

SENSE/STAGE - LOW COST, OPEN SOURCE WIRELESS SENSOR INFRASTRUCTURE FOR LIVE PERFORMANCE AND INTERACTIVE, REAL-TIME ENVIRONMENTS

*Marije A.J. Baalman, Vincent de Belleval
Christopher L. Salter*

Concordia University
Design and Computation Arts

*Joseph Malloch, Joseph Thibodeau
Marcelo M. Wanderley*

McGill University
Music Technology, IDMIL

ABSTRACT

SenseStage is a research-creation project to develop a wireless sensor network infrastructure for live performance and interactive, real-time environments. The project is motivated by the economic and technical constraints of live performance contexts and the lack of existing tools for artistic work with wireless sensing platforms. The development is situated within professional artistic contexts and tested in real world scenarios. In this paper we discuss our choice of wireless platform, the design of the hardware and firmware, battery options, and an evaluation of the data transmission quality within the wireless network. Additionally, software integration of the wireless platform with popular media programming environments is addressed, as well as evaluation and dissemination of the technology through workshops. Finally, we elaborate on the application of the hardware and software infrastructure in professional artistic projects: two dance performances, two media projects involving environmental data and an interactive, multi-sensory installation.

1. INTRODUCTION

SenseStage is a research-creation project to develop small, low cost and low power wireless sensor hardware together with software infrastructure specifically for use in live theater, dance and music performance as well as for the design of interactive, real-time environments involving distributed, heterogeneous sensing modalities. The project consists of three components:

- a series of small, battery powered wireless PCBs that can acquire and transmit input from a range of analog and digital sensors,
- an open source software environment that enables the real-time sharing of such sensor data among designers and
- plug in modules that enable the analysis of such sensor data streams in order to provide building blocks for the generation of complex dynamics for output media.

The project emerged from a desire to address a novel, emerging research field: distributed, wireless sensing networks for real-time composition using many forms of output media including sound, video, lighting, mechatronic and actuation devices and similar. The design of interactive environments using diverse output media increasingly involves the mapping of many channels of real-time sensor data to control the temporal behavior of such media. Standard mapping techniques with sensors that have been derived from the "instrument building" paradigm [7] usually address only small numbers of sensors or participants and may not scale well to larger spaces. Systems involving large numbers of sensors and participants are rare, custom-designed, and expensive [6]. Furthermore, while wireless sensors and wireless sensor networks (WSNs) are being increasingly deployed daily in areas such as health care, defense, seismology and home security, there are scant examples of such technologies in artistic projects simply due to the lack of available hardware/software infrastructure for artists to use. Most work in sensor networks has been in areas of applied technology development [10] without artistic aims or is restricted to lab settings. Based on these factors, SenseStage has developed a fully integrated hardware and software infrastructure that is intuitive to use by artists and designers, is scalable to many nodes and performs data acquisition, transmission, conditioning, sharing and compositional tasks all within the same system.

2. BACKGROUND AND MOTIVATION

Three specific factors have motivated the SenseStage project:

1) *Economic and technical constraints of live performance:* While there is increasing interest in the use of sensing technologies in live performance contexts (particularly theater, dance and music-theater), the economic and cultural constraints of live performance make the integration and use of such experimental technologies difficult. Long rehearsal periods and proper technical infrastructure necessary to test and use sensing systems are prohibitively expensive for artists and cultural institutions. Furthermore,

box office pressure forces presentation venues into adopting an industrial model of cultural production — show in, show out — leaving no room for the detailed exploration of new technological possibilities and the artistic impact they would yield. This is particularly evident in the extremely short technical integration periods (“tech week” or “technical rehearsals”) that are customary for theater, dance and music¹. Thus, the use of many sensing devices and software tools needs to be conditioned by flexibility, minimal pre-show setup time, quick deployment and use within a variety of stage conditions.

2) *Lack of tools for artistic use:* As previously stated, SenseStage emerged from a desire to address the emerging research field of ubiquitous computing within the artistic, real-time context of live performance and interactive environments. Although many groups are currently researching and developing WSNs, design decisions are normally motivated by engineering innovations thus leading to efficient yet, prohibitively expensive and complex systems out of the reach of artists. Furthermore, as will be detailed below, despite the high number of research initiatives currently taking place, there are disappointingly few wireless sensing platforms that are actually available for real world use or that are cost effective. In addition, there is a lack of software tools for interacting easily with the large amount of data produced by such distributed wireless systems, *especially* tools implemented in lingua franca programming languages and environments used by musicians, sound and media artists such as Processing, Max/MSP/Jitter, Supercollider, PureData and other environments supporting OpenSoundControl (OSC)². SenseStage seeks to develop a technological framework that eases the exchange of data between many diverse programming environments used for interactive sound and media projects in order that artists and designers with diverse practices can work efficiently on complex interactive projects in both development (i.e., rehearsal) and performance stages.

