

Western University

Scholarship@Western

---

Digitized Theses

Digitized Special Collections

---

2008

## Building Blocks for Adaptive Modular Sensing Systems

Anita Jain

*Western University*

Follow this and additional works at: <https://ir.lib.uwo.ca/digitizedtheses>

---

### Recommended Citation

Jain, Anita, "Building Blocks for Adaptive Modular Sensing Systems" (2008). *Digitized Theses*. 4065.  
<https://ir.lib.uwo.ca/digitizedtheses/4065>

This Thesis is brought to you for free and open access by the Digitized Special Collections at Scholarship@Western. It has been accepted for inclusion in Digitized Theses by an authorized administrator of Scholarship@Western. For more information, please contact [wlsadmin@uwo.ca](mailto:wlsadmin@uwo.ca).

# BUILDING BLOCKS FOR ADAPTIVE MODULAR SENSING SYSTEMS

(Thesis Format: Monograph)

by

**Anita Jain**

Graduate Program in Engineering Science  
Department of Mechanical and Materials Engineering

A thesis submitted in partial fulfillment  
of the requirements for the degree of  
Master of Engineering Science

Faculty of Graduate Studies  
The University of Western Ontario  
London, Ontario, Canada

© Anita Jain 2008

THE UNIVERSITY OF WESTERN ONTARIO  
FACULTY OF GRADUATE STUDIES

**CERTIFICATE OF EXAMINATION**

**Chief Adviser:**

\_\_\_\_\_  
Dr. Michael D. Naish

**Examining Board:**

\_\_\_\_\_  
Dr. Ralph O. Buchal

**Advisory Committee:**

\_\_\_\_\_  
Dr. Samuel F. Asokanthan

\_\_\_\_\_  
Dr. Rajni V. Patel

The thesis by  
**Anita Jain**

entitled:

**Building Blocks for Adaptive Modular Sensing Systems**

is accepted in partial fulfillment of the  
requirements for the degree of  
**Master of Engineering Science**

Date: \_\_\_\_\_

\_\_\_\_\_  
Chair of Examining Board  
Dr. John R. Dryden

# Abstract

This thesis contributes towards the development of systems and strategies by which sensor and actuator components can be combined to produce flexible and robust sensor systems for a given application. A set of intelligent modular blocks (building blocks) have been created from which composite sensors (made up of multiple sensor and actuator components) can be rapidly reconfigured for the construction of Adaptive Modular Sensing Systems. The composite systems are expected to prove useful in several application domains including industrial control, inspection systems, mobile robotics, monitoring and data acquisition.

The intelligent building blocks, referred to as transducer interface modules, contain embedded knowledge about their capabilities and how they can interact with other modules. These modules encapsulate a general purpose modular hardware architecture that provides an interface between the sensors, the actuators, and the communication medium. The geometry of each transducer interface module is a cube. A connector mechanism implemented on each face of the module enables physical connection of the modules. Each module provides a core functionality and can be connected to other modules to form more capable composite sensors. Once the modules are combined, the capabilities (e.g., range, resolution, sample rate, etc.) and functionality (e.g., temperature measurement) of the composite sensor is determined and communicated to other sensors in the environment.

For maximum flexibility, a distributed software architecture is executed on the blocks to enable automatic acquisition of configuration-specific algorithms. This logical algorithm imparts a collective identity to the composite group, and processes data based on the capabilities and functionalities of the transducers present in the system. A knowledge representation scheme allows each module in the composite group to store and communicate its functionality and capabilities to other connected modules in the system.

# Acknowledgements

This is a wonderful opportunity to thank the many people who made this thesis possible. First and foremost, I wish to extend my appreciation and gratitude to my supervisor Dr. Michael D. Naish for all his encouragement, expert guidance, keen supervision and highly interactive approach throughout my graduate program. It has been a great learning experience working under his supervision.

The financial support of Natural Sciences and Engineering Research Council of Canada (NSERC) is gratefully acknowledged.

I thank my colleagues, Andrew Lyle and Christopher Ward, at the Sensing and Mechatronic Systems (SAMS) Laboratory for all their support. It was an excellent experience coordinating with Andrew during the course of this project. Thanks to Chris for his advice and help with SolidWorks Models.

Thanks to my friends in London for making my stay an enjoyable experience. A very special thanks to Madhusudanan K Sampath for being a great friend and mentor during some of the most stressful times of my research work. Shilpi Goel deserves a special mention. Her positive outlook towards life is always inspiring.

Thanks to everyone at Electronics Shop at The University of Western Ontario for their help and invaluable advice on electrical designs.

Last but not the least, I would like to thank my family for all their support throughout my studies. I wish to extend my deepest gratitude to my parents, Vimal and Leela Jain, who have taught me the invaluable lesson of never giving up. Words fall short to express my thanks to them for their unfailing love, support, encouragement, understanding, and patience. Thanks to my sister Shweta for her support and encouragement. She provided me with the much needed entertainment during stressful times.

# Contents

<b>Certificate of Examination</b>	<b>ii</b>
<b>Abstract</b>	<b>iii</b>
<b>Acknowledgements</b>	<b>iv</b>
<b>Table of Contents</b>	<b>v</b>
<b>List of Figures</b>	<b>ix</b>
<b>List of Tables</b>	<b>xi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Sensing Technologies . . . . .	2
1.2 Wireless Sensing Networks . . . . .	3
1.3 Reconfigurable Sensing Systems . . . . .	4
1.4 Potential Benefits of Adaptive Modular Sensing Systems . . . . .	5
1.5 Project Scope and Objectives . . . . .	6
1.6 Contributions of Thesis . . . . .	7
1.7 Thesis Outline . . . . .	8
<b>2 Literature Review</b>	<b>10</b>

---

2.1	Sensor Classification . . . . .	10
2.2	Recent Trends in Sensors and Sensor Technology . . . . .	11
2.2.1	Smart Sensors . . . . .	11
2.2.2	Standards for Smart Sensors . . . . .	13
2.3	Sensor/Actuator Buses . . . . .	19
2.4	Existing Modular Sensing Systems . . . . .	20
2.5	Building Blocks for Sensing Systems . . . . .	23
<b>3</b>	<b>System Design</b>	<b>28</b>
3.1	Introduction . . . . .	28
3.2	Design and Implementation of the Hardware Architecture . . . . .	29
3.2.1	Controller/Communication Layer . . . . .	31
3.2.2	Face Connector Interface Layer . . . . .	37
3.2.3	Power Layer . . . . .	41
3.2.4	Transducer Interface Layer . . . . .	41
3.3	Geometry . . . . .	45
3.4	Face Connectors . . . . .	49
3.4.1	Mechanical Design . . . . .	49
3.4.2	Electrical Design . . . . .	51
3.4.3	Face Connector Protocol . . . . .	54
3.5	Application Programming Interface (API) . . . . .	57
3.6	Additional Modules . . . . .	58
3.6.1	Interconnect Modules . . . . .	58
3.6.2	Power Module . . . . .	59
3.6.3	Administration Module . . . . .	60
3.7	Knowledge Representation Scheme . . . . .	61

---

3.7.1	Basic TEDS . . . . .	62
3.7.2	Templates . . . . .	63
3.7.3	Interconnect TEDS . . . . .	65
3.8	Summary . . . . .	66
<b>4</b>	<b>System Validation</b>	<b>67</b>
4.1	Formation and Operation of a Composite System . . . . .	68
4.1.1	Experimental Set-up . . . . .	68
4.1.2	Discussion . . . . .	70
4.1.3	Summary . . . . .	71
4.2	Rapid Reconfigurability Feature . . . . .	75
4.2.1	Experimental Set-up . . . . .	76
4.2.2	Discussion . . . . .	77
4.2.3	Summary . . . . .	88
<b>5</b>	<b>Conclusions and Recommendations</b>	<b>89</b>
5.1	Summary and Conclusions . . . . .	90
5.2	Recommendations . . . . .	92
	<b>References</b>	<b>99</b>
	<b>Appendices</b>	<b>104</b>
<b>A</b>	<b>Transducer Electronic Datasheets (TEDS)</b>	<b>104</b>
A.1	Accelerometer TEDS . . . . .	105
A.2	LDR TEDS . . . . .	106
A.3	Servo motor TEDS . . . . .	107
A.4	Servo Control Template TEDS . . . . .	108
A.5	LCD TEDS . . . . .	110



---

A.6 LCD Merge Template TEDS . . . . .	111
<b>Curriculum Vitae</b>	<b>113</b>

# List of Figures

2.1	Model of Smart Sensor. . . . .	12
2.2	1451 family. . . . .	13
2.3	IEEE 1451 logical interfaces. . . . .	16
2.4	Block diagram of IEEE 1451.2 Smart Transducer Interface Module. . . .	17
3.1	Block diagram of the hardware architecture. . . . .	30
3.2	Various layers of the hardware stack. . . . .	31
3.3	Hardware stack. . . . .	32
3.4	Header board for LPC2148. . . . .	34
3.5	Block diagram of hardware architecture employing a multi-microcontroller system. . . . .	36
3.6	Module controller-transceiver interface. . . . .	37
3.7	Logic symbol of bilateral switch. . . . .	38
3.8	Block diagram of face connector interface layer. . . . .	39
3.9	Face connector interface board. . . . .	40
3.10	Block diagram of power layer. . . . .	41
3.11	Power layer board. . . . .	42
3.12	Block diagram of transducer interface layer. . . . .	43
3.13	Transducer interface layer. . . . .	45

3.14 Geometry of the module. . . . .	47
3.15 Side face of module. . . . .	47
3.16 Top face of module. . . . .	48
3.17 Bottom face of module. . . . .	48
3.18 Ideal design of the face clip. . . . .	50
3.19 Electrical connection mechanism. . . . .	52
3.20 Face connector. . . . .	54
3.21 Modules connected via face connectors. . . . .	55
3.22 Angled interconnect. . . . .	59
3.23 Translational interconnect. . . . .	60
3.24 An assembly of modules using interconnects . . . . .	61
4.1 LDR potential divider network. . . . .	69
4.2 Composite group consisting of accelerometer, LDR, and LCD. . . . .	72
4.3 Modules connected in a side-by-side configuration. . . . .	78
4.4 32 characters $\times$ 4 lines display system (Module A on left). . . . .	79
4.5 32 characters $\times$ 4 lines display system (Module B on left). . . . .	81
4.6 Modules connected in an above-and-below configuration. . . . .	82
4.7 16 characters $\times$ 8 lines display system (Module A above). . . . .	84
4.8 16 characters $\times$ 8 lines display system (Module B above). . . . .	86
5.1 Supervisory circuit with backup-battery. . . . .	94
5.2 Camera interface. . . . .	96
5.3 Limitation of the prototype face connector design. . . . .	97

# List of Tables

2.1	Sensor classification schemes. . . . .	11
2.2	Sensor Actuator Bus review. . . . .	18
3.1	Truth table of the bilateral switch. . . . .	38
3.2	Relative orientation and face clip connections. . . . .	57
3.3	Basic TEDS content . . . . .	63
3.4	AMSS TEDS content . . . . .	63
3.5	Examples of Standard template fields . . . . .	64
3.6	User defined template fields . . . . .	65
3.7	Interconnect TEDS content . . . . .	65
4.1	Side-by-side configuration (Module A on left). . . . .	78
4.2	Side-by-side configuration (Module B on left). . . . .	80
4.3	Above-and-below configuration (Module A on top). . . . .	83
4.4	Above-below configuration (Module B on top). . . . .	83

# Chapter 1

## Introduction

The sensor market is extremely diverse and is expected to grow to \$43 billion by 2008 [1]. Sensors are used in several industries and markets which include: machinery manufacturers and suppliers, processing industries, aircraft and ship building, construction sector, consumer and other electronics, and automotives, among others [2]. The rapid increase in the interest in sensors has been driven by numerous applications in which sensors can provide a large benefit. The use of sensors has undergone explosive growth due to the availability of processing devices to manipulate output signals and in addition to progress in communications technology. The role of sensors is becoming increasingly important as electronic and computer systems become more autonomous. Sensors are used in many devices and systems to collect parameters of interest or to identify the states of control, resulting in increased built-in intelligence. Accurate and robust sensing becomes critical for proper decision making and control since sensors perform a vital function in the operation of industrial systems and scientific equipment.

## 1.1 Sensing Technologies

Sensing technologies are well established and are widely applied in everyday life and industrial applications. The development of sensing technologies and integration between them is enabling major advances to be made in a wide range of sensor applications. In recent years, with the use of micro and nano technologies new families of semiconductor sensors are emerging such as smart and soft sensors and microelectromechanical systems (MEMS). A great deal of emphasis has been placed on the development of MEMS and smart sensors since the overlapping of these two technologies will enable innovation. With the emergence of the new generation of sensors, there will be a considerable change in how measurements are made, resulting in new approaches in instruments and instrumentation.

To utilize sensors in an increasingly large number of applications, it is required to keep track of sensor properties such as identification, calibration factors, correction/compensation factors, location, type, data sheets etc. Sensor signal data becomes useless if the sensor cannot be identified, verified to be operating within the manufacturers recommended signal range, or validated as operating within an allowed environment. Smart sensors provide many features required in order to further expand the application of sensing technologies in a cost effective manner. Some features of smart sensors include: electronic data sheets, self identification, smart calibration and compensation, digital sensor data, sensor communications for remote monitoring and remote configuration.

The IEEE 1451.4 smart sensor standard marked a huge advance in sensor technology and has been applied in various research and industrial applications. This standard combines the true robustness and cost-effectiveness of analog sensors with the intelligence of digital equipment. One of the most important aspects of IEEE 1451.4 is that it offers a standard interface and protocol by which a sensor can describe itself over a network.

## 1.2 Wireless Sensing Networks

Recent technological advances in micro electro-mechanical systems (MEMS) and integrated wireless sensing and communication are enabling dense wireless sensor networks to be realized. The emerging field of wireless sensor networks offer great promise for information capture and processing in both commercial and military applications. The extremely lightweight data gathering networks from Crossbow Technology [3], are the most popular in the field of the wireless sensor networks. Wireless sensor networks combine sensing, computation, and communication into a single device. These sensor networks involve deploying a large number of low cost small nodes which could either have a fixed location or be randomly deployed in order to sense changes in the environment and report them to other nodes in the network architecture. Sensors usually communicate with each other using a multi-hop approach. The data flow ends at nodes, which are generally called base stations (also known as sinks), that link the sensor network to other networks to disseminate the data sensed for further processing. The base stations are more capable than simple sensor nodes since they must do complex data processing. Sensor nodes can be deployed in hostile environments or over large geographical areas. Sensor networks have been useful in several domains including: environmental sensing, military monitoring, building monitoring, and health care.

A major challenge in the field of wireless sensing networks is that of power consumption. The overall power consumption affects the communication between nodes. To reduce power consumption, aggregation points are introduced in the network. This reduces the total number of messages transferred between nodes and saves some energy. These aggregation points are regular nodes that receive data from neighboring nodes, perform some kind of processing, and then forward the processed data to the next hop. In a similar method, sensor nodes are organized into clusters, each cluster having a cluster head as the leader. The data exchange within a cluster must travel through the cluster head,

which then is forwarded to a neighboring cluster head until it reaches its destination, the base station. Another method for saving energy is setting the nodes to go idle if they are not needed and wake them up when required.

A wireless sensor network is very similar to an ad-hoc wireless network in the sense that it employs the same multi-hop communication method. However, hundreds or thousands of nodes are often deployed in a sensor network, while an ad-hoc network usually involves only a few nodes. The nodes in a wireless sensor network are more restricted in terms of computational, energy and storage resources than ad-hoc network. Sensor nodes can be deployed in environments without the need for human intervention.

### 1.3 Reconfigurable Sensing Systems

Reconfigurable sensing systems have gained popularity in recent years. These systems allow the users to construct custom solutions using physical building blocks. A common feature of these building blocks is the ability to process and communicate with each other (using either wireless or wired communication). The functionality of the system is based on the physical connection between the blocks. Flexibility is a key feature of reconfigurable sensing systems. The functionality of the system can be modified or altered by connecting the blocks in different combinations.

Some physical building blocks available are I-BLOCKS [4] and Intelligent Artefacts (I-Artefacts) [5] (a Danish National Research Council project). These provide end-users with a direct approach to design new artefacts. They have led to the development of several generations of building blocks of which the first generation include light dependent resistors (LDRs), microphones, potentiometers, switches, servo motors, DC motors, LEDs, sound generators, etc. The second generation developed more complex building blocks including the multi-infrared sensors, ultrasound, and accelerometers. eBlocks [6, 7],



a set of electronic blocks, are another type of building blocks best suited for developing basic sensor systems involving logical transformations (e.g., AND, OR, NOT) and/or basic state functions (e.g., prolong, toggle, trip). Other systems include the modular system consisting of self-describing building blocks from the Mitsubishi Electric Research Laboratories (MERL). The implementation of BUG devices [8] is a more recent contribution in this area and deals with a collection of easy-to-use electronic modules that snap together in order to build gadgets as per user requirements. A detailed description of the functionality and components for each of these building blocks can be found in Chapter 2.

Most of the existing reconfigurable sensing systems such as [4, 5, 6, 7] have limited processing capabilities and are restricted to very few sensors and actuators. Hence, these systems can only be reconfigured for simple applications. On the other hand, recent efforts such as [8] feature a more complex processing platform and may be reconfigured for more complex applications. Nevertheless, a key limitation of these systems is the restriction of the number of building block connection combinations due to the limitation of the connector mechanism employed. In order to overcome this difficulty, the building blocks of the adaptive modular sensing discussed herein are designed in such a way that the system modules may be connected in a large number of different configurations.

## 1.4 Potential Benefits of Adaptive Modular Sensing Systems

Adaptive modular sensing systems can be widely used for a broad range of applications ranging from data acquisition systems to complex industrial applications. The sensing systems are adaptive because the functionality of a system can be changed by the simple addition or removal of building blocks. A key feature of the system is that it can be

very rapidly reconfigured for various applications of interest. In addition, to increase flexibility, the physical blocks can be connected in a number of different configurations. The design and implementation of a general-purpose hardware architecture provides an interface to a multitude of sensors and actuators, and also manages communication with other modules in the system. A distributed software architecture executed on the building blocks, enables execution of application specific system algorithms in order to allow the modules to be reconfigured for complex applications. Due to modular nature of the hardware and software architectures, the system can be easily extended to incorporate new technologies with minimal effort.

## 1.5 Project Scope and Objectives

The main focus of this research is to develop systems and strategies by which sensor and actuator components can be combined to produce flexible and robust sensor systems for a given application. A set of intelligent modular blocks (building blocks) will be created from which composite sensors (made up of multiple sensor and actuator components) may be rapidly reconfigured for different tasks. These components will contain embedded knowledge about their capabilities and how they can interact with other modules. Once the modules are combined, the capabilities of the composite sensor may be determined and communicated to other sensors in the environment. The system is robust since it is able to detect failures within the hardware and software architectures, as well as the addition or loss of modules in a group. For maximum flexibility, a distributed software architecture [9] will be implemented on the blocks to enable automatic execution of configuration-specific algorithms. Potential applications of composite sensing systems include industrial control, inspection systems, mobile robotics, monitoring and data acquisition.

The specific objectives of this research are outlined below:

1. To design and develop a modular hardware architecture that can be interfaced to a multitude of sensors and actuators.
2. To integrate the modular hardware platform with an independent distributed software control architecture.
3. To develop modules that encapsulate the hardware architecture and form the basic building blocks of an adaptive modular sensing system.
4. To design and develop a face connector mechanism to enable the physical connection of the transducer interface modules.
5. To implement schemes to enable a module to determine its position and orientation with respect to other modules present in the composite system.
6. To specify a knowledge representation scheme to allow the various modules in a composite system to communicate standard information required by a logical algorithm to assign a collective identity to the group.
7. To design and develop additional modules which may be required to make composite system completely functional.

## 1.6 Contributions of Thesis

The main contribution of this thesis is the design and development of a set of intelligent building blocks called transducer interface modules. These building blocks can be rapidly reconfigured for a given application. A modular hardware architecture, encapsulated in the transducer interface modules, is designed and implemented. A key contribution is the design and implementation of a face connector mechanism that enables the physical

connection between modules in four different orientations ( $90^\circ$  apart). Also, a knowledge representation scheme is implemented to enable the modules to communicate their capabilities and functionalities.

## 1.7 Thesis Outline

The organization of this thesis is outlined below:

Chapter 1 Introduction: Addresses the motivation, project scope, and objectives of this thesis.

Chapter 2 Literature Review: This chapter provides a discussion on various topics which include: sensor classification, recent trends in sensors and sensor technology, sensor actuator buses, existing modular sensing systems, and building blocks for sensing systems.

Chapter 3 System Design: Provides a detailed description of the system design. Includes hardware architecture design and electrical and mechanical designs of the transducer interface modules. Also, additional modules are discussed that may be required to make a composite system fully functional. A model of application programming interface is presented that enables the distributed software architecture to communicate with the hardware architecture. A knowledge representation scheme is discussed that enables the module to communicate its functionality (e.g., temperature measurement) and capabilities (e.g., range, resolution).

Chapter 4 System Validation: Presents various experimental results to demonstrate the features of hardware architecture, transducer interface modules, interconnect modules, and integration of hardware and software architectures.

---

Chapter 5 Conclusions and Recommendations: Concludes the thesis and highlights the contributions of the work. Suggestions for future work and improvements to the system are also provided.

# Chapter 2

## Literature Review

This chapter provides literature from a variety of topics related to the field of sensors, latest trends in sensor technologies, and existing sensing systems. The chapter is organized as follows. Various sensor classification schemes are discussed in 2.1. An overview on smart sensors including the IEEE 1451 standard for smart sensors is provided in Section 2.2. Section 2.3 outlines the commonly used sensor/actuator buses. A number of existing modular sensing platforms are discussed in Section 2.4. Section 2.5 presents a number of physical building blocks that can be reconfigured for different applications.

### 2.1 Sensor Classification

There are an estimated 50,000 to 100,000 different commercial sensors available on the market [10]. Diverse approaches have been adopted for classifying sensors due to the absence of a universal standard. Sensors can be classified based on physical process properties, sensor operational principles, or both [11]. One common method is based on determining the nature of the sensed parameter (such as temperature, pressure, etc.). A more popular approach is based on the nature of the output signals of sensors (analog and digital sensors). A flexible and comprehensive categorization scheme, useful for describing

Table 2.1: Sensor classification schemes.

Sensor classification scheme	Sensor type
Power supply requirements	Active sensors, Passive sensors
Nature of the output signal	Analog sensors, Digital sensors
Measurand	Environmental/Analytical, Level flow, Mechanical( Pressure/Strain), Position/Displacement, Optics/Images, Radiation/Heat flux, Sound/Vibration, Temperature
Signal conditioning circuits	Amplification, Level translation Galvanic isolation, Impedance transformation Linearization, Filtering
Measurement operation modes	Deflection mode, Null mode

and comparing sensors, is presented in [12]. An overview of the most commonly used sensor classification schemes is presented in Table 2.1.

## 2.2 Recent Trends in Sensors and Sensor Technology

The use of micro and nano technologies has led to the emergence of new families of semiconductor sensors such as smart and soft sensors and microelectromechanical systems (MEMS). MEMS technologies allow sensors to be miniaturized and for integration of sensor elements with microelectronic functions in minimal space. Also, it is possible to mass produce sensors more cost-effectively with the help of MEMS technologies [13].

### 2.2.1 Smart Sensors

The Institute of Electrical and Electronics Engineer (IEEE) defines a smart sensor as a sensor “that provides functions beyond those necessary for generating a correct representation of a sensed or controlled quantity. This function typically simplifies the integration of the transducer into applications in a networked environment” [14].

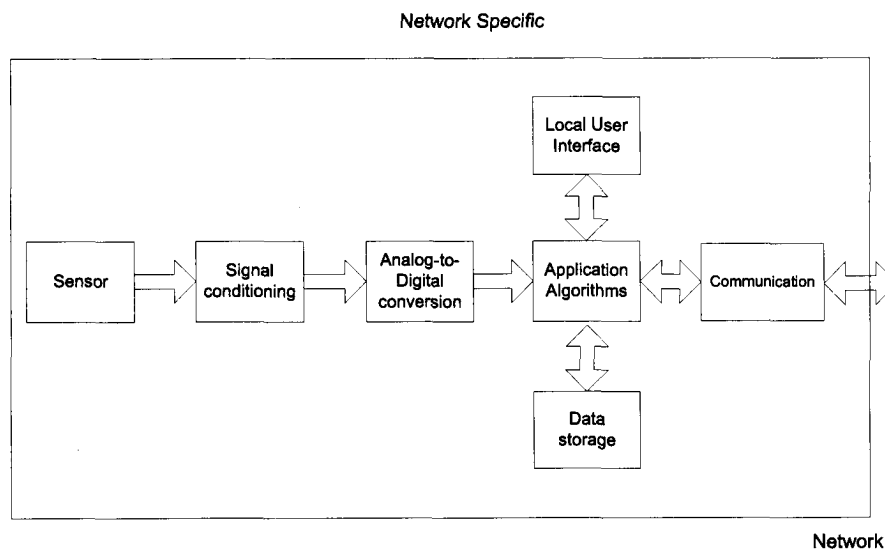


Figure 2.1: Model of Smart Sensor.

Smart sensors combine sensor, information processing, and communications technology [15]. These sensors contain local intelligence that equip sensors with advanced functionalities apart from sensing raw data only. Smart sensors make decisions based upon the data and commands received [16]. A decision may include driving an actuator, sending information back to the host, or communicating with other sensors. Smart sensors use a variety of communication protocols such as serial peripheral interface (SPI), integrated circuit (I<sup>2</sup>C), and Internet-based communication. A general model of smart sensor is shown in Figure 2.1.

