Musical MIDI Accompaniment

Tutorial

Bob van der Poel
Wynndel, BC, Canada

bob@mellowood.ca

November 7, 2010

Table Of Contents

Chapter 1
Introduction

This document is provided as an aid for the new *MᴍА* user to get started in a easy, gentle manner. It is a supplement to the main reference manual.

MᴍА is a complex and powerful program. We don't intend to show each and every possible command in this tutorial; we'll be quite happy if we show enough of the basics to get you started! And, please, read the reference manual!

We hope to cover the following topics:

♫ Installing *MᴍА* on your system,

♫ A step-by-step example showing how to create a simple backing track,

♫ An example showing how to create a library file.

The examples may include songs which are probably **not** in the public domain. Certainly "Happy Birthday" should be public domain, but isn't. User's should note[1] that the copyrighted songs in this document are examples only. It is probably not legal for you to copy it or play them. If this document is violating copyright by including any of the sheet music used in the examples, please let us know and it will be removed.

The example files in this text are included in the distribution in the `egs/tutorial` directory.

This is a **draft** document. Feedback is solicited!

---

[1]Bad pun intended!

Installation

For now, you'll have to rely on the reference manual and the READMEs for this.

Chapter 3

A Simple Example

3.1 Fella Bird, Basic Edition

Example 3.1: Sheet Music for a Popular Song

This example piece of music is shown in a "fake book" or "lead sheet" style. It shows the melody notes, lyrics and chords.

As a first try at making a MIDI accompaniment file, we have created the following file:

```
// Sample tutorial file
// Fella Bird, try 1

Tempo 120
Groove Rhumba
```

```
1       F
2       F
3       C7
4       F
5       F
6       F
7       C7
8       F
9       Gm
10      F
11      C7
12      F
13      Gm
14      F
15      C7
16      F  /  /  z!
```

After the comments in the file, the first line to note is:

`Tempo 120`

This sets the tempo, or speed, of the piece to 120 beats per minute. You may have to guess the tempo—very few pieces (especially in "fake book" style) will include a metronome setting. For slow pieces you might want to start with a tempo setting of "80"; for faster pieces, like swings, try around "150"; polkas and marches, which are really in $\frac{2}{2}$ time, may require fast sounding tempos like "250". Feel free to modify the tempo setting in the example files—you're not going to break anything.

Since we are using the standard library which was shipped with *MrA*, we can select one of the predefined grooves:

`Groove Rhumba`

Note that this command also sets up the "time signature" to $\frac{4}{4}$. Not quite the same as the "cut" time indicated, but close enough. And speaking of "close enough", we're using a rhumba rhythm instead of the merengue indicated in the score.

Finally, we have the chord information. The first bar (in the sheet music) indicates an "F" chord. So that's what we tell *MrA* to use:

`1 F`

We continue in a similar manner for the rest of the song.

We've simplifed the song a bit and show only one chord per bar. *MrA* automatically fills out the bars so that each has four chords. We could have entered the first bar as:

`2 F F F F`

or, more simply:

```
2 F / / /
```

and had the same result. But, who wants to do all that typing?

In the final bar, the last beat is a rest.

The line:

```
16 F / / z!
```

causes an "F" chord to be used on the first three beats, the fourth beat is completely silent due the to the "z!". Note the difference between a "z" and "z!"—the first silences all but the drum track, the second silences everything. See the reference manual for more details on "z".

Now, let's create our first MIDI file! If you are using the standard distribution, you should find a file "fella1.mma" in the tutorial directory. Change to this directory and type the command:

```
mma fella1
```

Your computer should run the python script called "mma" and process the file "fella1.mma". The MIDI file "fella1.mid" should be created. If this doesn't happen, please check your python installation, and make sure that you have installed "mma" where your shell can find it. If you are completely stuck, drop me an email.

Now, use your favorite MIDI file player to play the song.

How'd that sound? A piano doing some chords, a jazzy bass line, and a bit of drumming? Wonderful!

3.2   Fella Bird, Improved Edition

If you play the MIDI created in the previous section you'll find a number of things lacking. Certainly:

♫ It is hard to sing to because there is no aural indication of when to start,

♫ The whole piece is the same volume,

♫ It's a pretty short song, so let's play it twice.