3) *Real world testing scenarios:* Much of the research agenda for the project was driven by many years of artistic work and technological development of tools to facilitate the creation of interactive performances and installations with distributed sensing and which used mapping of such input data to complex parameter spaces for the control of sound and other media in real-time (e.g. *Schwelle* [1] and *TGarden*[12]). A key design element of the SenseStage project is thus to deploy SenseStage technologies into real world, professionally driven testing environments to see how such tools function “in the wild” and outside of the standard lab, demo-driven mode normally given to the presentation of new technologies.

¹http://en.wikipedia.org/wiki/Technical_rehearsal

²<http://opensoundcontrol.org/>

3. HARDWARE

3.1. Wireless sensing infrastructure

Our two main requirements for choosing a sensor node design were cost per unit — since low costs allow experimentation with large numbers of nodes — and immediate availability. Although none of the devices we investigated met these requirements, four stood out in particular and are described below.

The μ Parts³ [4] are small sensor nodes within the target price range (ca. 15 euro), but are not yet available to the general public. The sensor nodes have a fixed set of sensors (light, tilt, temperature, motion and acceleration) with little room for attaching additional sensors, making them less flexible than an open system. Their target applications are mostly slow environmental sensing with a minimum poll time of half a second.

The EcoMote⁴ [11] is a promising platform for our purpose, considering its small size and low battery consumption, but has not yet been made available for the general public, specifically not when we started the project.

The Tyndall Motes⁵ also provide an interesting modular approach and a small form factor (25mm by 25mm by 25mm), but also have not been available for wide distribution.

The Intel Motes⁶[9] are motes based on an ARM processor running TinyOS⁷ and using BlueTooth scatternet for wireless transmission, on which a sensor interface board can be stacked. Crossbow⁸ supplies various wireless network solutions, amongst which the IRIS (a successor of the MICA; and apparently a further development of the Intel Mote), which uses the Zigbee protocol for wireless transmission; the IRIS OEM Module is available for 29\$ per module, when bought in order sizes of 1000 units⁹. Crossbow’s marketing is clearly aimed at manufacturers of industrial applications of WSN’s.

BlueTooth based options were not considered, as the number of BlueTooth devices connected to the same host computer at the same time is limited to a maximum of 7 per BlueTooth receiver. Also the time needed for reconnecting, if a connection is lost, is too long for a performance context (up to 5 seconds for a default connection timeout). While the BlueTooth specifications are continuously improved and enhanced, its wide application in consumer electronics is a drawback for applications in artistic work as audience’s

³<http://particle.teco.edu/upart/>

⁴<http://www.ecomote.net/>

⁵<http://www.tyndall.ie/mai/>

WirelessSensorNetworksPrototypingPlatform_25mm.htm

⁶<http://techresearch.intel.com/articles/Exploratory/1503.htm>

⁷<http://www.tinyos.net/>

⁸www.xbow.com

⁹according to a press release; Crossbow’s website does not list prices.

gadgets (e.g. cell phones) may interfere with the wireless networking within the artwork.

3.2. Design of the SenseStage MiniBee

Our design goals were:

- Low cost
- Small form factor
- Flexible sensor configuration
- Usable for control of motors, LEDs, and other actuators.
- Operable in large groups (10+ nodes)
- Long battery life
- Ease of use
- Programmable, so that the board can take care of more logic and processing of data, if desired by the user

We have looked at two different wireless transmitters, the Nordic nRF24LE1¹⁰, and the XBee¹¹.

Because the nRF24LE1 has a steep development curve to work with, we decided to use the XBee in combination with Arduino¹², as the XBee already provided us with the needed ad-hoc network structure. Additionally, several other developers have documented their experience using XBees in conjunction with Arduinos, allowing us to skip some common development pitfalls.

