Wireless sensing for artistic applications, a reflection on Sense/Stage to motivate the design of the next stage

Marije Baalman
nescivi
sensestage@nescivi.nl

ABSTRACT

Academic research projects focusing on wireless sensor networks rarely live on after the funded research project has ended. In contrast, the Sense/Stage project has evolved over the past 6 years outside of an academic context and has been used in a multitude of artistic projects. This paper presents how the project has developed, the diversity of the projects that have been made with the technology, feedback from users on the system and an outline for the design of a successor to the current system.

Author Keywords

wireless systems, sensing/actuating platforms, physical computing, augmented instruments, new musical instruments, installations, dance, theatre, performance

ACM Classification

C.2.1 [Computer Systems Organization] Network Architecture and Design — Wireless communication J.5 [Computer Applications] Arts and Humanties — Performing arts K.6 [Management of Computing and Information Systems] H.5.2 [Information interfaces and presentation] User interfaces

1. INTRODUCTION

Wireless sensor network platforms and their use in artistic contexts have been a topic for research for many years in academic research projects. Very few of these academic research projects have resulted in wireless technology becoming available for artists (other than those involved in the project). Since the survey presented in our paper in 2010 [1] new platforms have become available. In particular within the context of the *Internet of Things*[1] ecology many new wireless platforms are emerging. An overview is given in section 5.

The Sense/Stage wireless module[1] was originally developed during a research-creation project at Concordia University between 2007 and 2010. Since then the module was taken into a small scale production and made available through a webshop. More than 500 modules have been sold and used in various projects around the world by artists in professional contexts. This paper reflects on the usage and developments that the module and its system have seen outside of the academic context and in professional artistic contexts.

In the following section the project is introduced; then the usage of the system out in the field is described and the feedback from users. Next the development of the project since its release in 2010 is discussed. The paper concludes with an overview of other wireless systems and the design considerations for a successor to the Sense/Stage MiniBee.

[1] http://www.itu.int/en/ITU-T/gsi/iot/Pages/default.aspx

NIME'17, May 15-19, 2017, Aalborg University Copenhagen, Denmark.

2. THE SENSE/STAGE PROJECT

The Sense/Stage research-creation project [1] was motivated by:

Economic & technical constraints of live performance — technology needs to be integrated and explored artistically in a relatively short time frame during rehearsals, because of the limited access to rehearsal spaces and getting artistic teams together. While touring, such systems need to be set up in a short amount of time, and have to be reliable in performance.

Lack of tools for artistic use — the need for a wireless system that integrates well with many common programming languages and environments used by musicians, sound and media artists.

Real world testing scenarios — the need for a system that works and is tested in the context of professional artistic work and presentation venues.

The project set out to develop a platform for wireless sensing that is accessible for artists to use, in terms of availability and affordability (cost) as well as usability. Not all artists have in-depth technical skills, or the money to hire these skills, which means that for a wireless sensing platform to be succesful in the artistic context, it has to be easy to set up and integrate well with the tools that artists are using already. On the other hand, it has to be flexible: different artistic projects need different sensors and/or actuators, even those you did not anticipate for while designing the hardware and firmware or those that are not available at the time of creating the platform (i.e. 'future-compatibility' is important).

The platform addresses these issues by providing standard firmware and accompanying software that allows for easy configuration for common sensing situations and integration with other software via the Open Sound Control[2] (OSC) protocol. At the same time, the firmware and software provides an interface to extend the modules with custom functionality.

[2] http://opensoundcontrol.org

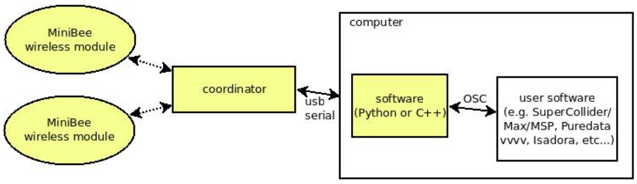

Figure 1: The Sense/Stage eco-system.