The following example has a number of improvements:

```
// Sample tutorial file
// Fella Bird, try 2

Tempo 120
Groove Metronome2-4

z * 2

Groove Rhumba
Repeat
```

```
Volume mp
Cresc mf 4

1       F
2       F
3       C7
4       F
5       F
6       F
7       C7
8       F
9       Gm
10      F
11      C7


Decresc p 4

12      F
13      Gm
14      F
15      C7


RepeatEnding

16      F  /  / z


RepeatEnd

17      F  /  / z!
cut -1
```

Try running *MMA* on this file and listen to it. Oh my, much better.

So, let's look at the changes.

```
Groove Metronome2-4
```

This sets the current "groove" to a metronome. To find this groove *MMA* will automatically process the library file "metronome".

The metronome groove consists of wood-blocks being struck in a $\frac{4}{4}$ pattern. In order to "sound" the metronome, we need to create an empty bar:

```
z * 2
```

Mind you, we could have specified a chord here and gotten the same result (the groove has no definitions for any instruments other than the wood-block, so nothing else will sound). But, it seems to be more clear to use a "z".

Okay, we lied. We didn't create an empty bar for the metronome. If you're sharp you'll notice that the "z" is followed by "* 2". This means to play this bar two times. So, you get 2 bars of metronome. Cool.

We've not made any changes to the chords, but a few little changes do help this simple piece. Remember the dull, single volume? Well, have a look at the "volume" directives we've included in this version. We start the song off at a moderate volume, and increase it in several steps. And, at the end we use a "decresc" to reduce the volume to "quiet" over the final 4 bars.

Please refer to the *M*♭*A* Reference Manual for salient details on the volume commands.

One minor change we did make in the chords is in bar 16. Note the change between this and bar 17. In the first we have a "z", the second a "z!". The difference between the two is that we want the precussion tracks to continue though the ending, but not on the final beat.

Finally, to ensure a complete cutoff in the final bar we have added a "cut" command. We're sure you'll find these minor changes a great improvement.

A More Complex Example

Fun time. We're going to take an old (slow) spiritual and surprise some people! We'll also learn about repeats, play with volumes, and do some interesting tempo things.

4.1 Deep River—A Swinging Spiritual

Example 4.1: A Public Domain Song

Rather than list the song separately, we've inter spaced the listing with textual comments. If you want to see the listing, it is in the **egs** directory.

We start out with a comment block.

```
// Deep River
```

Start off by selecting the 4 beat metronome.

```
Groove metronome2-4
```

We want to surprise our listeners a bit, so we start off with a "normal" tempo. The single "z" produces a 4 beat introduction.

```
Tempo 90
z
Groove FolkArticulated
```

As an introduction we play some nice, gentle chords for 4 bars.

```
1 F
2 Dm
3 Gm
4 C7
```

Surprise time. We are going to pump the tempo up to 140 BPM and select the Swing2 Groove.

```
Tempo 140
Groove Swing2
```

If we just did a switch of rhythm and tempo it'd be quite "interesting". We need some kind of a transition. Hmmm, what if we have two beats (not four) of swing drums? We could create a 2 beat measure, but it's quite cool to give the drummer a real workout. So, here we double the time and put in a single "z" bar.

```
Tempo *2
z
```

Now, we restore the tempo to the original 140.

```
Tempo *.5
```

Now, just like the sheet music we insert a repeat start.

```
Repeat
```

Set the volume, and advise *MA* that we'd like it get softer over the next 8 bars. Here we use the optional "start volume" setting in the DECRESC command. If we don't the current volume will be used, and right after a repeat that is not a good idea.

```
Decresc mf mp 8
```

This is straightforward chording.

```
5 F
6 Gm
7 Dm
8 Am C
9 F
10 Gm / Dm
```

```
11 F / C7
12 F
13 Dm
14 Am
```

Increase the volume over the next 4 bars, and more chording.

```
Cresc ff 4
```

```
15 Bb
16 Am / / C7
17 F
18 Gm
19 F / C7
20 Am / Dm C7
21 F
22 Gm
```

This is the first/second ending. Note how we've put a "2" at the end of the next line … this forces 2 repeats.

```
RepeatEnding 2
```

```
23 / / / C7
24 F
```

For the second and third time we play this we want a more interesting accompaniment. So, we select "Swing2Plus" which adds a clarinet doing its thing.

```
Groove Swing2Plus
```

This ends the repeated section.

```
RepeatEnd
```

For our 2 bar ending we select the "Swing2End" groove. Next, we ritard our tempo over the next 2 bars.

```
Groove Swing2End
Tempo -40 2
```

```
1 / / / C7
```

The "Swing2End" groove has a neat little saxophone scale. But only on the third and fourth bars. It is a four bar sequence and the first two give the sax player time to rest. The **Seq 3** forces the sequence to use the bar with the solo. We also set the solo to a louder volume.

```
Scale Volume ff
Seq 3
1 F
Fermata -1 1 200
```

To finish off the example, we add a bit of time to the last note with a **fermata** command.