Examples of other Arduino - XBee projects are the ArduinoXbeeShield¹³, created by Arduino together with Libelium, who use the board in their SquidMotes, a predecessor of the WaspMotes¹⁴. Both the XBeeShield and the WaspMotes are considerably larger than our board design for the SenseStage MiniBee. Another example is the Arduino XBee Interface Circuit (AXIC)¹⁵, which is a do-it-yourself solution. BlushingBoy sells the MiniBee R3 and MicroBee R3¹⁶, which can be used together with an Arduino Mini. More recently the RFBee¹⁷ has come out combining an RF radio with an Atmel chip, which is Arduino and XBee compatible.

We based our board design on the Arduino Mini Pro, being able to tap into many available firmware libraries, as well as the development and programming environment. Furthermore, Arduino is widely used in open source, artistic, physical computing contexts, so our board will be easy to use for this community.

The main focus then was to design a PCB design that was small, and to develop standard firmware that makes it

¹⁰<http://www.nordicsemi.com/index.cfm?obj=product&act=display&pro=95>

¹¹<http://www.digi.com/products/wireless/point-multipoint/xbee-series1-module.jsp#overview>

¹²<http://www.arduino.cc>

¹³<http://www.arduino.cc/en/Main/ArduinoXbeeShield>

¹⁴<http://www.libelium.com/products/waspmote>

¹⁵<http://132.208.118.245/~vitamin/tof/AXIC/>

¹⁶<http://blushingboy.org/content/microbee-r3>

¹⁷<http://www.seeedstudio.com>

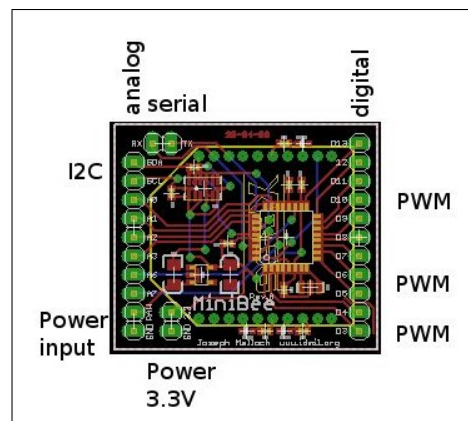


Figure 1. The SenseStage MiniBee PCB, rev. A. The XBee is mounted on the other side of the board. Changes in the second revision include smaller board-size and the footprint for a coin-cell.

easy to setup and use the boards, as well as exploring the use of and integration with the XBee wireless chips.

The PCB layout of the SenseStage MiniBee is shown in figure 1. The first board revision came to a unit cost price of about 32 CAD, excluding the XBee chip, for a manufacturing run of 100 boards (PCB creation, assembly and parts). With a larger manufacturing run and allowing for a longer assembly time, this per unit cost will drop considerably.

3.3. Battery choice

There is an obvious compromise to be made between battery size and battery capacity, and so we decided to consider two configurations: (1) for usage cases where size is not a concern, e.g. when the board is used for fixed environmental sensing, a large, high-capacity battery is best; and (2) for situations where size is important (mounted on a performer's body or a handheld instrument, for example) and the battery should be as small as possible. For logistical and ecological reasons, the battery should be rechargeable, and last at least as long as one rehearsal, i.e. approximately 5 to 6 hours. Our board draws about 70 mA of current, when used with a regular XBee, transmitting data every 50 ms, without activating a sleep mode.

The batteries we have tested include:

AA-sized Li-Ion 900 mAh¹⁸ Long battery life (almost 10 hours). Recharge time ca 3 hours¹⁹. Usable with a standard AA-battery holder. Towards the end of the battery life, the battery turns off at intervals of about 10 seconds and turns on again, rather than just turn off completely.

¹⁸Protected UltraFire 14500 AA sized 3.6V Li-Ion Rechargeable Battery 900 mAh CR14500

¹⁹with Ultrafire WF-138 3.6 volt Lithium-Ion AA / AAA battery charger from batteryjunction.com

```

#include <MiniBee.h>
char customData[2];
void customMsgParser( char * msg ){
    digitalWrite( 13, msg[2] );
}
void setup() {
    Bee.setCustomPin( 11, 2); // custom pin with data
    Bee.setCustomPin( 13, 0); // custom pin without data
    Bee.begin(19200);
    Bee.setCustomCall( &customMsgParser );
}
void loop() {
    Bee.addCustomData( customData );
    Bee.doLoopStep();
}

```

Figure 2. Firmware code example showing how to add custom functionality to the MiniBee library.

