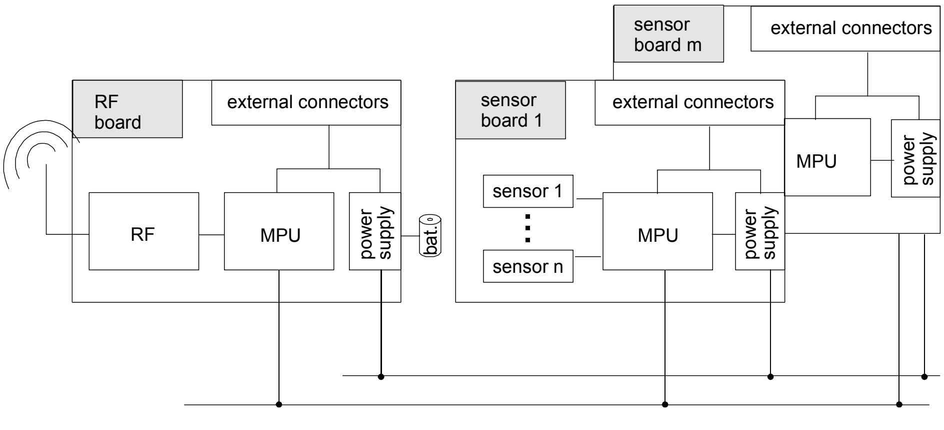

Figure 1 – Smart-Its Architecture

without the need for rewriting existing applications.

The architecture of the Smart-Its also seen in Figure 1 is as follows: a typical Smart-It contains one RF board for wireless high-level communication and zero to 16 sensor boards. All boards are interconnected via an inter-board bus that contains lines for data exchange and lines for control and power supply. All boards have the common operational characteristics that they contain their own power supply, their own processor, program and data memory and various connectors for attachment of additional hardware or to other computer devices (PCs, handhelds etc.). Essentially, communication boards contain a RF based wireless communication unit, while sensor boards possess a variety of sensors and actuators.

The architecture is open for other types of boards, e.g. high performance computing boards for specialized applications or memory boards. The complete Smart-Its hardware architecture is designed to provide maximum flexibility to be adapted to the special needs of applications and everyday objects to which a Smart-It is attached. A minimal Smart-It may consist of a RF board with probably only a small sensor connected via the external connector. More powerful Smart-Its may come with a collection of sensor boards connected to a RF board via the Inter-board interface.

3. Smart-Its Core Board

The Smart-Its communication board facilitates communication between Smart-Its but also contains basic sensing capabilities. It is able to run small applications without any additional hardware. Components on this board are the integrated microprocessor (Microprocessing Unit, MPU), radio frequency integrated circuit with attached digital potentiometer, power supply circuitry, a movement sensor and various connectors (see Figure 2).

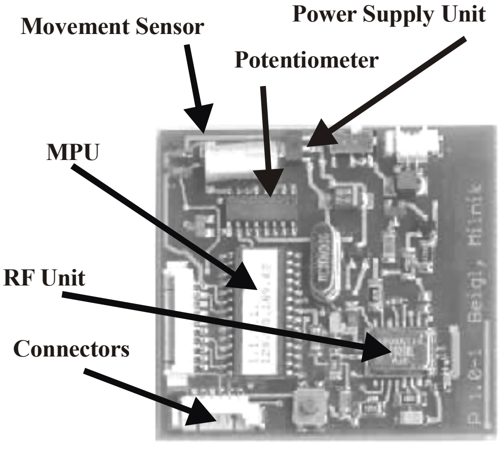

Figure 2 – Smart-Its Communication Board

3.1. Microprocessing Unit

The microprocessor on the communication board is responsible for running communication applications (e.g. the communication stack), simple sensor and actuator applications (e.g. through ball-switch sensors, piezo speaker) and for the transfer of data to and from sensor boards. As it should also be of small scale a stipulation for such a processor is reasonable computing power, integrated program and data memory and integrated I/O functionality.