The different modules of the smart transducer model can be grouped into functional units. The transducer signal conditioning and conversion modules can be grouped into a building block called a Smart Transducer Interface Module (STIM). Likewise, the application algorithm and network communication modules can be combined into a single entity called a Network Capable Application Processor (NCAP).

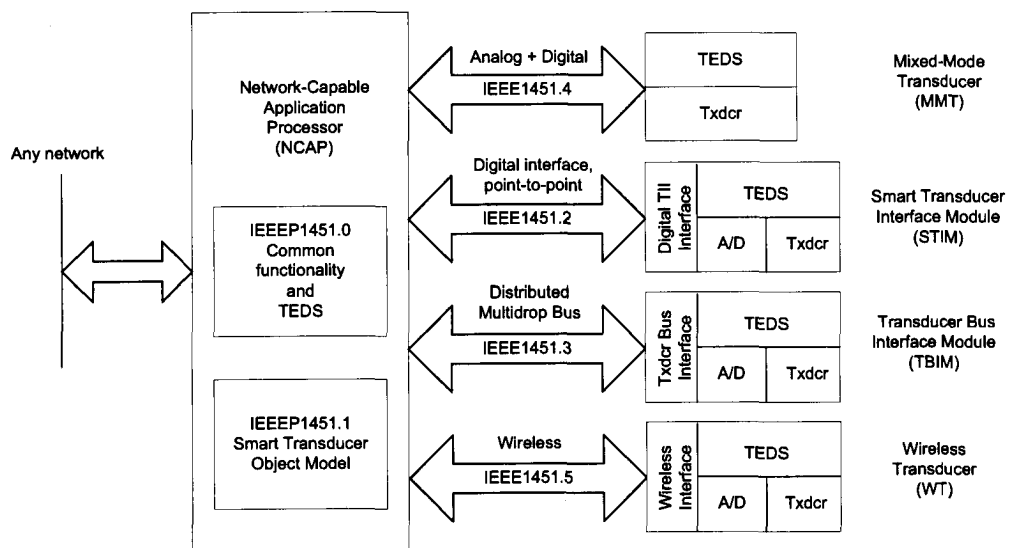
The main goal of smart sensor development is to improve the reliability and durability of these sensors [13]. The smart sensors should easily adapt to new functions and conditions during the operating phase.



### 2.2.2 Standards for Smart Sensors

The rapid development and emergence of smart sensor and field network technologies have made the networking of smart transducers a very economical and attractive solution for a broad range of measurement and control applications.

In order to standardize the smart sensor interface, the Technical Committee on Sensor Technology of the IEEE's Instrumentation and Measurement Society proposed a family of IEEE 1451 Standards, Figure 2.2. These standards specify a set of interfaces for connecting transducers to instruments, microprocessors, or field networks. The standards include digital, mixed-mode, distributed multi-drop, and wireless interfaces to accommodate the varying requirements of different sectors of the industry. A key concept in the IEEE 1451 standards is the Transducer Electronic Data Sheets (TEDS) which uses a standardized data format to represent device-specific information about the sensor such as manufacturer name, sensor types, serial number, and calibration data.



TII = Transducer Independent Interface  
Txdc = Transducer (Sensor or Actuator)

Figure 2.2: 1451 family.

The IEEE 1451 standard consists of eight parts as follows:

IEEE P1451.0: Common Functions, Communication Protocols, and Transducer Electronic Data Sheet (TEDS) Formats.

IEEE 1451.1: Network Capable Application Processor (NCAP) Information Model for Smart Transducers.

IEEE 1451.2: Transducer to Microprocessor Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats.

IEEE 1451.3: Digital Communication and Transducer Electronic Data Sheet (TEDS) Formats for Distributed Multidrop Systems.

IEEE 1451.4: Mixed-mode Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats.

IEEE P1451.5: Wireless Communication and Transducer Electronic Data Sheet (TEDS) Formats.

IEEE P1451.6: High-speed CANopen-based Transducer Network Interface for Intrinsically Safe and Non-intrinsically Safe Applications.

IEEE 1451.7: Transducers to Radio Frequency Identification (RFID) Systems Communication Protocols and Transducer Electronic Data Sheet Formats.

The following sections provide an overview of the various IEEE 1451 standards.

The IEEE P1451.0 standard defines a common set of functionality, which is independent of the physical communications media, that facilitates the interoperability among the various IEEE 1451 standards. The specified functionality includes the basic functions required to control and manage smart transducers, common communications protocols, and media-independent Transducer Electronic Data Sheet (TEDS) formats.

The IEEE 1451.1 standard establishes a Common Object Model description for networked transducers, defining the flexible, modular assembly of the network interface, measurement and control functions and transducer interface, in a NCAP. Thus, any transducer may be connected to any network with an appropriately configured NCAP. The IEEE 1451 logical interfaces are shown in Figure 2.3. The networked smart transducer object model provides two interfaces. The first interface is to the transducer block which encapsulates the details of the transducer hardware implementation within a simple programming model. This makes the sensor or actuator hardware interface look like an input/output (I/O) driver. The second interface is to the NCAP block which encapsulates the details of the different network protocol implementations behind a small set of communications methods.

The IEEE 1451.2 standard defines a transducer-to-microprocessor, serial communication protocol, allowing any transducer, or group of transducers, to receive and send digital data using a common interface. Any transducer can be adapted to the 1451.2 protocol with a Smart Transducer Interface Module (STIM). This standard introduced the concept of a TEDS attached to the transducer that stores transducer identification, calibration, and correction data. All communications between the STIM and the NCAP occurs through a ten wire digital interface known as the Transducer Independent Interface (TII). The TII is a 10 wire digital bus that incorporates a synchronous serial interface along with other lines for handshaking and support. The block diagram of IEEE 1451.2 is shown in Figure 2.4.

The IEEE 1451.3 defines a transducer bus for connecting transducer modules to a NCAP in a distributed multidrop fashion. Multiple transducer modules, called Transducer Bus Interface Modules (TBIM), can be connected to a NCAP via the bus.

The IEEE 1451.4 defines a mixed-mode transducer interface (MMI), which is used for connecting transducer modules, Mixed-mode Transducers (MMT), to an instrument, a

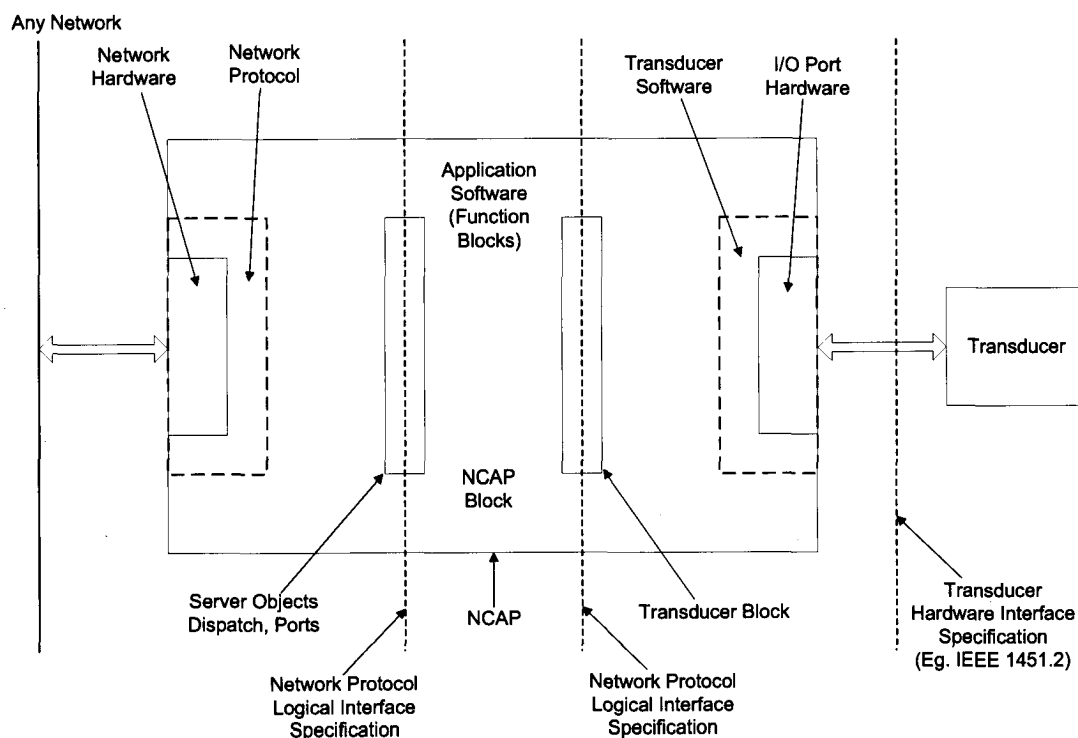


Figure 2.3: IEEE 1451 logical interfaces.

computer, or a NCAP.

The IEEE P1451.5 standard defines a wireless sensor communication interface standard that will leverage existing wireless communication technologies and protocol. The physical communication protocol(s) considered are IEEE 802.11 (WiFi), IEEE 802.15.1 (Bluetooth), IEEE 802.15.4 (WPAN-LR), and ultra wideband.

The IEEE P1451.6 standard establishes a CANopen-based network for multi-channel transducer modules and defines a safe CAN physical layer. It defines the mapping of IEEE 1451 transducer electronic data sheets (TEDS) to CANopen dictionary entries, as

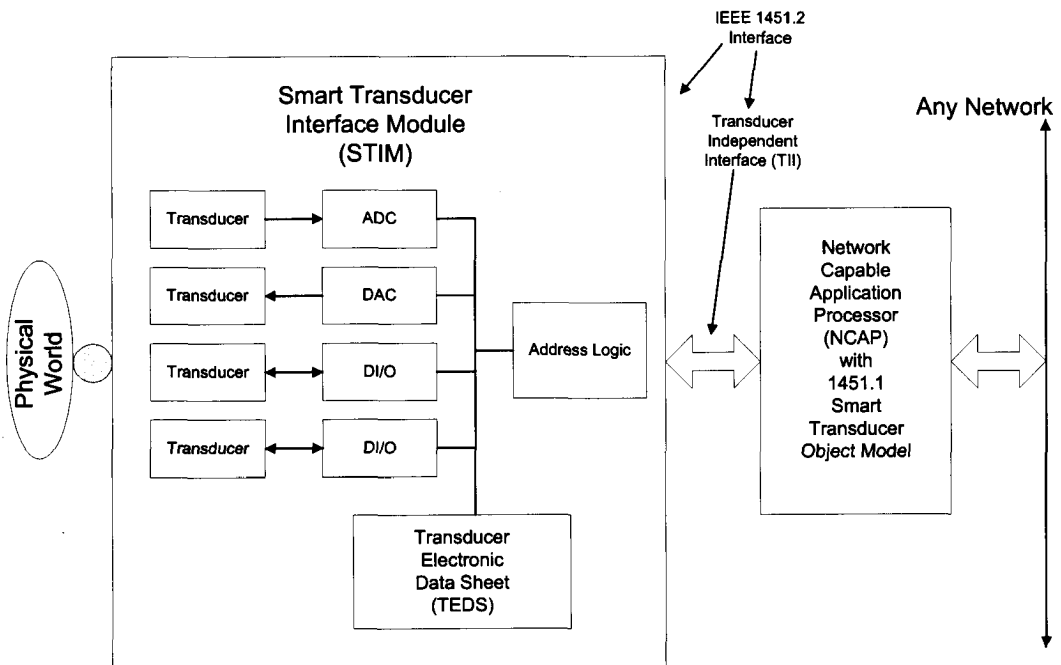


Figure 2.4: Block diagram of IEEE 1451.2 Smart Transducer Interface Module.

well as communication messages, process data, configuration parameter and diagnosis information. The proposed standard will allow development of lean gateways and cascaded transducer networks that combines the specifications of IEEE 1451 and CANopen.

The IEEE P1451.7 standard defines the communication methods and data formats for transducers communicating with RFID tags that follow the ISO/IEC 24753 standard. The purpose of this standard is to facilitate the interfacing and integration of transducers to the RFID tags and reporting the transducer data within the RFID infrastructure. The standard also provides a means for device and equipment interoperability.

Table 2.2: Sensor Actuator Bus review.

Type	AS-I	Interbus	CANBus	I <sup>2</sup> C	SPI
Developer	AS-Interface Consortium (Germany)	Phoenix	Intel and Bosch	Philips	Motorola
Cable	2-wire Unshielded (Yellow)	4-wire green	2-wire Balanced running over (STP)/(UTP)/ Ribbon cable	2-wire	4-wire
Connector	Via vampire taps or screw connectors	9 pin sub D-connector	9 pin D-connector		
Topology	Tree, Bus, Star	Bus, Tree, Star	Tree	Daisy chain	Daisy Chain
Signals	AS-I Specific	According to RS485	CANBus specific	SCL, SDA	MOSI, MISO, SS, SCL
Speed	167 kbps	500 kbps, 2 Mbps(max)	1 Mbps-30 m 500 kbps-100 m 250 kbps-250 m 125 kbps-500 m 62.5 kbps-1 km	100 kbps, 400 kbps 3.4 Mbps	10 Mbps
Maximum Length	300 m (with repeaters)	13 km	1 km	Maximum bus capacitance (400 pF) sets the length	
Maximum nodes	31 (V2.0) 62 (V2.1)	512	112 with 120 $\Omega$ termination	Maximum bus capacitance (400 pF) sets the number	

## 2.3 Sensor/Actuator Buses

A sensor/actuator bus (SAB), or sensor bus is defined as: "An interface specification that defines a common set of behavior and digital communications protocol to which smart [and dumb] sensors [and actuators] must conform if they are to remain (and function) within a well defined distributed control environment" [17]. Sensor/actuator buses are utilized for a variety of applications including controls, automotive, building automation, etc.

A sensor bus provides a consistent standard interface between actuators and sensors and the equipment's controller (computer) using a bus (or ring) interconnect. All data and control information are digitally encoded and transmitted along with the appropriate address and error correction information. A serial bus structure is the most efficient realization among different bus structures. The use of a bus results in a reduction in the amount of wiring required and the standardization of the electrical and physical interconnection [18]. This can lead to improved reliability due to a reduction in the number of connectors. The standard interface results in fewer wiring errors and supports interoperable components. Table 2.2 compares popular sensor actuator buses. Fieldbus is a generic-term which describes a digital communications network consisting of a bi-directional, multidrop, serial-bus, communications network used to link devices, such as controllers, transducers, actuators and sensors. AS-Interface, Interbus, CANbus, are few types of fieldbus. Other popular examples include: DeviceNet, LonWorks, PROFIBUS, etc.

The main goal for sensor bus development is to facilitate adaptation to a variety of communication networks and control systems. Standardized interfaces allow sensors to be connected to the network easily as a result. The installation in complex measuring and automation systems can be simplified, the intelligence of the system can be decentralized, and sensors can be used in multiple ways by various control systems [13].

## 2.4 Existing Modular Sensing Systems

In recent years, sensor systems have become extremely popular for an array of applications. Several embedded solutions have enabled the implementation of a broad range of modular reconfigurable sensing systems.

One of the best known modular sensing systems is the extremely lightweight data gathering wireless sensor network from Crossbow Technology [3]. A typical sensor network consists of dozens, hundreds, or even thousands of tiny, battery-powered computers, called Motes. The Motes monitor and collect environmental information such as light, temperature, humidity, and vibrations. The collected information is then relayed back to a central computing system for further processing via an adhoc network formed by the Motes.

Crossbow provides a hardware platform consisting of wireless Motes, sensor boards and gateways that result in a modular architecture which is optimized specifically for deeply embedded sensing networks. The sensing boards provide capabilities such as ambient light, barometric pressure, GPS, magnetic field, sound, humidity and temperature. The gateway boards include multiple interfaces such as Ethernet, WiFi, USB and serial in order to connect the sensing network to central processing system.

Crossbow's MICA family of wireless sensing network feature several products such as the Berkeley-style MICA Motes [19], implementing the Berkeley TinyOS operating system [20], and more recent ones such as the MICA2, MICAz, IRIS, Imote2. Recent products feature powerful capabilities such as superior software platform, high data rate radios, improved radio range, etc., that enhance the overall functionality of the wireless sensor networks.

All of the Mote modules include a fully programmable microcontroller, a two-way ISM band radio transceiver, and flash memory for over-the-air-programming and data logging of up to 100,000 measurements. The MICA2 Mote [21] consists of a single processor



board MPR400 based on the Atmega ATmega128L and a 868/916 MHz multi-channel transceiver with extended range. The IRIS [22] module utilizes XM2110CA processor based on the Atmel ATmega1281, while the MICAz [23] employs a MPR2400CA processor. The IRIS and MICAz modules work on the global 2.4 GHz ISM band with support for IEEE802.15.4/ZigBee. The Imote2 (IPR2400) [24] is an advanced wireless sensor node platform built around the low power PXA271 XScale processor and integrates an 802.15.4 radio (CC2420) with a built-in 2.4GHz antenna.

Crossbow also makes a software development platform called MoteWorks [25], based on the TinyOS 1.1, for wireless sensor network applications. It includes a collection of flexible software packages that enables simple out-of-the-box deployment of sensor systems for monitoring and alerting, as well as to powerful tools to empower custom development of pervasive sensory networks. MoteWorks supports all Crossbow MICA and IRIS series Mote hardware (IRIS, MICAz, MICA2, MICA2DOT) and sensor boards.

Another project sharing several features with Motes is the Smart-Its [26, 27]. Smart-Its are flexible self-contained units consisting of two separate components: a core board with a wireless transceiver and a standard sensor board providing support for rapid assembly. The sensor unit is responsible for data acquisition and has a dedicated processor for sensor control and extraction of generic features. The core unit controls the overall device, application-specific processing, and communication with other Smart-Its. The generic sensor boards can sense audio, light levels, acceleration, humidity, temperature, and pressure. Outputs include a speaker and light emitting diodes (LEDs). The capabilities of the Smart-Its can be extended by interfacing additional sensors and actuators. The Smart-Its prototypes are based on two different microcontroller platforms: Atmel and PIC. The Atmel platform utilizes Bluetooth integration, while the PIC-based platform is used in conjunction with RFM communication.

The Modular Architecture for Sensor Systems (MASS) [28], a more complex sensing

system, provides application flexibility due to its modular hardware and software design approach. MASS utilizes a multiprocessor hardware architecture operating under an intelligent distributed software control. It consists of a modular hardware platform comprised of four different modules: general purpose processor modules (GPPs), containing a powerful CPU or DSP for on-node processing and data fusion; wireless communication units; sensing modules; and power modules. A MASS node can be configured by stacking various modules together. MASS uses an I<sup>2</sup>C bus for inter-module communication and a wireless transceiver for inter-node communication. The distributed control software is integrated into each module on a separate microcontroller. Since the control in MASS is completely distributed, all of the modules can perform their tasks in parallel. The MASS has been implemented on a Cygnal C8051f125 microcontroller running  $\mu$ C/OS-II.

A more recent modular platform similar to the MASS architecture is the mPlatform [29]. The mPlatform is a reconfigurable modular platform that enables real-time processing on multiple heterogeneous processors. It consists of various stackable hardware modules such as general purpose processing boards, radio boards for wireless communication, sensor boards, and power boards for supplying power to a stack of modules. Each board (except the power board) has a local processor, thus enabling real-time event handling. Reconfigurability is supported by a scalable high performance communication bus interface that connects different modules of a node. The bus allows components of an application to span across different processors/modules. The local processor on each module interacts with a parallel bus through a bus controller, implemented in a low-power, high speed complex programmable logic device (CPLD). The CPLDs on the various modules are connected by a 24-bit wide data bus, thus enabling direct communication between any pair of modules on the bus. An asynchronous interface over an 8-bit wide data bus is implemented for the communication between the processor and the CPLD.

Most of the systems discussed above share many features including sensing units, a

core processing unit such as a microcontroller, a wireless transceiver unit, and a software architecture to manage the overall system. These systems also differ from each other due to the applications they were designed for. Motes have limited processing capability, making it extremely difficult to perform complicated in-network fusion or analysis. Similarly, a Smart-Its can only interpret its own sensor values, along with those gathered from other sensor devices, and wirelessly communicate the interpreted values to a more powerful processing unit to handle complex computational tasks. MASS and mPlatform are more capable as they implement a distributed software control architecture in which each module is self-processing.

In general, the majority of these systems focus primarily on sensing, none offer the ability to provide active sensing. That is, although the sensors and processing hardware are modular, no motion capabilities are provided.

## 2.5 Building Blocks for Sensing Systems

Several systems have been developed that enable the users to construct custom solutions using physical building blocks. Based on the physical connection of the blocks, the system assumes a functionality. These systems offer flexibility in that the functionality of the system can be changed by simply connecting the blocks in a different combination. This section provides a review of various physical building blocks.

I-BLOCKS [4] and Intelligent Artefacts (I-Artefacts) [5] (a Danish National Research Council project) provide physical building blocks that provide the end users with a straightforward technique to design new artefacts. Users of the I-BLOCKS and I-Artefacts systems can do “programming by building”. Construction with building blocks results not only in the development of a physical structure, but also in the development of a functionality of the physical structure. The behavior of such systems is defined by the

internal processing capability of each block present in the modular system and the communication capabilities between the blocks. The functionality of the system can be easily changed by manipulating the physical connection between the blocks.

Each block includes the standard functionality of being able to process and communicate and are extended with the addition of sensors and actuators as the input and output building blocks respectively. The electronic circuitry is housed inside a LEGO DUPLO brick. I-BLOCKS utilize a 28 pin, 8-bit CMOS flash PIC16F1876 microcontroller, while I-Artefacts employ a 40 pin, 8-bit CMOS flash PIC16F1876 microcontroller. The blocks communicate serially via four physical connectors (two male connectors at the top and two female connectors at the bottom). The first generation input building blocks included sensors such as light dependent resistors (LDRs), microphones, potentiometers, switches and the output building blocks included servo motors, DC motors, LEDs, sound generators, etc. More capable building blocks were developed in the second generation including multi-infrared sensors, ultrasound, and accelerometers.

eBlocks [6, 7], a set of electronic blocks, are best suited for developing basic sensor systems involving logical transformations (e.g., AND, OR, NOT) and/or basic state functions (e.g., prolong, toggle, trip). eBlock systems can be categorized into two types: integer-based systems that operate on number values (temperature, seconds of time, decibels, etc.) and Boolean-based systems that operate on yes/no values (motion/no motion, light/no light, etc.). Each eBlock has a fixed and particular functionality. The Boolean eBlocks consist of four types of blocks: sensors, logic/state, communication, and output. The sensing blocks include motion sensors, buttons, contact switches, etc. Output blocks include LEDs, beepers, etc. The logic/state block performs logical transformations. The communication block provides wireless point-to-point communication.

Triangles is a physical interface in the form of a construction kit of plastic triangles that connect together both physically and digitally with magnetic, conducting connectors

[30]. Triangles pieces are identical, flat, plastic equilateral triangles, each containing a PIC microprocessor. They can be tiled on a flat surface or used to construct three dimensional forms. When the pieces connect, information about their identities is exchanged, and messages are relayed to a computer. In this way, an application can determine relationships between all of the connected pieces at any time, and specific connections can trigger specific digital events. Triangles utilize conductive magnets in order to attach and remove the pieces.

Another modular system consists of the self-describing building blocks from the Mitsubishi Electric Research Laboratories (MERL). The building blocks form an object-modeling system that self describes the geometric structures into which the blocks are assembled [31]. Each block consists of connectors mounted on a printed circuit board that also accommodates a microcontroller (a PIC16C77), various passive components, and optional transducers and sensors. The blocks in an assembled structure use a distributed algorithm to discover how they are connected to their immediate neighbors and relay this information to a host computer. Each block acts as a self-configuring, store-and-forward computer. The host computer recovers the geometric structure of the assembled blocks from the block connectivity data and knowledge of the shape of each block. Based on recovered geometric structure, the host computer renders various styles to the assembled group. The group can be styled as a literal rendition where blocks resemble lego blocks or decorative rendition where each block is discovered automatically and styled accordingly.

An example of building blocks from the field of mobile robotics includes a reconfigurable modular drive joint [32], that can be used as the basis for building and configuring robotic and automated systems as an interconnected network of individual nodes. Each node represents a single modular joint and includes its own electronics and a set of sensors and actuator. The joints utilize a decentralized control system. A large number of different robot structures can be reconfigured using a small number of the modular joints.

BUG [8] devices from the Bugs Labs is a collection of easy-to-use electronic modules that snap together to build gadgets according to user requirements. BUGmodules add functionality to the BUG device. Each BUGmodule represents a specific gadget function such as a camera, video output, etc. These modules can be connected in different combinations to achieve various functionalities. BUGbase is the foundation of the BUG device. It is a fully programmable Linux computer implemented on the ARM1136JF-S-based microprocessor. The BUGbase also has a tripod mount and houses four connectors for users to combine BUGmodules in various configurations.

In general, the majority the existing reconfigurable sensing systems such as [4, 5, 6, 7] have limited processing capabilities and are restricted to simple sensors and actuators. Hence, these systems are suitable for a limited range of applications. The reconfigurable modular drive joint is also limited to a few sensors and actuators and is useful only for building robotic structures. BUG devices feature a more powerful processing platform and hence have the ability to be reconfigured for more complex applications. However, a key limitation of these systems is the restriction of the number of connection combinations of the building blocks due to limitations of the connector design.

In order to overcome this difficulty, the building blocks of the adaptive modular sensing systems presented herein are designed in such a way that they enable the connection of the modules of the system in different configurations. The objective is to create a set of intelligent modular sensing and actuating components. These modules can be rapidly reconfigured for various applications. The modular components are referred to as Transducer Interface Modules (TIMs). The TIMs each provide a single sensing or actuating function and have embedded knowledge of their processing and communication capabilities. The user has the freedom to connect these blocks in various configurations to create custom solutions. A distributed software architecture [9] executed on the blocks enables automatic acquisition of configuration-specific algorithms, enabling the modules to be

rapidly reconfigured for various applications. Potential applications of composite sensing systems include industrial control, inspection systems, mobile robotics, monitoring and data acquisition.