AAA-sized Li-Ion 500 mAh²⁰ Unfortunately, we found that a majority of these batteries would not recharge after using them only a few times.

18650 Li-Ion 2400 mAh²¹ Very long battery life (ca. 24 hours). Recharge time ca. 12 hours. Large battery, but useful for installations where size is not a constraint and recharging is part of the design, e.g. solar powered recharging.

Li-Poly Sparkfun 860mAh²² Long battery life (almost 10 hours). Recharge time ca. 2.5 hours²³. Battery is slightly larger than the board, but very flat.

Li-Ion coin cell Sparkfun 200mAh²⁴ Short battery life (almost 2 hours). Recharge time ca. 30 minutes²³. This coin cell fits within the footprint of the PCB so a battery clip will be added in the next revision of the board.

3.4. Firmware

The firmware is a library containing a collection of functions to handle wireless transmission and communication, sensor reading and basic read/write operation on any available pin of the MiniBee.

Currently the following sensors/actuators are supported by the firmware:

- Analog sensors (connected to the analog input pins, e.g. resistive sensors, analog accelerometers, infrared distance sensors)
- Digital sensors (on/off, e.g. buttons and switches)
- LIS302DL accelerometer²⁵, using I2C
- Relative humidity and temperature sensor²⁶

²⁰UltraFire 10440 AAA Li-Ion 3.6V Rechargeable Battery 500 mAh CR10440, together with “Protection circuit Module (PCB) for 3.6V(3.7V) Li-ion (18650/18500) cell Battery”

²¹Protected UltraFire 18650 3.6V Li-Ion rechargeable Battery 2400 mAh

²²Polymer Lithium Ion Batteries - 860mAh; Sparkfun no: PRT-00341

²³Using the LiPoly Fast Charger; SparkFun no: PRT-08293

²⁴Coin Cell Battery Rechargeable - 24.5mm; Sparkfun no: PRT-08818; together with “Protection circuit Module (PCB) for 3.6V(3.7V) Li-ion (18650/18500) cell Battery”

²⁵There is a footprint on the board for this sensor.

²⁶Sensirion SHT1x series

<i>Basic format</i>		escape (92) + (*message*) + delimiter (10)
<i>description</i>	<i>type</i>	<i>data</i>
Data output	'O'	node ID + msg ID + N values
Data	'd'*	node ID + msg ID + N values
Custom message	'E'	node ID + msg ID + (*data*)
Loopback	'L'	node ID + msg ID + onoff
Running	'R'	node ID + msg ID + onoff
Announce	'A'	
Quit	'Q'	
Serial number	's'*	Serial High (SH) + Serial Low (SL) + + library version + board revision
ID assignment	'I'	msg ID + SH + SL + node ID + (*config ID*)
Wait for config	'w'*	node ID + config ID
Configuration	'C'	config ID + configuration bytes
Confirm config	'c'*	node ID + config ID + smpMsg + msgInt + + datasize + outsize + (*custom*)
(*custom*)		N x (custom pin, data size)

Table 1. Message protocol between host and MiniBee nodes. The messages sent by the node are indicated with a * behind the message type.

- Ultrasound sensors²⁷
- PWM output (e.g. dimmable LEDs, motors)
- Digital output (on/off)

Sensors and actuators not supported by the library can be hooked into the library by setting custom options, thus extending the capabilities of the firmware, without losing the flexibility of the core functionality (see figure 2).

The serial protocol is based on the Serial Line Internet Protocol (SLIP), and is set up as simple as possible to ensure that data packets are small. It is built up as listed in table 1.

The firmware can be configured through a host computer which allows to quickly change its operation without having to physically reprogram the microcontroller. This approach is not unlike Firmata [13] with the difference that our firmware stores its latest configuration in the EEPROM of the minibee. Each time the MiniBee boots up, it can access its configuration through its EEPROM. Upon start it reads the serial number of the attached XBee²⁸. This information is relayed by the coordinator node to the host computer, which then assigns a unique node ID to the MiniBee and confirms what its configuration should be. The host computer software remembers the known boards and XBee serial numbers so node IDs are consistent for a project.

Using an 8 MHz clock on the board, the maximum baud rate that can be achieved reliably in combination with the XBee is 19200 baud. In a next revision we will include a faster (up to 12MHz) crystal, which will allow the use of higher baudrates and better timing accuracy.