We chose the microprocessor from Arizona Microchips PIC16F87x family of devices (a PIC16F876), an integrated microcontroller that is simple to configure and to program. It has 8k x 14 words of fast re-programmable flash program memory, 368 bytes data RAM and 256 byte EEPROM. At an external clock of 20Mhz the reduced instruction set CPU inside the microcontroler is able to perform 5 MIPS. It is also equipped with a built in UART for serial communication, A/D units for sensors and actuators, Timers, I^2C for connecting to other boards and hardware, SSP, etc.

Selection of the processor followed a pragmatic approach. There were several similar microcontrollers besides the PIC available. Our choice was mainly driven by the fact that for this processor a large community all over the world provides and shares experience, source code and hardware layouts. This allows users of our Smart-Its platform to easily implement additional hardware and software

Also the processor is still technically well positioned. Using the PIC based processor family allows us to rapidly develop projects in the area of Ubiquitous Computing like the MediaCup or MemoClip prototypes before.

Since Smart-Its are battery powered in most application scenarios, based on our Smart-Its scenario building experience, one important (if not the primary) goal is to keep the energy consumption as low as possible to achieve maximum lifetime. As Smart-Its are not only sensor and communication nodes but also run application programs it turns out that the core processor is the major energy-consuming component. Consequently, to save energy, applications should minimize the time of operation. This includes activity and idle operation of the microprocessor. Ideally, the processor should either run with full load or go to sleep mode.

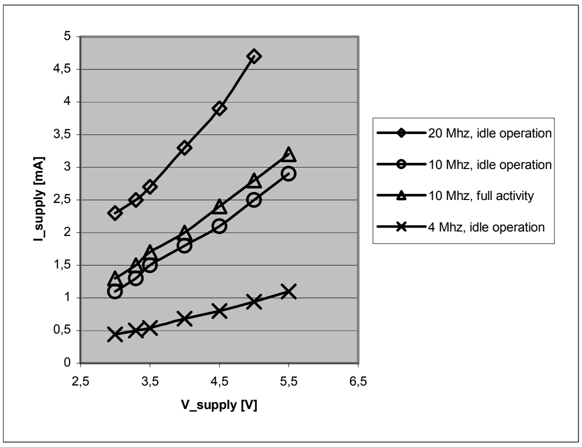

Figure 3 – Supply Current for PIC16F876

Figure 3 shows the supply current measured by us depending on the activity of the microprocessor. In this figure idle operation means, for example, active polling for an external signal. The energy consumption in idle state is nearly as high as in full operation. So an idle state should be replaced with the sleep mode (supply current: some µA) whenever possible. Oscillator considerations are important, when we talk about energy consumptions. There are different types of oscillator circuits possible with the PIC. It is possible to use either an external oscillator (high power consumption), a quartz crystal (moderate consumption) oscillator or a RC element (very low power). In our case, we decided not to use an external oscillator due to size and power considerations. The RC element was not feasible because the processor's driver circuit only supports frequencies up to 4 Mhz, which was found to be too low for major tasks such as communication or sensor reading. We therefore applied a quartz crystal, which also provided the precision required for high-speed communication tasks. The clock frequency should be kept as low as possible because the supply current increases with the higher clock as consequence of the capacities in the semiconductor die. As this is not possible in most Smart-Its application, further power optimisation can be accomplished by keeping the frequency variable, which is possible with some new PIC processors.

3.2. Power Supply Unit

Mobile devices need independent energy supplies, which often suggest the usage of batteries, and, furthermore, rechargeable units. For practical reasons the power supply unit should therefore be able to run with a large variety of batteries. As batteries come with various voltage levels a conversion and stabilization of the voltage level is required to assure a stable operation of the hardware. Additionally, the power supply unit should be not consume too much energy and should also not consume too much space on the board.

It is therefore advantageous to have a single power level on one board. It reduces the need for additional components (logic level converters) and circuit paths.
We identified four basic ways to build a stable power supply unit for Smart-Its.

Firstly, we used a simple voltage down regulator, which are often applied in all kinds of consumer electronic. These regulators only work as long as the input voltage is higher than the output voltage. The voltage difference is converted into heat. For a mobile device this would mean that a mentionable part of the energy is used for heating! Once the voltage level of the batteries falls below the target core voltage of e.g. 3V, the supply would become unstable or even shut down. Considering a battery of max. 6V, limiting the possible input voltage to 3V suggests that half of the capacity and lifetime of the mobile device is lost.