# Chapter 3

## System Design

### 3.1 Introduction

This chapter presents the design and implementation details of the transducer interface modules (TIMs). The transducer interface modules are a set of intelligent modular building blocks. These modules enable the construction of various sensor and actuator combinations to produce robust and flexible sensor systems for a given application. Each transducer interface module provides a core functionality. A transducer interface module can be connected to other modules to form more capable composite sensors (made up of multiple sensor and actuator components) that can be rapidly reconfigured for different tasks. These components contain embedded knowledge about their processing and communication capabilities. Once the modules are combined, the capabilities of the composite sensor is determined and communicated to other sensors in the environment. For maximum flexibility, a distributed software architecture [9] is executed on the blocks to enables automatic acquisition of configuration-specific algorithms. The composite sensing systems are useful in several application domains including industrial control, inspection systems, mobile robotics, monitoring, and data acquisition.



The organization of this chapter is as follows. Section 3.2 provides the design and implementation details of the hardware architecture interface of the transducer interface modules. The geometry of the transducer interface modules is discussed in Section 3.3. Face connectors, a key feature of transducer interface that enable the physical connection between the transducer interface modules, are discussed in Section 3.4. The mechanical and electrical designs of the face connectors are included. An application programming interface structure is discussed in Section 3.5. Section 3.6 present a description of additional modules which may be required to make a composite sensing system fully functional, depending on an application. A knowledge representation scheme to enable the modules to store and communicate their functionalities and capability is discussed in Section 3.7.

## 3.2 Design and Implementation of the Hardware Architecture

In this section, a detailed discussion related to the design and implementation of the hardware architecture of the TIMs is presented. The hardware architecture provides an interface between the sensors, the actuators, and the communication medium. Figure 3.1 shows the block diagram of the hardware architecture. Each TIM encapsulates the same hardware architecture. The main functions of the hardware architecture include: providing a generalized interface to a multitude of sensors and actuators, an interface between the face connectors and the module controller, and execute an independent distributed software architecture [9].

The functionalities of the hardware architecture are implemented in various layers. Based on the functions performed, the hardware platform is made up of the following layers:

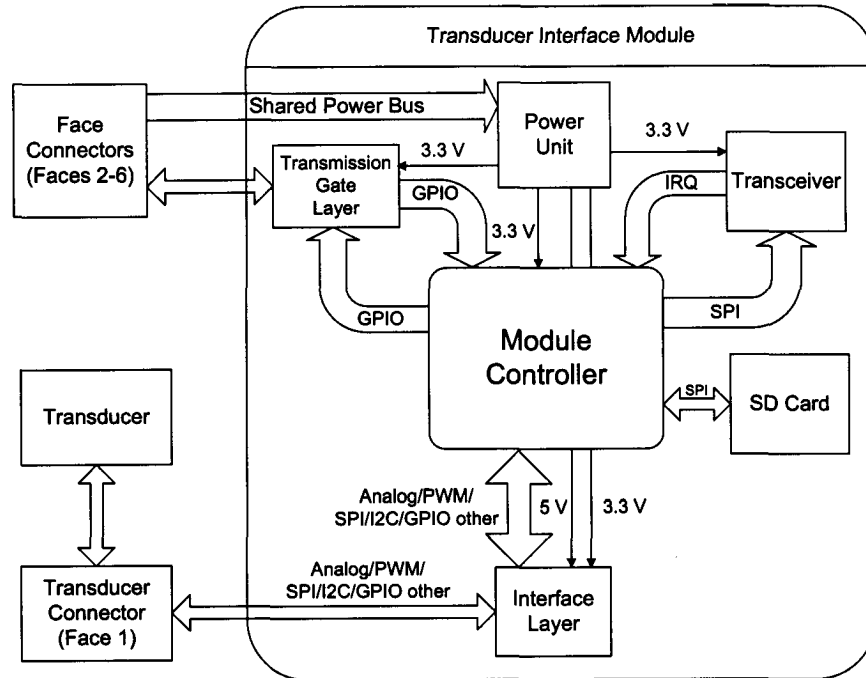


Figure 3.1: Block diagram of the hardware architecture.

- Controller/Communication layer
- Face connector interface layer
- Power supply layer
- Transducer interface layer

The layers are shown in Figure 3.2. These layers are stacked together to construct the hardware platform, as shown in Figure 3.3. The layered architecture approach offers several advantages. First, it simplifies the implementation of various layers. Second, modular nature of the hardware architecture facilitates the troubleshooting and debugging

of the hardware. Finally, the modular design allows the design of any layer to be modified without affecting other layers. The architecture can be expanded with minimum effort to include new functionalities. The following sections discuss in detail the functions performed by each layer.

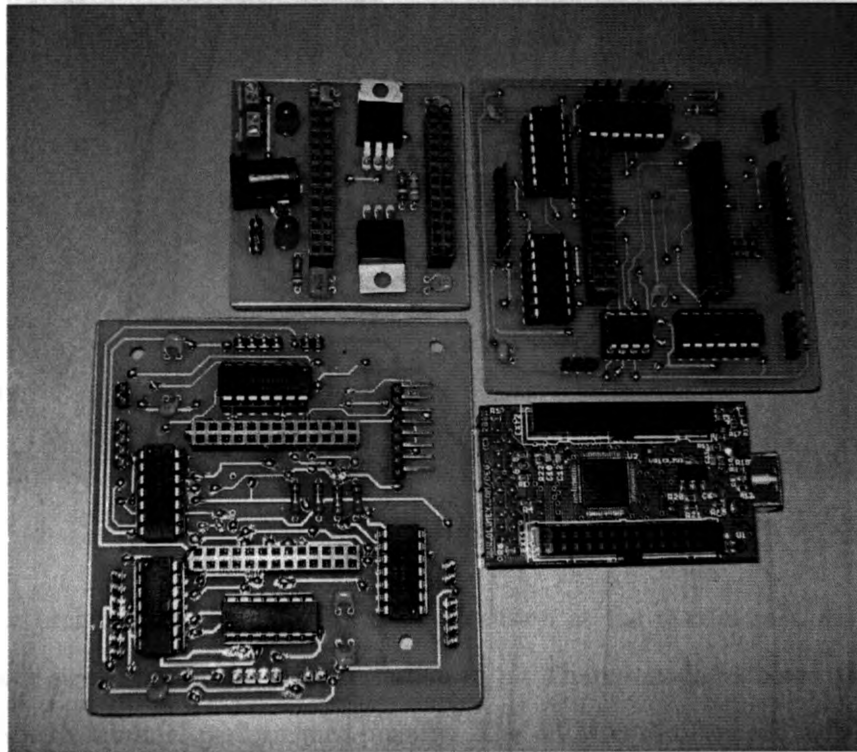


Figure 3.2: Various layers of the hardware stack.

### 3.2.1 Controller/Communication Layer

The core of this stackable architecture is the controller/communication layer. The LPC-2148 [33] from Philips Semiconductors is selected as the module controller for the prototype. It is a 16-bit/32-bit processor based on the ARM7TDMI-S central processing unit with real-time emulation and embedded trace support, that combine microcontroller with embedded high-speed flash memory. A 128-bit wide memory interface and a unique

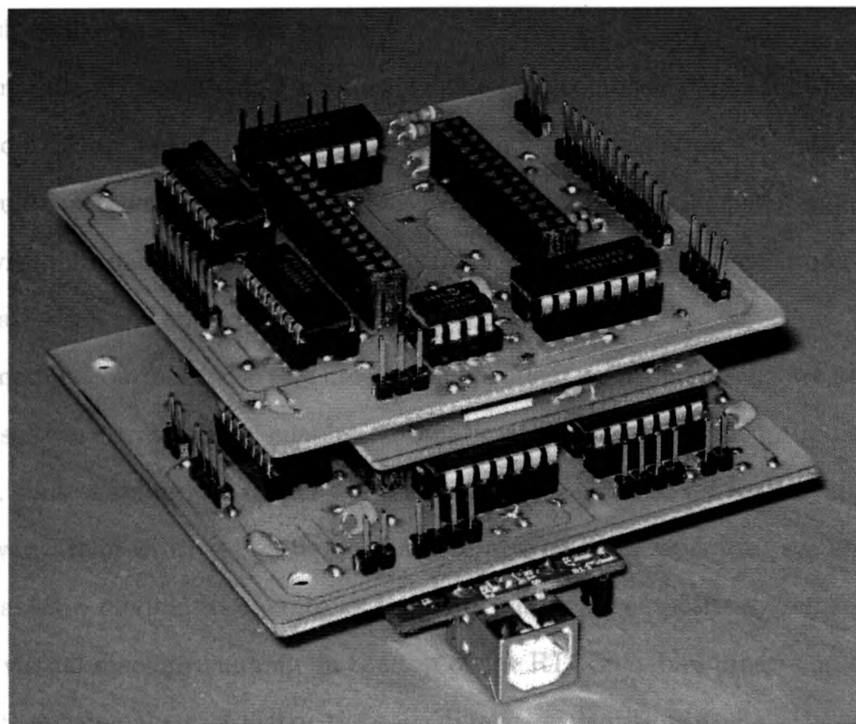


Figure 3.3: Hardware stack.

accelerator architecture enable 32-bit code execution at the maximum clock rate. For critical code size applications, the alternative 16-bit Thumb mode reduces code by more than 30% with minimal performance penalty. The ARM controllers are widely popular in the design of embedded applications due to their small size, low power consumption, and high performance.

The controller features serial communications interfaces ranging from a USB 2.0 Full-speed device, multiple Universal Asynchronous Receiver Transmitters (UARTs), Serial Peripheral Interface (SPI), Synchronous Serial Port (SSP), Inter-Integrated Circuit (I<sup>2</sup>C) bus and on-chip SRAM of up to 40 kB, thus providing both large buffer size and high processing power. It also includes various 32-bit timers, dual 10-bit analog to digital converters (ADCs), 10-bit digital to analog converter (DAC), pulse width modulation (PWM) channels and 45 general purpose input output (GPIO) lines with nine edge or

level sensitive external interrupt pins.

The module utilizes a single microcontroller to accomplish data acquisition, overall module control, application specific processing, and communication with other modules. The module controller processes the signals from the face connectors to determine the face connectivity information. The module controller is interfaced to a multitude of sensors and actuators via the transducer interface layer.

The module controller also executes an independent distributed software architecture [9]. The software architecture enables automatic acquisition of configuration-specific algorithms, thus enabling the modules to be rapidly reconfigured for various applications. The software architecture is built upon a pre-emptive real-time operating system (RTOS), which enables a module to operate as a part of multiple composite systems simultaneously. A virtual machine running on the top of the RTOS enables processing algorithms to be written using abstract instructions, enabling the algorithms to be written once and be independent of the hardware architecture thereafter. This allows the algorithms to be utilized on any hardware architecture that supports the virtual machine, even if these hardware architectures vary internally or are modified. The module controller enables drivers that provide an abstraction between the hardware and software architectures.

The LPC2148 controllers are available in a LQFP64 (low profile quad flat package; 64 leads; body  $10 \times 10 \times 1.4$  mm) package. In order to simplify testing and prototyping, a standard small header board for the LPC2148 integrated circuit (IC) from SparkFun Electronics [34] is utilized for the hardware stack. The header board features extension headers for all of the controller ports, reset circuit, on board voltage regulator, etc. The header board is shown in Figure 3.4.

The microcontroller is programmed using Flash Magic software [35]. Flash Magic, sponsored by NXP Semiconductors, is a PC tool for programming flash based microcontrollers from NXP using a serial protocol while in the target hardware. The software

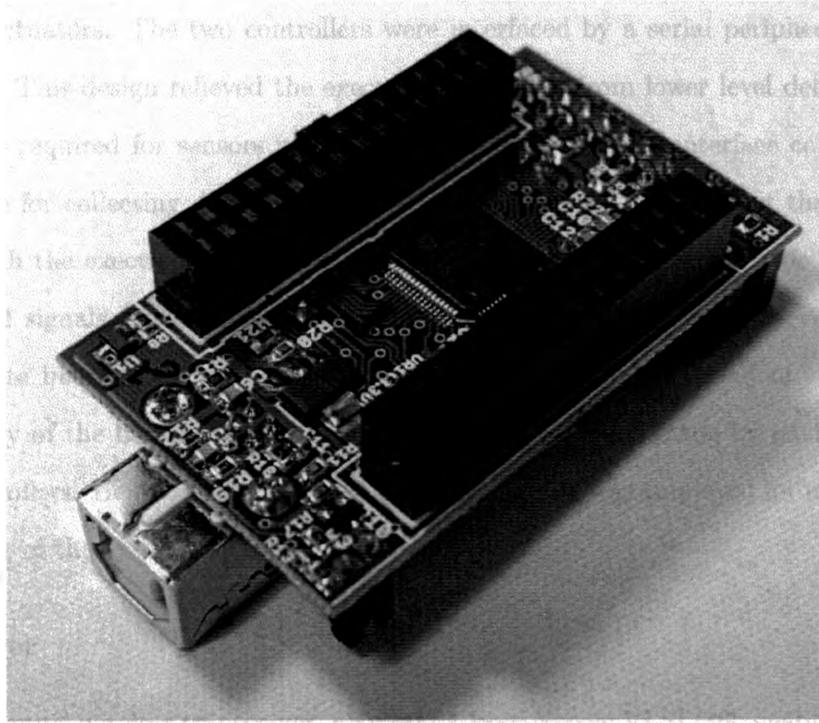


Figure 3.4: Header board for LPC2148.

loads the Intel hex file to the microcontroller by using its In-System Programming (ISP) mode communicating through the serial port. The ISP mode allows the microcontroller to communicate with host device (PC) through a serial port (RS232).

### Multi-microcontroller System

The initial architecture design utilized a multi-microcontroller system to perform the functions accomplished by a single module controller in the final design. As shown in Figure 3.5, the multi-microcontroller consisted of two microcontrollers: Execution controller and transducer interface controller. The execution controller, a powerful microcontroller, managed the system on the whole and executed the distributed software architecture [9].

The transducer interface controller was a low level microcontroller managing the sensors and actuators. The two controllers were interfaced by a serial peripheral interface (SPI) bus. This design relieved the execution controller from lower level details such as the drivers required for sensors and actuators. The transducer interface controller was responsible for collecting data from sensors and packaging it in a format that was compatible with the execution controller. Also, the transducer interface sensor had to send out control signals from the execution controller to the actuators. However, the data transfer rate between the two controllers was dependent on the speed of the SPI bus. The latency of the bus can be slow enough to offset the gains made by having multiple microcontrollers. Hence, a single powerful microcontroller was employed for more reliable functioning of the system.

### Transceiver

A multi-channel wireless transceiver, nRF24L01 from Nordic VLSI [36], enables the inter-module communication. The nRF24L01 is a single chip radio transceiver operating in the world wide 2.4–2.5 GHz ISM band utilizing Gaussian Frequency Shift Keying (GFSK) modulation technique. It is optimized for ultra-low power control and command applications. The transceiver consists of a fully integrated frequency synthesizer, a power amplifier, a crystal oscillator, a demodulator, modulator and Enhanced ShockBurst protocol engine. Output power, frequency channels, and protocol setup are easily programmable through a serial peripheral interface (SPI) bus. Current consumption is very low, only 9.0 mA at an output power of -6 dBm and 12.3 mA in receive mode. Built-in Power Down and Standby modes makes power saving easily realizable.

The nRF24L01 is interfaced to the module controller via the SPI Port 0. Since the transceiver remains active for most of the operation, the SPI Port 0 is exclusively used for transceiver. The interface between module controller and transceiver is shown in the

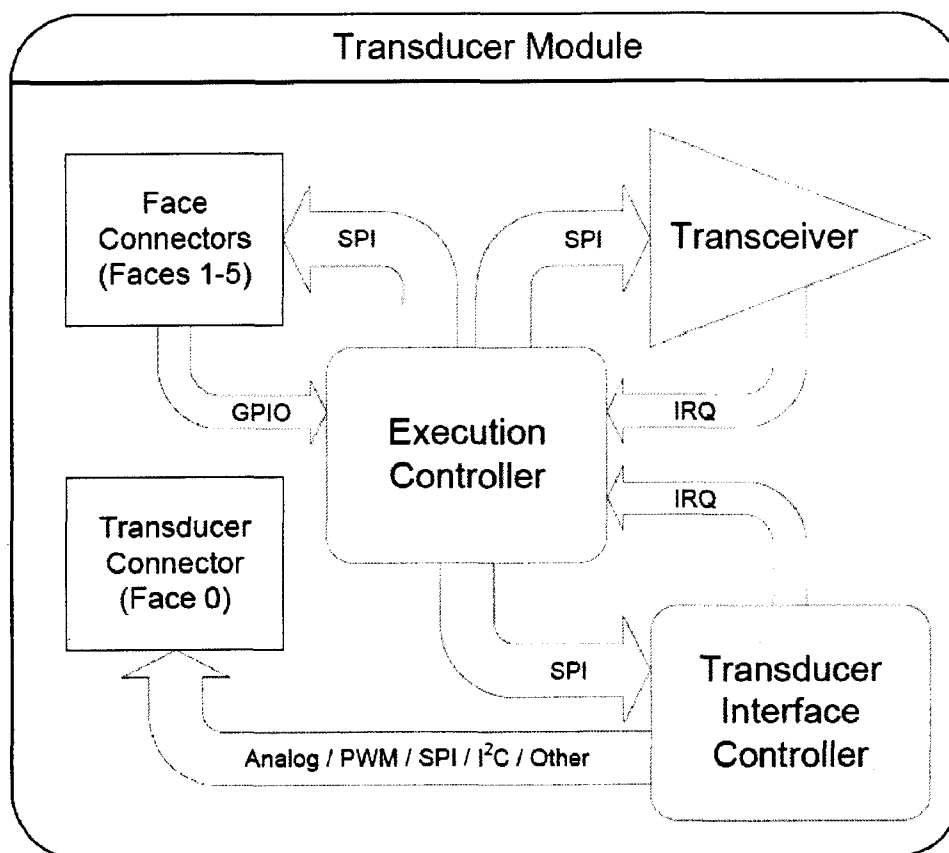


Figure 3.5: Block diagram of hardware architecture employing a multi-microcontroller system.

Figure 3.6.

Although there are several standard 2.4 GHz technologies such as Bluetooth, ZigBee and Wi-Fi, nRF24L01 is superior in terms of power consumption, bandwidth, protocol support, and overall implementation cost [37, 38] as compared to the other existing prototyping solutions for low power applications. nRF24L01 offers high speed, low power consumption and allows the implementation of custom protocols, thereby simplifying the overall design process.



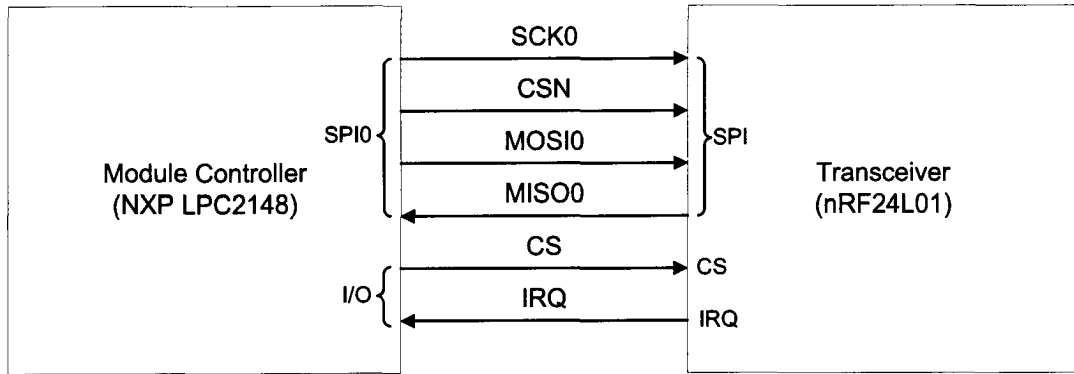


Figure 3.6: Module controller-transceiver interface.

### 3.2.2 Face Connector Interface Layer

The face connector interface layer provides an interface between the hardware architecture and the face connectors. The face connectors, discussed in detail in Section 3.4, enable the physical connection of the transducer interface modules in different orientations. The hardware architecture utilizes the face connector interface layer to detect connections between the modules and pose information of the connected modules.

The face connector interface layer uses a multiplexed bus consisting of four general purpose input output (GPIO) lines to enable the communication between the face connectors and the module controller. The multiplexed bus is implemented using quad bilateral switches. The layer consists of five CD4066 switches [39], one dedicated to each connectable face of the module. Each face connector requires four I/O lines in order to communicate with the module controller.

A bilateral switch uses a special complimentary metal oxide semiconductor (CMOS) circuit called a transmission gate. The circuit behaves as a single pole/single throw

(SPST) switch which is under electronic control. When the control signal, or enable, is high, the switch is closed, allowing signals to be transferred between the switch terminals. The switch is 'bilateral' because either terminal can be used as the input. In other words, current flow can be in either direction. A CD4066 consists of four bilateral switches and four enable lines to control the switches. The symbol of bilateral switch is shown in Figure 3.7. Table 3.1 shows truth table for the bilateral switch.

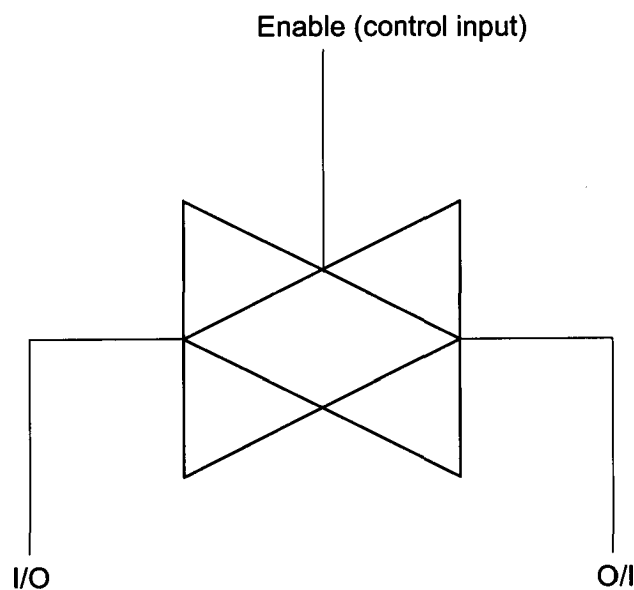


Figure 3.7: Logic symbol of bilateral switch.

Table 3.1: Truth table of the bilateral switch.

Enable	Switch function
High	ON
Low	OFF

The use of multiplexed bus results in a significant decrease in the number of I/O lines

required by the face connectors logic. Each face connector requires four I/O lines in order to communicate with the module controller. This implies the need of twenty I/O lines for five face connectors. The multiplexed bus allows the five face connectors to share four I/O lines. An additional five lines are required for selecting the switches. In order to enable a face connector, the dedicated switch is selected. Thus, a total of only nine lines are required to implement the overall logic for the face connectors. The block diagram of the face connector interface layer is shown in Figure 3.8.

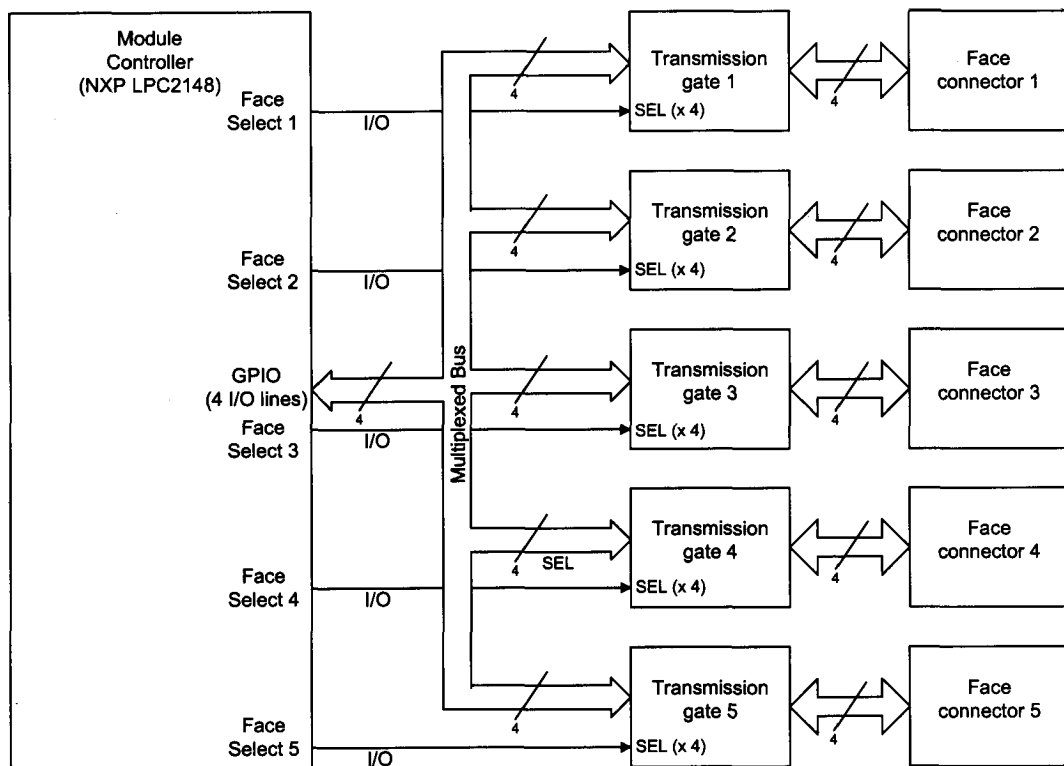


Figure 3.8: Block diagram of face connector interface layer.