Future work also includes writing a wireless bootloader to fully reprogram the SenseStage MiniBee without the need

²⁷The “Ultrasonic Ranger”, <http://www.robot-electronics.co.uk/htm/srf05tech.htm>

²⁸using the AT mode.

$\Delta T(ms)$	inputs	$\mu(ms)$	$\sigma(ms)$	$min(ms)$	$max(ms)$
25	1	28.61	3.73	0.08	74.96
25	4	28.63	5.3	0.13	743.06
25	8	30.66	3.64	0.19	81.58
25	19	36.78	3.21	0.38	90.19
50	1	53.11	4.73	0.08	241.85
50	4	53.15	6.09	0.28	106.82
50	8	55.19	3.27	19.21	129.79
100	1	103.39	3.77	34.46	147.3
100	4	103.3	3.64	0.64	144.77
100	8	108.33	4.45	0.77	156.03
100	19	114.34	3.51	64.8	163.13

Table 2. Results for one node sending data at different time intervals, and with different number of data bytes. See text for explanation of each column.

to manipulate the boards (see for example LadyAda²⁹). The extra components needed to do this will be added in the next revision of the board.

3.5. Evaluation of the wireless network

To evaluate the quality of the wireless sensor network we performed a series of tests regarding data arrival times, dependent on how many nodes are active at the same time, at which time intervals they are sending data, and the size of the data packets. The nodes in this test scenario are programmed with our firmware library, and then configured to sense a number of digital pins, whose values are then sent to the host computer (one byte per digital input). The sensing introduces a little bit of extra time between messages from the MiniBee.

This test was performed using various time intervals (25, 50 or 100 ms) between messages, numbers of nodes in the network (1 or 10) and amount of digital inputs (1, 4, 8, or 19³⁰). Each test was run for fifteen minutes.

The results for using 1 node at different time intervals is given in table 2. The first column indicates the time interval used in the firmware code, the second the number of digital inputs used. The next two columns give the mean and standard deviation of the arrival time intervals at the host computer. The last two columns the minimum and maximum measured time interval.

From the results we can see that the time intervals tend to be slightly longer than the firmware time interval; in particular we can see an increase in time when 8 or more digital pins are polled. The standard deviation is mostly between 3.5 and 4.5ms with some deviations where it goes up to 6ms.

The results for multiple active nodes are shown in table 3. Here we show a mean and standard deviation over the

²⁹<http://www.ladyada.net/make/xbee/arduino.html>

³⁰so packet size varying between 5 and 24 bytes; see table 1

number of active nodes of both the mean and standard deviation of the arrival times per node. It is clear that the average transmission time has increased, about 25 to 30ms later than expected for data sizes up to 8 inputs; and even more delay occurs for 19 inputs. This could indicate a loss of packets, as the network communication gets denser. While setting up the test for the multiple nodes it was notable that messages going from the host to the nodes were not always received, or the reply from the nodes did not arrive at the host.

In conclusion we can say that although the wireless boards will not give very accurate timing, they do provide a fairly continuous stream of data. In denser networks packets get lost or delayed in transmission. We will further investigate the issue of packet loss with another series of tests.

4. SOFTWARE

In order to make the data from the wireless sensor nodes available to several collaborators on a project simultaneously, we developed the SenseWorld DataNetwork. It is intended to facilitate the creation, rehearsal and performance of collaborative interactive media art works, by making the sharing of data (from sensors or internal processes) between collaborators easy, fast and flexible. Our aim is to support multiple media practices and allow different practitioners to use the software to which they are accustomed. The framework is intended to support *coordinated collaboration* with real-time data and multiple media types within a live interactive performance context.

The framework is different from the *KeyWorx*³¹ framework [5], which emphasizes net-based art and collaborative projects between different locations, and the McGill Digital Orchestra Tools³² [8], which primarily focus on the mapping and performance of monolithic digital musical instruments.

The final design criteria were to:

- Tight integration with the wireless sensing platform
- Allow reception of data from any node by any client (subscription)
- Allow transmission of data to any node by any client³³ (publication)
- Restore network and node configuration quickly
- Be usable within heterogeneous media software environments
- Enable collaboration between heterogeneous design practices
- Enable efficiency of collaboration within the limited timeframe of rehearsals

While the technical details of the implementation have already been discussed in [3], we have since continued the

³¹<http://www.keyworx.org>

³²There is a Max/MSP bridge between the SenseWorld DataNetwork and the Digital Orchestra Tools so they can be used together.