The second possibility is the usage of step down circuits. The higher input voltage is reduced via induction and efficiency over 90% can be achieved. Still, they only work as long as the input voltage is higher than the target voltage. Third, step up circuits work with switched inductivities that can generate highly inducted voltage peaks that can be collected by a capacity in order to then supply the circuit with a higher voltage. Forth, charge pump circuits can as well provide higher output than input voltages but do not reach the efficiency of step up converters.

We are using a Motorola MC33463, which has a very low bias current of some µA and a high efficiency when using high-quality coils. The step up IC is able to work with a minimum input voltage of approx. 800mV allowing the usage of almost all type of batteries and rechargables. In our experience the AAA size batteries provide the best trade-off in terms of supply duration and size.

3.3. Radio Link Unit

The Smart-Its feature a wireless, ad hoc network for communication of context. Hardware requirements for the design of the radio link were size, energy consumption, full control on the physical layer and mid-speed transmission bandwidth.

We opted for the TR1001 radio transceiver from RF Monolithics that seemed to best match the above requirements. We obviously share this opinion with several institutes over the world that designed similar boards and all independently selected this device (e.g. the Berkely Motes, SpotON). The TR1001 is available for the European Industrial, Scientific and Medical (ISM) band at 868Mhz or as TR1000 for the US ISM band at 915 MHz. It modulates Amplitude Shift Keying (ASK) or On/Off Keying (OOK) with a data rate of about 125kBit maximum. This cannot compete with W-LAN or bluetooth with respect to data rates, but it provides a very free programmable and controllable radio interface, which better fits the needs of research on low-level communication in comparison to other technology.

Critical design issues are the automatic gain control and the external data receiver circuit. These both must be designed in one step, as their functionality is highly interdependent. At the maximum emission power of 1mW, the transmission range in our office environment is about 20m, which provides a stable RF link in one or two rooms. A remarkable energy saving can be achieved by choosing a minimal required transmit power for a communication over RF.

The current into the TX input of the TR1001 sets the modulation depth and therefore the transmission power. The interrelation is visible in Figure 4. More current means a higher ASK amplitude. The emitted power can be adjusted over the full range until maximum output power. A digital potentiometer (Dallas Semoiconductor DS1803) is used to provide the different resistances for adjusting that input current. The value of the potentiometer can be changed between 0Ohms and 200kOhms in 256 steps from the processor by software via an I^2C interface. With the maximum resistance of 200kOhms the transmission range is limited to less than one meter.

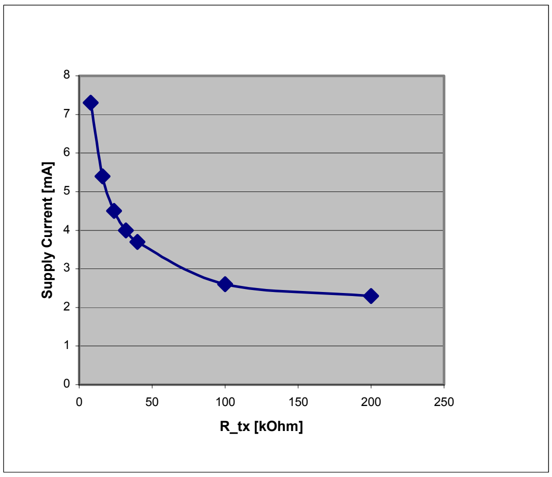

Figure 4 – Supply current for a "high" transmit level on TR1001 at 125kBit ASK mode

3.4. Movement Sensor

The idea to place sensing functionality in the core of the communication board was motivated by two factors. First, communication tasks in a Ubicomp environment can make use of changes in the environment. E.g. detecting a change of the position may also lead to a change in the network and may therefore require appropriate action. Second, simple sensing tasks can be implemented without attaching additional boards solely by using the communication board.