2.1 The wireless module and its eco-system

The wireless module consists of an Atmel328 microcontroller[3], an XBee[4] and an accelerometer (the ADXL345[5]) as its core components. Furthermore a pin header gives access to 6 analog inputs, 8 digital IO pins and the two wire interface, so that users can connect additional sensors and actuators for their project. The module has a small footprint, which allows it to be worn on the body or embedded in small objects.

Along with the hardware module, firmware has been developed that handles the wireless communication via the XBee modules and allows for configuration of the module through the wireless connection. This ensures that for most use cases, the user does not need to write firmware herself, but can instead interface with the sensors and actuators through software. The configuration of the modules is made in an XML file and data is received and sent via OSC. This shortens the development time for projects considerably: rather than struggling for many hours with technology, artists can start using their sensors within minutes of installing the software. Once the software is installed and a configuration file is made, the user can send and receive wireless data within seconds of turning the module on.

The software is crossplatform and written in Python, and acts as a bridge between the user's software and the wireless network of modules. For each module a configuration can be made through an XML file; the software then ensures that as wireless modules are turned on, a negotiation is made for addressing and configuration of the module and the user gets notifications about the configuration process via OSC. Once the configuration is done, the user receives the OSC data from the module, or can send data to the module via OSC, e.g. to control LEDs or motors.

The hardware and firmware are based on the popular Arduino[6] platform. This means that artists and developers familiar with that platform can leverage their experience to connect sensors and actuators to the module, or to create customised firmware for the module.

2.2 Dissemination of the project

After the finalisation of the academic research-creation project, the issue of availability to artists was addressed. Small series of the board were manufactured (100 to 200 modules per batch) and sold as individual units or kits through an online webshop. While small scale, this approach ensured that the module was available for artists for a reasonable price (comparable to regular Arduino boards). By hosting workshops in various artist-run spaces or in schools, a user base was slowly built up. Further advertisement was done by word-of-mouth, or via online means.

[3]http://www.atmel.com/devices/ATMEGA328.aspx

[4]https://www.digi.com/lp/xbee

[5]http://www.analog.com/en/products/mems/accelerometers/adxl345.html

[6]http://arduino.cc

3. USAGE OF THE SYSTEM

In this section the target audience, usage scenarios, various projects made with Sense/Stage and user feedback are discussed.

3.1 Target audience and usage scenarios

The target audience for the Sense/Stage MiniBee consists of artists and students. In general people who want to make interactive works, but do not have a lot of experience with electronics or (low-level) firmware programming.

The typical usage scenarios are:

prototyping : see how a sensor works, get a quick sense of what data a sensor produces by feeding the data into a familiar environment for making sound, light, video, or different media. Being able to try out different sensors before deciding which ones will actually be part of the final project.

making an instrument : once decisions are made on what sensors or actuators to use in a project, the wireless board needs to be embedded into an instrument: mounted on an acoustic musical instrument, inside an object, or embedded in clothing or other wearables.

on stage and on tour : once the instrument is made, it needs to go on stage (or in an exhibition) and tour. It needs to be reliable, quick to setup, and last over time.

Examples of what people have been using the Sense/Stage MiniBee for:

- interactive dance with body-worn sensors, where the sensors control sound, light and/or video;
- digital musical instruments: embedded in some sort of object that becomes the instrument;
- augmented musical instruments: mounted on an acoustical instrument, so that the sensors can be used to process the sound of the acoustic instrument;
- light instruments: wireless control of lights, customising firmware to enable different kinds of light behaviours;
- outdoor projects: mounted on kites, or bmx bikes;
- in installations: to control robots, embedded in cocktail shakers, mounted on steel cables, in models of plants;
- in workshops with students learning about physical computing and digital media;
- in workshops with students learning about e-textiles and embedding sensors in costumes.

