IOSR Journal of Engineering (IOSRJEN)
e-ISSN: 2250-3021, p-ISSN: 2278-8719, www.iosrjen.org
Volume 2, Issue 9 (September 2012), PP 11-15

Iobs—Adding Intelligence to Real-Life Objects

Sahithi Govindaraju[1], Krishna Prasad Satamraju[2]
[1] *Department of ECE, Vasireddy Venkatadri Institute of Technology, Nambur, Guntur (dt), A.P., India*
[2] *Department of ECE, Vasireddy Venkatadri Institute of Technology, Nambur, Guntur (dt), A.P., India*

Abstract*—Ubiquitous computing systems are useful in developing technology to realize a vision of computation everywhere, where computer technology seamlessly integrates into everyday life, supporting users in their daily tasks. By embedding sensors, computation, and communication into common artifacts, future computing applications can adapt to human users rather than the other way around. Consequently, exploring novel ubiquitous computing systems and applications inevitably requires prototyping not only software but also physical components.* ***Intelligent Objects (Iobs)*** *are hardware and software components for augmenting physical objects with embedded processing and interaction. These components are embedded devices that interact with their environment through a configurable collection of sensors and actuators.*

Keywords*—Intelligent objects, Prototyping, Seamless Integration, Sensors, Ubiquitous Computing*

I. INTRODUCTION

Ubiquitous computing is the method of enhancing computer use by making many computers available throughout the physical environment, but making them effectively invisible to the user. The transition from mainframe to personal computing was marked by ‘human integration’, considering human users no longer as peripheral but as integral in computer applications. Similarly, the current transition from personal to ubiquitous computing is marked by ‘physical integration’, considering the physical world around computer and user as integral part of the overall system.

A far-reaching approach to achieve such physical integration is to embed computing into the objects and artefacts that are subject of everyday activity. The objective of this paper is to develop a range of small, embedded devices as platforms for augmentation and interconnection of artefacts. We describe some of the work we conducted for computer-augmentation of everyday artefacts. We begin with a brief introduction about ubiquitous computing and then elaborate the prototyping of Iobs through a series of applications demonstrating the incorporation of intelligence to the artefacts.

II. MOTIVATION

1. Ubiquitous Computing

As pointed out by Weiser and Brown [1],”Ubiquitous Computing is fundamentally characterized by the connection of artefacts in the real world with computation”. Artefacts are commonly defined as ‘something created by human for a practical purpose’ and it is compelling to build on these familiar purposes while enabling new applications on the basis of embedded computing and communication. Artefacts thus augmented become intelligent objects that can be tied directly into software processes to overcome the media break between physical flow of activity and related flow of information.

In the emerging interactive environments, intelligent objects may embody physical I/O to be enabled as tangible user interface objects that facilitate richer interactions between people and their environments. Computer and communication hardware has become so small and inexpensive to consider their embedding in everyday objects. As a consequence it is expected that networked intelligent objects will give rise to new types of application and in particular such that are more tightly coupled with activity in the physical world.

New hardware systems design for ubiquitous computing has been oriented towards experimental platforms for systems and applications of invisibility. New chips have been less important than combinations of existing components that create experimental opportunities.

2. Related Work

The first ubiquitous computing technology to be deployed was the Liveboard [2], which is now a Xerox product. Miniaturization of components is presently reaching a stage at which it becomes practical and affordable to embed processing, networking and physical interaction into even the most mundane objects [3]. This has inspired a range of design examples built over the last years to explore application opportunities and technology design challenges. Few examples are the Mediacup (a coffee-cup that autonomously computes its use context from embedded sensors, and serves itself to potential applications in the local environment [4]), the StrataDrawer (a chest of drawers that tracks its physical contents to provide new forms of user interaction [5]), and the Pin&Play noticeboard (a board that has smart pushpins autonomously asserting priorities to visually alert users [6]). These examples are generally one-off prototypes and can only provide very limited insight into applications and challenges that may emerge with more pervasive networking of smart objects. Investigation of applications that involve a larger number and diversity of smart objects has so far been hindered by the lack of a suitable hardware/software platform.