From analysing our settings we found the detection of simple movements of the object as the most interesting sensor value for most applications running on a sensor board. We embedded on every Smart-Its communication board a ball switch that can be used to detect movement and movement patterns. The sensor itself consists of a small metal cylinder containing a ball able to shortcut the cylinder's jacket with a metal contact in the middle of the cylinder. The ball switch is also used as a component to trigger the wake up event for the processor in sleep mode. It is important to note that the ball switch as a sensor does not consume any energy, but can still trigger an event.

3.5. Actuators

Actuators are used to signal certain states within the program execution on the core board and can be used for debugging during software development process. They can indicate e.g. errors or RF link states either visually, audible or tactile. Besides visible LED type actuators Smart-Its have a possibility to drive an external speaker or vibration component. In most cases we used a piezo speaker because of size and low power consumption.

3.6. Interconnection

Connectors provide additional flexibility to hardware. For Smart-Its flexibility was needed to

- Attach different types of batteries according to the needs of the application, especially the run time
- Attach other boards (e.g. Sensor boards) to the communication board
- Attach other hardware modules (Sensors, Actuators) to the communication board
- Attach the communication board to other computers, e.g. PCs or PDAs
- In-Circuit program the processor

We identified five interfaces that have to be implemented to fulfil the above functionality: Power-In for the batteries, communication to other boards via I^2C, an industrial standard 2 wire bus to interconnect various types of hardware devices at 100 or 400 kHz speed. Beside these, various input/output (I/O) lines for additional sensors, serial lines to attach the board to a PC and a programming connector are of need.

To reduce the overall number of connectors we implemented 3 connectors on the Smart-Its communication board. A 2-pin power supply connector, a 6-pin programming connector and a 10-pin feature connector, which provides an I^2C, a serial and an I/O interface.

We used 1.25 mm pitch connectors from Molex, which offer a good ratio of available pins and packaging for the connector. These Molex connectors can be interconnected through cables. Such a cable connection allowed us to place various boards in different places, e.g. to place the communication board hidden in an object and to locate the sensor board on a position that best fits the need of the sensors.

Robustness of the mechanical components of the connectors turned out to be the main issue using cable connectors. While running our experiments, cable connections emerged to introduce problems due to heavy use of the connectors especially with the power and programming connectors. Here the mechanical stress on the cable connectors resulted in unstable interconnections. Nevertheless the interconnection using Molex connectors and cables worked well in application scenarios, where the connectors were only infrequently plugged and unplugged.

4. Sensor Board

The sensor board (Figure 5) provides a collection of sensors that we found of general use for Ubicomp applications. The selection was informed by previous projects, such as TEA, MediaCup or MemoClip.

Besides the sensors, a sensor board contains its own central processing unit, various connectors and a power supply unit. We opted for the Arizona Microchip PIC16F877, a MPU from the same family as the processor on the communication board but providing more I/O lines. Also, the power supply unit has the same characteristics as the communication board. The sensor board is supposed to run the sensor application and acquire the sensor values. The sensor board is intended to be attached to the communication board but can also be used as a generic sensor modules for other devices e.g. embedded computer boards, PDAs or PCs.

The core board provides the RF communication services for the sensor board. We describe the different sensors used in detail in the following sections.

4.1. Connectors

For the same reason as the communication board, the sensor board also requires connectors to be flexible in extending its functionality and to provide its functionality to other devices. Basically, the sensor board uses the same connectors. Additionally, connectors for high-speed serial connections up to 1 Mbit/s and regulated power out for 3.3V and 5 V are added. Also, the I/O lines where extended with lines for directly attaching analog sensors to the board.

4.2. Acceleration Sensor

In our experience, movement is one of the most important contexts, as it is related to changes in the environment and to the usage of an object. To allow for sophisticated movement detection an acceleration sensor is implemented on the board. Because of the size and the power consumption, only MEMS types of acceleration sensors are applicable on a small wireless device. Because of its simple use, small outline, low power consumption and good resolution we selected the ADXL202 from Analog devices.

This two-axis accelerometer comes with a resolution better than 5mg and a maximum range of 2g. The signal output is either an analogue signal or a square wave signal that represents the value in its duty cycle. We use the square wave signal output because it is much easier to take a precise time with a micro controller than to handle analogue signals. The maximum update rate is about 1kHz. Configuring the ADXL there is a trade off between update rate and noise in the output signal. The best noise reduction leads to a power up time of about 52ms until the square wave output has stabilized.

