A New Wi-Fi based Platform for Wireless Sensor Data Collection

Jim Torresen, Yngve Hafting and Kristian Nymoen
Department of Informatics, University of Oslo
P.O. Box 1080 Blindern, N-0316 Oslo, Norway
{jimtoer,yngveha,krisny}@ifi.uio.no

ABSTRACT

A custom designed WLAN (Wireless Local Area Network) based sensor interface is presented in this paper. It is aimed at wirelessly interfacing a large variety of sensors to supplement built-in sensors in smart phones and media players. The target application area is collection of human related motions and condition to be applied in musical applications. The interface is based on commercially available units and allows for up to nine sensors. The benefit of using WLAN based communication is high data rate with low latency. Our experiments show that the average transmission time is less than 2ms for a single sensor. Further, it is operational for a whole day without battery recharging.

Keywords

wireless communication, sensor data collection, WLAN, Arduino

1. INTRODUCTION

There has in recent years been an increasing interest in using body movement for controlling interactive systems, including musical applications. A number of larger research projects have investigated such potentials, e.g. the MEGA project [3], Sound to Sense – Sense to Sound ($S2S^2$) [9], Gesture Controlled Audio Systems (ConGAS) [5], the Sonic Interaction Design project [2], etc. There is now also a commercial impact of such thoughts, with Nintendo Wii, Microsoft Kinect and Apple's iPhone as examples of exploitation of the potential of motion sensing in interactive devices.

There are numerous challenges when it comes to developing such interactive systems, including the development of smaller, faster, cheaper and more precise sensor systems. Another important factor is to reduce the latency in the systems, to ensure that the user of an interactive system gets an immediate response to an action being carried out.

An important feature of many new digital interfaces for musical expression is wireless sensing. Smartphones and media players typically have a large range of built-in sensors but external sensing allows for sensors being distributed on the body and allows for a larger extent of expressivity. The ideal solution would be designing an interface being able to connect many sensors, being small, fast, inexpensive, accurate/precise, consume little battery power, and, perhaps the most important: be reliable in all sorts of performance contexts.

Figure 1: WLAN based sensor interface with 3D-printed casing.

One problem with some media devices like Apple's iOS-devices is that Apple does not allow end users to connect hardware via the embedded connector, without being part of their *MFi-programme* (MFi is abbreviation for Made For iPhone/Pod/Pad). The MFi-programme is only commercially available and employs an identification chip made by Apple, which effectively puts it out of the question for academic users. In order to get around this problem, wireless connection is the only option. However, it has the benefit that porting an application with an external interface to another operating system, would not require change of connectors and hardware configuration.

In this paper we report on a project aimed at creating a reliable and powerful wireless sensor platform to be used with a media player, see Figure 1. The target is (inter)active musical applications to be controlled by human motion and condition. High speed is often critical when designing musical *instruments*, where a total latency of approx. 10 ms may be the upper limit [11]. However, in adaptive music devices, aimed at giving the user an active listening experience, the tolerance for latency may be considerably higher [6, 8].

The paper starts with introducing some wireless standards, followed by a brief overview in Section 3 of the core components that our platform is based on: the Arduino microcontroller board, Wi-Fi transceiver and accessories. Section 4 reports on communication latency for the sensor platform.

2. BACKGROUND

The de facto standard for short range wireless transmission is Bluetooth (IEEE 802.15.1), a technology which is currently embedded in a variety of commercial devices, ranging from computer mice to mobile phones, cameras, printers, etc. While it certainly works in many contexts, our experience is that Bluetooth is not reliable enough for music-

Table 1: Comparison of ZigBee, Bluetooth and WLAN, based on [4, 7]

	WLAN	Bluetooth	ZigBee
Band	2.4, 3.6 and 5 GHz	2.4 GHz	2.4 GHz, 868/915 MHz
Power	500 mW	100 mW	30 mW
Battery life	Hours	Days – months	6 months – 2 years
Range	30–70	10–30 m	10–75 m
Data rate	1–150Mbps	1–3 Mbps	25–250 Kbps
Network	Ad hoc, P2P	Ad hoc, P2P, star	Mesh, ad hoc, star
Security	WEP(128-bit)/WPA2(256-bit)	128-bit encryption	128-bit encryption
Wake and transmit	10 ms	3 s	15 ms

