DUL Radio: A light-weight, wireless toolkit for sketching in hardware

Martin Brynskov
brynskov@imv.au.dk

Rasmus B. Lunding
rasl@cs.au.dk

Lasse Steenbock Vestergaard
lassesvestergaard@gmail.com

Information and Media Studies
Center for Digital Urban Living
Aarhus University
Helsingforsgade 14
DK-8200 Aarhus
Denmark

TEI 2011, January 23-26, 2011, Funchal, Madeira, Portugal

Abstract
In this paper we present the first version of *DUL Radio*, a small, wireless toolkit for sketching sensor-based interaction. It is a concrete attempt to develop a platform that balances ease of use (learning, setup, initialization), size, speed, flexibility and cost, aimed at wearable and ultra-mobile prototyping where fast reaction is needed (e.g. in controlling sound). The target audiences include designers, students, artists etc. with minimal programming and hardware skills. This presentation covers our motivations for creating the toolkit, specifications, test results, comparison to related products, and plans for usage and further development.

Keywords
sketching, toolkit, ultra-mobile, wireless

ACM Classification Keywords
H.5.2 [Information Interfaces and Presentation]: User Interfaces—prototyping; C.5.3 [Computer System Implementation]: Microcomputers—microprocessors

General Terms
hardware, prototyping, interaction design, tools, teaching

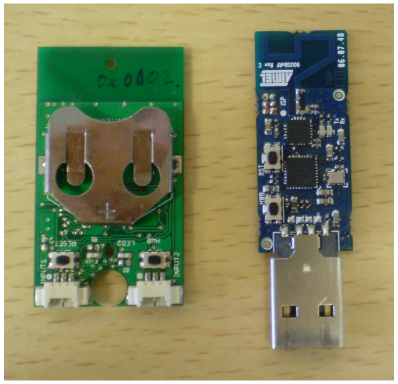


Figure 1. The DUL Radio board (left) and receiver (right). The board measures 2.5 x 4.3 x 0.6 cm (1 x 1.7 x 0.2 inches) including antenna, two analog ports, an accelerometer, and a battery (3 V lithium cell) – compare to the standard USB Type A plug (right).

Introduction

One of the main obstacles in tangible and embedded interaction design is the complexity of tools and competences required for sketching and casual prototyping. Whether for educational[1], designerly[2], or artistic[3] purposes, various other projects have addressed this challenge, and a range of tools and platforms have emerged. *DUL Radio* (Fig. 1) is a concrete attempt to develop a platform for sensor-based interaction design that is both (a) easy to use and (b) has a specific feature set not typically present in a single, unified system.

Ease of use applies to the point of view of the prospective interaction designer (whether formally trained or not): no steep learning curve, easy initialization and set-up, requiring no or minimal programming skills or hardware experience. Thereby, we aim to accommodate users who are not (yet) technically skilled, typically designers, students, and artists.

The feature set of any platform represents trade-offs as a result of different developer traditions, standards, needs, end-product perspectives, and even ethics. But in general, the development environments tend to favor either ease of use or low-level hardware control.

DUL Radio seeks to balance size, speed, flexibility, and cost optimized for wearable and ultra-mobile prototyping scenarios where fast reaction is needed (e.g. in controlling sound and triggering events). It aims to be efficient and flexible while still favoring ease of use.

Specifications

DUL Radio consists of a hardware kit and an accompanying software part.

Hardware

Hardware-wise, the toolkit is made up of

- a wireless data-transmitter (the DUL Radio board),
- a receiver (standard product)[4],
- various sensors.

SENSOR BOARD SPECIFICATIONS

- Physical size: 2.5 x 4.3 x 0.6 cm (1 x 1.7 x 0.2 inches) including antenna, two analog ports, an accelerometer, and a battery.
- Power: 3 V lithium cell.
- Processor: Atmel ATMega168.
- Transceiver: Nordic nRF24L01 (2.4GHz).
- Accellerometer (ACC), digital: Analog Devices ADXL345 (3D g-force, tilt, single/double-click, and free-fall).
- 2 x 10 bit analog input ports (ADC).
- Optionally one of the analog input ports can switch to a pwm-controlled output (not implemented in the current version).

Each receiver can communicate with up to 4 sensor boards simultaneously (multi-performer operational), thus giving a total of 8 analog inputs + 4 ACCs per receiver. Since the sensor board is equipped with transceivers (capable of transmitting as well as receiving), DUL Radio can potentially be used for node-based setups, were the transmitters are turned to stand-alone data-handling entities, that communicate in chains or groups and make decisions based on on-board code. It can be developed to meet the ZigBee standard. See Fig. 2 for a test setup.