We found that acceleration is an excellent source for generating shared context of physically collocated objects, e.g. for theSmart-It friends [12] application.

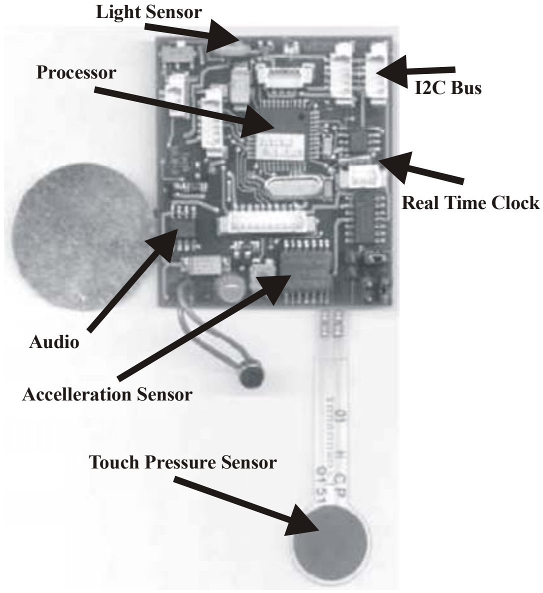

Figure 5 – Smart-Its Sensor Board

4.3. Illumination Sensor

Light level and frequency is mainly used for detecting environment condition. Precision is less important than efficiency – especially power consumption – robustness, scale and flexibility in choice of the detected frequency spectrum. Such flexibility allows use of the sensor for detecting sunlight, artificial light conditions or infrared beams.

To minimize the need for additional components we selected a light sensor with integrated filter and amplification. We settled on a cheap and easily available sensor, which directly converts the light level into a voltage. The sensor is available for various light frequency spectrums including daylight and infrared and with different sensitivities and velocities. The major problem of this small sensor is that it runs into saturation when exposed to strong light. As the application field for an object was often known before the problem was fixed by using an additional optical filter.

Using multiple light sensors for various light frequencies was found to be very helpful for situation detection. Detecting visible and infrared light at one time helps e.g. to discover the type of the light source (bulb, sun, LED…) because of the different power distribution over the spectrum. Detecting a light level of an object illuminated by fluorescent tubes means to average over at least one period of the 50Hz because the light level changes during one period.

4.4. Audio Sensor

Audio signals can be used to detect various situations and contexts. As processing and storing audio signals is difficult using small MPUs as in Smart-Its, special care has to be taken to deliver easy-to-process audio signals. Technically this means that the audio sensor should provide a linear and noise-free audio signal to avoid additional stress of the processor with adaptation or filtering tasks.

Using audio signals for recognition, environment analysis or any other application in ubiquitous computing, there are three major steps necessary: First, the choice of the microphone, second, the choice of the amplifier and last the A/D conversion.

The choice of the microphone narrows as soon as the conditions for lightweight, small size and mobility are kept in mind. These are perfectly realized in capacitive microphone. A critical design issues for the amplifier circuit was to cut out the noise on the power supply (what you cannot avoid if using a step up converter) and low energy consumption.

We use a Linear Devices LM4880 as amplifier and a high linear capacitive microphone with low noise. It was selected especially for size and ease of usage. For A/D conversion the internal analogue sub group of the PIC microprocessor was considered satisfactory, which can provide up to 35kHz sampling frequency with a resolution of 10 bits.

We found that getting audio information from the environment can be useful for understanding the context of a mobile device. Simple calculations of e.g. the sound level can lead to conclusions on the activity level in a certain surrounding. Sensitivity of the microphone was of high importance for those applications.

4.5. Magnetic Field Sensor

The natural magnetic field can give an indication of the orientation of a device or the co-location of two devices. There are two main types of magnetic field sensors available today off-the-shelf, for measuring these natural magnetic fields. First, sensors based on a ferromagnetic material called Permalloy, which shows a relatively strong magneto resistive effect that changes the resistance depending on the applied magnetic field. The second type of sensors measure the magnetic field based on the Hall effect with the help of two or more silicon Hall plates. Both sensor types are rather insensitive to measured physical value compared to other type of sensors. This makes it difficult to analyse the value with justifiable complexity for a small board. Our main decision criteria were therefore efficiency of the design, energy consumption and size.