A CD4066 has four select lines for the four bilateral switches, so that each switch can be controlled individually. For the face connectors, the select lines of each CD4066 are tied together and controlled by a dedicated I/O line of the module controller. This

ensures that all the four switches of a CD4066 dedicated to a face connector are enabled or disabled at the same time. In order to select or deselect a face connector, the module controller enables or disables the respective CD4066 by sending appropriate signal on the I/O line controlling the select lines. Once a face connector is selected, the module controller sends and receives signals from the face connector. These signals are handled by a face connector protocol, discussed in Section 3.4.3, to detect connectivity between modules and the relative pose of the connected modules. The process of enabling or disabling the switches to control the face connectors ensures that only one face connector is using the multiplexed bus at any given time, thus preventing conflict between the face connectors. In order to protect the controller I/O lines from excessive current flow, current limiting resistors are connected in series between the I/O lines and the switches.

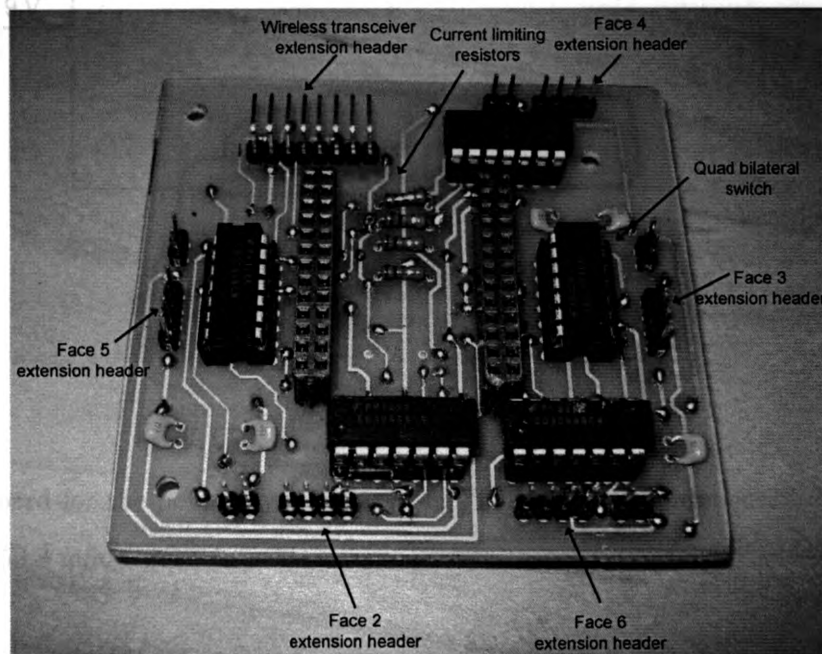


Figure 3.9: Face connector interface board.

The board for the face connector interface layer is shown in Figure 3.9. The overall dimension of the board is  $80.01 \times 78.74$  mm.

### 3.2.3 Power Layer

The power layer forms the third layer of the hardware architecture. It supplies voltage and current to the various layers in the hardware architecture. This layer performs voltage regulation to convert the input voltage from a battery or shared powered modules to both 3.3 V and 5 V. The block diagram of the power layer is shown in Figure 3.10.

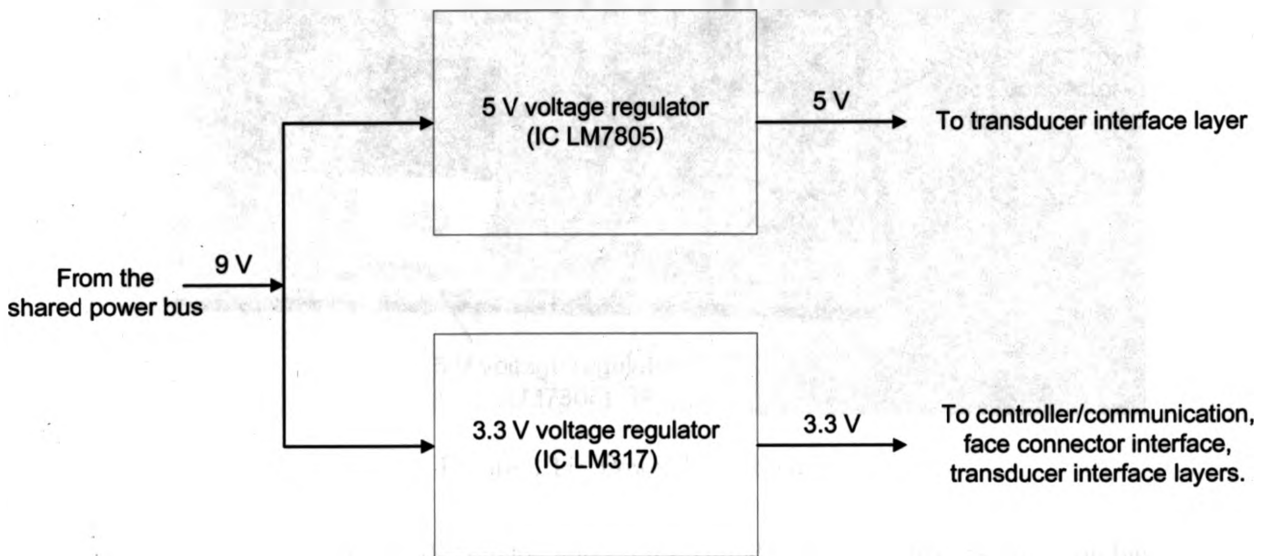


Figure 3.10: Block diagram of power layer.

The board for the power layer is shown in Figure 3.11. The dimension of the board is  $50.8 \times 50.8$  mm.

### 3.2.4 Transducer Interface Layer

The final layer of the hardware platform is the transducer interface layer that supports a broad range of sensors and actuators. The block diagram of transducer interface layer is shown in Figure 3.12. As discussed in Chapter 2, sensors can be classified based on the

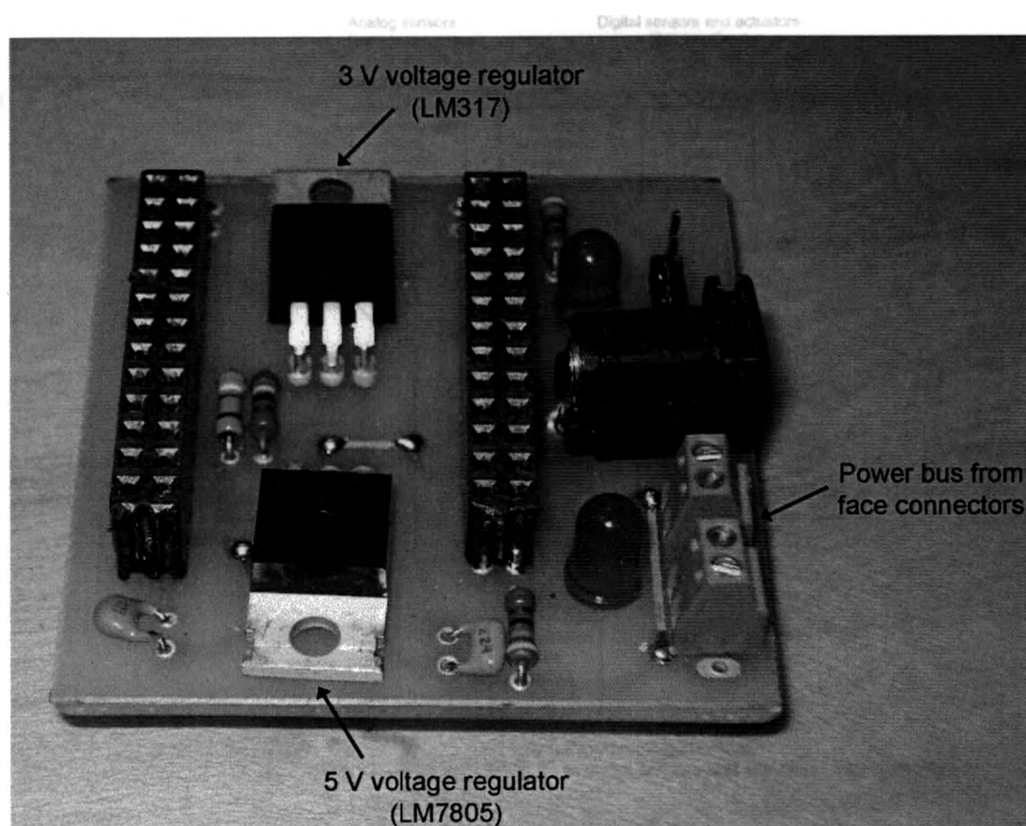


Figure 3.11: Power layer board.

nature of their output signals (analog and digital signals). Sensor outputs are available in widely varying formats. The most common analog sensor output types include voltage, current (0–20 mA and 4–20 mA) and frequency (Amplitude modulated (AM), frequency modulated (FM), pulse width modulated (PWM)). As the full-scale outputs of most analog sensors are relatively small, appropriate signal conditioning is required. Also, analog signals need to be digitized in order to be communicated to the microcontroller.

The transducer interface layer requires all analog signals to be represented as a voltage signal. Hence, signal conversions such as current to voltage or frequency to voltage are performed externally. The interface layer performs basic analog signal conditioning such as amplification using programmable gain amplifiers (PGAs). The PGAs are suited for

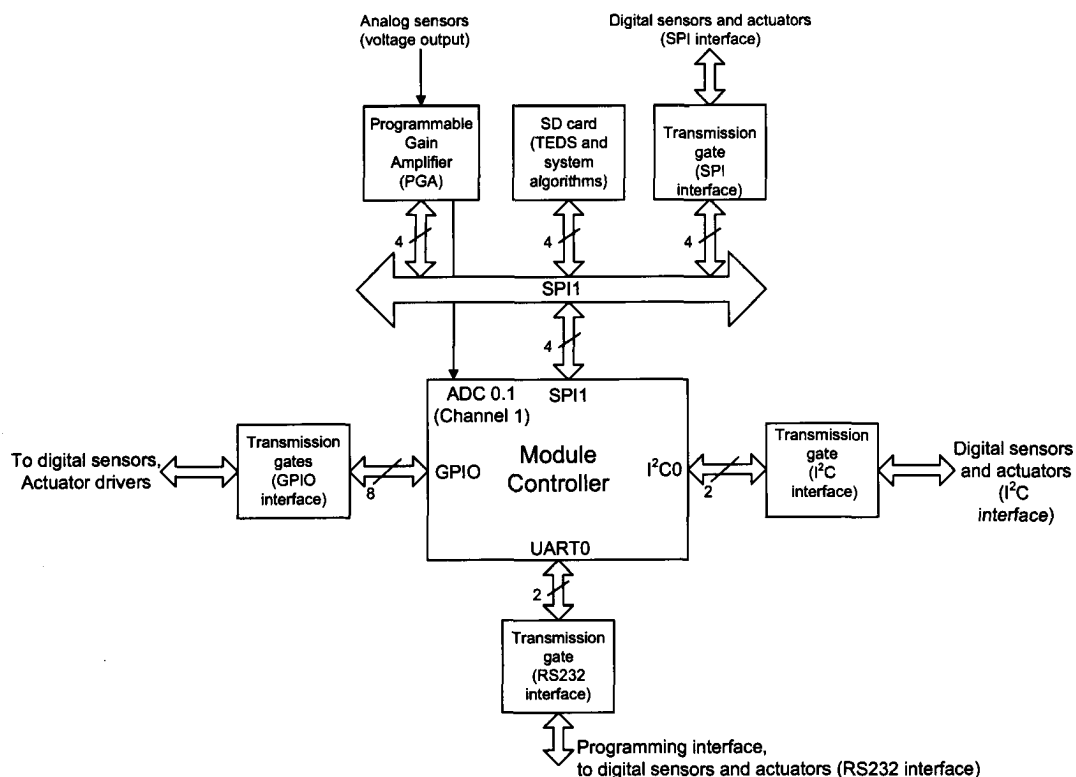


Figure 3.12: Block diagram of transducer interface layer.

embedded applications as gains can be easily programmed using a microcontroller. Since the gain is programmable, a single PGA can amplify a wide range of inputs. The PGAs from Microchip [40] are selected for the design. The single channel PGA can be configured for gains from 1 to 32. The PGA is programmable over serial peripheral interface (SPI) bus. The amplified signal is digitized by 10-bit analog-to-digital converter (ADC) on the module controller. The use of the on chip ADC serves to minimize the number of hardware components required by the module. The 10-bit ADC provides a resolution of 3.2226 mV/code over the range of 0–3.3 V.

Digital sensors include a digital interface that allows communication with a microcontroller. The output of digital sensors can have different formats such as parallel interface (eight or sixteen digital input output lines), a serial interface (for example following the

RS232 standard), or a synchronous serial interface (for example serial peripheral interface (SPI)). The interface reports sensor readings back to the microcontroller, in addition to receiving control instructions from the microcontroller to set the parameters of the sensors. Digital sensors can be interfaced directly to the module controller. The module controller provides interface to popular buses such as serial peripheral interface (SPI) [41], and inter-integrated circuit (I<sup>2</sup>C) [42]. Since I<sup>2</sup>C lines are open drain, external pull up resistors (each 4.7 k $\Omega$ ) are connected between the supply and the bus lines in order to provide an output functionality. Additional interfaces include RS232 and general purpose input output (GPIO). The GPIO interface (consisting of eight I/O lines) allows non-standard transducers to be interfaced to the module controller. Depending on the power requirements, the interface layer may send the control signals to actuators directly or direct them to appropriate external hardware drivers.

The transducer interface layer also provides an interface to the Secure Digital card (SD card) from Sandisk, a form of flash memory. For compatibility with existing controllers, the SanDisk SD Card offers, in addition to the SD Card interface, an alternate communication protocol based on the SPI standard. The SD card stores the Transducer Electronic Data Sheet (TEDS) of the module and serves as the database of system algorithms. Based on the functionality of the composite group, the module controller executes the appropriate system algorithm. The SD card is extremely small in size (32  $\times$  24  $\times$  2.1 mm ) and provides large storage capacities (64 MB–2 GB, recent cards come with 4 GB capacity).

The transducer interface layer also provides extension headers for 3.3 V and 5 V power supplies. These supplies can be used to power connected transducers.

Figure 3.13 shows the board for transducer interface layer. The overall dimension of board is 75  $\times$  75 mm.



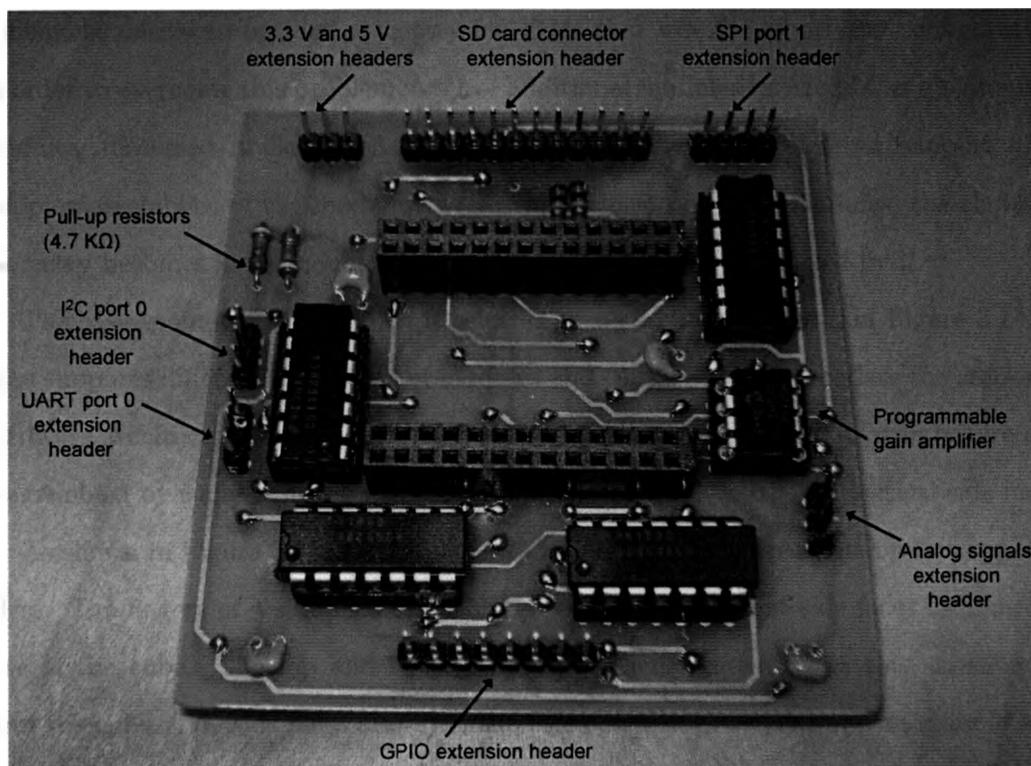


Figure 3.13: Transducer interface layer.

### 3.3 Geometry

The hardware stack is encapsulated in a case which serves as the platform for interfacing sensors and actuators. The geometry of the case plays an important role in determining the flexibility of connecting modules in different configurations. Several factors were taken into account while selecting the case geometry. These include the number of assemblies supported by a particular geometry type, the stability of the module, and the ease with which hardware can be accommodated inside the module. Regular polyhedra emerge as the strongest candidates due to the fact that they can support a number of different orientations. Also, regular polyhedra are easier to work with since all of the faces are identical. Although regular polyhedra support a number of different assemblies, it is

difficult to derive an optimal geometry for TIMs that would support any configuration. In order to overcome this problem, a set of additional modules referred to as interconnect modules, discussed in Section 3.6.1, are designed. These modules can support more complex assemblies of the modules. With the inclusion of these modules, the choice of geometry becomes independent of the number of assemblies supported by it.

The case geometry of the TIMs is selected to be a cube, as shown in Figure 3.14. A cube supports different assemblies of modules and can easily accommodate the stackable hardware architecture discussed in Section 3.2. The cube itself is modular in nature and is assembled by connecting six faces: top face, bottom face, and four identical side faces.

As shown in Figure 3.15, a side face has three fingers and three slots on two opposite edges. The fingers of a side face fix into the slots of the adjacent side faces forming the case of the cube. The top and bottom faces are then screwed on to this assembly to form the cube. The side faces also have the face connector mechanism consisting of four face clips to enable the physical connection of the modules and a central power contact implementing a power bus.

The top face, shown in Figure 3.16, is recessed and is reserved for connection to a sensor or actuator. This design allows sensors to be securely housed inside the module. Other sensors that need to be exposed to their environment, such as cameras and temperature sensors, can be placed on the top of the face. Also, there is a slot on the top face to mount the SD card on the module.

The bottom face, shown in Figure 3.17, has the same face connection mechanism as the side faces to enable the physical connection of the modules. It also has four posts on which the hardware stack, discussed in Section 3.2, is mounted.

The dimension of the prototype module depends on the internal hardware stack size. The overall dimensions of prototype module are  $110 \times 110 \times 110$  mm, without the clips. The clips protrude from each side by 10 mm.; therefore, the distance between the centres

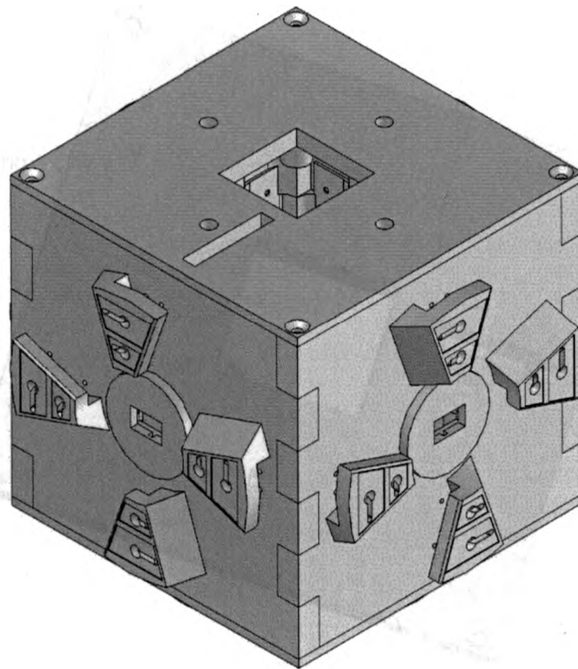


Figure 3.14: Geometry of the module.

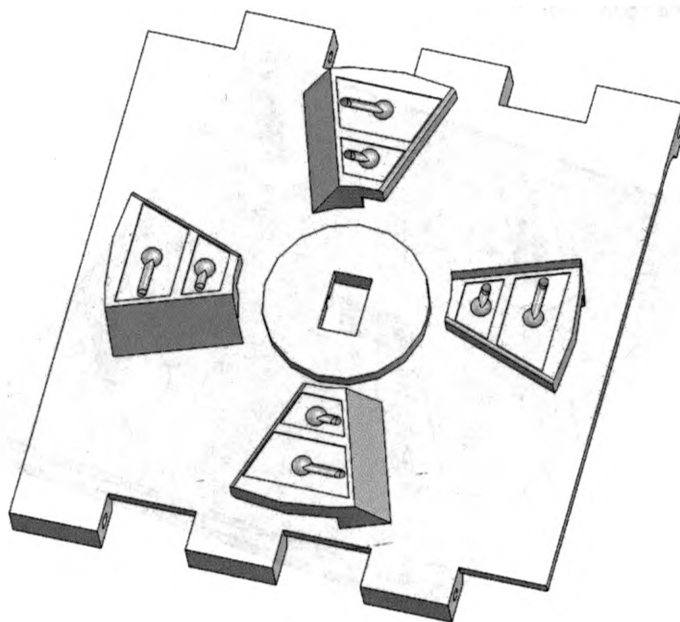


Figure 3.15: Side face of module.

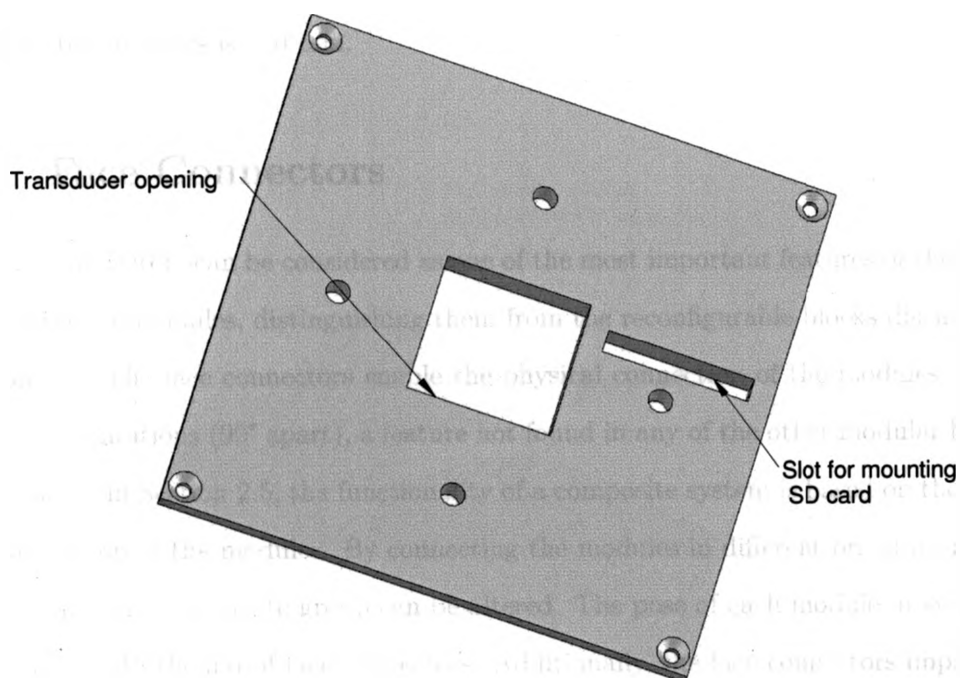


Figure 3.16: Top face of module.

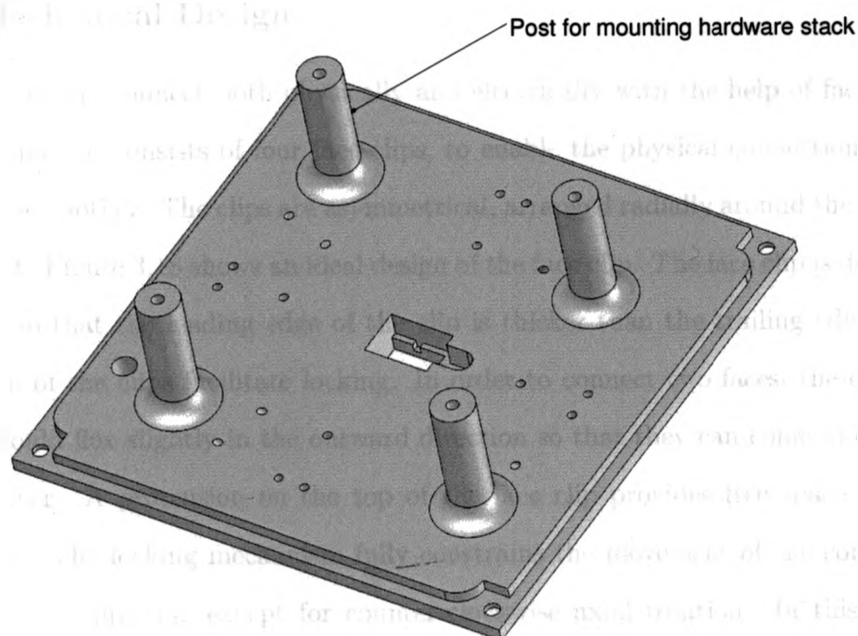


Figure 3.17: Bottom face of module.

of connected modules is 120 mm.