3.2 Artistic projects

To give an idea of the diversity of the projects made with Sense/Stage, a few projects are highlighted here with a short description:

- Hayes' vibrotactile feedback device, which is used as a means of coupling the instrument and performer, for networked improvisation and as a cueing system for acoustic ensembles.
- Roosna & Flak's "Blood Music"[7], a dance performance where the body becomes an instrument (see figure 2).
- Salter, Baalman and Spier's "N-Polytope"[8], a large-scale spectacular light and sound performance combining lighting, sound, sensing and machine learning, where over 40 modules are used.

[7]http://www.roosnaflak.com/#!performances/c14o2

[8]http://chrissalter.com/projects/n-polytope-behaviors-in-light-and-sound-after-iannis-xenakis/

Figure 2: The Sense/Stage MiniBee worn on ankles and wrists in Roosna & Flak's Blood Music.

Figure 3: The Sense/Stage MiniBee on a cello's bow in Otto's Fello.

- Vandoren's "Integration04"[9], a live performance where the performer manipulates ephemeral audiovisual shapes as if they were tangible materials. Here, a combination is used of the wireless MiniBees with bend sensors connected to them, and tracking using Kinect camera systems.
- Rayzhekov and Köller's "10VE"[10], a participatory biofeedback and movement composition for two actuators and an audience.
- Marangoni's "Echo Moiré"[11], a robotic opera-ballet in which a pair of loudspeaker vehicles is employed to play a room as a musical instrument.
- Baalman's "Wind Instrument"[12], where sensors are mounted on a two-line kite. The data is used to manipulate the sound of the kitelines in realtime.
- Nuages Gris' fashion show "Immaterialicious"[13], which was a collaboration between fashion designers and audiovisual artists exploring the concept of a digital fashion identity.
- Otto's "Fello"[14], which connects his cello and computer as one instrument, using the bow as a special interface (see figure 3).

3.3 Users' experiences

In 2016 an extensive questionnaire[15] was created to gather users' experiences with the current system and their suggestions for improvements. The feedback on this questionnaire was combined with the feedback gathered over the years from users via email and in workshops. The main purpose of the questionnaire was to inventorise the experiences and desires of users to inform our design of a successor to the MiniBee.

In general users were happy with the size of the module, allowing easy integration in their projects (for example to wear it on the body). The standard battery connector (a JST connector) was found difficult to use: it is hard to remove the battery from its socket, with a possible risk of breaking the cables to the battery. An on-board battery charger was considered a desirable feature on the board by some respondents, while other users remarked that it was useful to be able to exchange the battery (i.e. not having to wait for a module to charge).

On the software side, the installation process of the software was challenging for a lot of users, as it involves using a command line. This was an issue for the first install and when the software needed to be reinstalled or installed on another computer. Many users required guidance through this process. Once they made their notes on how to start the software and configure it, the usage was quite straightforward and reliable. Users asked for more concrete examples of configurations and more extensive documentation about the features of the software. The firmware customisation options turned out to be a feature that is only used by users with a more technical background.

In the workshops around the Sense/Stage MiniBee we were able to get participants to use wireless sensor data within their software environments within the time frame of a few hours, and focus on creative use of the data and exploring the artistic possibilities. The most issues people found were with the documentation of the project and they have made suggestions on improving it with a clearer structure like a user's manual, tutorials on how to achieve particular tasks, and possibly video tutorials. The feedback from the users showed that they have been able to realise their projects with the Sense/Stage MiniBees and found it reliable enough to use on tour.

The ease with which artists have been able to realise their projects and tour with them shows that the Sense/Stage MiniBee reaches its goal of being a valuable tool for artistic use and addresses the issue of economic and technical constraints of the artistic practice. In particular, artists/users with no prior experience in interactive or wireless technologies were able to start exploring the artistic possibilities within a very short amount of time.

At the same time the feedback from the users indicated that a continued availability of such a platform is very important to support current and future artistic projects.

4. ONGOING DEVELOPMENT

The wireless module and its associated software have been in continuous development over the past 6 years. The improvements made were driven by the developer's own artistic needs and experiences with the system, as well as by the exchange with various users and collaboration with them to address issues that they ran into.