³³only one client can set data to a specific node at a time.

ΔT	inputs	$\mu(ms)$		$\sigma(ms)$		min	max
		μ_μ	σ_μ	μ_σ	σ_σ		
25	1	49.846	9.095	62.75	25.914	0.18	3259.28
25	4	54.479	12.484	68.69	30.553	0.2	2417.82
25	8	62.101	14.805	86.885	40.791	0.22	5651.99
25	19	114.484	37.846	202.893	96.057	0.4	5200.68
50	1	72.385	4.916	60.951	15.562	0.96	1795.95
50	4	76.439	14.087	72.101	24.722	0.21	3499.91
50	8	80.297	8.16	80.145	26.102	1.44	3840
50	19	134.073	33.525	202.398	102.075	0.94	12451.75
100	1	119.482	3.8	55.577	11.516	61.95	2117.35
100	4	120.671	4.074	61	11.118	64.29	2647.43
100	8	125.433	4.135	69.92	13.765	5.56	2617.17

Table 3. Results for multiple nodes sending data at different time intervals, and with different number of data bytes. For the tests at 25ms 8 nodes were active; for 50 and 100ms 10 nodes were used. See text for explanation of each column.

development and evaluation of the software framework, and created clients for more environments, so that there are now client implementations available for SuperCollider, Max/MSP, PureData, Processing, Java, and C++ (including an example of how to integrate this with OpenFrameworks) [2]. In addition, a standalone host for OSX is available.

We have specifically integrated the connection to the SenseStage MiniBee network, by providing options to configure, send and receive data from the sensor network, through the host of the DataNetwork. On the other hand, the DataNetwork can be used independently, to share any kind of data between clients.

5. SENSESTAGE WORKSHOP

The first SenseStage workshop³⁴ was held in May 2009 at Concordia University, as a test case for using many sensor nodes in one space, as well as having a number of artists, unfamiliar with the specifics of the technology and coming from diverse artistic and technical backgrounds, use the DataNetwork simultaneously. Participants were able to employ all available data in various projects, which were developed over the course of one week. While this workshop served as a test case for evaluating the use of our hardware and software, we were also interested in how participants would make artistic use of the potentials of the system.

The workshop resulted in five group projects, with groups consisting of 3 to 5 collaborators, using light, sound, animation and video as output media. The groups were able to go very quickly from concept to experimenting with various sensor modalities and using the data to drive the output media. Encouragingly, the participants seemed to be primarily concerned with “what to do with the data” rather than “how to get the data”. The results of this workshop also highlighted the need for more sophisticated ways of dealing with

sensor data, for combining, conditioning, and processing of multiple data streams.

Several more SenseStage workshops are planned for 2010 and 2011 as a means to familiarize people with the SenseStage infrastructure, as well as further explore issues in mapping and using large numbers of datastreams.

6. USAGE CASES

In this section, we will discuss several projects in which the SenseStage technology has been used. All of these projects have been, or are being shown at international festivals and multiple venues, thus fulfilling the criterium that the technology has to be useable in a real world, professional artistic environment.

6.1. Dance: Schwelle and Chronotopia

The dance performance Schwelle [1] had been developed before the SenseStage project began and has informed many of the design decisions made during the SenseStage project, so we are currently discussing how the infrastructure can be adapted and improved to use our new technology for future performances. The performance involves 3 accelerometers on the body (originally wireless Create USB interfaces (CUI) with MicroChip RF chips), 1 accelerometer in an object (originally a WiiMote), and 3 light sensing boards (originally wired CUIs) placed at various places in the room. Furthermore, there is activation of custom lights and motors in one part of the stage. Using the SenseStage MiniBees, we will be able to use now 3 separate sensing boards on the body, saving us problems of wiring along the body; instead of using the WiiMote for the accelerometer in the object, we can now use a SenseStage MiniBee, which we can wake up from a sleep mode, once we need the sensing in the box; this will save us various problems of having to make the WiiMote set up a BlueTooth connection by push-

³⁴<http://sensestage.hexagram.ca/workshop/>



Figure 3. A still from the dance performance Chronotopia with the Attakkalari Centre for Movement.

ing buttons, while it is packed inside a box. For the sensing boards inside the room, we will also be spared of the wiring. Where we were previously using a custom OSC-namespace for this piece, we can now use the DataNetwork clients instead, and gain much more robustness with regard to reconnecting; also it will be much faster to add new data to exchange between the interactive light controls and the sound control, should we feel the need to do so.

