

The 1997 Mathews Radio-Baton & Improvisation Modes

From the *Proceedings* of the 1997 International Computer Music Conference – Thessaloniki Greece

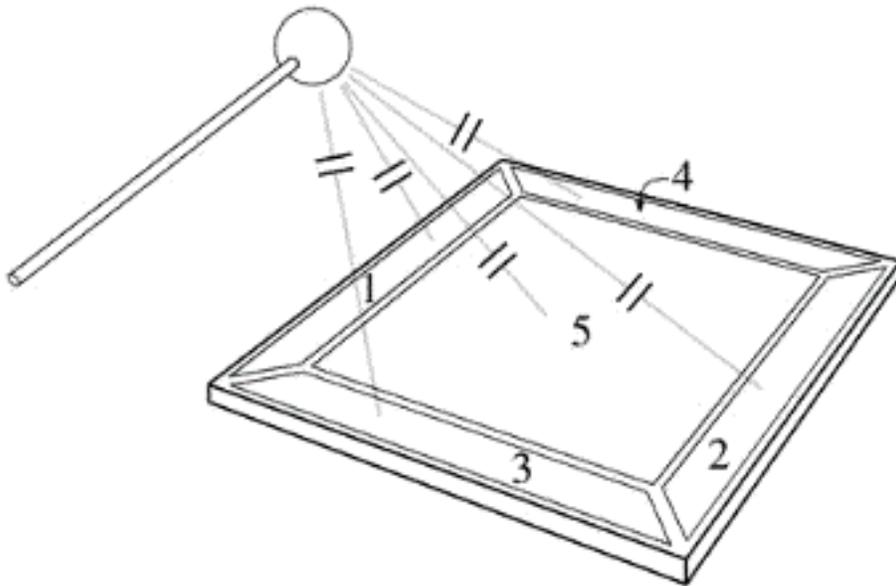
Richard Boulanger & Max Mathews

rboulanger@berklee.edu & mvm@ccrma.stanford.edu

Berklee College of Music & Stanford University

Introduction

The Radio-Baton is a controller for live computer music performances. It tracks the motions, in three dimensional space, of the ends of two Batons which are held in the hands of a performer. The X, Y and Z trajectories of each Baton are used to control the performance. The Radio-Baton is a MIDI instrument in the sense that it has MIDI input, output, and thru connectors. All electrical communication with the Baton is done over MIDI cables using standard MIDI conventions. The Baton was designed to work with MIDI synthesizers and MIDI-based sequencing and programming software.