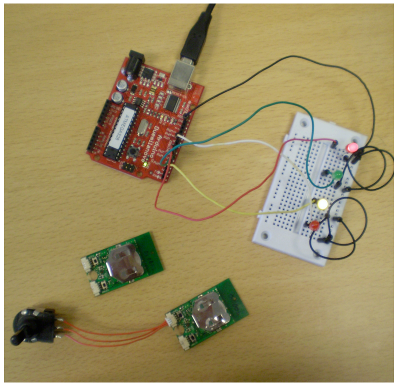


Figure 2. Test setup. Two sensor boards (bottom/middle), one with an external sensor (potentiometer, far left) transmitting wirelessly to at laptop with a receiver (not visible). The signal is sent via the laptop to an Arduino board (top) which is connected to four LEDs. The LEDs represent the state of the two accelerometers and two analog ports.

Software
The software in the toolkit can be separated into two groups: development software (bootloader etc. for the platform) and the user interface (for the interaction designer).

DEVELOPMENT SOFTWARE
The DUL Radio features a custom built bootloader and other pieces of software to make the basic functionality and configuration accessible in a convenient way. Parameters that can be set in software include:

- de/activate ports (ADCs, ACC)
- wake mode (by ADC, ACC or both)
- threshold for wake, timeout for sleep
- data format (binary or ASCII)
- sample rate
- choose transmitting radio channel

Apart from the software that is required for basic functionality, additional software can be be developed and downloaded to the EEPROM on-board.

USER INTERFACE
The Ease of use-aspect essentially deals with the development of the interface through which the user configures and controls the hardware. PicoBoard[5] using the Scratch programming interface is a notable example of an easy-to-use interface to embedded programming.

The DUL Radio targets adults who know how to operate computers and use ordinary application software, thus the user interface does not have to be overly simple, but it should allow the user/designer to focus on the essentials: The design-process and realization through triggering of events and handling of sensor/actuator data.

The sensory-data transmitted is made available to the receiving system on a serial port. The current setup uses a standard Type A USB receiver attached to a laptop (Mac, Linux or PC works fine), but that could as well be any other USB host-enabled device, e.g. an Arduino or a USB On-The-Go-enabled[6] mobile phone (e.g. Nokia N8). The data can then be picked up by any application capable of listening to a serial port, e.g. Processing or a Java program.

For our purposes, and at this early stage, we are using the Max[7] environment which is widely used by audio designers and performance artists. In order to test the platform, a java-external handling the data-strings and three simple Max patches were developed (Fig. 3-5).

Tests
We have run a series of initial performance tests on the first version. The sensor boards will run about +5 days (+120 hours) with continuous wireless data transfer from the two ADCs running at 50 Hz. This is a high-performance scenario. With only ACC on and lower transfer rate, battery life will be considerably longer. The antenna range is about 10-15 m (30-50 feet) indoor (with lots of RF devices around) and 45-70 m (150-230 feet) outdoor.

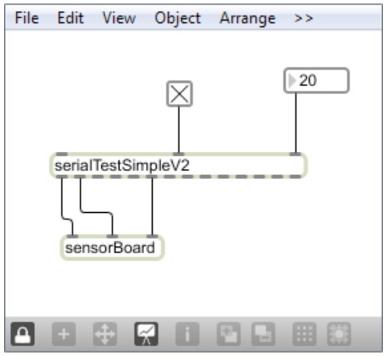
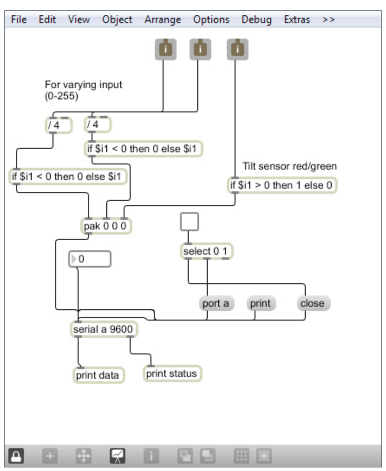


Figure 4. This patch connects the patch that handles sensor board input (Fig. 3) and the patch that handles output to an Arduino board.

Figure 5. This patch converts the data from the sensor board to numbers that the Arduino board understands and sends it to the Arduino board.