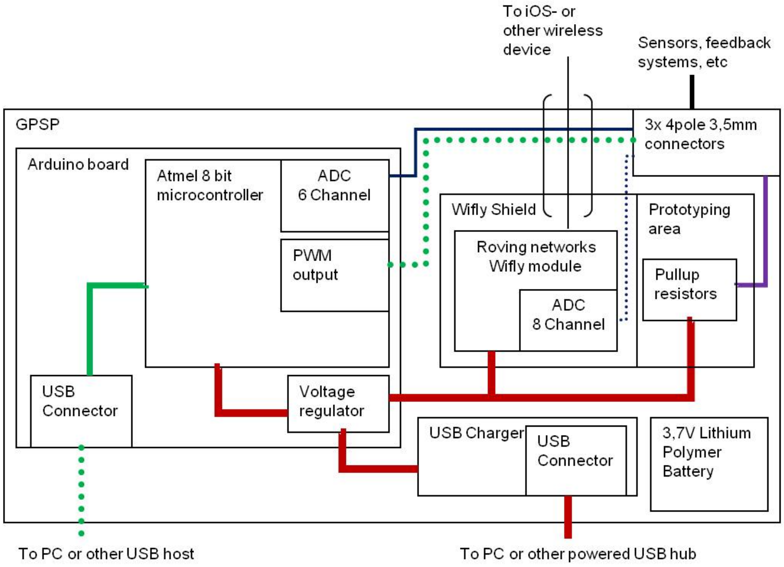

Figure 2: The general purpose sensor platform (GPSP).

related applications. An alternative solution is to use ZigBee (IEEE 802.15.4), which allows for creating systems that are smaller and less power-consuming than Bluetooth, while at the same time being reliable [10]. However, the data rate is limited and few smartphones and laptops have ZigBee transceivers included. WLAN (IEEE 802.11) transceivers on the other hand are typically embedded in most portable media devices and represent a much more powerful and reliable way of communication. However, such systems are also typically larger, more expensive, and more power-consuming than Bluetooth systems. See Table 1 for a comparison of WLAN with ZigBee and Bluetooth [4, 7].

IEEE 802.11 is a set of standards for implementing Wireless Local Area Network (WLAN) computer communication in the 2.4, 3.6 and 5 GHz frequency bands. The base version of the standard IEEE 802.11-2007 with subsequent amendments, represents the basis for wireless network products using the Wi-Fi brand name. This is a standard for wirelessly connecting electronic devices. Many such devices like personal computer, video game console, smartphone, tablet and digital audio player are today equipped with such a communication unit. Wi-Fi units can connect to a network resource such as the Internet via a wireless network access point. It has a range of about 30 meters indoors (less if obstructions like brick walls) and a larger range outdoors [4].

3. A PLATFORM FOR INTERFACING SENSORS

In this section the developed sensors interface is outlined. We have focused on the need for flexibility when it comes to number and variants of sensors to be connected.

This system, which we call the General Purpose Sensor Platform (GPSP) – see Figure 1, is intended to wirelessly collect sensor data. It can either be used as a standalone unit or as we plan, in addition to the sensory input available within a media device. We would apply iPod Touch devices and work with interfacing sensors that may be relevant for (inter)active music systems, e.g. external force sensors applied in shoes. A sensor interface is typically a compromise between the number of sensors connected, communication time, cost and size. By using flexible prototyping boards and rapid prototyping, our platform represents a highly flexible system for collecting data relevant in various music contexts.

The GPSP consists of a box containing a sensor input and microcontroller unit, a unit for wireless communication, a power supply unit, a battery, and connectors, as shown in Figure 2. The weight of the GPSP is approximately 120g. Figure 3 shows a picture of the inside of the GPSP. The following subsections cover the GPSP in detail.

3.1 Microcontroller

A microcontroller unit was included to allow for performing computational tasks based on the sensory input before forwarding data to the main unit. For example, it is possible to check whether the input from each sensor has changed significantly enough to be forwarded or undertake some simple processing of the sensor input to reduce data to be transmitted. We chose an Arduino [1] board for several reasons:

- The Arduino platform is widely used; thus much sample code and many examples of use are available on

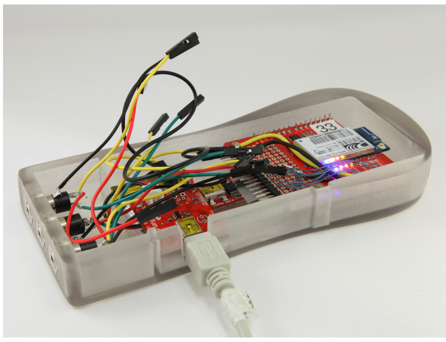

Figure 3: Picture of the GPSP.

the Internet.

- A family of development boards are available in different sizes and shapes. This allows for starting out with one board, and then later changing to a smaller or larger one depending on whether more IO or using less space and weight are desirable.
- The boards can be powered over USB, or by batteries.
- Using the Arduino Uno or Pro, there is a much more powerful, yet pin-compatible development board available (FEZ-Domino).

In our current platform, we use an Arduino Pro board containing an Atmel ATmega328 microcontroller.