## 3.4 Face Connectors

The face connectors can be considered as one of the most important features of the transducer interface modules, distinguishing them from the reconfigurable blocks discussed in Section 2.5. The face connectors enable the physical connection of the modules in four different orientations (90° apart), a feature not found in any of the other modular blocks. As discussed in Section 2.5, the functionality of a composite system is based on the physical connection of the modules. By connecting the modules in different orientations, the functionality of a composite group can be altered. The pose of each module in system is determined with the aid of face connectors. Additionally, the face connectors implement a power bus that can be shared between the connected modules.

### 3.4.1 Mechanical Design

The face connectors connect both physically and electrically with the help of face clips. Each face connector consists of four face clips, to enable the physical connection of the modules to one another. The clips are asymmetrical, arranged radially around the central power contact. Figure 3.18 shows an ideal design of the face clip. The face clip is designed at an angle, so that the leading edge of the clip is thicker than the trailing edge. The angled design of the clips facilitate locking. In order to connect two faces, the clips on both faces would flex slightly in the outward direction so that they can come in contact with each other. A protrusion on the top of the face clip provides free space for the clip to deflect. The locking mechanism fully constrains the movement of the connected modules in every direction except for counter-clockwise axial rotation. In this mode, movement is constrained by the clips, which present spring and frictional resistance to

applied counter-clockwise torque.

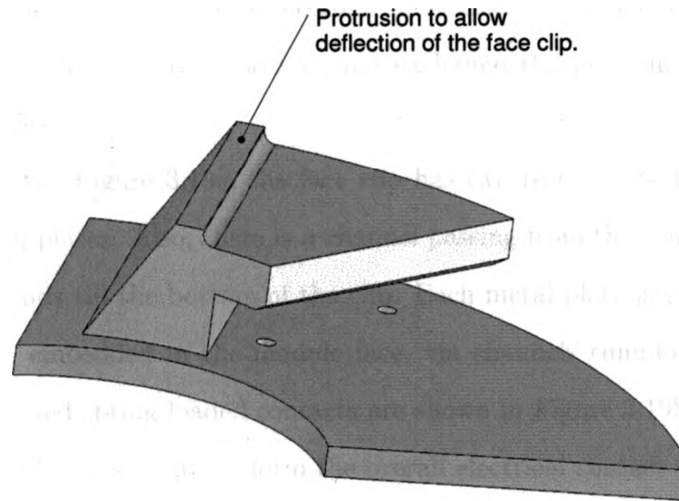


Figure 3.18: Ideal design of the face clip.

The modules were manufactured using the fused deposition modeling (FDM) rapid prototyping technique. Due to rigidity of the material (acrylonitrile butadiene styrene (ABS)) used in the manufacturing process, the face clip is designed to be flat. In this case, the counter-clockwise axial rotation is constrained solely by the frictional resistance of the clips to applied counter-clockwise torque. In order to connect two modules, one module is held at a  $45^\circ$  angle with respect to the other, and rotated clockwise to bring the clips into contact and lock the two modules. Should the clips prove insufficient for a particular application, rods may be inserted into holes drilled through each face.

### 3.4.2 Electrical Design

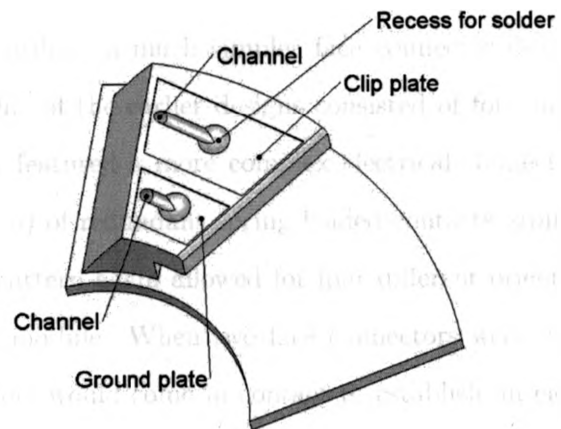
An electrical connection mechanism, consisting of metal plates and spring loaded contacts, is implemented on the face connectors. This mechanism enables a TIM to detect connections on its faces, and determine and exchange the pose information with the connected modules.

As shown in the Figure 3.19a, the face clip has two recessed portions to house two conducting metal plates. Also, there is a channel passing from the center of the recessed portion and extends till the bottom of the clip. Each metal plate is connected to spring loaded contacts, embedded in the module face, via channels running through the face clips. The embedded spring loaded contacts are shown in Figure 3.19b. The metal plate and the embedded spring contacts form the overall electrical contact for the clip. When two clips come in contact, the metal plate on either clip presses against the spring loaded contacts, thus establishing an electrical connection between face connectors.

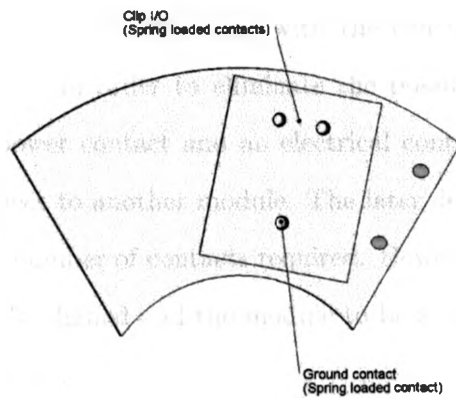
The first metal plate, labeled clip plate, and the corresponding embedded spring loaded contacts form the input-output channel for the face clip. The four channels from the four face clips of a face connector are connected to the face connector interface layer via the multiplexed bus, discussed in Section 3.2.2. Once a face connector is selected, the module controller sends and receives signals from the face connector via these channels. The control signals are then utilized by face connector protocol, discussed in Section 3.4.3, to detect connectivity between modules and the relative pose of the connected modules.

The second metal plate, labeled ground plate, and the corresponding embedded spring loaded contacts form the ground contact for face connector. The ground contacts on all the faces are connected internally to implement a common ground plane. The central power contact is a leaf spring contact from Keystone Electronics. The power contacts on all the faces are connected together.

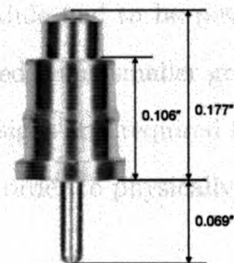
The spring loaded contacts utilized in the prototype design are from Mill-Max Manu-



(a) Face clip.



(b) Embedded spring loaded contacts.



(c) Spring loaded contact.

Figure 3.19: Electrical connection mechanism.



facturing Corporation [43]. The diameter of the contact is 1.066 mm. In order to ensure a reliable connection between the metal plate and spring loaded contact, redundant contacts can be embedded in the module face. The prototype design utilizes two spring loaded contacts.

The final prototype utilizes a much simpler face connector design as compared to the previous designs. One of the earlier designs consisted of four mechanical face clips as discussed before, but featured a more complex electrical connection mechanism. It consisted of a grid ( $5 \times 5$ ) of redundant spring loaded contacts around a central power contact. The mirrored pattern again allowed for four different orientations of a module with respect to another module. When two face connectors were in contact, the grids on the two face connectors would come in contact to establish an electrical connection. The contacts were designed to implement the asynchronous RS232 serial communication protocol. However, the design was complex due to the number of electrical contacts. Also, when two modules were physically connected, the electrical contacts of one face connector had to perfectly align with the exact same electrical contacts on the other face connector. In order to eliminate the possibility of accidental connection between the central power contact and an electrical contact, the module had to be powered off so as to connect to another module. The later designs featured much smaller grids, thus reducing the number of contacts required. However, these designs still required the grids to be perfectly aligned and the module to be switched off in order to physically connect to other modules.

The present design utilizes the face clips for electrical connection, thus eliminating the grid completely. This avoids the problems of grid alignment and accidental shorting of contacts by the power contact.

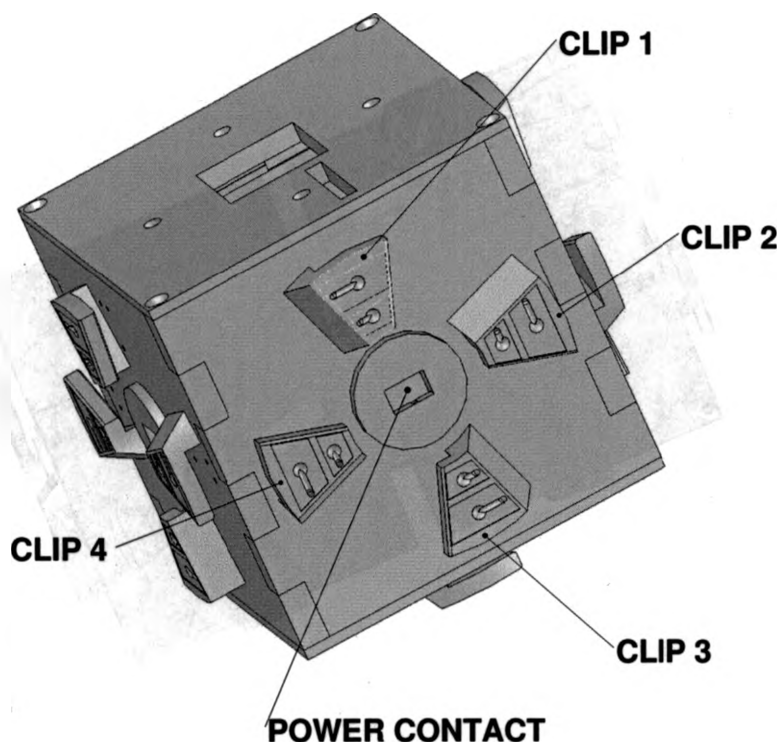


Figure 3.20: Face connector.

### 3.4.3 Face Connector Protocol

The functionality of a composite system is determined by the pose of the connected modules. The overall pose of the system can be established by determining the relative orientations and face connections (faces that are in contact when two modules are physically connected) between all the modules present in a composite system.

In order to determine the relative orientation and face connection information, the module controller executes the face connector protocol. The face connector protocol is based on the standard asynchronous RS-232 serial communication protocol. Asynchronous communication implies transmission of data without the use of an external clock signal. Any timing required to recover data from the information is encoded within

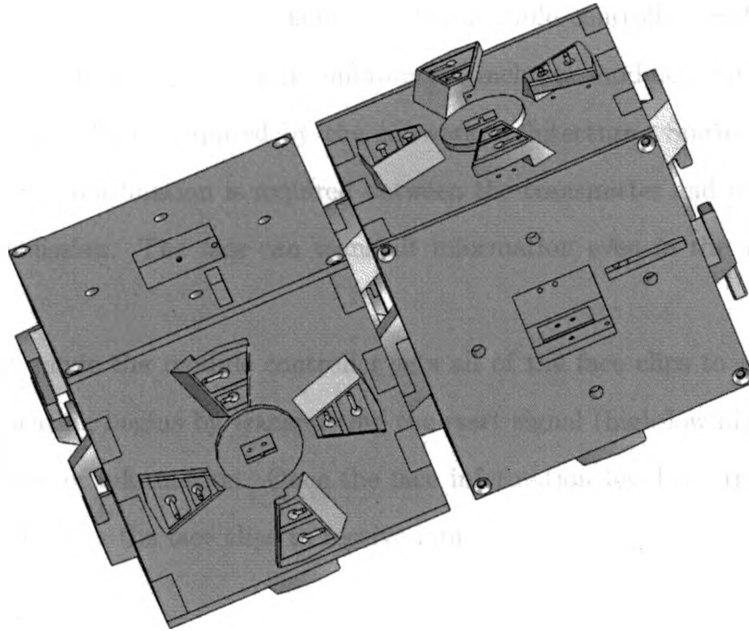


Figure 3.21: Modules connected via face connectors.

the signals.

The face connector interface layer, as discussed in Section 3.2.2 , provides an interface between the face clips and the module controller. The module controller sends and receives control signals from the face clips via the face connector interface layer. The module controller executes the face connector protocol and processes these signals to determine face connectivity and relative orientation information.

The face connector protocol consists of two main functions: transmit and receive modes. The faces remain mainly in receive mode, monitoring the clips for any incoming data. Occasionally, the faces switch to transmit mode to send out the face information.

### Transmit Mode

In this mode, the face acts as a transmitter. The module controller sends out multiple bytes of information on Clip 1. This information includes module address, face number, and additional bytes required by the software architecture. Since the protocol is asynchronous, no coordination is required between the transmitter and receiver prior to the data transmission. The face can transmit information even in the absence of any connection.

In transmit mode the module controller sets all of the face clips to act as outputs. Data communication begins by transmitting the start signal (high-low-high bit pattern) followed by bytes of information. Once the face information has been transmitted, the module controller sets the face clips to receive input.

### Receive Mode

In this mode, the face acts as a receiver. The module controller sets all of the face clips to input and then monitors the face clips for any incoming information. Since the protocol is asynchronous, the receiver has to synchronize itself with the incoming data. The receiver synchronizes itself with the transmitter by detecting the start symbol (high-low-high bit pattern). The module controller reads the face information from the face clip on which the start signal is detected. In order to read the incoming information reliably, the module controller starts reading the data after a delay of one half of the bit duration.

The relative orientation between the connected modules is also detected in the receive mode. As discussed before, the face clips enable the physical connection of the modules to one another in four different orientations (90° apart). Depending on the orientation, the face clips of the receiver get connected to face Clip 1 of the transmitter in one of the four possible combinations as shown in Table 3.2. The relative orientation between the connected faces can be established by determining the face clip on the receiver that

comes in contact with the Clip 1 of the transmitter. The module controller detects the relative orientation by determining the face clip on which the start symbol is detected.

Table 3.2: Relative orientation and face clip connections.

Relative Orientation	Transmitter face Clip 1 $\rightarrow$ Receiver face clip
0°	Clip 1 $\rightarrow$ Clip 1
-90°	Clip 1 $\rightarrow$ Clip 2
180°	Clip 1 $\rightarrow$ Clip 3
90°	Clip 1 $\rightarrow$ Clip 4

### 3.5 Application Programming Interface (API)

A standard application programming interface (API) provides a set of smart functions that are utilized by the distributed software architecture [9] running on the top of the hardware architecture to access the hardware drivers. The introduction of hardware drivers results in shielding of the underlying hardware architecture details from the distributed software architecture, thus allowing decoupling of hardware and software architectures. The diverse hardware drivers enables the associated transducer interface modules to support an unchanging API that allows the software architecture to form a composite module. With hardware drivers a portability layer is defined, which offers stable API towards software architecture enabling platform transparency and translates software application and real-time operating system (RTOS) requirements to hardware architecture [44].

The hardware drivers allow software architecture to create, access, modify, obtain results from a composite group. A sensor driver reads data from a physical sensor device, formats the data into the internal form, and publishes the data in the inter-process communication queues. Each sensor driver is a separate process. A sensor driver contains software to initialize the measurement process and to obtain sensor data [45].

## 3.6 Additional Modules

In order to make a fully functional composite sensing system, a few additional module types may be required. These additional modules include interconnect modules, power modules, and administration modules. The modules do not perform sensing and actuation tasks and, depending on the application, may be absent altogether.

### 3.6.1 Interconnect Modules

To support more complex configurations, an assembly of connected modules may require angular or translational offsets. An efficient way of realizing complex assemblies of modules is through utilizing generic interconnect modules. The interconnect modules enable both physical and functional connectivity. Interconnect modules are available in various shapes and provide a variety of angular and translational offsets (e.g., 45°, 30°, and 11 cm). Figure 3.22 shows an angular interconnect. An example of a translational interconnect is shown in Figure 3.23. More complex assemblies of modules can be created by connecting different types of interconnects together, shown in Figure 3.24.

Prototype interconnects are passive modules utilizing the modular hardware platform, discussed in Section 3.2. As the interconnect modules are passive, they are powered by the transducer interface modules to which they are connected. Each interconnect has a dedicated SD card that stores a datasheet describing the attributes of the associated interconnect. Interconnects use the same face connection mechanism as the module face connectors, discussed in Section 3.4. The wireless unit is optional for the interconnects as all communication on interconnects occurs through face connectors. This results in reducing power consumption and system cost.

Once interconnects are connected to modules, they must inform the modules about their basic properties. In addition, the interconnects need to identify if they are a part of

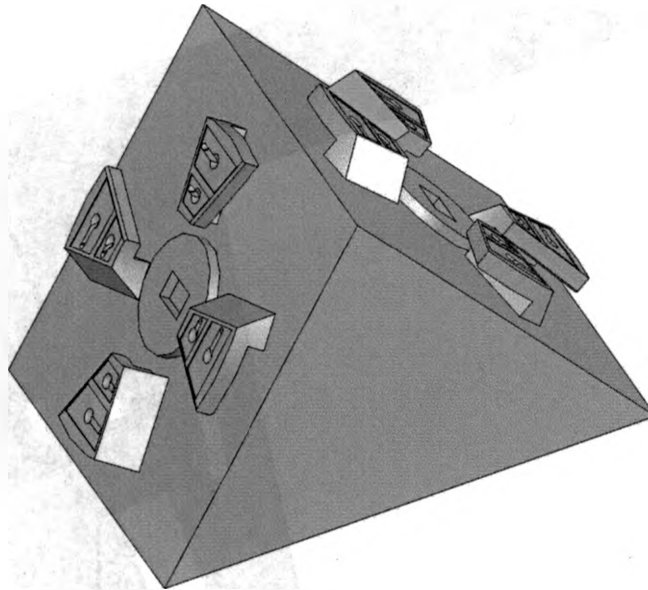


Figure 3.22: Angled interconnect.

an assembly of interconnects or simply connected directly between two modules. An assembly of interconnects assumes a single identity representing the overall offset provided.

### 3.6.2 Power Module

This module supplies power to the various modules in a sensing system, and should maintain a regulated output voltage under variations in supply voltage or load. The design of a power module is not fixed. Power modules may be based on rechargeable batteries or derive power from a wall outlet. They may also include additional hardware to perform signal conditioning such as rectification and filtering to provide a constant DC output. Depending on the application, a composite system may require only one power

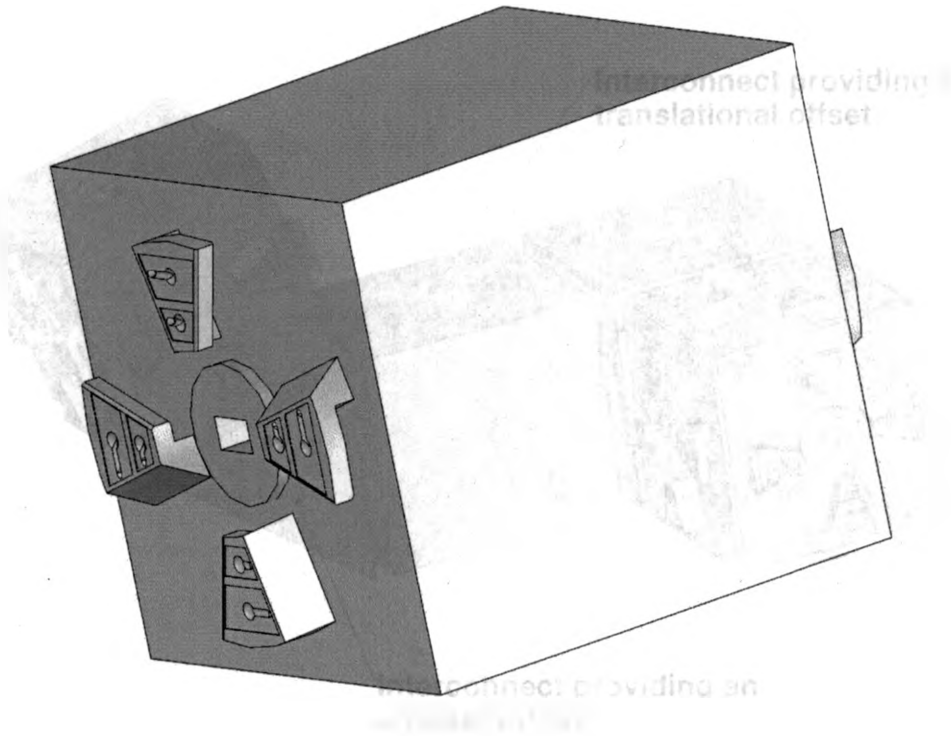


Figure 3.23: Translational interconnect.

module capable of supporting all modules, or require power modules to be connected in parallel in order to increase the available current. TIMs can utilize and share these power sources using the common power bus on their face connectors.

### 3.6.3 Administration Module

This module can be viewed as a system manager used to allow a user to interact with and control TIMs in its vicinity through a graphical user interface. It may be a self-contained unit consisting of a power supply, a module controller, and a transceiver, or be implemented as part of a computer system. Its administrative functionality is provided by the software architecture running on it. Using the software architecture described in



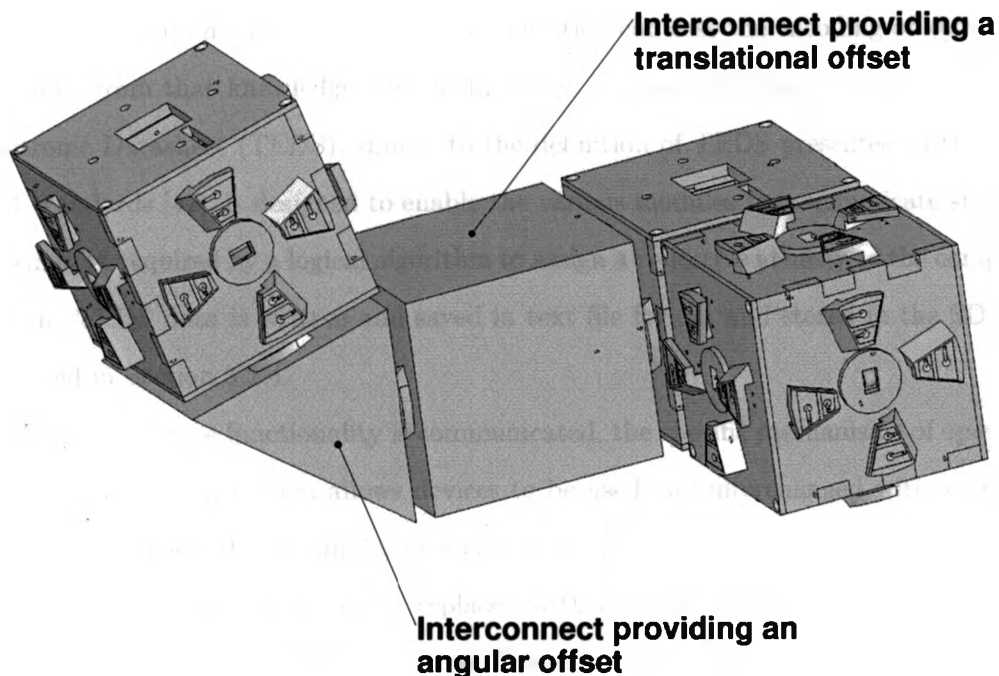


Figure 3.24: An assembly of modules using interconnects

[9], an administration module detects the presence of TIMs in its vicinity by listening for packet transmissions on a special control channel, and also serves as a local repository for processing algorithms used to implement the cooperative behavior of TIMs.

### 3.7 Knowledge Representation Scheme

In order to make a composite system functional, a logical algorithm [9] is loaded into each module in the system. This logical algorithm imparts a collective identity to the composite group, and processes data based on the capabilities (e.g., range, resolution, sample rate, etc.) and functionalities (e.g., temperature measurement) of the transducers present in the composite system. A knowledge representation scheme allows each mod-

ule in the composite group to store and communicate its functionality and capabilities to other connected modules in the system. The selection of knowledge representation scheme is of vital importance as good representation can ease the encoding of knowledge, reasoning from that knowledge and modification of that knowledge [46]. A Transducer Electronic Datasheet (TEDS), similar to the definition of TEDS presented in the IEEE 1451 standards [47], is designed to enable the various modules to communicate standard information required by a logical algorithm to assign a collective identity to the composite system. TEDS data is written and saved in text file format and stored in the SD card, discussed in Section 3.2.4.

While the device functionality is communicated, the specific mechanisms of operation are not. This encapsulation allows devices to be used and interchanged without regard to their intricacies. For example, in a composite group a temperature sensing module with a specific sensor type can be replaced with another temperature sensing module interfaced to an entirely different sensing element without affecting the overall operation of the group.

### 3.7.1 Basic TEDS

In an adaptive modular system, each transducer interface module is uniquely identifiable by a basic TEDS (see Table 3.3) and an AMSS (adaptive modular sensing systems) TEDS. While the definition of basic TEDS is similar to the one described in IEEE 1451 standards, AMSS TEDS has been uniquely defined for adaptive modular sensing systems. Both the TEDS contain fields of variable length. These two TEDS are a minimum requirement for any transducer interface module. In AMSS TEDS, the `ModuleDataWidth` and `ModuleDataHeight` fields are specified to define the different types of outputs from various transducers in a consistent manner.

Table 3.3: Basic TEDS content

Manufacturer ID	Model number	Version letter	Version number	Serial number
-----------------	--------------	----------------	----------------	---------------

Table 3.4: AMSS TEDS content

Field name	Description
ModuleAddress	unique 64-bit address
ModuleType	Sensor/Actuator/Interconnect
ModuleClass	Functionality of the transducer
ModuleDatatype	Float, string, etc
ModuleDataWidth	
ModuleDataHeight	
PrimaryHandlerName	Label assigned to the transducer

### 3.7.2 Templates

The remaining data contained in the TEDS may be defined by a standard IEEE template, a manufacturer published template, or a user defined template. Templates are used to read or write TEDS data, and describe the memory structure which defines the transducer identity, and provides transducer specific data. The IEEE 1451.4 defines standard templates, containing a standard set of properties such as sensitivity and mapping properties, electrical signal properties, and calibration properties, for common classes of transducers. Each property has several fields that provide a complete description of the transducer characteristics. A standard set of fields remain fixed for each template. Table 3.5 lists a sample of the standard fields which are common to all transducers.