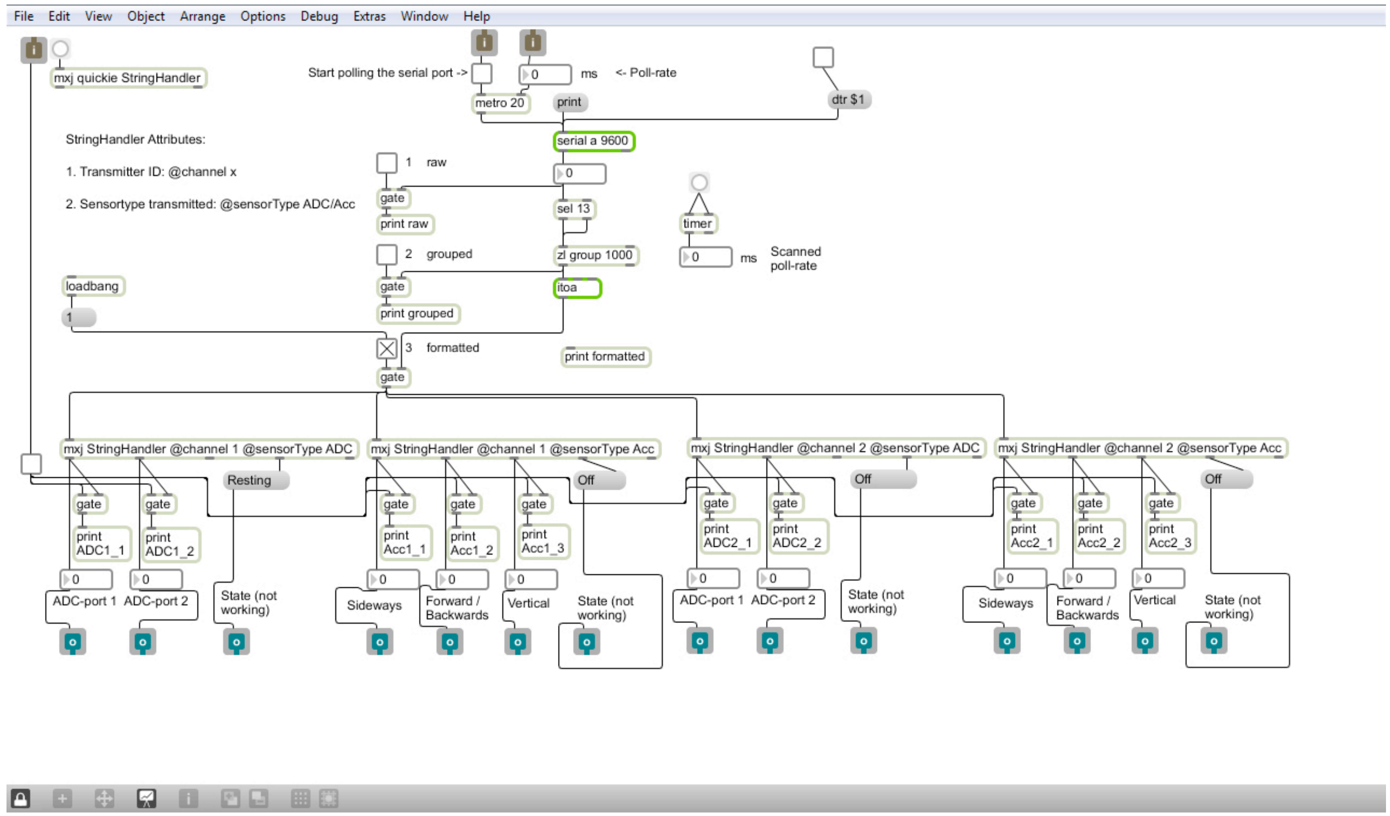


Figure 3. Max[7] patch that gets input from the sensor board, separates ADC from ACC data and sends the split data to specific outlets. See also Fig. 2 (above) for the physical setup.

Comparison with related products

In 2006, addressing a design issue requiring data communication from several moving sensors in a room (equipped in shoes) wirelessly and at a fast rate (fast enough to track step-dance-like behavior) and small enough to build into the shoes so no visual cue would reveal the embedded enhanced features of them, we realized that such a design task was not easily met with off-the-shelf products. Mobility, pervasiveness, and size where key features, so constructing a responsive floor-layer was not an option. A system designed to address this and similar tasks was not readily available to us. So we had to build our own.

We made an early prototype system which supported the task. It consisted of several small transmitter units equipped with one analog input transmitting to one central receiver. The system described in this paper is a semi-standardized and generalized product designed with the early prototype system as an inspiration, and with the aim of addressing the two key areas: ease of use and the specific features we needed. One might add, that in order to be useful in teaching, it should also be available in large quantities and at a reasonable price. Four years is many life-times in terms of technological products. Our updated survey reveals that products once marketed with success are now obsolete, and new products have entered the market. As it turns out, a 1-to-1 comparison is not possible. Different design features are favored over others in the different products. But an attempt to compare relevant products shows the following systems, grouped in two categories.