3. Prototyping of Iobs

3.1 Hardware Prototyping

Iobs are small, embedded context-aware devices that integrate sensing, actuation, processing, and communication (Fig 1). They're customized for each application and exist in many different configurations. The hardware design allows for a large degree of flexibility in terms of the type and number of sensors and actuators. In addition, developers can tailor an Iob to support different means of communication including wired and wireless networks. A flexible hardware design, combined with high-level software abstractions and development tools, lets developers rapidly design new device configurations. The Iob design's flexibility lets developers use an Iob for a variety of purposes.

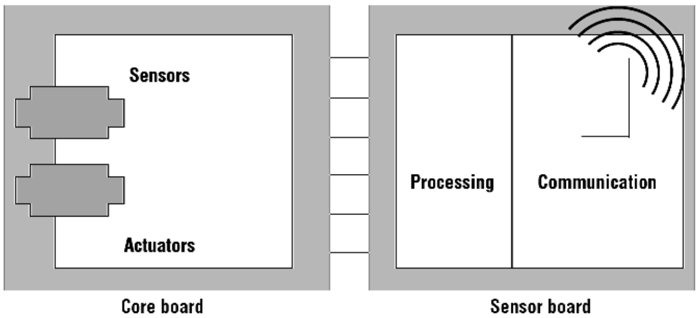
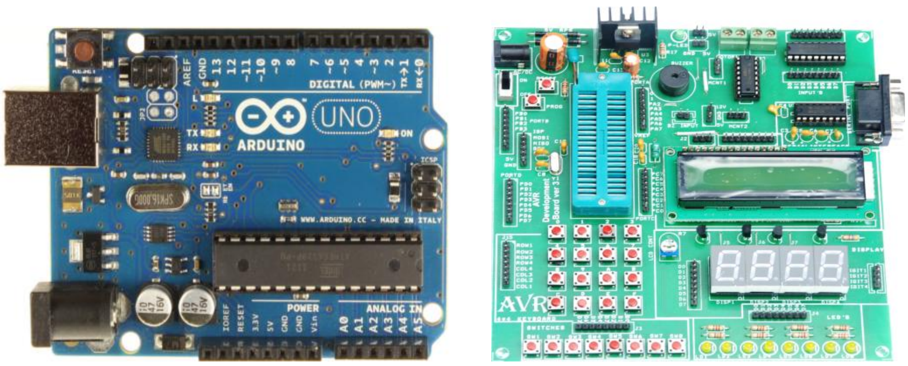


Figure 1 *Iobs Architecture integrate physical I/O devices with a processing environment and wireless Communication*



(a) Arduino UNO Board *b) HPS AVR Board*

Figure 2 Hardware used in our application development

We designed our models based on Arduino UNO board [7] and HPS AVR board V3 [8]. The Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers.

The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer.

3.2 Software Prototyping

Intelligent Objects are designed for embedded and highly applied use. The technology concept foresees that the software executed on the device is task-specific in the sense that is customized for a particular physical object. This involves management of subtasks for physical I/O, communication and processing of events, but it does not involve support for concurrent user-level applications that would compete over system resources. The required system software is provided in libraries and becomes compiled into applications in the development process.

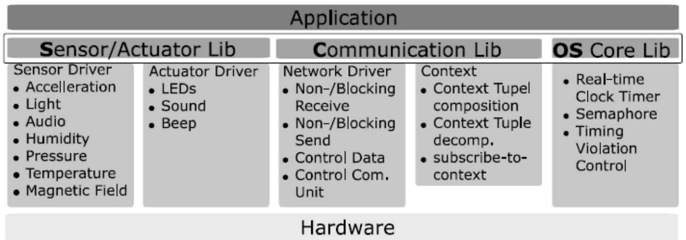


Table 1 Software Interface for Intelligent objects

As shown in Table 1, the system software interface is based on three libraries with device drivers for each hardware component. These provide hardware abstraction but also support fine-grained control, for instance over individual sensors. In addition to libraries for the physical I/O and communication subsystems, basic core functions are provided for coordination of access to resources. The processor platform does not support context switching but separate cooperative tasks are supported with semaphores. Core functionality also includes real time clock and calendar, as well as notification of timing violations.