BLOCK DIAGRAM OF RADIO-BATON

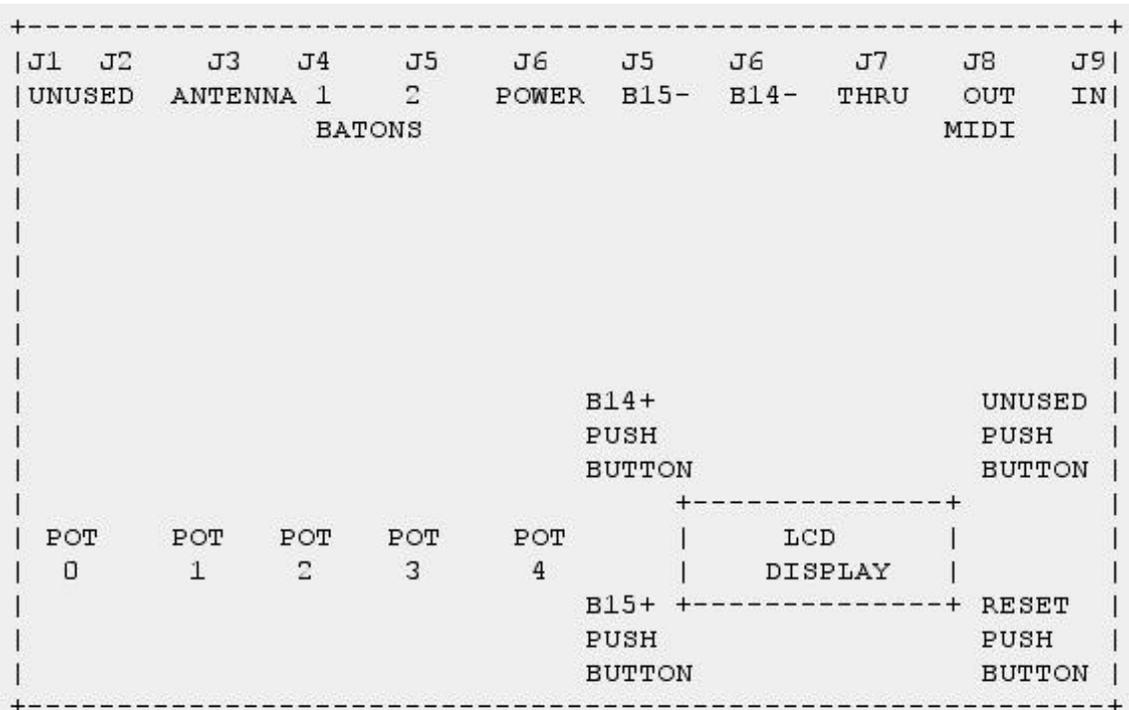
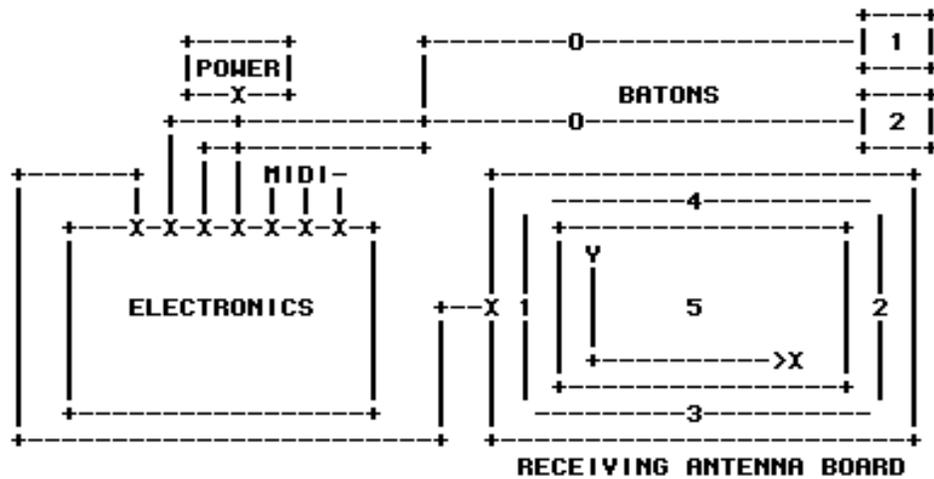


FIGURE 1: SKETCH OF BATON ELECTRONICS BOX--TOP VIEW

How the Radio-Baton Works

The Radio-Baton uses a simple technique to determine the XYZ coordinates of the batons. At the end of each baton is a small radio transmitting antenna. On the receiving antenna surface are 5 receiving antennas as sketched on the figure--four long thin antennas arranged along the four edges of the board and one large antenna covering the entire center area of the board. The closer a baton is to a given receiver, the stronger the signal at that receiver. By comparing the signal strengths at the #1 and #2 antennas, the computer in the electronics box can determine the X position of the baton. Comparing

the #3 and #4 strengths gives the Y position. The #5 strength gives the height above the board or Z position. The two batons operate at different frequencies and thus can be independently tracked.

A low frequency of about 50kHz is used for the radio signals. It is appropriate to describe the Radio-Baton as a capacitance sensor. The radio-frequency signal is really used as a method of measuring the capacitance between a transmitting antenna electrode and a receiving antenna electrode. The measurement technique is simple and robust. The relationship between antenna signal strengths and XYZ coordinates is complex but the Radio-Baton computer is good at making the needed conversions. A table lookup procedure is used to speed the calculation.

The Radio-Baton can compute the positions of the batons every 4 milliseconds. The accuracy of measurements is about 1 part in 100. The X and Y information is linearized and mapped onto the standard MIDI range of 0-127. The Z information is not linearized. The Z data decreases from about 100 when a Baton is on the antenna surface to about when the Baton is about 3 feet above the antenna surface.

