EoBody : a Follow-up to AtoMIC Pro's Technology

Emmanuel Fléty
Engineer - Creation Department
Ircam - Centre Pompidou – 1 pl. Igor Stravinsky
75004 Paris – France
+33/1 44 78 48 43
emmanuel.flety@ircam.fr

Marc Sirguy
Director
EoWave CEO - 7, rue Rouvet
75019 Paris - France
+33/1 40 05 17 18
mesi@mesi.fr

ABSTRACT
Ircam has been deeply involved into gesture analysis and sensing for about four years now, as several artistic projects demonstrate. Ircam has often been solicited for sharing software and hardware tools for gesture sensing, especially devices for the acquisition and conversion of sensor data, such as the *AtoMIC Pro* [1][2]. This demo-paper describes the recent design of a new sensor to MIDI interface called *EoBody*[1]

Keywords
Gestural controller, Sensor, MIDI, Computer Music.

1. INTRODUCTION
The design of our previous sensor digitizing system aimed to be as versatile as possible [2]. This attempt was quite successful, thus satisfying : it showed that despite of the numerous contexts in which the device was used, the multiple functionalities solved the problem. The multi-configuration feature was also a real asset, especially for universities and art schools where several students shared the same unit. Nevertheless, *AtoMIC Pro* also appeared to be over-dimensioned for some applications and also quite expensive to build (and so to sell). So, our next wish was to design another analog to MIDI interface more adapted to electronic music and to end-users who would like to be able to experiment with sensors in their home-studio at a reasonable price (half the price of high-ends existing interfaces).

2. NEW DESIGN, OTHER COMPROMISE
The goal of this design was not to replace *AtoMIC Pro*. In order to reduce size and cost, we had to choose what to keep and to suppress from our original design.

We really meant to keep attractive features from *AtoMIC Pro* but the cost aspects lead us to remove the LCD and so the onboard configuration system[2]. Of course we kept the facility of a stand alone mode : once configured, the setup is stored in a non-volatile memory. Only one configuration setup can be stored in the unit at this time, but libraries can be stored on the host computer that runs the configuration editor.

The number of inputs has been decreased to 16, but we added 3 onboard potentiometers (knobs) and 4 buttons (switches) : thus, the user does not need to waste any analog inputs for implementing some "classic" sensors. The buttons and potentiometers, like the analog inputs, can be mapped to any MIDI message.

Figure 1 : The EoBody unit

All these significant modifications and restrictions might lead us to think this interface was getting closer and closer to other commercial products such as the I-Cube system. Nevertheless, what motivated the design of such an interface was the latency and sample rate aspect. As a matter of fact, existing systems are still clocked at a sampling period varying between 1 and 4 ms (or more) and we really wanted to keep the converter running as fast as possible : the fewer sensors you use, the faster it goes. The quality of the sampled gesture depends on this [3].

While considering sampling performances we also got interested in the portability aspect. Thus, we decided to port into the C language some software modules from AtoMIC Pro. In order to keep the system fast, critical sections (sampling and interrupts, among others) were written in assembler, but most of the code is easily portable and upgradeable.

The consequence of this port is a slight increase of the scan latency which now reaches 150 µs per active input[3] which leads us to a total scanning latency of 2.5 ms with all the 16 inputs activated.

Table 1 : Comparison of quoted Analog to MIDI Converters (adapted from [3])

Interface	*AtoMIC Pro*	*EoBody*	*Digitizer*
Manufacturer	Ircam	Ircam – EoWave	Infusion Systems
Platform	Any	Any	Any
Max SR [Hz]	1000 / Active inputs	900 / Active inputs	244 / 240 (12/7 bits)
Analog IN	32	16 + 3 pot.	32
Digital IN	8	4 switches	-
Input Res.	10/7 bits	10/7 bits	12/7 bits
Outputs	switches + 4 MIDI	MIDI + merger	8 switches + MIDI
Size (HWD) [mm]	38x165x225	30x160x115	34x121x94

1 http://www.forumnet.ircam.fr/
http://www.eowave.com/human/eobody.html
2 The 4 line LCD represents about one fourth of the manufacturing cost.
3 See the comparison chart (Table 1)

2. TECHNIQUES & IMPROVEMENTS

2.1 Sensor bandwidth

Designing a "light" version of a device does not mean removing all interesting aspects of the regular version. Since MIDI is (still) deadly slow compared to the amount of data we want to export, we kept our implementation of the noise gate algorithm and the sub-sampling process [2]. Thus, priority can be given to sensor, and gesture (or sensor) noise can be removed, with significant gain on the dataflow bandwidth. This feature also allows the user to adjust a pertinence criteria on each sensor, reducing the post-processing of the digitized sensor.