Compile this example and play it. Follow along on both the sheet music and the *MA* file so you understand what's happening.

4.2   Future Directions

Well, that's two examples.

Certainly, the ideas here only scratch the surface of what *MA* can do. We, as developers, can only hope that we've included enough commands and options for you to create wonderful music. And hope that if we haven't, that you let us know.

May we suggest that you list and examine some of the sample songs—then, it's all up to you. Have fun!

Library File Creation

You should know that the GROOVE definition engine is a very powerful tool that has almost unlimited possibilities. It is impossible to explain all those possibilities in a short tutorial. The flexibility of *MℓA* allows you to do the same things in different ways; the example you'll see here is just one way to create a GROOVE. We have tried to keep it simple and organized it in different sections so that you understand how the GROOVE was built.[1]

The day that you decide to create your own "proper" GROOVEs you'll have to read the manual in detail!

5.1 Let's Create Our First GROOVE

The GROOVE we will create will be a simple $\frac{4}{4}$ rock GROOVE.

The first thing we have to do set the timing for $\frac{4}{4}$ and initialize some variables (those are explained in detail in the *MℓA* manual)
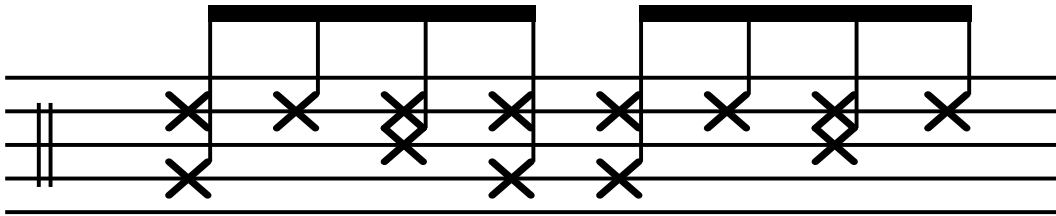
```
SeqClear
SeqSize 1
Time 4
Timesig 4 4
```

Now let start to create our patterns.

5.1.1 The Drummer

For the drummer we will use the following pattern:

---

[1]This chapter was written by Rony Steelandt of Kara Music Production with minor spelling and grammar fixes by Bob van der Poel.

As we can see this is a pretty simple pattern.

We have:

♫ A kick drum on beats 1, 2 and, 3

♫ A snare on beat 2 and 4

♫ A closed highhat every eight note

Before we actually define the drum-instruments we will define the patterns that we will use:

```
Begin Drum Define
    D1 1 8 90 ; 2.5 8 90 ; 4 8 90
```

So what have we done?

Well, we have just translated what we have seen on the score. We have created a pattern D1 with a 8th note on beats 1, 2.5 and 4 and a velocity of 90.[2]

Using the same logic we can create the pattern for the snare drum.

```
S1 2 8 90 ; 4 8 90
```

For the closed HiHat pattern we will use a little trick to avoid having to enter 8 notes

```
CH1 1 8 90
C1 CH1 * 8
```

As you can see we just define one note on the first beat and multiplied it by 8, which will result in eight notes.

And we close the definition by adding

```
End
```

Now that we have the patterns for our drum, we will create the different instruments.[3]

─────────────────────────────

[2]See *MMA* Reference Manual section 4.1.8 foo here32 for more details.
[3]See *MMA* Reference Manual section 4.1.9 for more details.

```
Begin Drum-Kick
  Tone KickDrum1
  Sequence D1
End

Begin Drum-Snare
  Tone SnareDrum1
  Sequence S1
End

Begin Drum-HH
  Tone ClosedHiHat
  Sequence C1
End
```
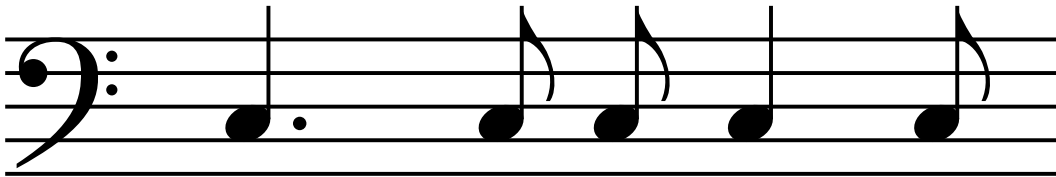
5.1.2   The Bass Player

Now that we have a drummer, let's have a look at the bass player for our GROOVE.