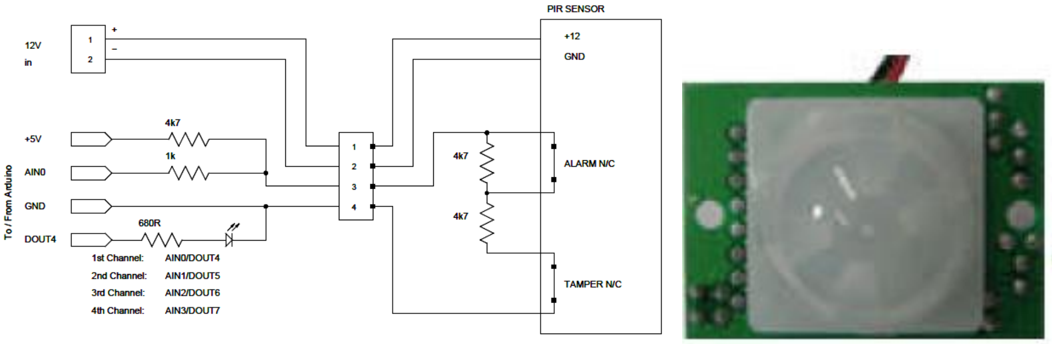
The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. A *SoftwareSerial* library allows for serial communication on any of the Uno's digital pins. The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus. For SPI communication, use the SPI library.

III. IOBS IN REALITY

In this section, we are going to present few Iobs that are handy in our daily lives. We used Arduino and HPS boards for the applications development. The Arduino Uno can be programmed with the Arduino software. The ATmega328 on the Arduino Uno comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol.

1. Intelligent Stroller

Most of the women are busy with their house-hold works. A baby in a pram or stroller should not be left unattended - as the pram or stroller may tip over or may roll-down. This intelligent stroller will alert the mother when there is a movement in the stroller. In this simple project, we'll build a motion-sensing alarm using a PIR (passive infrared) sensor which is connected to the stroller and an Arduino microcontroller (Fig 3).



(a) Circuit Diagram *(b) PIR*

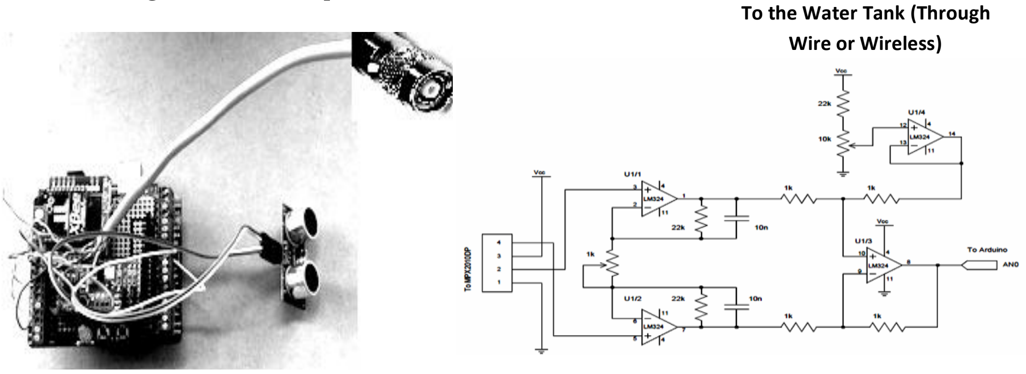
Fig 3 Motion-Sensing Alarm Using A PIR

Whenever there is a movement in the stroller, PIR sends the information to the Arduino which alerts the mother through buzzer.

2. Intelligent Water Tank

Water is a precious resource in many parts of the world, and many people rely on water tanks to supplement their water supply by storing collected rainwater or water pumped from a well or bore. One way of measuring the tank depth is by placing a series of conductive pickups at various heights inside the tank and measure the resistance between them.

This project works a little differently. It uses a device called a differential pressure transducer to measure the water pressure at the bottom of the tank, and from that to calculate how full the tank is. Water pressure increases by about 9.8067kPa per meter of depth so a full tank 2m tall will have a pressure at the bottom of about 19.6134kPa above ambient atmospheric pressure. The "above ambient atmospheric pressure" part is important: it's not enough to simply measure the pressure at the bottom of the tank because varying climate conditions will alter the reading. That's why this project uses a "differential" pressure transducer that has two inlets. By leaving one inlet open to the atmosphere and connecting the other to the bottom of the tank the transducer will output the difference between the two, automatically compensating for varying air pressure and giving a constant reading for constant depth.