In Chronotopia (see figure 3), a dance performance by the Bangalore (India) based Attakkalari Centre for Movement, and in collaboration with visual artist Chris Ziegler, we used the wireless technology for controlling a matrix of 6 by 6 cold cathode fluorescent lights (CCFL), and 3 handheld CCFL lights. Since the power required for the light matrix is quite high, it cannot be battery-powered, but the use of wireless technology freed us from running cable between the light matrix and the computer controlling it. Given the short setup time in theaters (usually just one day), and especially in technically challenging environments as India, this was a considerable advantage. For the 3 handheld objects, wireless control was critical, as the objects are carried across the stage by the performers during the show as part of the dramaturgy of the piece. Within the light control setup itself, the DataNetwork was used extensively to exchange data between different portions of the setup, such as the motion tracking data (from a camera looking down at the stage), and pitch and beat tracking data extracted in real-time from the soundtrack. We also needed to exchange data between the light control and the interactive video, both for synchronisation of cues with the soundtrack (using frame-time of the playback, published as data on the network), and for connecting the intensity of the lights to the video image (the light control was publishing the maximum output value of all the lights in the matrix onto the network, which was used to control the brightness of the video image).

6.2. Environmental: MARIN and Arctic Perspective

In two artistic projects dealing with environmental data, MARIN³⁵ and Arctic Perspective Initiative³⁶, the SenseStage MiniBees were used to gather environmental data, such as temperature, humidity, light and air quality, as well as 3-axis acceleration. The DataNetwork was used to access the data for real-time use, and to gather and log all the data to file for artists to use at a later time for visualisation and sonification.

From an expedition to Nunavut in Northern Canada in Summer 2009 as part of the Arctic Perspective Initiative project we learned that the range of the XBee and XBeePros is very much dependent on the environmental conditions. In the outside conditions there, the achieved range of transmission was only a few hundred meters, only about a fifth of the range specified on the datasheet of the XBeePro. At ranges larger than about 50 m. they are extremely affected by blockage, e.g. from the body. In indoor situations, the radio waves will be reflected by objects and walls and such blockage may be mitigated.

Additionally, the batteries lost charge faster in colder conditions (about 6°C) resulting in shorter battery life.

Another issue was that it was difficult to power the host computer, receiving the data from the MiniBees, using solar power, because of foggy and cloudy days. For this reason a kind of datalogger approach for the MiniBees, which stores data locally and sends it to a host when the host is online, could be useful for this kind of application.

6.3. Installations: JND/Semblance

JND/Semblance is an interactive installation that explores the phenomenon of cross modal perception — the ways in which one sense impression affects our perception of another sense. The installation comprises a modular, portable environment, which is outfitted with devices that produce subtle levels of tactile, auditory, visual and olfactory feedback for the visitors, including a floor of vibrotactile actuators that participants lie on, peripheral levels of light, scent and audio sources, which generate frequencies on the thresholds of seeing, hearing and smelling.

In JND/Semblance the SenseStage MiniBees are used for gathering floor pressure sensing data. The SenseWorld DataNetwork is used to gather the raw sensor data, to extract features from it, and to establish flexible mappings to light, sound, and vibration on a platform on which the visitor is lying down.

³⁵“M.A.R.I.N. (Media Art Research Interdisciplinary Network) is a networked residency and research initiative, integrating artistic and scientific research on ecology of the marine and cultural ecosystems.” (from <http://marin.cc/>).

³⁶“The Arctic Perspective Initiative (API) is a non-profit, international group of individuals and organizations whose goal is to promote the creation of open authoring, communications and dissemination infrastructures for the circumpolar region.” (from <http://www.arcticperspective.org>).

7. CONCLUSIONS AND FUTURE WORK

We have presented SenseStage, an integrated hardware and software infrastructure for wireless mesh-networked sensing, actuating, data sharing and composition within interactive media contexts. The infrastructure is unique as it integrates hardware and software, and makes sensor data and other data easily available for all collaborators in a heterogeneous media project, within each collaborator's preferred software environment.