[9] http://www.dietervandoren.net/index.php?/project/integration04/

[10] http://raijekov.cc/10ve

[11] http://instrumentinventors.org/work/echo-moire/

[12] https://marijebaalman.eu/wind_instruments/2016/08/20/wind-instruments.html

[13] https://dezwijger.nl/programma/immaterialicious

[14] http://andiotto.com/fello

[15] http://docs.sensestage.eu/old/questionnaire

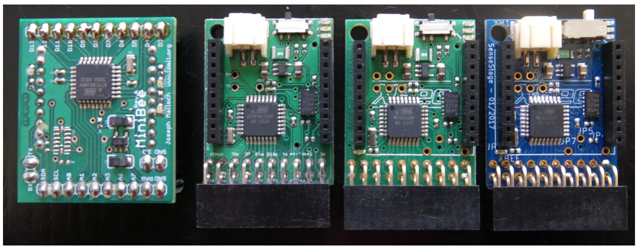

Figure 4: The different hardware revisions of the MiniBee; from left to right revision A, B, D and F.

4.1 Improvements to the hardware

Since its initial design the board has had four revisions. The first revision (from version A to B) involved a considerable change in the layout of the board with a different breakout header: instead of two headers on opposite sides of the board, a doublerow header at the edge of the board was chosen. This allowed for an easier embedding in other projects: a ribbon cable or a header could be used to connect the wireless microcontroller to user made extension boards.

In the revision from B to D an additional status LED was added, which could be used in the module's firmware. The pins needed for programming the module were broken out, so that it became easier to prepare the module with a pogo-pin board before shipping the modules out.

Revisions A and B were manufactured within the scope of the Sense/Stage research project in North America. The modules that were sold to customers through the webshop were all manufactured in the Netherlands and of revision D, up to 2016 and of revision F since 2017.

In the revision from D to F feedback from the users was addressed, while trying to keep the module compatible with revision D. The changes include a more sturdy on/off switch and a voltage regulator with higher current supply. Solder pads have been added on the bottom that break out power supply, the switch and the interrupt pins of the accelerometer. Also pin D2 which is used to control the sleep mode of the XBee was exchanged with pin D7, as pin D2 on the microcontroller has an interrupt capability, which was found useful to be user accessible (e.g. for connecting encoders).

4.2 Improvements to the firmware

The most important update to the firmware code was to use the API mode (instead of the transparent (AT) mode) of the XBee chip. This made wireless communication between the modules and the coordinator node much more robust.

The API of the library has been expanded to allow for further custom behaviours and extensions of the library. In particular when sending data to the module, it was found that an additional confirmation of reception of the data greatly improves the reliability of the communication. Also, it was found that for smooth dimming and pulsing outputs on the pulse width modulation (PWM) outputs, it was useful to implement possible curves in the firmware and send customised messages with parameters for these curves and triggers to start them, rather than sending continuous updates of the desired PWM value.

4.3 Improvements to the software

In the initial version of the software, we relied on the data sharing network that was also developed during the Sense/Stage research project [2]. Over time, it became clear that the data sharing network was not a necessary feature for most projects using the Sense/Stage MiniBee, so the default mode was set to the plain OSC mode.

Other software improvements include the switch to the API mode of the XBees and numerous updates to enhance the user interface and improve the reliability of communication with the modules.

For one particular project, the software needed to run on an embedded computer. It turned out that the Python software was too CPU-intensive for this application and the software was rewritten in C++. This greatly improved efficiency and allowed the software to run smoothly on the embedded platform. On the other hand, the cross platform distribution of this version is harder, given the available work power to compile the program for different operating systems.

4.4 Maintenance issues

Since the end of the research-creation project, the development of the project has been done by only one person, although a lot of work was leveraged from other open source projects by using existing libraries, looking at other open source hardware designs and example firmware for different projects. While a lot of new features and bug fixes were made while working with users, only few users were able to directly contribute to the technology by submitting bug fixes or adding new features themselves. This may be due to the small user base and due to a lack of an online community to exchange knowledge between users. The platform has drawn those users who are not tech-savvy to get started with wireless sensing quickly; perhaps potential users who have technical experience are more likely to build their own systems.