We decided upon a Honeywell HMC1022, an optional magnetic field sensor, that can be applied to the connectors of the Smart-Its. It is a 2-axis Permalloy based sensor. With a strong current pulse (500mA for 2µs) one can define the main magnetization direction within the Permalloy film. Compared to other equivalent sensors the set/reset current needed is quite low. The output voltage is 5mV/gauss (the earth's magnetic field is 0.5 gauss) at 5V supply voltage, which is higher than with comparable Hall effect based sensor devices. The output voltage has to be amplified through a low noise amplifier before given to the A/D converter. We use the INA122 instrumentation amplifier because of its very low noise and quiescent current (60µA) and simple gain set by only one additional resistor.

Our experience shows that the overall achieved precision is about one degree in orientation, which is enough for most application settings. There are serious impacts on the performance of measuring orientation coming from electrical devices in vicinity of the sensor that superimpose the earth's magnetic field.

4.6. Temperature Sensor

Temperature values are important for measuring the internal temperature of an object (e.g. for a liquid in a cup) and to measure the external temperature of the environment. For both measurements flexibility of use and robustness is more important than precision of the device.

Temperature can be measured using two principles. First, a materials characteristic can be measured that changes according to the temperature – e.g. a resistor may change its value according to the temperature. Such sensors require contact between the sensor and the object. Second, the infrared radiation can be measured. Here the sensor is required to have a minimal distance from the object that should be measured and the measurement is restricted to non-metallic objects.

For our general-purpose sensor board this was found too restrictive. To measure an environmental or object temperature, we used the Dallas DS1621 digital thermometer. It has a wide measuring range from -55°C to +125°C with 1°C resolution and is integrated in a small 8-pin SOIC case. The chip can be controlled over the I^2C communication bus, so no additional parts are needed. This feature also allows us to use multiple temperature chips on one Smart-It. Also, with the DS1621 no calibration has to be done; conversion and temperature calculation is completed within the sensor and takes 1000ms. The temperature can then be read out over I^2C in degrees centigrade; no A/D conversion is needed.

We found, that the slow read out brought a remarkable energy effort because the sensor had to be powered over the 1s conversion time. On the other hand, the simple use of multiple temperature sensors on one board allowed us to measure temperature in the environment and on multiple parts of the object at the same time.

4.7. Pressure Sensor

Touch sensors are useful for direct or indirect user interaction. Situations like “object is lying on the desk” can be detected with the help of this sensor.

Small outline and low energy consumption were our main criteria for selecting the pressure sensor. From the sensors available today only foil type pressure sensors are fulfilling both criteria. We selected the FSR-151AS from IEE, touch sensor, providing a changing resistance from 3k to 1MOhm over the applied touch pressure. The range includes weight equivalents from 1g to some kg. This changing resistor is then translated to a voltage over a resistor bridge. The mapping from applied force to resistor values is not linear and varies a lot between equal sensors.

Our experience shows that the sensor was not designed for exact measuring of weights. It was found to be usable for binary decisions like if an object is touched or not or lying somewhere with it’s own weight pressing on the sensor. In most scenario this characteristics was found to be sufficient. Furthermore the sensor could also form buttons that have some trigger level that distinguish between different push forces. The ranges fit quite well into the area of the force of human hands.

4.8. Comparison of Sensors

Table 1 shows a comparison of the used sensors. Measurements were taken for turn on times, power consumption and package size. The values in bold font were determined experimentally. The values in normal font style were taken from datasheets. Special interest lies on the column with the mean power for continuous measurement. This value was calculated on an update rate of 10ms. This interval is considered as real time for human interaction. The sensors are being shut down between measurements if possible realizing a power management like in [13]. The software architecture for the Smart-Its is also aware of power management [14] and takes care of energy issues especially in the sensor drivers and communication protocol. There are huge differences of mean power between the sensors. This must be kept in mind when designing applications for ubicomp devices with limited energy resources.