#### User Defined Template

In addition to the fields found in a standard TEDS, an additional field is appended to the standard TEDS in order to define a position and orientation of a transducer with respect to its associated transducer interface module, shown in Table 3.6. This field is referred

Table 3.5: Examples of Standard template fields

Sensitivity and Mapping Properties	
Field name	Description
Sens	Sensitivity of transducer
Sens@Ref	Sensitivity of transducer at reference conditions
Reffreq	Reference frequency (f ref)
RefTemp	Reference temperature (T ref)
Sign	Phase inversion (0° or 180°)
Direction	Direction, or axis, of sensitivity (x, y, or z)
MapMeth	Mapping method of physical to electrical units
MinPhysVal	Minimum value of physical measurement/control range
MaxPhysVal	Maximum value of physical measurement/control range
MinElecVal	Minimum value of electrical signal range
MaxElecVal	Maximum value of electrical signal range
Electrical Signal Properties	
Field name	Description
ElecSigType	Type of electrical signal (enumerated)
RespTime	Response time
ACDCCoupling	Coupling of electrical signal (AC or DC)
SensorImped	Electrical impedance of sensor (of each element in case of bridge)
DiscSigType	Discrete signal type
DiscSigAmpl	Discrete signal voltage amplitude
PulseMeasType	Pulse signal measurement type (frequency, period, count, etc.)
Gain	Gain of preamplifier
Filter	Indicates selectable filter
TempCoef	Temperature coefficient
Calibration Properties	
Field name	Description
CalDate	The date of the last calibration
CalInitials	Calibration initials
CalPeriod	Amount of time recommended between calibrations

Table 3.6: User defined template fields

Pose Properties	
Field name	Description
%Position	$x_i, y_i, z_i$ , coordinates of transducer with respect to each face ( $i = 1, 2, \dots, 6$ )
%Orientation	Orientation ( $\theta_i, \gamma_i, \alpha_i$ ) of transducer with respect to each face ( $i = 1, 2, \dots, 6$ )

to as the DefaultPose field. As discussed in Section 3.3, the geometry of the transducer interface modules is a cube. This field is then utilized by initial transformation matrices defined for all the faces to determine the pose information of the transducer with respect to all the faces of the module.

### 3.7.3 Interconnect TEDS

As discussed in Section 3.6.1, the attributes of the interconnect module are stored in a data sheet. The basic fields of the interconnect TEDS are same as the ones described in the adaptive modular sensing systems (AMSS) TEDS. An additional field called InitialTransformFace2 is defined, which describes the angular or translational offset provided by the interconnect. An example of interconnect TEDS is shown in Table 3.7.

Table 3.7: Interconnect TEDS content

Field name	Description
ModuleAddress	unique 64-bit address
ModuleType	Interconnect
ModuleClass	None
PrimaryHandlerName	intconn
InitialTransformFace2	$x_i, y_i, z_i, \theta_i, \gamma_i, \alpha_i$

## 3.8 Summary

This chapter provided a detailed description on the transducer interface modules which serve as the basic building blocks for adaptive modular sensing systems. The design and implementation of the hardware architecture encapsulated in the transducer interface modules was presented. The functions of the various layers of the modular hardware architecture were discussed in detail. The design of face connectors that enable the physical connection between the modules was discussed. In addition to mechanical and electrical designs, a face connector protocol was described. The face connector protocol allowed the modules to detect connections and determining and exchanging of pose information between the connected modules. An application programming interface (API) was presented, which provided a set of smart functions to allow the distributed software architecture [9] to create, access, modify, obtain results from a composite group. Also, a set of additional modules were discussed which may be required to make a composite system completely functional. Finally, a knowledge representation scheme was presented to enable the modules to store and communicate their capabilities and functionalities.

# Chapter 4

## System Validation

This chapter presents examples to validate how the system presented in Chapter 3 meet the specific objectives outlined in Chapter 1. These examples illustrate the formation and functionality of a composite group of modules. An important aim of these examples is to demonstrate how the various features of the TIMs operate. These include: testing of the general purpose hardware architecture stack, functioning of the face connectors (electrical and mechanical designs), and verification of the face connector protocol. The successful implementation of these examples relies on the proper integration of the hardware and the software architectures.

The first example demonstrates the formation and operation of a composite system. A composite system is constructed using voltage sensors and an actuator. The position of the actuator is controlled by the sensors. This example shows how a collective identity is assigned to the composite system. The second example is more comprehensive, demonstrating the rapid reconfigurability feature of the composite systems. In this example a display system is constructed by physically connecting two liquid crystal display (LCD) modules. The display format of the system is extended by connecting the modules in side-by-side or above-below configurations.

## 4.1 Formation and Operation of a Composite System

This example demonstrates the formation and operation of a composite group. It shows how different modules, each providing a core piece of functionality, are connected to form more a capable composite sensor. Once the modules are combined, the capabilities and functionality of the composite sensor is determined and communicated to other modules in the environment. Based on this information, the distributed software architecture [9] assigns a collective identity to the composite system.

This example also assesses the proper operation of the following features of the system:

- The ability to interface with a variety of sensors and actuators.
- Use of specific interface types on transducer interface layer (hardware architecture stack) including: the analog interface, the serial peripheral interface (SPI) bus (digital interface), the pulse width modulation (PWM) output, and the SD card interface.
- The knowledge representation scheme used to describe the functionality and capabilities of the various modules.
- The wireless transceiver.

### 4.1.1 Experimental Set-up

The experimental set-up consisted of a light dependent resistor (LDR) (analog sensor), an accelerometer (digital sensor), and a servo motor (actuator). Each transducer was interfaced to one TIM.

The LDR was connected in series with a 10 k $\Omega$  resistor to form a potential network, as shown in Figure 4.1. The analog output from the potential divider network was fed to the programmable gain amplifier (PGA) on the transducer interface layer of a TIM. The



gain of the amplifier was set to one. The analog output of the PGA was digitized by the on-board ADC. A liquid crystal display (LCD) device was interfaced to this module to display the output values from the various sensors in the system.

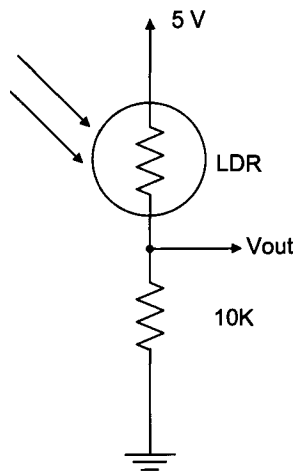


Figure 4.1: LDR potential divider network.

An LIS3LV02DQ [48] was selected as the accelerometer. It is a three-axis digital output linear accelerometer that includes a sensing element and an IC interface that can output the measured acceleration signals to through an I<sup>2</sup>C/SPI serial interface. The SPI interface of the accelerometer was interfaced to the SPI Port 1 on the transducer interface layer of a TIM. Also, a data logger (virtual sensor) was created and stored in the SD card interfaced to the transducer interface layer of this TIM. The purpose of the logger was to store the output values from the various sensors in the system.

The actuator was the servo motor. It was controlled using pulse width modulation (PWM) output from a TIM. The internal servo loop operates in 20 ms blocks (50 Hz). Every 20 ms the servo requires a positive going pulse. The period of the pulse determines

the position of the servo head. For the servo motor used in this demonstration pulse-width of 1.5 ms corresponds to the 90° position of the servo head (middle).

The knowledge representation scheme was described in order to allow the each module to store and communicate its functionality and capabilities. Each module was assigned a unique identity by basic and adaptive modular sensing system (AMSS) TEDS. A standard template describing the attributes of each sensor and actuator was also defined. The overall TEDS information for each module was stored in the SD card interfaced to the transducer interface layer of the associated TIM. The TEDS for the accelerometer, the LDR, and the servo motor are presented in Appendix A.1, A.2, and A.3 respectively. In addition, a TEDS called as servo control template (Appendix A.4) was defined for the composite group. This TEDS contains a Roles field that was utilized by the software architecture to determine the types of sensors and actuators that could form a composite group.

The distributed software architecture executed on the modules enabled the automatic acquisition of configuration-specific algorithms. These algorithms facilitated the assumption of a collective identity by the composite group. A collective identity was assigned by the software architecture on the basis of standard information communicated by the knowledge representation scheme.

The aim of the group was to control the position of the servo head using the voltage sensors present in the system. The composite system averaged the outputs from all the voltage output sensors present in the composite system to set the PWM signal that controls the position of the servo head.

#### **4.1.2 Discussion**

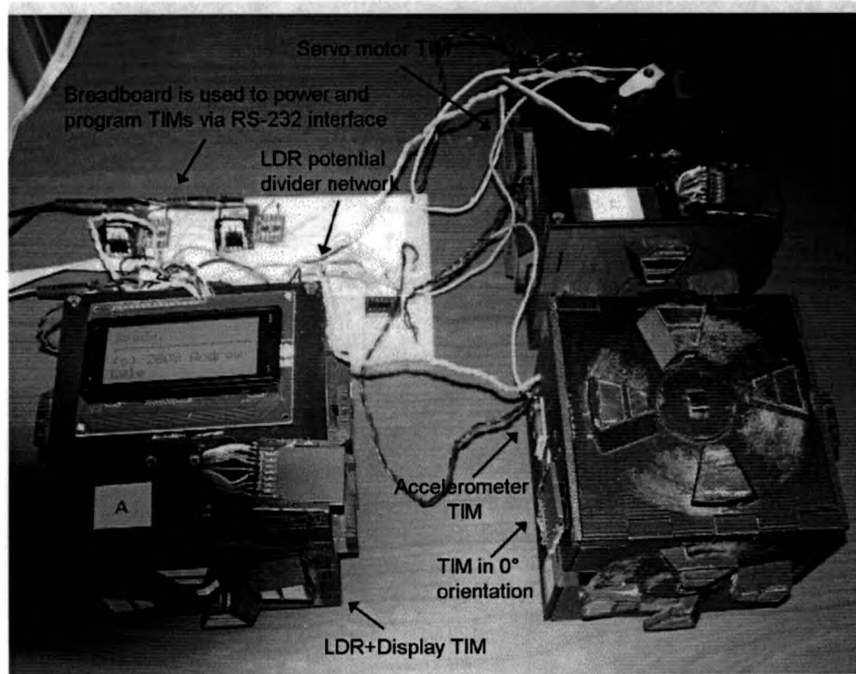
On initialization, the modules communicated their functionality and capabilities via the wireless transceivers. The distributed software architecture, executed on the TIMs, identi-

fied the modules present in the system based on this information and assigned a collective identity to the system. The assumption of collective identity enabled the acquisition of the algorithm which allowed the position of servo head to be controlled by averaging the output voltages from sensor modules. In this case, the module with the lowest address gathered the voltage outputs, executed the averaging algorithm, and controlled the position of the servo head.

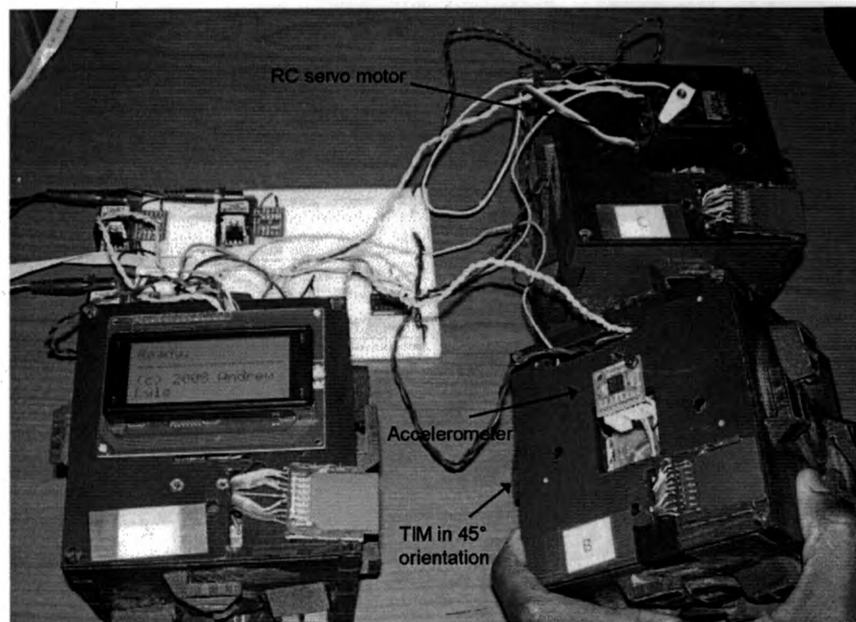
The following figures show the various positions of the servo head based on the averaged output voltage.

### **4.1.3 Summary**

This experiment demonstrated the formation and functioning of a composite group. A composite group was formed using two sensors and an actuator. It was shown how the distributed software architecture assigned a collective identity to the system based on the functionality and capabilities of the transducers present in the system. Additionally, the example also validated the analog interface, the SPI interface, and the PWM output on the transducer interface layer of the hardware architecture stack. The wireless transceivers on the modules facilitated inter-module communication. This example demonstrated the successful integration of the hardware and the software architectures.

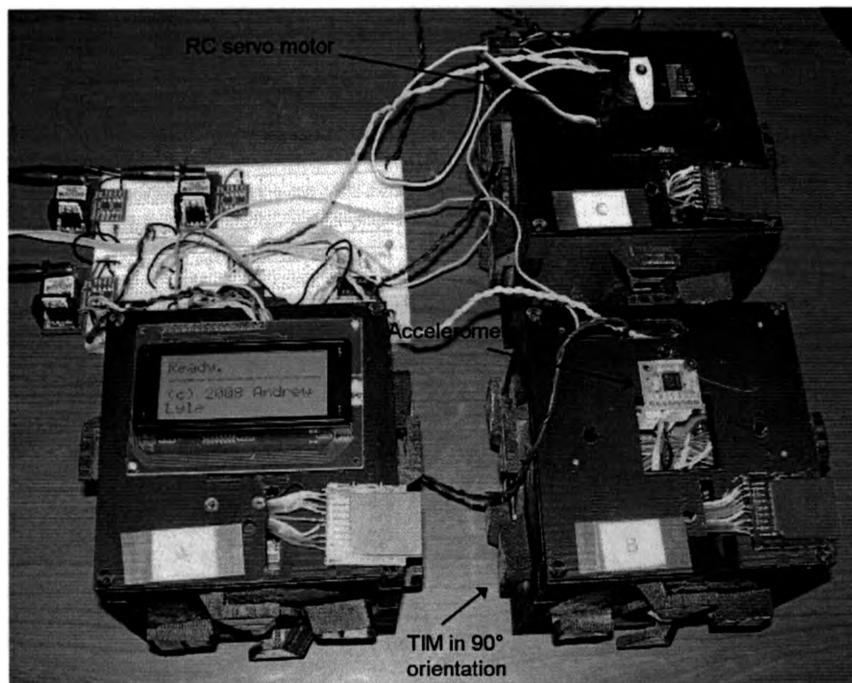


(a) 0° position of the servo head



(b) 45° position of the servo head

Figure 4.2: Composite group consisting of accelerometer, LDR, and LCD.

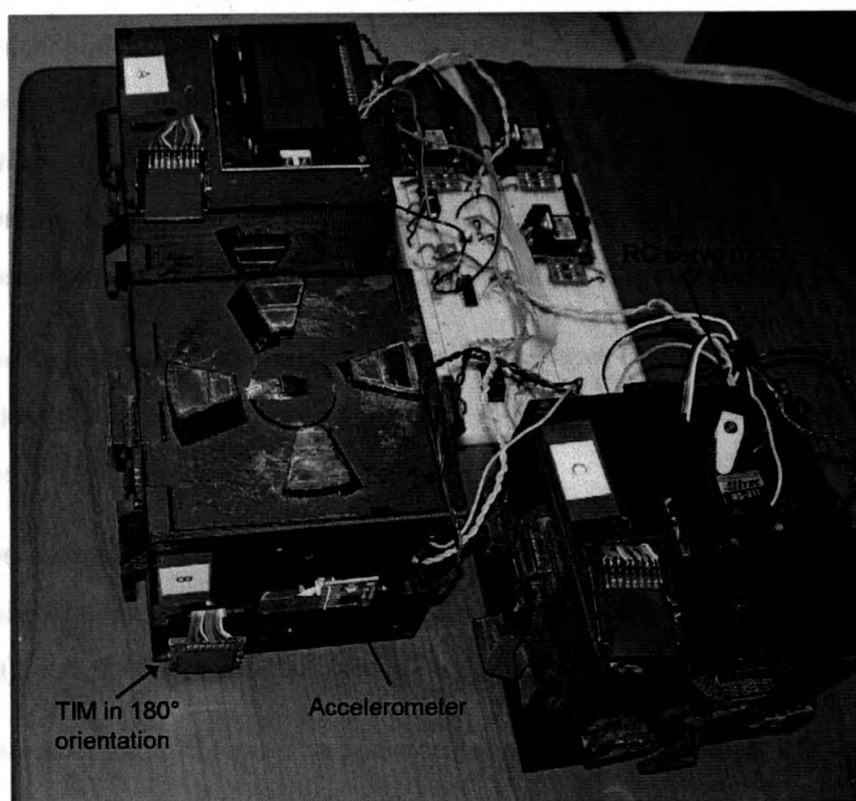


(c) 90° position of the servo head



(d) 135° position of the servo head

Figure 4.2: Composite group consisting of accelerometer, LDR, and LCD.



(e) 180° position of the servo head

Figure 4.2: Composite group consisting of accelerometer, LDR, and LCD.

## 4.2 Rapid Reconfigurability Feature

As discussed in Section 3.4, the functionality of a composite system may be dependent on the physical connection between the modules forming the system. Based on how the modules are physically connected, the position and orientation (pose) information is calculated for each module present in the system. The pose information may then be used to determine the functionality of the composite system. The functionality can be changed by simply changing the physical connection between the modules. The following example demonstrates this feature and hence proves that the TIMs can be rapidly reconfigured for various applications by changing the physical connection between the modules.

This example also assesses the proper operation of the following features of the system:

- Specific interface types on transducer interface layer (hardware architecture stack) which included: the general purpose I/O (GPIO) interface, and the SD card interface .
- Operation of the face connectors which included: the face connector interface layer of the hardware architecture, the design of face connectors (electrical and mechanical), and the face connector protocol.
- Determination of the position and orientation (pose information) of the connected modules.
- The knowledge representation scheme used to describe the functionality and capabilities of the modules.
- The wireless transceiver.

### 4.2.1 Experimental Set-up

In this example, a display system was constructed using two TIMs, each interfaced to a liquid crystal display (LCD) device. The display format of each LCD device is 16 characters  $\times$  4 lines and is compatible with the HD44780 Character LCD, an industry standard. Each device was connected to the general purpose I/O (GPIO) interface of transducer interface layer (hardware architecture stack).

The knowledge representation scheme was described in order to allow the LCD modules to store and communicate their functionality and capabilities. Each LCD module was assigned a unique identity by basic and adaptive modular sensing system (AMSS) TEDS. A standard template describing the attributes of LCD was also defined. The LCD TEDS are presented in Appendix A.5. Also, a LCD Merge Template TEDS (Appendix A.6) was defined for the composite group. This TEDS contains an additional field referred to as Roles, which is utilized by the software architecture to determine the type of sensors and actuators that can form a composite group. The overall TEDS information was stored in the SD card interfaced to the transducer interface layer.

The distributed software architecture executed on LCD modules enabled the automatic acquisition of configuration-specific algorithms. These algorithms facilitated the assumption of a collective identity by a composite group of LCD modules. The collective identity was assigned on the basis of standard information communicated by the knowledge representation scheme.

The overall display format was extended by connecting the modules side-by-side or above-and-below. In case of the side-by-side configuration, the display format could be increased to 32 characters  $\times$  4 lines. On the other hand, the display format of system could be extended to 16 characters  $\times$  8 lines by connecting the modules above-and-below.

The physical connection between the LCD modules determined the overall pose information of the system. This information was then utilized by the configuration-specific-



algorithm to establish the functionality (side-by-side or above-and-below) of the system. The software architecture assigned the TIM with lowest module address as the group leader. The group leader served as the reference for calculation of the system pose.

### 4.2.2 Discussion

#### Side-by-side Configuration (32 characters $\times$ 4 lines)

In this case a display system was constructed by connecting two LCD modules side-by-side to increase the display format to 32 characters  $\times$  4 lines. Figure 4.3 shows the modules connected in side-by-side configuration. The faces that came into contact when modules were connected in this configuration were: Face 2 of the left module and Face 4 of the right module. Both the modules executed the face connection protocol, discussed in Section 3.4, and determined and exchanged the the pose information. The group leader, based on the received pose information, determined whether the other module was to the right or the left.

#### Case 1: Module A on left

Figure 4.4 shows the side-by-side configuration with Module A on the left. The initial state of the modules in this configuration is shown in Figure 4.4a. On initialization, either module could assume the identity of group leader (based on the module assigned the lowest address). The software architecture assigns functionality to the system based on the calculated pose. This information is shown in Table 4.1. In the first case, Module A assumes the role of group leader and calculates the position and orientation of Module B. Since Module B is on the right, its position with respect to the centre of Module A is +120 mm. In the second case, Module B assumes the role of group leader and calculates the position of Module A with respect to its centre as -120 mm. In both cases the modules are connected in the same orientation. The face information including: the module address, the face number, and the orientation is determined and communicated by

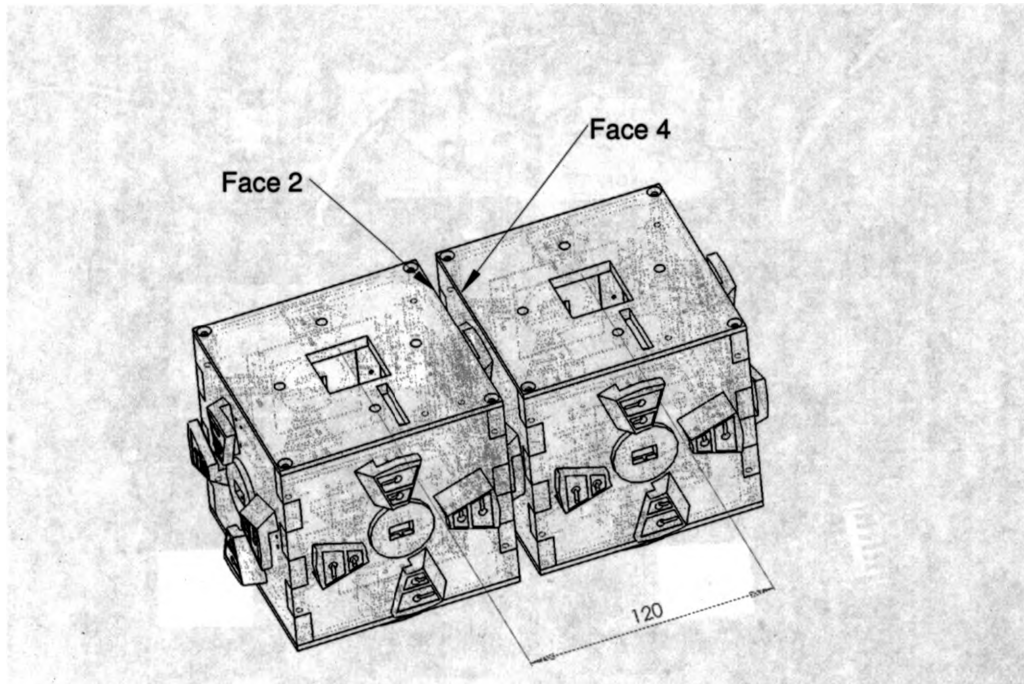
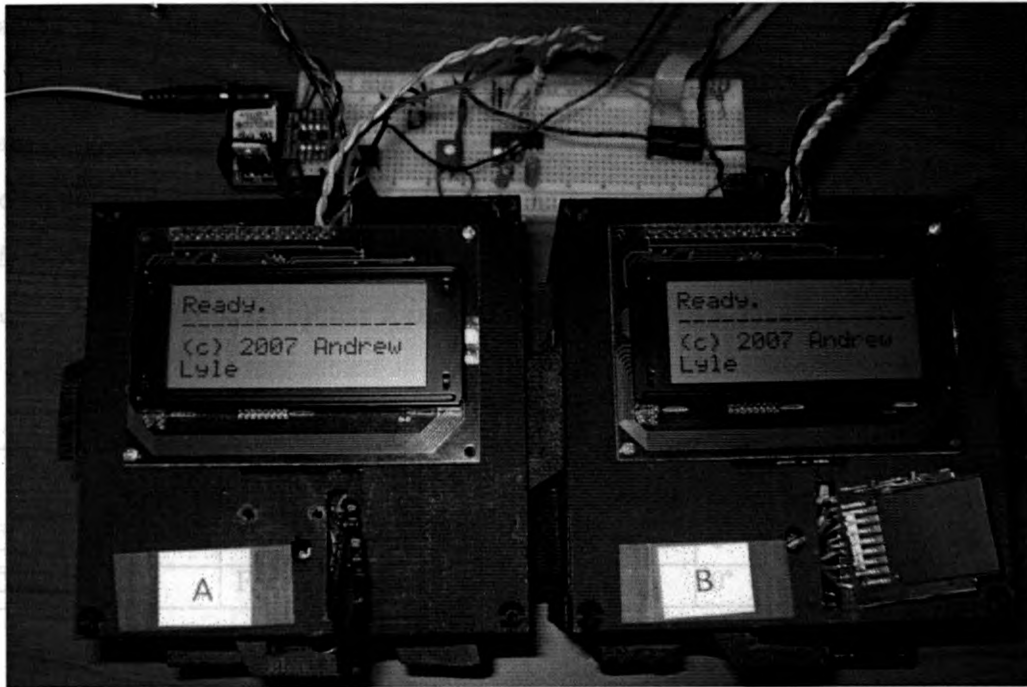


Figure 4.3: Modules connected in a side-by-side configuration.