Examples of how to interface with different environments have been supplied by the community, e.g. for Max/MSP, Isadora and vvvv.

In the maintenance of the software the main difficulty has been to support three different operating systems, and their changing versions.

Finally, keeping documentation up to date with the latest versions of the hardware, firmware and software has been a challenge due to time constraints.

5. AVAILABLE WIRELESS SYSTEMS

Since our initial investigation of available wireless sensing platforms [1], a number of new projects have emerged and become available. They are discussed here in the context of the design considerations for the succesor of the Sense/Stage MiniBee.

- The X-OSC[16] module is a Wifi-module that can be configured via a web interface. Data is received and sent via the OSC-protocol. The module is larger than the Sense/Stage MiniBee and needs a slightly larger battery. The project is not open source, which limits its extensibility. The cost of the module is around 160 GBP.
- The ESP8266[17] is a small and cheap Wifi module (prices ranging from 5 to 20 euro). The module can be used standalone with very limited sensing capability or in conjunction with other microcontrollers. The documentation is aimed at engineers, which makes the module not directly usable by most artists. There are examples available for using the module with Arduino.
- The Simblee module[18] by RFDigital is intended to make embedded devices using Bluetooth Low Energy connections easier for everyone: hardware hackers,

[16]http://x-io.co.uk/x-osc

[17]http://esp8266.com/

[18]https://learn.sparkfun.com/tutorials/simblee-concepts

app developers, students, makers, engineers, and anyone else who wants to leverage their smartphone via BLE.

- The Nordic nRF24L01+[19] is a wireless module that operates in the 2.4 GHz range. Examples are available for using the unit with Arduino. The firmware of the module itself is not open source and cannot be changed. The onboard firmware supports receiving data from up to 6 modules.
- The Pinoccio[20] is a wireless module based on the Atmega2564RFR2[21] chip, which uses an 802.15.4 protocol in the 2.4 GHz band (like XBee does).

This overview is not exhaustive, but reflects a number of different categories of devices available. The X-OSC is accessible for artists through its easy configuration and clear documentation. The ESP8266, the nRF24L01+ and the Simblee module are interesting as devices, but the accessibility for artists is low: users still need to write firmware and software to interface with the modules.

In our original design requirements [1] we wanted to be able to use many nodes at the same time. That makes the nRF24L01+ and the Bluetooth solution not attractive, given the limit to connect or listen to 6 or 7 modules at the same time. Wifi as a solution for wireless communication seems attractive, but has a number of drawbacks in practice:

- a Wifi connection can take time to set up, and
- there is the danger that the presence of many other Wifi-enabled devices in the same space (e.g. mobile phones of audience) can influence the setup time negatively, or cause the connection to be lost.
- Power consumption for Wifi is relatively higher than for other wireless units.
- A practical issue with using Wifi is that while working on a project, you cannot look at online resources for documentation and tutorials; especially with modern computers that do not have an ethernet port anymore, this is a problem.

The Pinoccio project stands out, given that it is completely open source and the Atmega2564RFR2 supports the use of many nodes at the same time in the same space. The target market of the Pinoccio was aimed at the *Internet of Things*, which generally deals with much slower data rates than the realtime sensor communication needed by the artistic projects that Sense/Stage is trying to enable. The project is unfortunately no longer maintained and the forum and support pages are no longer available. On the positive side, the Pinoccio project is open source, so it is possible to build upon the work done on this module for a successor of the MiniBee.

The Sense/Stage MiniBee consists of an Atmega328 microcontroller in combination with an (off-the-shelf) XBee module[22]. The disadvantage of using the XBee module is the additional cost of the XBee module, as well as the fact that the firmware of the XBee itself is not open source. With the Atmega2564RFR2 chip a solution is possible that will remove the dependency on the XBee, which will both reduce the cost of the module and the size (the overall module will be flatter), and allow the firmware to be completely open source. On the downside, the module will need to be certified for use in various countries.