Those are the notes the bass player plays on a C chord. He builds a nice riff using only the root of the chord.

This is pretty simple to translate in *MtA* syntax, using what we have learned to create the drum pattern.

So we go up in our text file to the point where we created the patterns for the drum (the line after End) and we add:

```
Begin Bass Define
  B1 1 4+8 1 90 ; 2.5 8 1 90 ; 3 8 1 90 ; 3.5 4 1 90 ; 4.5 8 1 90
End
```

You will already recognize the timing of the riff, the duration and the velocity of the notes. The only item that is different for a bass player is that we add in the pitch definition for the note to be played.

If we look at the first note definition we see that beat 1 is a note with duration 4+8 (dotted quarter),[4] the note to play is the first (root) of the chord and the volume or velocity of the note is 90.[5]

[4]When specifying the duration of a note the following are equal: "4+8", "4." or "8+8+8".

[5]See *MtA* Reference Manual section 4.1.1 for more detail.

Now the only thing left is to add the bass player at the end of the file, the same way we did with the drums.

```
Begin Bass-Simple
  Voice AcousticBass⁶
  Sequence B1
End
```

5.1.3   The Piano

As a last example, we will add a piano player with a simple riff to our GROOVE.
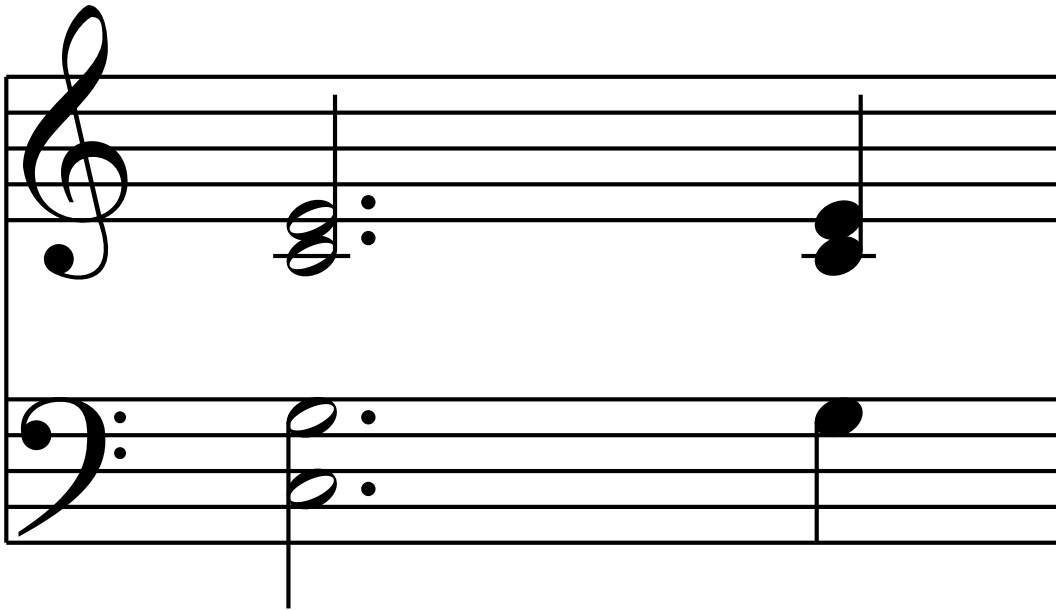
Creating piano patterns is a bit different from other players as they (most of the time) play with both hands. As we can see on the above score (using only a C chord), our piano player will play the second inversion of the chord with his right hand as a dotted halfs and a fourth note and with the left hand he plays the root on the start of the bar also as a dotted half note.

There are a couple of ways in *MₘA* to define this, what we explain here is just one of the possibilities.

We will define both "hands" of the piano player separately.[7]

Let's start with the easiest, the left hand. Actually if you think a bit about it, there is not a big difference between a bass player and the left hand of a piano player. Following this logic, we will simply define a new pattern and attach it to a new bass player called Bass-LeftHandPiano.