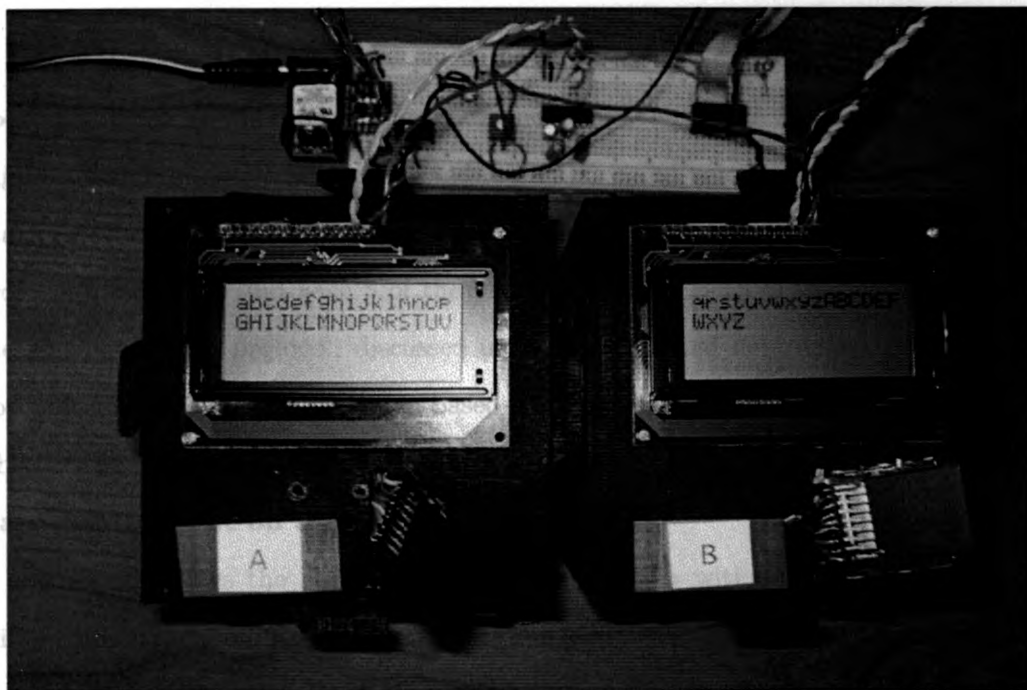
the face connector protocol. The face number information is then utilized to determine the position information. In this way, the overall pose information is calculated for the system. The software architecture assigns side-by-side display functionality to the modules, thus extending the display format to 32 characters  $\times$  4 lines. The extended display format is shown in Figure 4.4b. The output remains the same in both cases.

Table 4.1: Side-by-side configuration (Module A on left).

	Group leader	Left face	Right face	Position w.r.t. leader	Orientation w.r.t. leader	Display sequence
Case 1 a	A	Face 2	Face 4	+120 mm	0°	Left to Right
Case 1 b	B	Face 2	Face 4	-120 mm	0°	Left to Right



(a) Initial state of the system.



(b) Extended display system.

Figure 4.4: 32 characters  $\times$  4 lines display system (Module A on left).

### Case 2: Module B on left

Figure 4.5 shows the side-by-side configuration with module B on left. The initial state of the modules in this configuration is shown in Figure 4.5a. Again, the software architecture assigns a collective identity based on calculated pose information. The pose information is determined in a similar manner as discussed in Case 1 and represented in Table 4.2. The extended display format is shown in Figure 4.5b.

Table 4.2: Side-by-side configuration (Module B on left).

	Group leader	Left face	Right face	Position w.r.t. leader	Orientation w.r.t. leader	Display sequence
Case 2 a	A	Face 2	Face 4	-120 mm	0°	Left to Right
Case 2 b	B	Face 2	Face 4	+120 mm	0°	Left to Right

### Above-and-below Configuration (16 characters $\times$ 8 lines)

In this case, a display system was constructed by connecting two LCD modules in the above-and-below configuration to increase the display format to 16 characters  $\times$  8 lines. Figure 4.6 shows the modules connected in an above-and-below configuration. The faces that came into contact when modules were connected in this configuration were: Face 3 of the top module and Face 5 of the bottom module. Both of the modules executed the face connection protocol, discussed in Section 3.4, and determined and exchanged pose information. The group leader, based on the received pose information, determined whether the other module was above or below.

### Case 3: Module A on top

Figure 4.7 shows the above-and-below configuration with Module A on top. The initial state of the modules in this configuration is shown in Figure 4.7a. On initialization, either module could assume the identity of group leader (based on the module assigned the lowest address). The software architecture assigns functionality to the system based



(a) Initial state of the system.



(b) Extended display system.

Figure 4.5: 32 characters  $\times$  4 lines display system (Module B on left).

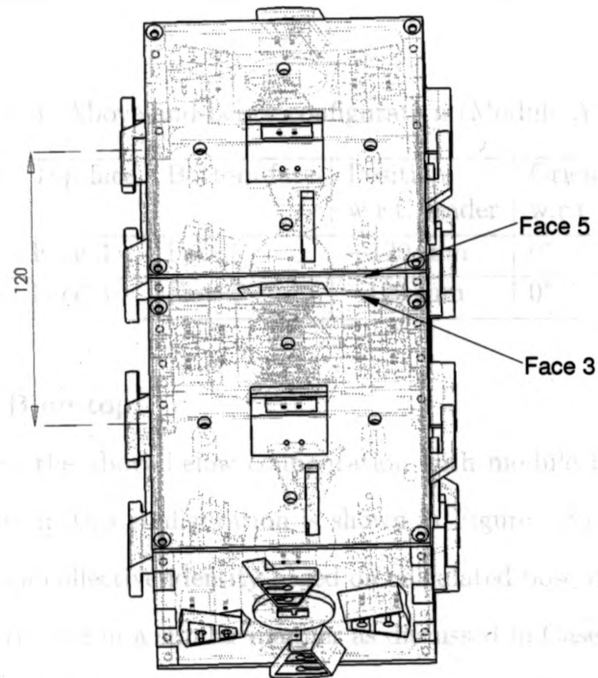


Figure 4.6: Modules connected in an above-and-below configuration.

on the calculated pose. This information is shown in Table 4.3. In the first case, Module A assumes the role of group leader and calculates the position and orientation of Module B. Since Module B is below, its position with respect to the centre of Module A is -120 mm. In the second case, Module B assumes the role of group leader and calculates the position of Module A with respect to its centre as +120 mm. In both cases, the modules are connected in the same orientation. The face information including: the module address, the face number, and the orientation is determined and communicated by the face connector protocol. The face number information is then used to calculate pose information for the system. The software architecture assigns above-and-below display functionality to the modules, thus extending the display format to 16 characters  $\times$  8 lines.

The extended display format is shown in Figure 4.7b. The output remains the same in both cases.

Table 4.3: Above-and-below configuration (Module A on top).

	Group leader	Top face	Bottom face	Position w.r.t. leader	Orientation w.r.t. leader	Display sequence
Case 3 a	A	Face 3	Face 5	+120 mm	0°	Top to bottom
Case 3 b	B	Face 3	Face 5	-120 mm	0°	Top to bottom

#### Case 4: Module B on top

Figure 4.8 shows the above-below configuration with module B on top. The initial state of the modules in this configuration is shown in Figure 4.8a. Again, the software architecture assigns a collective identity based on calculated pose information. The pose information is determined in a similar manner as discussed in Case 4 and represented in Table 4.4. The extended display format is shown in Figure 4.8b.

Table 4.4: Above-below configuration (Module B on top).

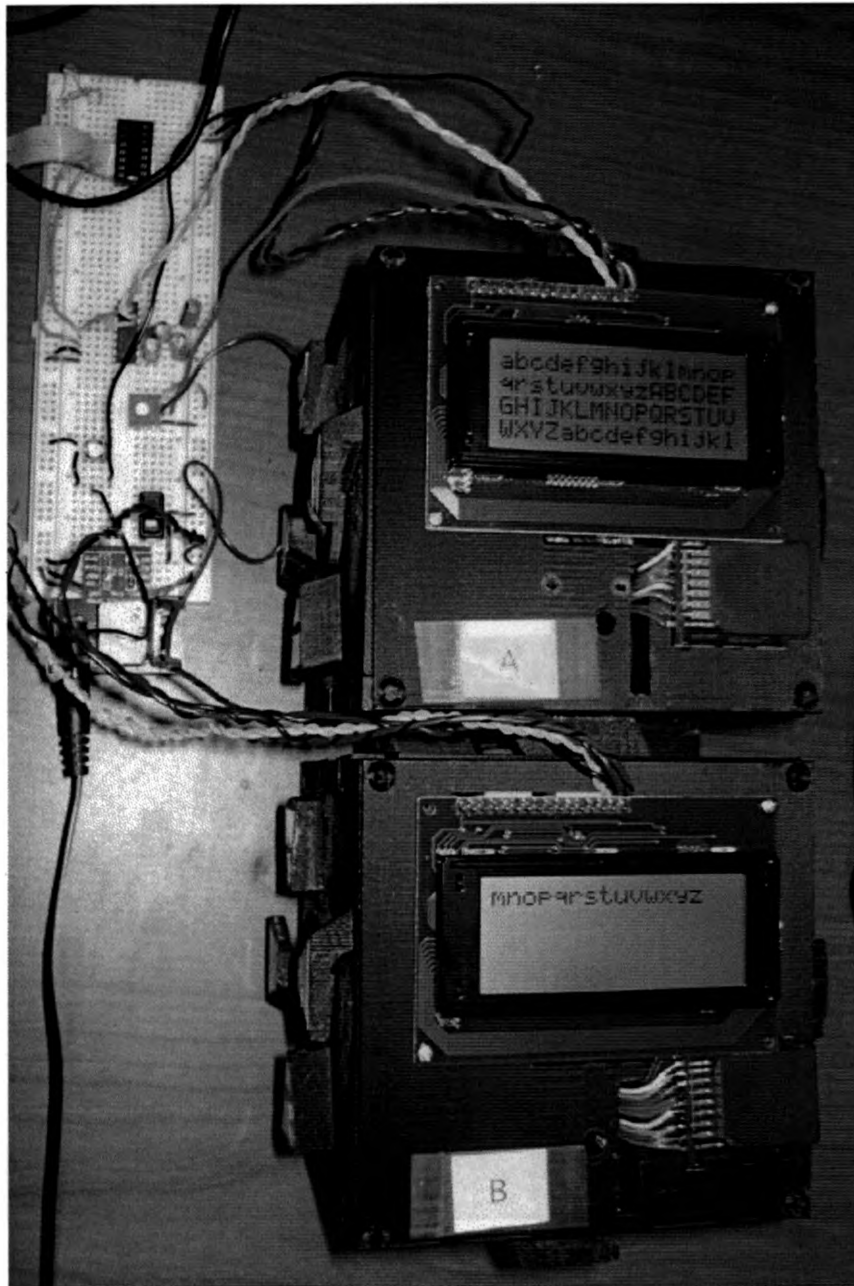
	Group leader	Top face	Bottom face	Position w.r.t. leader	Orientation w.r.t. leader	Display sequence
Case 4 a	A	Face 3	Face 5	-120 mm	0°	Top to bottom
Case 4 b	B	Face 3	Face 5	+120 mm	0°	Top to bottom



(a) Initial state of the system.

Figure 4.7: 16 characters  $\times$  8 lines display system (Module A above).





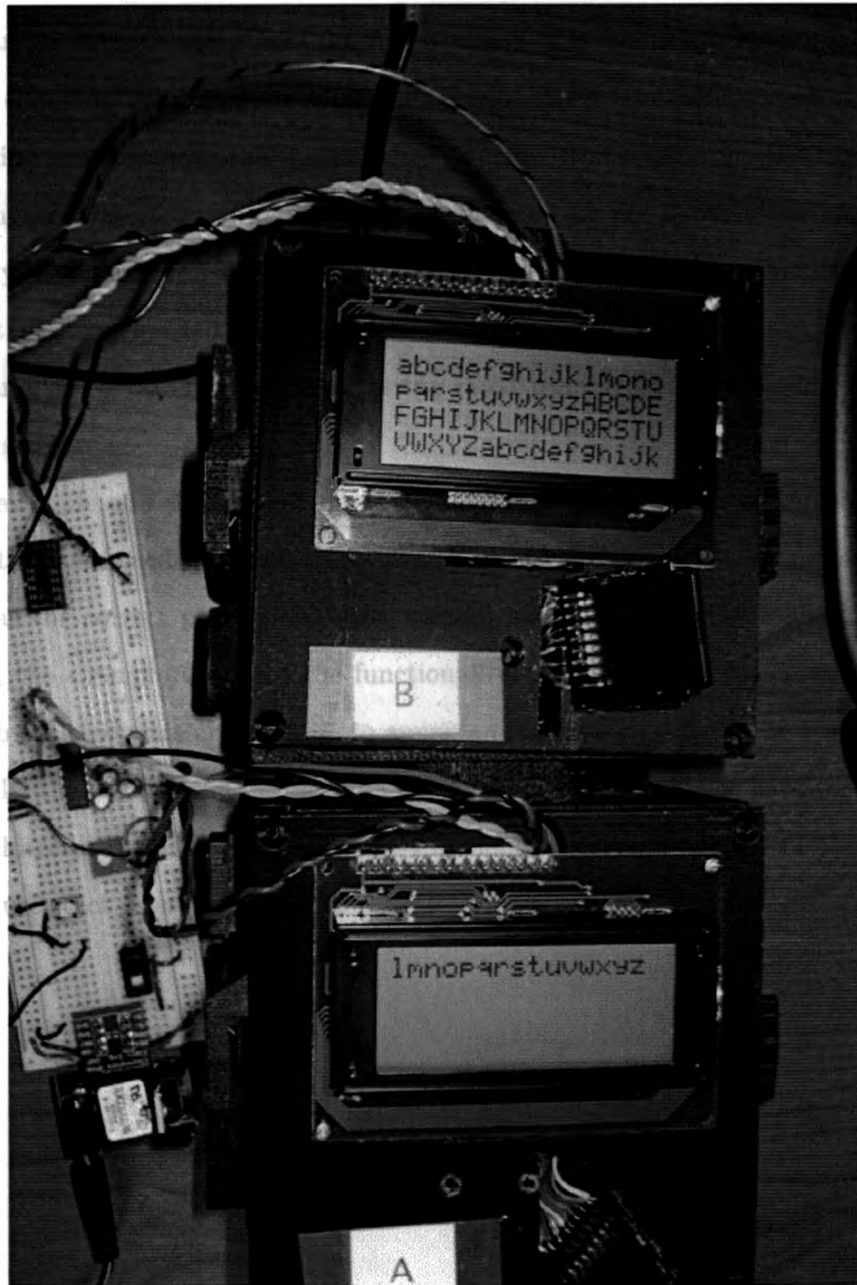
(b) Extended display system.

Figure 4.7: 16 characters  $\times$  8 lines display system (Module A above).



(a) Initial state of the system.

Figure 4.8: 16 characters  $\times$  8 lines display system (Module B above).



(b) Extended display system.

Figure 4.8: 16 characters  $\times$  8 lines display system (Module B above).

### 4.2.3 Summary

This example demonstrated how the functionality of a composite system could be altered by changing the physical connection between the transducer interface modules. In side-by-side configuration, the functionality remained the same when the modules were interchanged. The modules were rapidly reconfigured for the above-below configuration by simply changing the physical connection between the modules. Again in this case, the functionality remained the same when the position of the modules was interchanged. This example successfully demonstrates the rapid reconfigurability feature of the composite systems.

The example also proves that the system assumes a collective identity based on the pose of the connected modules. The operation of the face connector mechanism was verified as the correct pose information was calculated in all of the configurations. In addition, the example validated the functionality of the wireless transceiver which was utilized by the software architecture to assign a collective identity to the system.

Finally, the example established the successful integration of the hardware and software architectures and verified that the functioning of various layers of the hardware architecture stack operated properly.

## Chapter 5

# Conclusions and Recommendations

This thesis presented a set of intelligent modular blocks (building blocks), referred to as transducer interface modules (TIMs), from which composite sensors (made up of multiple sensor and actuator components) can be rapidly reconfigured for the construction of Adaptive Modular Sensing Systems. Specifically, the following objectives were considered:

1. To design and develop a modular hardware architecture that can be interfaced to a multitude of sensors and actuators.
2. To integrate the modular hardware platform with an independent distributed software control architecture.
3. To develop modules that encapsulate the hardware architecture and form the basic building blocks of an adaptive modular sensing system.
4. To design and develop a face connector mechanism to enable the physical connection of the transducer interface modules.
5. To implement schemes to enable a module to determine its position and orientation with respect to other modules present in the composite system.

6. To specify a knowledge representation scheme to allow the various modules in a composite system to communicate standard information required by a logical algorithm to assign a collective identity to the group.
7. To design and develop additional modules which may be required to make the composite system completely functional.

Each of these objectives were considered and implemented to some degree. As specified, transducer interface modules form the building blocks of adaptive modular sensing systems. The blocks encapsulate a general purpose hardware architecture that can be interfaced to a multitude of sensors and actuators. A face connector mechanism has been designed and implemented on the modules to enable the physical connection of the building blocks. A knowledge representation scheme has been defined to allow the modules to store and communicate their functionality and capabilities.

## 5.1 Summary and Conclusions

TIMs are a set of intelligent building blocks, created for the development of systems and strategies by which sensor and actuator components can be combined to produce flexible and robust sensor systems. These blocks can be rapidly reconfigured for various applications including industrial control, inspection systems, mobile robotics, monitoring, and data acquisition. TIMs contain embedded knowledge about their capabilities and how they can interact with other modules. Once the modules are combined, the capabilities (e.g., range, resolution, sample rate, etc.) and functionality (e.g., temperature measurement) of the composite sensor is determined and communicated to other sensors in the environment.

TIMs encapsulate a general purpose hardware architecture that has been designed and developed to provide an interface between the sensors, the actuators, and the com-

munication medium. The architecture is a stackable system consisting of four layers: a controller/communication layer, a face connector interface layer, a power supply layer, and a transducer interface layer. The architecture is scalable, modular, and can be interfaced to a multitude of sensors and actuators.

A face connector mechanism has been designed and developed for the TIMs to enable the physical connection of the building blocks. The face connectors allow the modules to be connected in four different orientations (90° apart). A face connector protocol has been specified to determine the position and the orientation of the connected modules. The face connectors also implement a power bus that can be shared between the connected modules.

A knowledge representation scheme has been specified that allows each module to store and communicate its functionality and capabilities to other modules in the system. The definition of this scheme is similar to that of the Transducer Electronic Datasheet (TEDS) presented in the IEEE 1451 standards. The knowledge representation scheme includes basic and AMSS TEDS that assign a unique identity to each module. Additional templates (standard as well as user-defined) are included to define the various attributes of transducers. The knowledge representation scheme is designed to enable the various modules to communicate standard information required by a logical algorithm in order to assign a collective identity to the composite system.

The distributed software architecture [9] executed on the blocks enables automatic acquisition of configuration-specific algorithms. The logical algorithm imparts a collective identity to the composite group, and processes data based on the standard information communicated by the knowledge representation scheme.

An application programming interface (API) has been defined to provide a set of smart functions that are utilized by the distributed software architecture running on the top of the hardware architecture to access the hardware drivers. The hardware drivers

shield the details of the underlying hardware architecture from the distributed software architecture. Hardware drivers have been defined for various sensors and actuators.

A set of additional modules has been specified which may be required to make a composite system completely functional. These include: interconnect modules, power modules, and administrative modules. More complex assemblies of modules can be created using interconnect modules. These modules may be available in various shapes and provide a variety of angular and translational offsets. Power modules may be used to power the various modules in a system. These modules may be based on rechargeable batteries or derive power from a wall outlet and include additional circuitry to maintain a constant output voltage. The administrative module may be viewed as a system manager used to allow a user to interact with and control TIMs in its vicinity through a graphical user interface.

## 5.2 Recommendations

The current TIM prototypes are large, expensive, and are typically suited for simple applications. A number of features can be included or changed to make the existing prototype more capable and suited for large-scale production.

The time and cost of manufacturing the modules can be significantly cut down by reducing the size of the module. The overall size of the module is dependent on the dimensions of the encapsulated modular hardware stack. In order to minimize the size of the module, the dimensions of the hardware stack need to be reduced. The hardware stack consists of two-layer PCB boards, populated with dual in-line package (DIP) components.

A major disadvantage of DIP components is that they occupy more board area as compared to a number of available packages. As the pin count of a DIP increases, its PCB space utilization efficiency drops sharply [49]. On the other hand, the surface mountable



packages such as small outlines (SOs), thin small outlines (TSOPs), quad flat packages (QFPs), leaded and leadless chip carriers (plastic leaded chip carrier (PLCC), leaded ceramic chip carrier (LCCC)), and ball grid arrays (BGAs) are relatively thinner and smaller and occupy much lesser board area as compared to standard DIP components. As an example, a small outline package (SOP) component is roughly one-eighth the size of its standard DIP equivalent and occupies almost 30–50% less board area.

Multi-layer boards provide a number of advantages over two-layer boards, including increased space utilization and a corresponding reduction in physical size. A multi-layered board enables the implementation of dedicated ground planes, thereby reducing the complexity of routing, and avoiding ground loops. The power and ground connections are available at any location on the board simply by connecting through to the appropriate layer. Thus, space on the board which would otherwise be required for power and ground buses is available for other uses. Also, power distribution can be managed efficiently with multi-layered boards due to the fact that different supplies can be routed as different planes. A combination of surface mountable components and multi-layer PCB boards can result in a much smaller hardware stack, thereby reducing the size of the module considerably.

Additional hardware circuits can be included in the modular stack design in order to provide a smooth transition between internal (e.g., batteries) and external power sources (e.g., a shared power module). A supervisory circuit with battery backup can be used for this purpose. The circuit can detect and monitor an external power source and in the absence of any external source switch to a backup battery. Figure 5.1 shows a block diagram of this circuit. The output of the circuit is directed to the voltage regulators to obtain 3.3 V and 5 V output.

An important aspect of this work is to implement a generalized hardware architecture that can be interfaced to a multitude of sensors and actuators with minimum overhead.

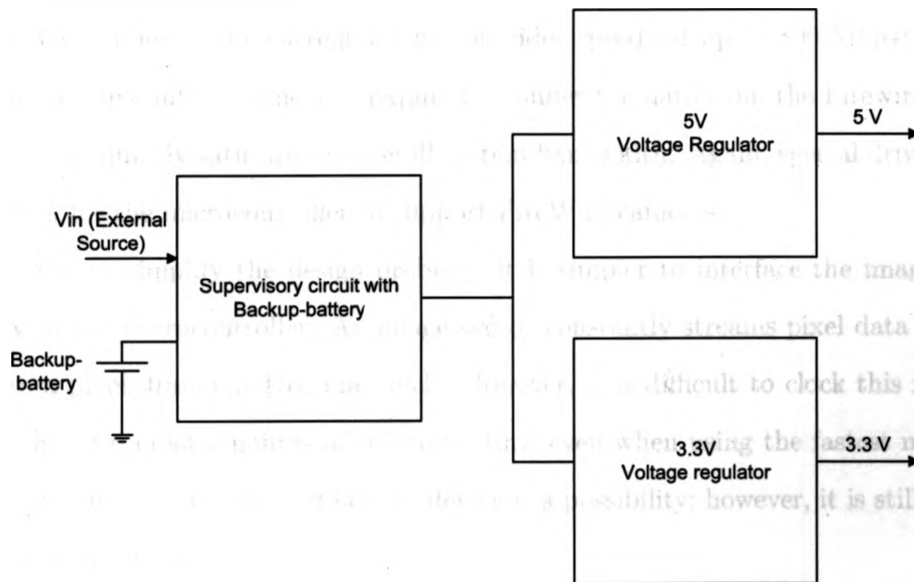


Figure 5.1: Supervisory circuit with backup-battery.

This includes high-end devices such as cameras, providing a resolution of at least  $640 \times 480$ . The choices for cameras include: USB, Firewire, and embedded cameras.

A USB camera requires an additional controller (e.g., MAX3420E) in order to be interfaced to the microcontroller. However, USB camera requires a custom driver to be installed on the microcontroller when plugged in for the first time. While Windows includes special built-in drivers for USB devices, there are no such standard drivers available for microcontrollers. Hence, a special driver has to be written for the microcontroller to support USB cameras. A possible solution is to reverse engineer the software of an available camera driver and then apply the same to the microcontroller. However, this may result in a comprehensive code requiring lot of memory.

The FireWire (IEEE 1394) cameras are easy to use but are better suited for PC based applications. The major limitation in interfacing the FireWire camera to the embedded

system is the inability of the existing microcontrollers to support the high speeds offered by FireWire cameras (the current version provides speeds of up to 800 Mbps). Also, a considerable amount of memory is required to buffer the data from the Firewire camera and they can quickly saturate the overall system bandwidth. Again, special driver has to be written for the microcontroller to support FireWire cameras.

In order to simplify the design problem, it is simpler to interface the image sensor directly to the microcontroller. An image sensor constantly streams pixel data synchronized to a pixel, frame and/or line clock. However, it is difficult to clock this raw data since it uses up great amounts of controller time even when using the fastest microcontrollers. Reducing the frame rate of the device is a possibility; however, it is still difficult to support the speed.

Figure 5.2 shows one possible solution for interfacing an image sensor directly to the microcontroller. The hardware interface includes a microcontroller and an external RAM memory. The microcontroller can be programmed to feed the data from the image sensor into RAM. The parameters of the image sensors can be controlled by a serial bus (such as I<sup>2</sup>C ).