[19] `http://www.nordicsemi.com/eng/Products/2.4GHz-RF/nRF24L01P`

[20] `http://pinocc.io/`

[21] `http://www.atmel.com/devices/ATMEGA2564RFR2.aspx`

[22] The XBee itself has limited sensing and actuation capabilities and is therefor not considered as a standalone module.

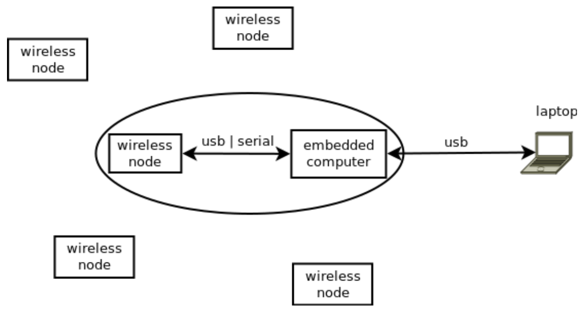

Figure 5: The setup for the new Sense/Stage system

6. DESIGN OUTLINE FOR THE NEXT STAGE

The design outline for the new version was made in a brainstorm session with the author, two experienced users of the Sense/Stage MiniBees and an electronics designer. In the new design we wanted to address the various issues that users mentioned in their feedback, keep the strengths of the MiniBee and address the issues of maintainability of such an eco-system by a small amount of developers.

The basic setup for the system (illustrated in figure 5) consists of a base station with a receiver node to which the other wireless nodes can connect. The base station is connected to a user's laptop and is interfaced via a web interface. The other wireless nodes communicate with this base station, but can also be configured to communicate with each other directly.

6.1 Wireless node

The core design goals for the wireless node are to be small, low cost, enable reliable bidirectional wireless communication, enable the use of multiple nodes in the same space, and use standardised headers for connectivity. The user can hook up sensors and actuators to the wireless module and send and receive the data wireless. Also the module can be used as a serial USB module.

The wireless node and the receiver node are the same hardware — so there is just one board design and the modules are interchangeable.

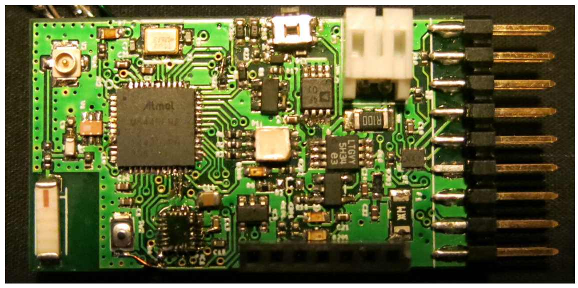

Figure 6: First prototype of the new wireless node.

The main features of the board are an onboard 9 degrees of freedom sensor and a high resolution (16bit) 4 channel ADC, that can also be used in a differential mode, with a programmable gain factor. For powering the device a Li-Ion battery, a user supplied power source or a USB battery can be used. The board also features an onboard charger for the Li-Ion battery.

For the antenna, there are two options: either to use the chip antenna that is onboard, or to connect an external antenna with a U.FL connector (see top left on the board in figure 6). This is particularly important when it is expected that the wireless signal to the chip antenna will be obstructed by either the object that the board is embedded in, or by the body. Furthermore, the pin header has a connection to replace the onboard power button with an external button, so that the module can easily be embedded inside a larger object.

While designing the PCB, it was decided to put the USB interface on a small daughter board to prevent the module from becoming too large. In figure 6 the first prototype of the wireless node is shown.