Triggers

In addition to providing XYZ data, one of the most important functions of the Radio-Baton is to send triggers over MIDI. A trigger can be generated when either Baton touches an invisible plane which can be positioned at various heights above the antenna board. This plane is called the HIT-LEVEL. In order to avoid DOUBLE-TRIGGERING, a second plane is also positioned slightly above the HIT plane. This plane is called the SET-LEVEL. A Baton must be raised above the SET-LEVEL before a second trigger can be generated. This method for generating triggers and avoiding double triggers using a HIT-LEVEL and a SET-LEVEL works well for most musical purposes.

Principal Programs

The processor in the Radio-Baton can execute a number of different programs but the two main modes are the IMPROV mode and the CONDUCTOR mode. The IMPROV mode is used for executing improvisations and algorithmic programs and the CONDUCTOR mode is used to play traditional scores.

The IMPROV Mode

In the IMPROV mode, the Radio-Baton is a simple controller which sends XYZ position information and triggers to a control computer. The control computer interprets this information musically according to whatever program is being run in the control computer.

The CONDUCTOR Mode

When the Conductor Program is running in the Radio-Baton, a control computer loads a score into the processor in the Radio-Baton. The Baton plays this score according to the conductor-gestures made by the performer. In response, the Radio-Baton sends MIDI commands from its MIDI out connector to a synthesizer which plays the notes.

Typical Commands

When the Radio-Baton is powered up or reset, it's processor enters a polling loop looking for commands received at it's MIDI input terminal. Most commands are encoded as MIDI system-exclusive messages which contain only an operation code as a single information character. For most commands the Radio-Baton sends a confirmation message from its MIDI output to the control computer.

TEST

| | | | |
|------------------|-----|---|-----|
| Received Message | 240 | 1 | 247 |
| Response | 240 | 1 | 247 |

This command is used to test whether the Baton is working and whether the MIDI is correctly patched.

UART

| | | | |
|------------------|-----|----|-----|
| Received Message | 240 | 33 | 247 |
| Response | 240 | 33 | 247 |

This command puts the Baton into the UART mode in which all MIDI characters received are simply transmitted from the MIDI out jack. The Radio-Baton can only be gotten out of the UART mode by pushing the reset push button or button B15+ on the Baton box. When B15+ is pushed the Baton transmits:

240 44 247

to confirm that it has left the UART mode.

IMPROV

| | | | |
|------------------|-----|----|-----|
| Received Message | 240 | 46 | 247 |
| Response | 240 | 46 | 247 |

This command puts the Baton into the IMPROV mode where it functions as a simple controller.

PLAY

| | | | |
|------------------|------|---|-----|
| Received Message | 240 | 4 | 247 |
| Response | none | | |

This command puts the Baton into the CONDUCTOR mode where it will play a specified score under the expressive control of baton gestures. This mode has already been described in other publications and will not be further discussed here.

The IMPROV Protocol

When in the IMPROV mode, the Baton functions as a simple controller and sends triggers and other controller information from its MIDI out jack. Usually a computer receives this information. The task of making musical sense from the controller information falls entirely on the computer, its program, and its programmer. The conversation between the Baton processor and the computer is two-way. For example, in order to reduce the load on the midi cable, the Baton only sends the XYZ position data for the Batons when this information is requested by the computer.

Although a number of examples are provided with the system, the user is expected to program their own improvisation algorithms. To assist in this task, a support program and template are provided. Starting with the template, the user merely needs to fill in the blank functions. In addition to those functions provided in the template, the user can

access various functions for sending MIDI commands to a synthesizer and can read a millisecond clock which is provided automatically by the support program.