We are currently revising the hardware and firmware design, including options to configure and program the boards wirelessly. We plan to have the board available for sale in the second half of 2010. Our future research will focus on techniques for composing and creating with the many streams of realtime data available from such a dense network of sensors.

8. ACKNOWLEDGEMENTS

This work was supported by grants from the Social Sciences and Humanities Research Council of Canada and the Hexagram Institute for Research/Creation in Media Arts and Sciences, Montréal, QC, Canada.

Thanks to Matt Biederman for his feedback on the use of the SenseStage MiniBees in the Arctic Perspective Initiative project.

Thanks to Brett Bergmann, Harry Smoak and Elio Bidinost, as well as all the SenseStage Workshop participants, for their input and feedback.

The *SenseWorld DataNetwork* is available from <http://sensestage.hexagram.ca>. It is released as open source software under the GNU/General Public License.

9. REFERENCES

- [1] M. A. Baalman, D. Moody-Grigsby, and C. L. Salter, "Schwelle: Sensor augmented, adaptive sound design for live theater performance," in *Proceedings of NIME 2007 New Interfaces for Musical Expression, New York, NY, USA, 2007*.
- [2] M. A. Baalman, H. C. Smoak, V. de Belleval, B. Bergmann, C. L. Salter, J. Malloch, J. Thibodeau, and M. Wanderley, "Sense/Stage — low cost, open source wireless sensor and data sharing infrastructure for live performance and interactive realtime environments," in *Linux Audio Conference 2010, Utrecht, The Netherlands, 2010*.
- [3] M. A. Baalman, H. C. Smoak, C. L. Salter, J. Malloch, and M. Wanderley, "Sharing data in collaborative, interactive performances: the SenseWorld DataNetwork," in *Proceedings of NIME 2009 New Interfaces for Musical Expression, Pittsburgh, PA, USA, 2009*.
- [4] M. Beigl, C. Decker, A. Krohn, T. Riedel, and T. Zimmer, "uParts: Low cost sensor networks at scale." in *UbiComp 2005, 2005*.
- [5] S. Doruff, "Collaborative praxis: The making of the keyworx platform," in *aRT&D: Research and Development in the Arts*, J. Brouwer, A. Mulder, and A. Nigten, Eds. V2/NAI Publishers, Rotterdam, 2005. [Online]. Available: <http://spresearch.waag.org/images/CollabPraxis05.pdf>
- [6] M. Feldmeier and J. A. Paradiso, "An interactive environment for large groups with giveaway wireless motion sensors." *Computer Music Journal*, vol. 31, no. 1, pp. 50–67, Spring 2007.
- [7] A. Hunt, M. M. Wanderley, and M. Paradiso, "The importance of mapping in musical instrument design," *Journal of New Music Research*, pp. 429–440, 2003.
- [8] J. Malloch, S. Sinclair, and M. M. Wanderley, "A network-based framework for collaborative development and performance of digital musical instruments," in *Computer Music Modeling and Retrieval. Sense of Sounds: 4th International Symposium, CMMR 2007, Copenhagen, Denmark, August 2007, Revised Papers*, ser. Lecture Notes in Computer Science, R. Kronland-Martinet, S. Ystad, and K. Jensen, Eds. Springer, 2008, no. ISBN 978-3540850342.
- [9] L. Nachman, R. Kling, R. Adler, J. Huang, and V. Hummel, "The intel mote platform: a bluetooth-based sensor network for industrial monitoring." in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN 2005) (Los Angeles, CA, April 2005), 2005*.
- [10] J. Paradiso, "Sensor architectures for interactive environments." in *Augmented Materials & Smart Objects: Building Ambient Intelligence Through Microsystems Technologies*, K. Deleaney, Ed. Springer, 2008, ch. 3.
- [11] C. Park and P. H. Chou, "Eco: Ultra-wearable and expandable wireless sensor platform," in *Third International Workshop on Body Sensor Networks (BSN 2006)*.
- [12] J. Ryan and C. L. Salter, "Tgarden: Wearable instruments and augmented physicality," in *Proceedings of the 2003 Conference on New Instruments for Musical Expression (NIME-03), Montreal, CA, 2003*.
- [13] H.-C. Steiner, "Firmata: Towards making microcontrollers act like extensions of the computer," in *Proceedings of NIME 2009 New Interfaces for Musical Expression, Pittsburgh, PA, USA, 2009*.