(a) Circuit Diagram *(b) Implemention with Arduino*

Fig 4 Intelligent Water Tank

The data sent by the water tank can be collected by arduino either through wire or wireless. Depending on the level, the arduino can turn the buzzer on when the water level decreases below threshold.

3. Intelligent Optical heart rate Pulse Indiactor

This is another application that will help heart patients who are busy at work and cannot recognize the change in their heart-rate. In an optical heart-rate pulse sensor, light is shot into a finger tip or ear lobe. The light either bounces back to a light sensor, or gets absorbed by blood cells. This information can be sent either through a wire or wireless medium to the Arduino (Fig 5), which processes the data and if the beat is abnormal then it signals (either through buzzer or voice) the patient to take a corresponding pill.

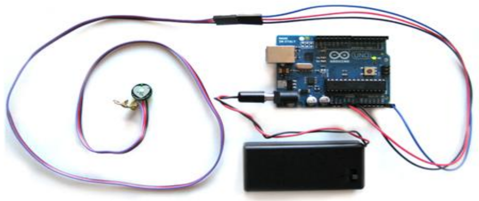


Fig 5 Intelligent Optical heart rate Pulse Indiactor

4. Intelligent Chair

This is another interesting project that will incorporate intelligence to the sitting chair. This chair uses a weighing sensor. The project works in two ways. The first one is ***user mode*** in which arduino maintains data about the weight of a person who regularly uses the chair. The second one is ***general mode*** in which any person can use the chair.

When operated in user-mode, the arduino maintains weight information about the regular user. If the user put up some weight, the chair immediately warns him that he is putting up some weight and displays his weight either through voice or signals (such as buzzer beep indicating weight increase and an LCD or LED 7-segment display to display the weight).

In the general mode, the arduino will not maintain any information. If the person sitting on the chair has a weight greater than the weight that is tolerable by the chair, it will beep.

IV. CONCLUSION

In this paper, we introduced Intelligent Objects (Iobs), a distinct technology concept in which computing is decentralized and placed in the background of physical artefacts. The concept is fully implemented in a platform that has become deployed in domestic area. The concept of Iobs can be extended further and can be implemented in real-time applications in industrial and military areas.

REFERENCES

[1] M. Weiser and J. S. Brown. The Coming Age of Calm Technology, October 1996.
[2] Elrod, Bruce, Gold, Goldberg, Halasz, Janssen, Lee, McCall, Pedersen, Pier, Tang, and Welch. Liveboard: a large interactive display supporting group meetings, presentations and remote collaboration. pp. 599-607. CHI ’92 Conference proceedings. May 1992. ACM, New York, NY.
[3] H.-W. Gellersen, A. Schmidt, M. Beigl. Multi-Sensor Context-Awareness in Mobile Devices and Smart Artifacts, Mobile Networks and Applications (MONET), Kluwer, Oct 2002.
[4] Beigl M., Gellersen H., Schmidt A. MediaCups: Experience with Design and Use of Computer-Augmented Everyday Objects, Computer Networks, Vol.35, No. 4, March 2001, Elsevier, pp 401-409
[5] Siio, T. Digital Decor. http://siio.ele.eng.tamagawa.ac.jp/ projects/decor/index.html
[6] Van Laerhoven, K., Schmidt, A. and Gellersen, H. Pin&Play: Networking Objects through Pins. Proc. Ubicomp 2002, Gothenburg, Sept 2002, pp. 219-229.
[7] Official website of Arduino http://arduino.cc/en/Main/ArduinoBoardUno
[8] Official website of HPS Development boards www.hpselectronics.com /Development Boards

Sahithi Govindaraju is currently pursuing her B. Tech degree in the Department of ECE, Vasireddy Venkatadri Institute of Technology, Nambur, Guntur (dt), A.P., India, affiliated to JNTU, Kakinada. Her areas of interests include Embedded systems, and Linux based device driver development.

Krishna Prasad Satamraju obtained his M. Tech degree from SRM university, with a specialization in Digital Communications and Networking. He is currently working as Assistant professor in the Department of ECE, Vasireddy Venkatadri Institute of Technology, Nambur, Guntur (dt), A.P., India. His research areas include Embedded systems, Linux based device driver development, Mobile Ad hoc Networks.