The IMPROV Programming Template in the C Language

```
#include <stdio.h>
#include <conio.h>
#include "drivers.h"
#include "define.h"
#include "impr.h"
/*-----initialization algorithms-----*/
initialization(){
/*-----main loop algorithms -----*/
mainloopalgorithms(){
/*-----triggered algorithms-----*/
stick1trig(){
stick2trig(){
b14plustrig(){
b15plustrig(){
b14minusup(){};
b14minusdown(){};
b15minusup(){};
b15minusdown(){};
noteplay(){
/*-----poll response algorithms-----*/
stick1pollresponce(){
stick2pollresponce(){
potspollresponce(){
/*----additional functions and variables provided by the support program---*/
m_send(a)-----send midi character
m_poll()-----receive midi character
m_play(chan,keyno,keyvel)-----play note
m_cont(chan,controller,value)---control change
m_pc(chan,program)-----program change
t_time-----clock in milliseconds
```

A Simple Improvisation Program:

Play Notes with Baton 1 and Sustain them with the Pedal

```
#include <stdio.h>
#include <conio.h>
#include "drivers.h"
#include "define.h"
#include "impr.h"
short keyno;
/*-----initialization algorithms-----*/
initialization(){
    printf("Baton1 Trigger plays note on MIDI channel#1: X Axis = Note\n");
    m_pc2(0,25); printf("Initial Patch is #25 on MIDI Channel #1\n");
}
/*-----triggered algorithms-----*/
stick1trig(){
    m_play2(0,keyno,0);
    keyno=44+xlt/3;
    m_play2(0,keyno,127);
    printf("keyno= %d\n",keyno);
}
b14minusup(){m_cont2(0,64,0);    printf("pedal b14- up\n"); }
b14minusdown(){m_cont2(0,64,127);printf("pedal b14- down\n");}
```

Boulanger – Radio Sonata #1

Mvt 1 Opening Scored in the Mathews Conductor Program

```
I K0# v30
q0 h11 v127 c7 q0 h12 v110 c7 q1 h11 x2 -c10
q2 h12 x2 c10 q3 h11 y2 c11 q4 h12 y2 c11
2 o84 h11 ky1 t88
3 o36 h11 ky1
4 o36 h12 ky1 t81
16 h16 t2
*0*
0 /....
*1*
0 /.....
3 !FFGab
4 !#F!#G#a
*2*
0 /..... .. /.....
2 f^e.....r.. #Gg
*3*
0 ..... .. /..... ..
2 .....r.. #Aa.....r..
*4*
0 /..... .. /.....
2 o60Cb.....r.. !D#C
*5*
0 ..... ..
2 .....r..
3 .....r..
4 .....r..
*7*
0 /.... .. /.....
4 F.... .. #a
```

Conducting a MIDIFILE with the Mathews Conductor Program

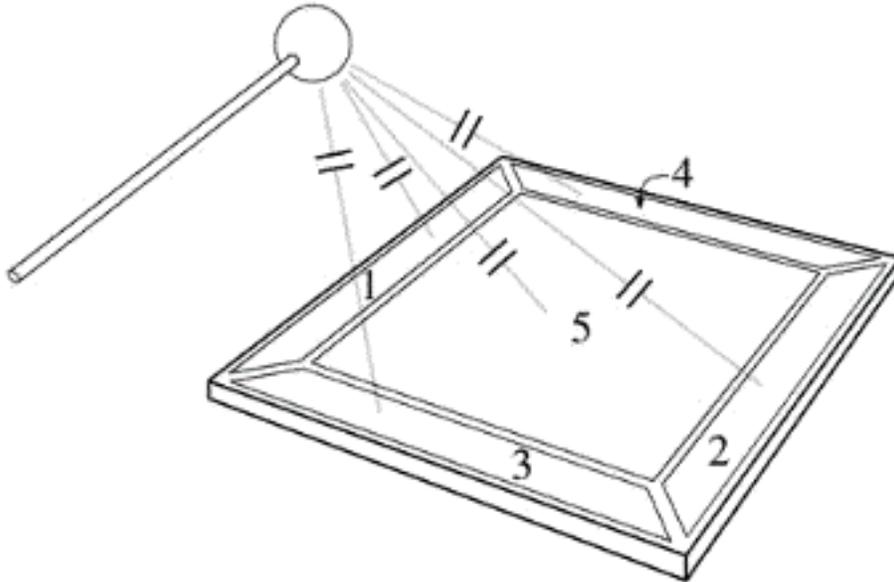
```
I *Initialize the Baton*
v100 *Set initial Tempo*
Z110 *Reduce Velocity from MIDI File*
q0 h7 v100 c7 *On MIDI Channels 7-10
q0 h8 v120 c7 *Assign Controller 7 Constants with
Patchcords*
q0 h9 v127 c7
q0 h10 v110 c7
1 h7 kv100 t66 q2 h7 y2 c11 *Send Programs & Map Controllers to Axis*
2 h8 kv100 t114 q3 h8 y2 c11
3 h9 kv100 t12 q4 h9 y2 c11
4 h10 kv100 t1 q5 h10 x2 c11
16 h16 t0 *Send Program Change #1 on Channel 16*
*1*
0
M *Jump to the MIDI File*
0
```