6.2 Base station

The concept for the base station is inspired by the Bela project[23][3, 4], where the embedded platform can be programmed and configured via a web interface; also the documentation will be available through this interface. One wireless node is connected via USB-serial to an embedded computer which runs the software needed to act as a bridge between the user's own software and the wireless modules. The embedded computer is hooked up to the user's computer via USB and shows up as a networking device. The user can configure the wireless modules through a web interface and select a configuration to use at startup; the data from the wireless modules is then received in the user's software via OSC. As all the software is running on the embedded system, the user does not need to install any special software on his own computer; this means that only one platform needs to be maintained by the developers, and this platform is a controlled environment. For users it means that the use of the system is not dependent on upgrades on their own computer, or they can easily switch to another computer. To summarise, the base station:

- consists of an embedded computer (e.g. BeagleBone, Raspberri Pi, PiZero);
- includes a wireless node to talk to the other nodes;
- runs the software that talks to the firmware on the wireless node;
- connects to user's computer via USB and shows up as an IP interface;
- has a web interface for configuring the functionality of the wireless nodes in the network;
- has a web interface to make custom firmware for wireless nodes;
- has browsable documentation that corresponds to the version of software that is installed on it.

6.3 Firmware and software

For the firmware we identified three different usage levels: the basic level to support the most common use cases, an intermediate level where firmware can be customised through templates that support special sensors or actuators, and the expert level that provides an API to the firmware library to make completely custom firmware.

The software bridge will talk via a serial interface with a wireless node (that acts as coordinator in the network), manage the configuration of the nodes (config file load/save; osc update) and provide the communication interface to other programs via OSC. Optionally, also MIDI and HID protocols can be enabled.

6.4 Documentation and community

The documentation structure will be completely revised and consist of both reference documentation as well as tutorials to accomplish particular tasks. The documentation will be available both online (for the latest version) and on the base station (matching the version of the software that is running). In addition a PDF will be available as a basic manual to get started.

An online forum will provide a platform for discussion, where users are able to ask questions. This ensures that the answers to particular questions will also be available to other users, and users can help each other.

[23] `http://bela.io/`

7. CONCLUSION

The continued interest of artists in the Sense/Stage MiniBee makes clear that there is a demand amongst artists for an affordable and accessible wireless platform. The module and its associated firmware and software have shown that the Sense/Stage platform has a number of advantages over other systems that are on the market. The core advantage is that the system is open enough for artists to customise it to their needs, while at the same time being easy to use: to set up and get started because of the preprogrammed firmware and the associated software.

The design for the successor of the MiniBee takes into consideration the various issues that came up over a period of 6 years being used out in the field by various artists for a wide range of projects, while at the same time addresses the issues of sustainability by a small crew of developers.

8. ACKNOWLEDGEMENTS

The development work on the new module is funded by the Creative Industry Fund NL in the project *Sense the next Stage*. A special thanks goes to Till Bovermann, Jonathan Reus and Simon Claessen for their collaboration on this project, as well as to all the users that provided feedback through the online questionnaire, their feedback and continued support of the project, as well as to all the places where Sense/Stage workshops were hosted over the past years. The development work on the XthSense[24] project with Marco Donnarumma has informed the design for the new module immensely.

The Sense/Stage project is documented at `https://sensestage.eu`.

9. REFERENCES

[1] M. A. Baalman, V. de Belleval, C. L. Salter, J. Malloch, J. Thibodeau, and M. M. Wanderley. Sense/Stage — low cost, open source wireless sensor and data sharing infrastructure for live performance and interactive realtime environments. In *International Computer Music Conference (ICMC) 2010*, New York City / Stony Brook, NY, USA, June 2010.

[2] M. A. Baalman, H. C. Smoak, C. L. Salter, J. Malloch, and M. Wanderley. Sharing data in collaborative, interactive performances: the SenseWorld DataNetwork. In *New Interfaces for Musical Expression (NIME 2009)*, Pittsburgh, PA, USA, 2009.

[3] A. McPherson and V. Zappi. An environment for submillisecond-latency audio and sensor processing on beaglebone black. In *Proc. AES 138th Convention*, Warsaw, Poland, 2015.

[4] G. Moro, S. A. Bin, R. H. Jack, C. Heinrichs, and A. McPherson. Making high-performance embedded instruments with bela and pure data. In *International Conference on Live Interfaces (ICLI)*, Brighton, UK, 2016.

[24] `http://www.xth.io/`