We already know how to do this.

---

[6]For drums we use TONE for other instruments we use VOICE.

[7]As many things in *MₘA* there are different ways to define that kind of pattern. If you're curious, look at the Reference Manual section 14.4, page 91.

In the pattern section of our GROOVE we will add:

```
L1 1 2+4 1 90
```

Remember: First beat, half+fourth note, root of the chord, velocity 90.

And we will create a new bass player

```
Begin Bass-LeftHandPiano
  Voice Piano1
  Sequence L1
  Octave 3 // This a new command, but simple to understand
End
```

Now the right hand. What the right hand of the piano player does is more the playing of chords than playing simple notes. That's exactly what we will tell *MMA*: play chords.

On the example score we see that he plays the chord in his first inversion form, but for the sake of simplicity of this "Getting started" document we are just ignoring this and we will define the standard form of the chord.[8]

Again, we go up in the definition section of our GROOVE and under the bass patterns we will enter the chord definition

```
Begin Chord Define
  C1 1 2+4 80 ; 4 4 80 [9]
End
```

As you can see the chord definition structure is a bit different from what we have done until now.

Chord definitions are done as follows: Name (C1), Beat (1), and the note volume or velocity. It is possible to specify different velocities for each note in a chord (see page 4.1.2 in the Reference Manual) and even disable certain notes or limit the range of chord.

Finally we need our piano right hand player

```
Begin Chord-RightHandPiano
  Voice Piano1
  Sequence C1
End
```

5.1.4   Putting It All Together

We have just created a trio to play Drums, Bass and Piano for us.

We have to give a name to the created GROOVE, so that we can use it in a song:

---

[8]If you are curious about how to define inverted chords, see the Reference Manual section 14.5, page 92.

[9]This is not exactly right, it will play the 7th on a C7 chord too, but this is beyond the scope of this tutorial.

```
DefGroove Myrock1
```

is added at the end of the file.

The file that we created should look like this[10]

```
SeqClear
  SeqSize 1
  Timesig 4 4

Begin Drum Define
  D1 1 8 90 ; 2.5 8 90 ; 4 8 90
  S1 2 8 90 ; 4 8 90
  CH1 1 8 90
  C1 CH1 * 8
End

Begin Bass Define
  B1 1 4+8 1 90 ; 2.5 8 1 90 ; 3 8 1 90 ; 3.5 4 1 90 ; 4.5 8 1 90
  L1 1 2+4 1 90
End

Begin Chord Define
  C1 1 2+4 80; 4 4 80
End

Begin Drum-Kick
  Tone KickDrum1
  Sequence D1
End

Begin Drum-Snare
  Tone SnareDrum1
  Sequence S1
End

Begin Drum-HH
  Tone ClosedHiHat
  Sequence C1
End

Begin Bass-Simple
  Voice AcousticBass
  Sequence B1
```

---

[10]If you are really lazy and don't want to type the text, you download the file here: `http://www.kara-moon.com/MMA/tut1.mma`.

```
End

Begin Bass-LeftHandPiano
  Voice Piano1
  Sequence L1
  Octave 3 // This a new command, but simple to understand
End

Begin Chord-RightHandPiano
  Voice Piano1
  Sequence C1
End

DefGroove Myrock1
```

5.2 How To Use Our First GROOVE

Save the text file we just created in the lib-directory of *MtA* under the name myrock.mma.

First thing we have to do is tell *MtA* to update its database so that our new GROOVE can be found.

Go to a command prompt and enter: $ mma -g

Now start a new textfile to create a song and start the file with:

```
Groove Myrock1
Tempo 120

// Enter chords here
```

Save the file as Mysong.mma and compile it as explained in the first part of this tutorial.

5.3 Where Do We Go From Here

As you probably noticed, the style that we have created here sounds very simple and lacks variation ...an intro, a break and an ending .... Don't panic, all those things are possible but beyond the scope of this small "Getting Started".

Now before you do anything else: **READ THE MANUAL!**

This is the best advice you can get. The author of this program has put a lot of effort in the development process so that the software can be as powerful and flexible as possible. Years of development went into this product, so don't think you will learn it in a day.

Also read the README files that are delivered with the distribution. You can find valuable information in these.[11]

Have fun with *MiA*!

---

[11]Amongst others, some very good information about fake books.