Max V. Mathews was born in Columbus, Nebraska, on November 13, 1926. He studied electrical engineering at the California Institute of Technology and the Massachusetts Institute of Technology receiving a Sc.D. in 1954. He worked in acoustic research at AT&T Bell Laboratories from 1955 to 1987 where he wrote programs (Music I through Music V and Groove) for playing music on digital computers. He has been called the father of computer music. In 1987 he joined the Stanford University Music

Department in the Center for Computer Research in Music and Acoustics (CCRMA) as Professor of Music (Research). Here he developed a new pickup for electron violins and real-time computer systems for music performance. His latest system is the Radio Baton and CONDUCTOR & IMPROV Programs.

Richard Boulanger (b.1956) is a composer and performer of interactive computer music. He did his Ph.D. at the University of California in San Diego and worked there with Roger Reynolds and Dick Moore. For more than a decade now, he has collaborated extensively with Max Mathews on the Radio Baton and Barry Vercoe on Csound. Currently, Boulanger is a Professor of Music Synthesis at the Berklee College of Music in Boston.

The Mathews Radio-Baton - 1997



The Mathews Radio-Baton is a controller for musical performances. It consists of two batons, a receiving antenna board, and an electronics box. This device tracks the motion of the two batons as they are moved by a musician in three-dimensional space above the receiving antenna.

The electronics box contains a processor which analyses the gestures made by the musician and uses this information to control the music being played by a synthesizer. The electronics contains a standard set of midi in, out, and thru connectors. All communication between the Radio-Baton and the synthesizer is done via MIDI. The Radio-Baton can interact with any computer that can send and receive MIDI.

The Radio-Baton operates in two principal modes – the *Conductor Mode* and the *Improv Mode*. In the **Conductor Mode**, the Radio-Baton simulates an orchestra. The musician loads a score of the piece to be played from a Computer into the Radio-Baton. She then uses one baton to beat time and thus to control the tempo of the performance and the other baton to control the dynamics, balance, and timbres of the voices.

In the **Improv Mode**, the Radio-Baton serves as a simple controller which sends triggers and the x,y,z positions of the two batons to a computer. The musician must write a program in the computer to interpret this information and to send MIDI commands to play music on a synthesizer. The Improv Mode can run with any computer (for example, a MacIntosh running the *Max* program). It is a more general mode than the Conductor Mode, but it requires that the musician write the complete program to make a musical interpretation of her gestures.

A Radio-Baton can be obtained from the MARMAX Company for \$1200 plus shipping costs and any applicable sales tax. Included in the price are:

- The Radio-Baton electronics box, antenna board, power supply, two batons, and interconnecting cables.

- A copy of the Conductor Program (*Bat.exe*) which runs on a PC computer under either the DOS or the Windows 95 operating system and the source code (*Bat.c*) for the Bat program. The source code is written in the Microsoft C language.
- An instruction manual and a floppy disk of sample scores for the Conductor Mode.

In contrast to the Improv Mode, which can be used with any computer that speaks MIDI, the Conductor program runs **ONLY** on a PC computer and requires a Midiator ms124 MIDI interface. The ms124 connects to an rs232 serial port on the computer and can be obtained from Key Electronics 7515 Chapel Ave., Fort Worth, TX USA 76116

To obtain more information about the Radio-Baton contact MARMAX and ask for a copy of the instruction manual or contact us via email at m.v.mathews@worldnet.att.net

MARMAX 223 PRECITA AVE. SAN FRANCISCO CA 94110 415-821-4661