2.2 Running status

Strongly concerned by the temporal resolution of gestural acquisition, we have implemented this very well known technique, introduced into the MIDI standard a few years after the birth of the protocol[4] : it consists in suppressing the status byte of a MIDI message[5] if it hasn't changed from the previous message. Since *EoBody* is mainly designed to export continuous values thought *Control Change* MIDI message, we can easily assume that most sensors might be configured to talk on the same MIDI channel[6]. The status byte remains the same for the scan of a whole array of sensors and can thus be omitted.

Our *running status* algorithm actually sends/repeats the status byte every 32 sent messages to avoid a major drop when the status does not change. Improvement can easily be calculated, on 32 messages, as the following table shows :

Table 2 : Running status improvement

	No Running Status	Running Status
1st message	3 bytes	3 bytes
messages 2 to 32	3 bytes	2 bytes
	Total = 96	Total = 67

The bandwidth thus increases of :

$$100 \times (1 - \frac{67}{96}) = 31\%$$

2.3 High resolution vs. standard MIDI

Again, considering temporal resolution aspects, it was obviously not a good idea to send high resolution data (10 bit wide in our case) through System Exclusive MIDI messages, the involved number of bytes being a disaster for the transmission time of the digitized info. So, we rather preferred single or combination of standard MIDI messages like *control change* (7 LSBs + 3 MSBs) or *pitch bend* (the 10 bits being mapped on the 10 MSBs of the message).

2.4 Sensors' range

To minimize the difference of sensors range, the voltage reference of the A/D converter is still accessible, via a trimmer. Despite the fact that signal windowing is unavailable, signal zooming can be achieved using the 10 bit resolution and then scaling the value elsewhere.

2.5 Connectors and plugs

A known problem with *AtoMIC Pro* was the "complex" connector used to connect sensors (i.e. a *Sub-D* plug). We still think that wiring is extremely important, and that's why *EoBody* still uses *Sub-D* connectors, with locks. However, to keep plugging sensors easy, we include in the *EoBody* package 2 splitter cables that distribute the 15 pin *Sub-D* male plug to 8 regular *jack* plugs. Thus, the user can choose between easy (but quite bad quality, we all know how good a *jack* plug is) connection for experimenting, and very secure wiring for a performance or an installation. We also underline that exporting the *jack* plugs through a splitter cable allowed us to reduce the size of the housing box[7]. Moreover, it is easier to repair a splitter cable with regular connectors[8], than having to open the whole box for repairs.

The danger of using jack connections would be sensors hot-plugging. People who have experimented expression with pedals know the problem well : inserting the male jack plug with the power on creates a little short-circuit that sometimes makes the fuse burn. Most systems actually have a current- limiting circuit to avoid this, but it is impossible to simply implement in our device, since we do not know how much current the connected "sensor" will sink. So, we have chosen to use a *Polyswitch*™ fuse : it allows transitory peaks of current, but cuts off the power supply if too much current is drawn[9] : there is no risk of flashing fuse after fuse just by connecting a sensor.

3. REFERENCES

[1] Fléty, E., "Interactive devices for gestural acquisition in the musical live performance context." *Proceeding of SCI,* SCI 2001 – Orlando, Florida – USA, pp 535-540.

[2] Fléty, E., "AtoMIC Pro : a Multiple Sensor Acquisition Device." *Proceedings of NIME,* NIME 2002 – Dublin, MIT Europe – Ireland, pp. 96-101.

[3] Wanderley, M., "Performer-Instrument interaction : applications to gestural control of sound synthesis". *Doctor's Degree Thesis*, Université Paris 6 – France, pp 41-42.

4 www.midi.org

5 Most MIDI messages are composed of three bytes, one being the status, i.e. the command to be executed (note generation, parameter modification etc.), and the two others being the data (7 bit long) bytes of the message.

6 Even if the MIDI channel can be individually set, input per input.

7 Made of steel for a better hardness.

8 *Sub-D* and *jack* plug can be found anywhere and are really cheap.

9 The resistance of the *Polyswitch* increases with temperature. It rearms itself when the temperature decreases (thermal fuse).