3.2 Wireless Transceiver

The Wi-Fi transceiver is a Sparkfun WiFly shield, featuring a Roving networks WiFly GSX module. This module supports both the creation of its own ad-hoc networks and connecting to other networks. The module itself can be configured wirelessly, and it does provide analogue and digital IO connections which means that it can be used standalone, without the Arduino board – if desirable. Using the wireless module alone will not allow for local computation, but it may allow for higher throughput, since the microcontroller part of the chain is removed.

The WiFly shield is designed using a printed circuit board with matching pin slots to the Arduino Uno and Pro. The shield configuration makes it very easy to attach to the Arduino circuit board. Together with the Arduino Pro, their combined height is about 13mm, less than the height needed for the connectors at the end of the device. The drawback of using the shield is that it only allows for using three of the eight analogue inputs available on the WiFly device mounted on the shield. Connecting the WiFly device directly to the Arduino board would be possible and give a more compact system. However, it would be a less robust connection and the lack an interface chip on the shield would reduce communication speed to the Arduino microcontroller.

The WiFly shield also contains a small prototyping board area for soldering connecting wires, resistors, and other hole-mounted components. Using wireless communication will impact the response time and the throughput for the system, compared to a wired system. Thus, if higher communication speed is needed, we may use the USB or a serial connection to the Arduino board. However, this will drastically limit the types of devices we can connect to. E.g. most media devices will not allow for such a connection. If, on the other hand, we can settle with lower throughput than WLAN provides, we can also look into replacing the WiFly module with a Bluetooth or Zigbee module. Selecting such an option may reduce the power usage significantly but with the drawbacks outlined in Section 2. Thus, it seems that WLAN offers the greatest flexibility for our setup.

3.3 Power Supply and Batteries

In order to make the device able to work without being physically connected to other devices, we added a rechargeable lithium polymer battery together with a USB charging circuit board. Using this combination to power the system allows for continuous operation, both while the battery is charging when connected to a USB port, and when the USB cable is detached. The battery has a capacity of 850mAh and a weight of about 18.5g. It can easily be changed to both smaller and bigger variants.

3.4 Connectors

The connectors chosen for this prototype are panel mount 4 pole 3.5mm sockets. They were chosen because of their sturdiness. For each of these connectors, we can have up to three sensors connected to one plug. With three of these, we may connect up to nine sensors, utilizing both the six analog inputs at the Arduino board and the three available on the WiFly shield. The connectors may also easily be replaced with similar sized 3 pole sockets, commonly used for stereo headphones. This may be desirable, since their plug counterpart is available in versions that are easier to solder than the 4 poled version. The USB charger and the Arduino board both have USB connectors ready to use.

3.5 The Casing

The casing is designed using 3D design software, and printed on a 3D printer. Designing a new casing and making a new prototype is a task that can be performed in as little as one day, depending on the complexity of the structure chosen. Thus, we have the option to change the design all the way until we have the final version ready.

4. TIMING AND BATTERY TESTING OF THE PLATFORM

To test the real-time performance of the sensor platform, we set up a simple test with the GPSP device and a PC to measure the latency introduced by sending data using UDP (User Datagram Protocol) packets over WLAN. As shown in Figure 4, the test setup consisted of the GPSP device receiving packets from the PC and acknowledging the receipt back to the PC. The PC would then compute the "time of flight" between sending and receiving each UDP packet.

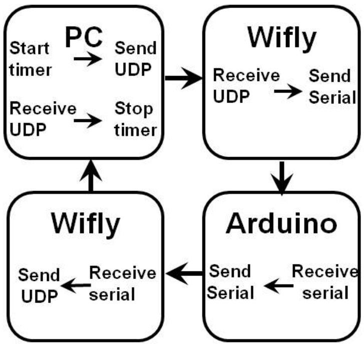

Figure 4: Flow chart of packet sending between PC and GPSP.

Detection of packet loss was also possible since the byte value sent by the PC was incremented for each packet. An ad-hoc network created by the PC was applied for the

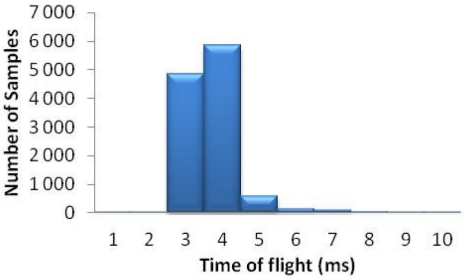

Figure 5: The time of flight distribution for sending packets of one byte from a PC until acknowledge is received by the PC.

	A	B	C
Interference	Small	Medium	Medium
Packet interval (ms)	5	5	3.1
Packet loss (%)	0	0	0.12
Avr. time (ms)	3.75	3.80	6.60
Std deviation	1.01	5.53	13.47
95% Percentile (ms)	5.0	5.0	14.0
Max delay (ms)	19	106	149

Table 2: Results from three different tests of WLAN packet transmission.

test rather than using an external router (wireless network access point) that would induce further delays, typically adding 4ms to the delays reported below according to our measurements.