Products based on the IEEE 802.15 classification of wireless personal area networks (WPAN) – primarily Bluetooth (IEEE 802.15.1) and ZigBee (IEEE 802.15.4) – are typically systems that are prepared for OEM products or prototyping schemes. Power-consumption, advanced time-out algorithms, and on-board sensors are attractive features. However, the usability for non-programmers and non-engineers is next to none. We shall call this category "Middleware". Comparatively, searching for consumer-oriented products that meet our needs are also in vain. These products focus on ease of use but are typically not equipped with power-saving features, wireless RF data transmission etc. We shall call these plug-and-play solutions "Off-the-shelf products".

The following is a list of products and platforms available on the market relating to DUL-Radio:

Off-the-Shelf products (as of nov. 2010)

- **Eowave: eobody2 hf:** Wireless interface with 16 sensors per transmitter, 1 transmitter per 1 receiver. 12 bit, meets IEEE 802.15.4 (ZigBee) standard.
Size: Large (pouch size: 105 x 95 x 10 mm, board size: 6.5 x 6 x 1 cm incl. battery).
Battery lifetime: not estimated.
- **Eroktronix: MidiTron:** Wireless Interface to MIDI. Up to 10 analog or 20 digital inputs, 900 MHz FM (not ZigBee), 10 bit.
Size: Transmitter ca. 6 x 3.8 x 3.8 cm (incl. battery).
Battery lifetime: not estimated.
- **Infusionsystems: Wi-microDig:** Wireless Bluetooth interface, 8 analog inputs, 10 bit.
Size: 2.5 x 2.5 x 8 cm (incl. battery).
Battery lifetime: estimated to 4 hours (9 V).

Middleware (as of nov. 2010)

- **avrfreaks: iDwaRF:** Sensor-node for wireless sensor networks.

- **TECO PARTiCLE: zPart:** ZigBee, prepared for add-on sensors, formerly Smart-Its. Size: 3.5 x 2.8 cm.
- **gumstix: Overo Wireless:** Multifunctional interface with 6 analog inputs, 6 pwm-outputs, camera input, based on Overo Fire, bluetooth or FM (900 MHz). Size: 5.8 x 1.7 x 0.4 cm (without battery), no direct connections; needs external modules for sensors etc.
- **Phidgets SBC:** Fully functional single-board computer, 64 MB RAM. 8 analog inputs, 8 pwm-outputs, 8 digital ports. Wireless connection based on 900 MHz FM. Size: no measurements, but relatively big.

Arduino

An obvious candidate is the Arduino platform which we would categorize as in between middleware and off-the-shelf. It is not that easy to use, and wireless setups require extra components.

- **Arduino Fio:** The Fio is a controller equipped with 8 analog inputs (10 bit), 14 digital inputs (of which 2 can be flipped to pwm-outputs). Not wireless. Like Arduino boards in general, an Xbee ZigBee transceiver-module is required, and another Xbee must be used. Size: 6.8 x 2.8 cm (not incl. battery).

There are other categories, e.g. teaching and highly specialized OEM systems (weapons, tracking etc.) which we have looked at but not found fulfilling.

Conclusion and future development

We have presented the DUL Radio which is now in version 1. It aims to be easy to use for tech novices while still retaining a feature set only found in less accessible products, according to our survey. The toolkit has reached a level of hardware maturity where we can begin to scale up the prototyping workshops and use it in teaching. With a scheduled larger-scale production, the price will come down. At the moment, production and handling is relatively expensive, while the components are fairly cheap. After a phase of field testing, we will consolidate the prototypes on the user interface software, moving from ad hoc Max patches to better designed applications. We look forward to getting feedback on the platform, both from our internal use and from others.

Acknowledgements

Our sincere thanks to Jesper Nielsen and Jacob Andersen at the Alexandra Institute who did much of the development. Research funded by the Danish Council for Strategic Research, 09-063245, (Digital Urban Living) & EU Regional Fund/Ebst.dk grant 08-0018.

References

[1] Resnick, M., Martin, F., Sargent, R. and Silverman, B. Programmable bricks: Toys to think with. *Ibm Syst J*, 35, 3-4 (1996), 443-452.
[2] Gross, M.D. and Do, E.Y.-L. Demonstrating the electronic cocktail napkin: A paper-like interface for early design. *Proc. CHI 1996*. ACM (1996), 5-6.
[3] Mellis, D.A., Banzi, M., Cuartielles, D. and Igoe, T. Arduino: An Open Electronics Prototyping Platform. *Proc. CHI 2007* (2007)
[4] Atmel, ATAVRUSBRF01 USB 2.4 GHz, http://www.atmel.com/dyn/products/tools_card.asp?tool_id=4322.
[5] Playful Invention, PicoBoard, http://www.picocricket.com/picoboard.html.
[6] USB On-The-Go, http://www.usb.org/developers/onthego/.
[7] Cycling '74, Max, http://cycling74.com/products/maxmspjitter/.