The LPC2148 header board utilized for the prototype modular hardware stack has very limited RAM (32 Kbytes) with no provisions for extending the memory. The header board does not provide access to the break out pins for interfacing external RAM, hence the system imposes a memory limitation. This can be overcome by using the latest prototyping boards (LPC-E2214, LPC-E2294, LPC-P2378) which provide an external memory bus.

The diameter of the electrical contacts (spring loaded contacts) utilized for the face connectors was extremely small and hence redundant contacts had to be embedded in the face to ensure connectivity. This design can be made more reliable by using larger contacts.

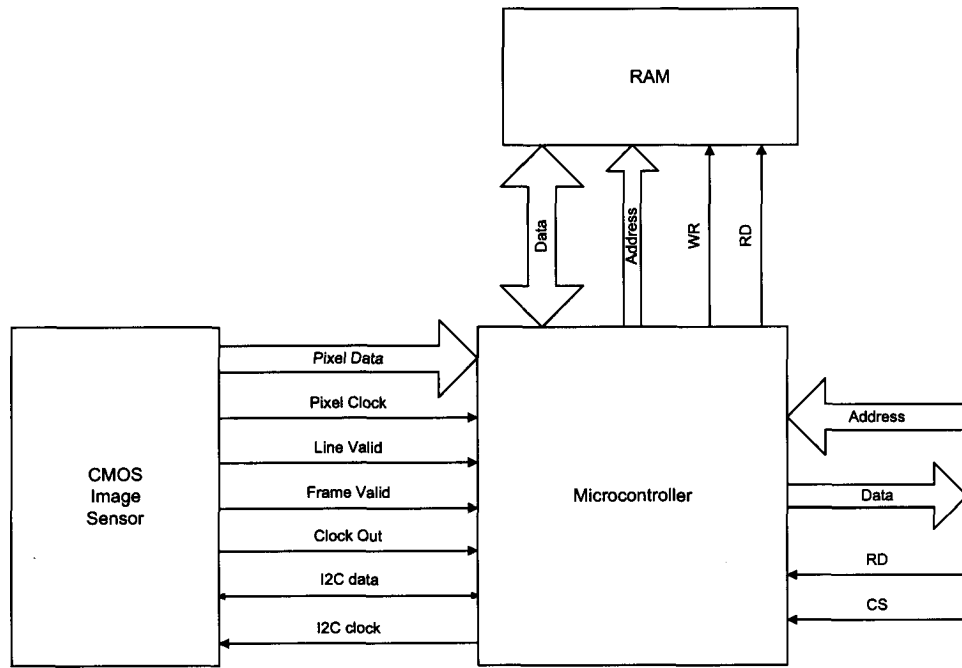


Figure 5.2: Camera interface.

The prototype face connector design supports a number of different assemblies of module. However, it is not suitable for building a dense assembly of modules, an example of which is shown in the Figure 5.3. This is due to the fact that the modules have to be held at an angle with respect to each other to facilitate locking. One possible way of overcoming this problem is to mount the clips on a rotating disc. In this case the disc, and not the module, needs to be rotated for a connection. This design would allow the modules to be tightly packed. However, it requires a more complex and possibly a motorized face assembly. The implementation of a motorized face assembly would also result in extending the number of configurations in which the modules can be connected. Hence, modules could be reconfigured for more complex applications.

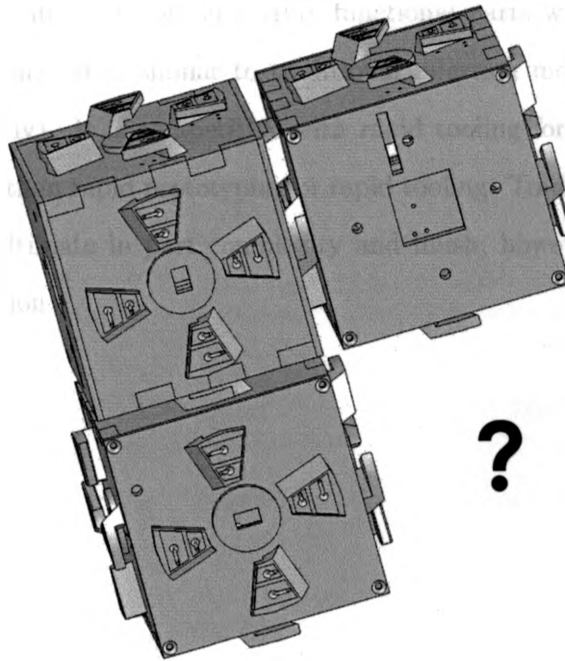


Figure 5.3: Limitation of the prototype face connector design.

The modules were manufactured using the fused deposition modeling (FDM) rapid prototyping technique. Although it is one of the quickest methods to reproduce very complex shapes, the tolerances are generally greater than 0.1 mm. The process is inexpensive as long as only a few parts are needed. However, the cost of rapid prototyping increases rapidly with the larger volumes of parts, as all the parts have to be made from scratch. Also, only a limited range of materials can be utilized for manufacturing and the parts are typically left with a coarse finish. Rapid tooling and rapid injection molding can sometimes produce better quality parts than rapid prototyping. Rapid tooling is slower and more costly than rapid prototyping due to the extra step required to create a tool from the original prototype. The need to create molds also increases up-front cost and

can limit the complexity of shapes that can be effectively duplicated. Rapid injection molding uses metal molds to produce truly functional parts with good finish and in a wide variety of resins. It is similar to traditional injection molding (though far faster and much less costly). It is competitive with rapid tooling for speed and offers better economies of scale than rapid prototyping or rapid tooling. Traditional injection molding can produce the ultimate in part complexity and finish; however, it is best suited for large-scale production.

## References

- [1] K. Lee, "IEEE 1451: A standard in support of smart transducer networking," in *Proceedings of the IEEE Instrumentation and Measurement Technology Conference*, vol. 2, (Baltimore, MD USA), pp. 525–528, May 1–4, 2000.
- [2] K.-F. Reinhart and M. Illing, *Sensors for automotive applications*, vol. 4, ch. Automotive sensor market, pp. 5–20. Wiley-VCH, 2003.
- [3] Crossbow Technology Inc., "Crossbow wireless sensor networks." [Online]. Available: [www.xbow.com](http://www.xbow.com), 2007.
- [4] T. D. Ngo and H. H. Lund, "Modular artefacts," in *Proceedings of the 18th European Conference on Object-Oriented Programming (ECOOP) 2004 workshop: components-oriented approaches to context-aware computing*, (Oslo, Norway), June 14–18, 2004.
- [5] H. H. Lund, "Intelligent Artefacts," in *Sugisaka and Tanaka (eds.) Proceedings of 8th International Symposium on Artificial Life and Robotics (ISAROB)*, (Oita, Japan), 2003.
- [6] S. Cotterell, K. Downey, and F. Vahid, "Applications and experiments with eBlocks — electronic blocks for basic sensor-based systems," in *Proceedings of the IEEE Sensor and Ad Hoc Communications and Networks (SECON)*, (Santa Clara, CA, USA), pp. 7–15, October 4–7, 2004.
- [7] S. Cotterell, R. Mannion, F. Vahid, and H. Hsieh, "eBlocks — an enabling technology for basic sensor based systems," in *Information Processing in Sensor Networks (IPSN) Track on Sensor Platform, Tools, and Design Methods for Networked Embedded Systems (SPOTS)*, (Los Angeles, CA, USA), pp. 422–427, April 2005.
- [8] "The Lego of Gadgets." [Online]. Available: <http://buglabs.net/products>, 2008.
- [9] A. C. Lyle, "A software architecture for adaptive modular sensing systems," Master's thesis, The University of Western Ontario, 2008.
- [10] M. Korsten and P. Regtien, "A knowledge based system for automatic selection of industrial sensors," in *Proceedings of the 14th International Conference on Systems Engineering (ICSE)*, vol. 1, (Coventry, UK), pp. 329–334, September 12–14, 2000.

- [11] H. Eren, *The Engineering Handbook*, ch. Sensors. CRC Press LLC, 2005.
- [12] R. M. White, "A sensor classification scheme," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 34, pp. 124–126, March 1987.
- [13] Intechno Consulting, "Sensor markets 2008." [Online]. Available: <http://www.intechnoconsulting.com/>, February 2008.
- [14] IEEE1451 committee, "IEEE standard 1451.2-1997, standard for a smart transducer interface for sensors and actuators — transducer to microprocessor communication protocols and transducers electronic data sheet (TEDS)," (Piscataway, NJ, USA), Institute of Electrical and Electronics Engineers, Inc., September 26, 1997.
- [15] Y. Zhang, Y. Gu, V. Vlatkovic, and X. Wang, "Progress of smart sensor and smart sensor networks," in *Proceedings of the fifth World Congress on Intelligent Control and Automation*, vol. 4, (Hangzhou, P.R. China), pp. 3600–3606, June 15–19 2004.
- [16] E. Jacobsen, "The building blocks of a smart sensor for distributed control networks," in *Proceedings of the IEEE Technical Applications Conference Northcon/96*, (Seattle, Washington, USA), pp. 285–290, November 4–6, 1996.
- [17] S. Soloman, *Sensors Handbook*, ch. A process for selecting a commercial sensor actuator sensor bus as an industry interoperable standard, pp. 29.1–29.15. McGraw-Hill Professional, 1998.
- [18] R. S. Gyurcsik, W. C. Lamb, and J. R. Moyne, "Sensor bus control networks in semiconductor processing equipment," in *Proceedings of the SPIE on Manufacturing Process Control for Microelectronic Devices and Circuits*, vol. 2336, (Austin, TX, USA), pp. 141–144, October 21, 1994.
- [19] Crossbow Technology Inc., "Motes, smart dust sensors, wireless sensor networks." [Online]. Available: [http://www.xbow.com/Products/Wireless\\_Sensor\\_Networks.htm](http://www.xbow.com/Products/Wireless_Sensor_Networks.htm), 2004.
- [20] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, "System architecture directions for networked sensors," in *Architectural Support for Programming Languages and Operating Systems*, (Cambridge, MA, USA), pp. 93–104, November 2000.
- [21] Crossbow Technology, Inc., "MICA2 — Wireless Measurement System." [Online]. Available: [www.xbow.com/Products/Product\\_pdf\\_files/Wireless\\_pdf/MICA2\\_Datasheet.pdf](http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICA2_Datasheet.pdf), 2007.
- [22] Crossbow Technology Inc., "IRIS Wireless Measurement System." [Online]. Available: [http://www.xbow.com/Products/Product\\_pdf\\_files/Wireless\\_pdf/IRIS\\_Datasheet.pdf](http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/IRIS_Datasheet.pdf), 2007.



- [23] Crossbow Technology Inc., "MICAz Wireless Measurement System." [Online]. Available: <http://www.davidson.com.au/products/wireless/crossbow/pdf/micz.pdf>, 2007.
- [24] Crossbow Technology Inc., "Imote2 High Performance Wireless Sensor Network Node." [Online]. Available: [http://www.xbow.com/Products/Product\\_pdf\\_files/Wireless\\_pdf/Imote2\\_Datasheet.pdf](http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/Imote2_Datasheet.pdf), 2007.
- [25] Crossbow Technology Inc., "MOTWORKS." [Online]. Available: [www.xbow.com/Products/Product\\_pdf\\_files/Wireless\\_pdf/MoteWorks\\_OEM\\_Edition.pdf](http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MoteWorks_OEM_Edition.pdf), 2007.
- [26] L. E. Holmquist, H.-W. Gellersen, G. Kortuem, S. Antifakos, F. Michahelles, B. Schiele, M. Beigl, and R. Maze, "Building intelligent environments with Smart-Its," *IEEE Computer Graphics and Applications*, vol. 24, pp. 56–64, Jan–Feb 2004.
- [27] L. E. Holmquist, F. Mattern, B. Schiele, P. Alahuhta, M. Beigl, and H.-W. Gellersen, "Smart-Its Friends: A technique for users to easily establish connections between smart artefacts." [Online]. Available: <http://www.vs.inf.ethz.ch/res/papers/smf.pdf>, 2006.
- [28] N. Edmonds, D. Stark, and J. Davis, "MASS: modular architecture for sensor systems," in *Proceedings of the Fourth International Symposium on Information Processing in Sensor Networks*, (Los Angeles, CA, USA), pp. 393–397, April 25–27, 2005.
- [29] D. Lymberopoulos, N. B. Priyantha, and F. Zhao, "mPlatform: A Reconfigurable Architecture and Efficient Data Sharing Mechanism for Modular Sensor Nodes," in *Proceedings of the 6th international conference on Information processing in sensor networks (IPSN)*, no. 128–137, (Cambridge, MA, USA), April 25–27, 2007.
- [30] M. G. Gorbet and M. Orth, "Triangles: Design of a physical/digital construction kit," in *Proceedings of the second conference on Designing interactive systems: processes, practices, methods, and techniques*, (Amsterdam, The Netherlands), pp. 125–128, August 18–20, 1997.
- [31] D. Anderson, J. L. Frankel, J. Marks, D. Leigh, K. Ryall, E. Sullivan, and J. Yedidia, "Building Virtual Structures With Physical Blocks," in *Proceedings of the 12th annual ACM symposium on User interface software and technology*, (Asheville, NC, USA), November 7–10, 1999.
- [32] A. Goldenberg, N. Kircanski, M. Kircanski, and A. Seshan, "Reconfigurable modular joint and robots produced therefrom, U.S. Patent #6,084,373." [Online]. Available: <http://www.patentstorm.us/patents/6084373/claims.html>, July 2000.

- [33] Philips Semiconductors, "LPC2141/42/44/46/48 Single-chip 16-bit/32-bit microcontrollers; up to 512 kB flash with ISP/IAP, USB 2.0 full-speed device, 10-bit ADC and DAC," August 2006.
- [34] Sparkfun Electronics, "Header board for LPC2148." [Online]. Available: <http://www.sparkfun.com/>, December 2006.
- [35] Embedded Systems Academy, "Flash Magic Programming tool sponsored by NXP Semiconductors." [Online]. Available: <http://www.flashmagictool.com/>, September 2007.
- [36] Nordic Semiconductor, "Single chip 2.4 GHz transceiver nRF24L01," March 2006.
- [37] D. Zhao and C. Peng, "A small low-power reliable communication module in a wireless monitoring system," in *Proceedings of the first International Conference of Bioinformatics and Biomedical Engineering (ICBBE)*, (Wuhan, China), pp. 1194–1197, July 6–8, 2007.
- [38] RF Globalnet, "Nordic 2.4 GHz transceivers used to display and navigate content stored on iPods." [Online]. Available: <http://www.rfglobalnet.com/Content/news/article.asp>, April 14 2007.
- [39] Fairchild Semiconductor, "CD4066BC Quad Bilateral Switch." [Online]. Available: <http://www.fairchildsemi.com/ds/CD/CD4066BC.pdf>, October 2005.
- [40] Microchip, "Single-Ended, Rail-to-Rail I/O, Low Gain PGA MCP6S21/2/6/8." [Online]. Available: <http://www.microchip.com/>, March 2007.
- [41] Motorola, Inc., "SPI block guide V03.06," February 2003.
- [42] Philips Semiconductor, "The I<sup>2</sup>C-bus specification," January 2000.
- [43] Mill-Max, "Spring loaded connectors." [Online]. Available: [www.mill-max.com](http://www.mill-max.com), January 2008.
- [44] F. Pospiech and S. Olsen, "Embedded software in the SoC world. How HdS helps to face the HW and SW design challenge," in *Proceedings of the IEEE 2003 Custom Integrated Circuits Conference*, (San Jose, CA, USA), pp. 653–658, September 21–24, 2003.
- [45] J. Blumenthal, F. Golatowski, M. Haase, and M. Handy, *Embedded systems handbook*, ch. Software development for large-scale wireless sensor networks, pp. 40–1–40–26. CRC Press, 2006.
- [46] K. Reddy, C.S. Reddy K, and P. Reddy, *Knowledge based computer systems: International conference KBCS '89*, ch. Implementation of conceptual graphs using frames in lead, pp. 213–229. Springer-Verlag, 1989.

- [47] National Instruments, "An overview of IEEE 1451.4 Transducer Electronic Datasheets (TEDS)." [Online]. Available: [http://standards.ieee.org/regauth/1451/IEEE\\_1451d4\\_Templates\\\_Tutorial\\_061804.pdf](http://standards.ieee.org/regauth/1451/IEEE_1451d4_Templates\_Tutorial_061804.pdf).
- [48] STMicroelectronics, "LIS3LV02DQ MEMS inertial sensor 3-Axis - 2g/6g digital output low voltage linear accelerometer." [Online]. Available: <http://www.st.com/stonline/products/literature/ds/11115/lis3lv02dq.pdf>, July 2007.
- [49] E. R. Hnatek, *Integrated circuit quality and reliability*, ch. Back-end fabrication operations, pp. 99–176. CRC Press, 1999.

# Appendix A

## Transducer Electronic Datasheets (TEDS)

This chapter presents the various TEDS used in Chapter 4. The basic and AMSS TEDS assign a unique identity to each module, while the standard templates describe the attributes of transducers.

## A.1 Accelerometer TEDS

In this section, the accelerometer TEDS is defined. Basic and AMSS TEDS assign unique identity to the TIM. The standard template defines attributes of the accelerometer.

```
# Accelerometer Module TEDS File
```

```
# -----
```

```
# AMSS TEDS
```

```
# -----
```

```
ModuleAddress      1000000000000001
ModuleType          1    # Sensor
ModuleClass         10    # Acceleration
ModuleDataType      100  # Float
ModuleDataTypeWidth  3    #
ModuleDataTypeHeight 1
PrimaryHandlerName  accel
```

```
# Basic TEDS
```

```
# -----
```

```
ManufacturerID      1
ModelNumber          1
VersionLetter        A
VersionNumber        1
SerialNumber         1
```

```
# Standard Template TEDS
```

```
# -----
```

```
TemplateID          30    # Voltage Template
ElecSigType          0    # Voltage Sensor
MapMeth              0    # Linear
SelectExcitation     1    # Include Excitation/Power Requirements
ExciteAmplNom        3.3  # Volts
ExciteAmplMin        3.3  # Volts
ExciteAmplMax        3.3  # Volts
ExciteType           0    # DC
ExciteCurrentDraw    1.0  # Amps
CalDate              1
CalInitials          NUL
CalPeriod            365  # Days
MeasID               1
```

## A.2 LDR TEDS

In this section, the LDR TEDS is defined. Basic and AMSS TEDS assign unique identity to the TIM. The standard template defines attributes of the LDR.

```
# LDR Module TEDS File
```

```
# -----
```

```
# AMSS TEDS
```

```
# -----
```

```
ModuleAddress      10000000000000000006
```

```
ModuleType         1   # Sensor
```

```
ModuleClass        8   # Voltage
```

```
ModuleDataType     100 # Float
```

```
ModuleDataTypeWidth 1
```

```
ModuleDataTypeHeight 1
```

```
PrimaryHandlerName ldr
```

```
# Basic TEDS
```

```
# -----
```

```
ManufacturerID     1
```

```
ModelNumber        2
```

```
VersionLetter      A
```

```
VersionNumber      1
```

```
SerialNumber       2
```

```
# Standard Template TEDS
```

```
# -----
```

```
TemplateID         30 # Voltage Template
```

```
ElecSigType        0 # Voltage Sensor
```

```
MapMeth            0 # Linear
```

```
SelectExcitation   1 # Include Excitation/Power Requirements
```

```
ExciteAmplNom      3.3 # Volts
```

```
ExciteAmplMin      3.3 # Volts
```

```
ExciteAmplMax      3.3 # Volts
```

```
ExciteType         0 # DC
```

```
ExciteCurrentDraw  1.0 # Amps
```

```
CalDate            1
```

```
CalInitials        NUL
```

```
CalPeriod          365 # Days
```

```
MeasID             1
```

## A.3 Servo motor TEDS

In this section, the servo motor TEDS is defined. Basic and AMSS TEDS assign unique identity to the TIM. The standard template defines attributes of the servo motor.

```
# Servo Module TEDS File
# -----

# AMSS TEDS
# -----
ModuleAddress      10000000000000007
ModuleType         2   # Actuator
ModuleClass        20   # Rotation
ModuleDataType     100 # Float
ModuleDataTypeWidth 1
ModuleDataTypeHeight 1
PrimaryHandlerName servo

# Basic TEDS
# -----
ManufacturerID     1
ModelNumber        3
VersionLetter      A
VersionNumber      1
SerialNumber       3

# Standard Template TEDS
# -----
TemplateID         30 # Voltage Template
ElecSigType        0 # Voltage Sensor
MapMeth            0 # Linear
SelectExcitation   1 # Include Excitation/Power Requirements
ExciteAmplNom      3.3 # Volts
ExciteAmplMin      3.3 # Volts
ExciteAmplMax      3.3 # Volts
ExciteType         0 # DC
ExciteCurrentDraw  1.0 # Amps
CalDate            1
CalInitials        NUL
CalPeriod          365 # Days
MeasID             1
```

## A.4 Servo Control Template TEDS

In this section, TEDS is defined for composite group made up of an accelerometer TIM, an LDR TIM, and a servo motor TIM. In addition to the basic and AMSS TEDS and the standard template, a Roles template is defined. The Roles template is used by the software architecture to identify the types of sensors and actuators that can be used to form a composite group.

```
# Servo Control Template TEDS File
```

```
# -----
```

```
# AMSS TEDS
```

```
# -----
```

```
ModuleType          2  # Actuator
ModuleClass         20  # Rotation
ModuleDataType     100 # Float
ModuleDataTypeWidth  1
ModuleDataTypeHeight 1
```

```
# Basic TEDS
```

```
# -----
```

```
ManufacturerID      1
ModelNumber         5
VersionLetter       A
VersionNumber       1
SerialNumber        5
```

```
# Standard Template TEDS
```

```
# -----
```

```
TemplateID          30  # Voltage Template
ElecSigType         0  # Voltage Sensor
MapMeth             0  # Linear
SelectExcitation    1  # Include Excitation/Power Requirements
ExciteAmplNom       3.3 # Volts
ExciteAmplMin       3.3 # Volts
ExciteAmplMax       3.3 # Volts
ExciteType          0  # DC
ExciteCurrentDraw   1.0 # Amps
CalDate             1
CalInitials         NUL
CalPeriod           365 # Days
MeasID              1
```



## # Roles

# -----

```
Role 1
RoleAssignmentLimit >=1
RoleConnectionType 7 # Any
RoleModuleType 1 # Sensor
RoleModuleClass 18 # Acceleration/Voltage
RoleModuleDataType 100 # Float
RoleModuleDataTypeWidth >=1
RoleModuleDataTypeHeight ==1
```

```
Role 2
RoleAssignmentLimit >=1
RoleConnectionType 7 # Any
RoleModuleType 2 # Actuator
RoleModuleClass 20 # Rotation
RoleModuleDataType 100 # Float
RoleModuleDataTypeWidth ==1
RoleModuleDataTypeHeight ==1
```

## A.5 LCD TEDS

In this section the LCD TEDS is defined. Basic and AMSS TEDS assign unique identity to the TIM. The standard template defines attributes of the LCD.

```
# LCD Module TEDS File
# -----
```

```
# AMSS TEDS
# -----
```

```
ModuleAddress      10000000000000005
ModuleType         2   # Actuator
ModuleClass        4   # Text
ModuleDataType     800 # String
ModuleDataTypeWidth 16
ModuleDataTypeHeight 4
PrimaryHandlerName lcd
```

```
# Basic TEDS
# -----
```

```
ManufacturerID     1
ModelNumber        4
VersionLetter      A
VersionNumber      1
SerialNumber       4
```

```
# Standard Template TEDS
# -----
```

```
TemplateID         30 # Voltage Template
ElecSigType        0 # Voltage Sensor
MapMeth            0 # Linear
SelectExcitation   1 # Include Excitation/Power Requirements
ExciteAmplNom      3.3 # Volts
ExciteAmplMin      3.3 # Volts
ExciteAmplMax      3.3 # Volts
ExciteType         0 # DC
ExciteCurrentDraw  1.0 # Amps
CalDate            1
CalInitials        NUL
CalPeriod          365 # Days
MeasID             1
```

## A.6 LCD Merge Template TEDS

In this section, TEDS is defined for composite group made up of LCD modules. In addition to the basic and AMSS TEDS and the standard template, a Roles template is defined. The Roles template is used by the software architecture to identify the types of sensors and actuators that can be used to form a composite group.

```
# LCD Merge Template TEDS File
```

```
# -----
```

```
# AMSS TEDS
```

```
# -----
```

```
ModuleType          2  # Actuator
ModuleClass         4  # Text
ModuleDataType      800 # String
ModuleDataTypeWidth  1
ModuleDataTypeHeight 1
```

```
# Basic TEDS
```

```
# -----
```

```
ManufacturerID      1
ModelNumber         6
VersionLetter       A
VersionNumber       1
SerialNumber        6
```

```
# Standard Template TEDS
```

```
# -----
```

```
TemplateID          30 # Voltage Template
ElecSigType         0  # Voltage Sensor
MapMeth             0  # Linear
SelectExcitation    1  # Include Excitation/Power Requirements
ExciteAmplNom       3.3 # Volts
ExciteAmplMin       3.3 # Volts
ExciteAmplMax       3.3 # Volts
ExciteType          0  # DC
ExciteCurrentDraw   1.0 # Amps
CalDate             1
CalInitials         NUL
CalPeriod           365 # Days
MeasID              1
```

# Roles

# -----

Role 1

RoleAssignmentLimit &gt;=1

RoleConnectionType 3 # Local/Physical

RoleModuleType 2 # Actuator

RoleModuleClass 4 # Text

RoleModuleDataType 800 # String

RoleModuleDataTypeWidth &gt;=1

RoleModuleDataTypeHeight &gt;=1