5. Summary and Outlook

The current status of the Smart-Its boards is sufficient to run larger application tests and to hand out hardware and software to application designers and developers. Up to now, 160 of these Smart-Its are built and working in various places and are delivered to

Values: Datasheet/ **Experiment**	Power up time	Current for measurement	Approx. acquisition time for a	Approx. acquisition energy for a	Approx. mean acquisition	Component package size	Price for single quantities
Light Sensor TSL250 (sensing office light; 50Hz lamp)	**500µs**	900µA/ **100µA (bright) - 760µA (dark)**	**21ms (Period of 50Hz)**	**53µJ**	**2.5mW**	4.85mm x 4.80mm x 2.74mm = 63.8 mm^3	4,30 €
Light Sensor TSL250 (sensing daylight)	**500µs**	900µA/ **100µA (bright) - 760µA (dark)**	**600µs**	**1.8µJ**	**125µW**	4.85mm x 4.80mm x 2.74mm = 63.8 mm^3	4,30 €
Acceleration Sensor ADXL202ae with C_filt=330nF,	**7ms ramp up, 50ms stabilize**	600µA/ **640µA**	**59ms**	**188µJ**	**3.2mW**	4.50mm x 5.0mm x 1.98mm = 44.6mm^3	16 €
Acceleration Sensor ADXL202ae with C_filt=3.3nF,	830µs/**50µs ramp up, 4ms stabilize**	600µA/ **640µA**	**6ms**	**19µJ**	**1.9mW**	4.50mm x 5.0mm x 1.98mm = 44.6mm^3	16 €
Temperatur Sensor DS1621	**150µs (I2C data), 1s temperature**	1000µA/ **450µA**	**1000ms**	**2250µJ**	**2.3mW**	8-Pin SOIC	4,50 €
Magnetic Field Sensor Honeywell HMC1022 and opamp INA122	**2µs pulse**	**2* 5mA (bridge current) +2A**	**4us**	**42µJ**	**4.2mW**	16-Pin SOIC	36 €
Capacitive Microphone and Amplifier lm4880	**1ms**	3,6mA/ **1600µA**	**20ms (stationarity of speech)**	**160µJ**	**8mW**	h=2.95mm; r=3.00mm; 83.4mm^3; 8-Pin SOIC	6,10 €
Touch Pressure Sensor FSR-151AS and opamp lm324	**10us**	700µA/ **600µA**	**12µs**	**156nJ**	**16µW**	51.65mm x 15.23mm x 0.43mm = 338mm^3; 14-Pin SOIC	5,00 €

Table 1 – Comparison of Sensors used on the Smart-Its Sensor Board

partners within Europe.

Still there are several topics on the wish list of our agenda. Firstly, although the delivered Smart-Its provide a large platform that gives us substantial feedback we would like to distribute the platform even wider to collect more user experience.

Secondly, there are special hardware issues where we would like to have a second look into. One is the inter-board and intra-board communication of Smart-Its. An interesting value by which to compare different types of busses is the bandwidth / power consumption factor. We will have a look into this factor for different types of busses suitable for the use in our setting. Another topic will be the RF communication. There is now a new family of RF transceivers available that would allow us to raise the RF bandwidth up to 1 Mbit/s requiring only minor changes in the hardware and software design. Also a further size reduction of the hardware is on our agenda. This allows us then to integrate Smart-Its in even smaller objects.

Large settings of objects equipped with a computer can be used for studies of how often context events arrive and how often they are distributed. Because there is currently no published information around what the behaviour would look like we will start to collect this data in a real setting and make this data available afterwards. Such data can than be used for more reliable simulations of such environments.

Several topics are open in development support. We will soon finish a development environment that lets an application developer maintain and program Smart-Its directly from his PC without removing the Smart-It from the object or attach some cables. Through an optional network backend infrastructure programming and maintaining Smart-Its will also be possible remotely via the Internet. We are also working on additional administrative and repository services to keep track of the Smart-Its and to collect data (e.g. context data that is used as user input) for later reviewing.

Widely distributed and mobile networks need new concepts for management and administration. Out Future research will also focus on these subjects in ubiquitous computing.