The test was repeated until the best settings for the network were found, including a data rate of 6Mbit/s. By undertaking the test in an office environment with multiple networks sharing frequencies, the delay and packet loss varied with the activity on other networks. We used the inSSIDer program to monitor the network frequency usage and to select the channel that would most likely have the least interference. With limited interference, we got the delay time distribution as seen in Figure 5. The measurement started after the network initialization was finished. In this test, a new packet was sent every 5ms for a total of 57s, 11326 packets in total. The average time of flight was 3.75ms with a standard deviation of 1.01ms. This time is for sending both to and back from the GPSP, thus, the real sensor reading time would be half of the times reported here, i.e. an average delay of less than 2ms.

Even with more interference, most packets came through. Table 2 summarizes the results from three different tests with variation in network interference and packet transmission interval, respectively. The first column (A) reports numbers for the test in Figure 5. In the test with larger interference (B), the average delay only increased to 3.80ms but the standard deviation was substantially larger (5.53ms). If the packet interval is shrunken down to 3.1ms together with having interference, the timing gets similarly worse as seen in the table (C). For experiments A and B, we see that 95% of the packets arrive in 5ms or less, indicating a satisfactory reliability for real-time musical performance. However, to reduce the maximum delay, it would be important to select the channel with least interference.

The tests above report only for packets of one byte each. Typically, we would like to send data from several sensors at the same time. Each UDP packet would easily contain all samples in a single packet. This includes Open Sound Control (OSC) messages which we have implemented in our system. Further, the A/D conversion time is marginal. The transfer of data from the Arduino board to the WiFly module is through an SPI-to-UART chip on the WiFly board. By adjusting the data rate in that unit, we have achieved sending samples from all the six Arduino sensors and the timing data wrapped in a 24 byte OSC message every 6th millisecond. This resulted in a total delay from sampling to reception of data from all six sensors of 8ms given low network interference. We expect increased data rate is possible by further optimizing the SPI chip settings as well as upgrading to one of the more powerful microcontroller boards being available.

A potential challenge with WLAN systems is large power consumption and demand for frequent battery re-charging. Thus, a test was undertaken with continuous transmission of data from six sensors connected to the six A/D converters on the Arduino board (sampling rate of about 40Hz). The battery lasted 15 hours which indicates that the device can be applied for a whole day without recharging[1].

Future work consists of further improving our system and incorporating it into music related applications.

5. ACKNOWLEDGMENTS

The research is funded European Union Seventh Framework Programme under grant agreement no 257906, www.epics-project.eu

6. REFERENCES

[1] Arduino web page. www.arduino.cc.

[2] Sonic interaction design. sid.soundobject.org.

[3] A. Camurri, G. De Poli, A. Friberg, M. Leman, and G. Volpe. The MEGA project: Analysis and synthesis of multisensory expressive gesture in performing art applications. *Journal of New Music Research*, 34(1):5–21, 2005.

[4] Y. Chen, D. Lymberopoulos, J. Liu, and B. Priyantha. FM-based indoor localization. In *Proc. of 10th Int. Conference on Mobile Systems, Applications and Services (MobiSys 2012)*. ACM, 2012.

[5] R. I. Godøy and M. Leman. *Musical Gestures: Sound, Movement, and Meaning.* Routledge, New York, 2010.

[6] M. Goto. Active music listening interfaces based on signal processing. In *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, volume 4, pages IV–1441. IEEE, 2007.

[7] J. Kooker. Bluetooth, zigbee, and wibree: A comparison of wpan technologies, www.cse.ucsd.edu/classes/fa08/cse237a/topicresearch.

[8] B. Moens, L. van Noorden, and M. Leman. D-jogger: Syncing music with walking. In *Proceedings of SMC Conference 2010*, pages 451–456.

[9] X. Serra, M. Leman, and G. Widmer. A roadmap for sound and music computing. Technical report, Sound to Sense, Sense to Sound, 2007.

[10] J. Torresen, Ø. N. Hauback, D. Overholt, and A. R. Jensenius. Development and evaluation of a ZigFlea-based wireless transceiver board for CUI32. In *Proceedings of the International Conference on New Interfaces For Musical Expression (NIME)*, pages 409–412, Ann Arbor, Michigan, 2012.

[11] D. L. Wessel and M. Wright. Problems and prospects for intimate musical control of computers. In *Proc. of the 2001 Int. Conference on New Interfaces for Musical Expression (NIME)*. ACM Press.

---

[1] The is based on the assumption that a typical application would not need continuous low latency throughout a whole day.