6. Acknowledgements

The Smart-Its project is funded by the Commission of the European Union as part of the research initiative “The Disappearing Computer” (contract IST-2000-25428). General information on the project is available at http://www.smart-its.org.

7. References

[1] Hans-W. Gellersen, Michael Beigl, and Holger Krull, ”The MediaCup: Awareness Technology embedded in an Everyday Object”, *Handheld & Ubiqutious Computing, Lecture notes in computer science*; Vol 1707, ISBN 3-540-66550-1; H-W Gellersen ed., Springer, 1999, pp 308-310.

[2] Michael Beigl, “Memoclip: A location based Remambrance Appliance”, *2th International Symposium on Handheld and Ubiquitous Computing (HUC2000)*, Bristol, UK, Sept. 25-27, 2000 and Personal Technologies Vol. 4 No. 4 2000, Springer Press, pp. 230-234.

[3] Roy Want, Bill N. Schilit, Norman I. Adams, Rich Gold, Karin Petersen, David Goldberg, John R Ellis, Mark Weiser, ”An overview of the PARCTAB Ubiquitous Computing experiment”,*IEEE Personal Communications*, 2(6), 1995, Seite 28-43.

[4] Gregory D. Abowd, Christopher G. Atkeson, Aaron F. Bobick, Irfan A. Essa, Blair MacIntyre, Elizabeth D. Mynatt and Thad E. Starner: Living Laboratories,”The Future Computing Environments Group at the Georgia Institute of Technology”, *Proceedings of the 2000 Conference on Human Factors in Computing Systems*

[5] Sue Long, Dietmar Aust, Gregory D. Abowd and Chris Atkeson.Cyberguide,”Prototyping Context-Aware Mobile Applications”, *Proceedings of 1996 Conference on Human Factors in Computing Systems*, Vancouver, CA, April 13-18, 1996. pp. 293-294.

[6] Jeffrey Hightower and Gaetano Borriello,”Location Systems for Ubiquitous Computing”, Computer, vol. 34, no. 8, *IEEE Computer Society Press*, Aug. 2001, pp. 57-66

[7] Jason Hill, Robert Szewczyk, Alec Woo, Seth Hollar, David Culler, and Kristofer Pister, ”System architecture directions for networked sensors”, *Proceedings of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems*, Cambridge, MA, USA, Nov. 2000, ACM, pp. 93--104.

[8] Gray Girling, Jennifer Li Kam Wa, Paul Osborn, Radina Stefanova,”The Design and Implementation of a Low Power Ad Hoc Protocol Stack.” *IEEE Wireless Communications and Networking Conference*, September 2000, Chicago

[9] Carson Reynolds and Rosalind W. Picard,”Designing for Affective Interactions”, *Proceedings from the 9th International Conference on Human- Computer Interaction*, New Orleans, August 2001.

[10] Hans-W. Gellersen, Michael Beigl and Albrecht Schmidt, "Sensor-based Context-Awareness for Situated Computing", *Workshop on Software Engineering and Pervasive Computing SEWPC00* at ICSE 2000, Limerick, Ireland, June 2000.

[11] Michael Beigl, “Point & Click – Interaction in Smart Environments”*, 1th International Symposium on Handheld and Ubiquitous Computing (HUC99)*, Karlsruhe, Germany, 1999 & Lecture notes in computer science; Vol 1707, H-W Gellersen ed, ISBN 3-540-66550-1; Springer, 1999, pp 311-314.

[12] Lars Erik Holmquist, Friedemann Mattern, Bernt Schiele, Petteri Alahuhta, Michael Beigl and Hans-W. Gellersen, “Smart-Its Friends: A Technique for Users to Easily Establish Connections between Smart Artefacts”, *Ubicomp 2001*, Springer-Verlag LNCS 2201, pp. 116-122, 2001.

[13] L. Benini, A. Bogiolo, G. De Micheli, “A Survey of design techniques for system-level dynamic power management”, *IEEE Transactions on VLSI Systems*, 8(3), June 2000.

[14] Gerard J.M. Smit, Paul J.M. Havinga, “A survey of energy saving techniques for mobile computers”, *Moby Dick technical report*, 1997.