**Specification for the Music Macro Language to MIDI Project**

**Overview**

The music macro language (MML) is a music description language that has been used in one form or another since 1978. There has never been any official specification, so each version and implementation varies slightly, and over the years the language has improved and evolved. My project is to write a program that can convert a MML text file into a MIDI file.

**Programming Language Choice and Aids Used**

I have decided that the C programming language is the most appropriate choice for me to do this project. This is because I am most comfortable with writing C than any other language, and it has time-tested compiler compiler software available, which is particularly useful for the challenges presented by the project.

I will use two compiler compilers to help me write the program: the first is lex, a lexical analyser generator that takes a language specifying file and outputs C code that can scan it. The second is yacc – this is a parser generator that takes its own language specifying file and, again, generates C code that can read it. Lex and yacc are very often used together and perform best in this configuration.

I’ve decided to not use any libraries for this project aside from the standard C libraries, as I feel like this will make the completion of the project a more valuable and informative experience. One effect of not using libraries is that I will have to write my own code to write a MIDI file.

To compile my software I will use the GNU Compiler Collection (GCC) called by makefiles. This is a method I have used many times before, and is a standard way to keep a handle on the minutiae of compilation, such as optimisation levels.

As the project is quite complicated, and will likely have a large source code, version control will be essential. For this, I will use git and github to allow me to work on the project from multiple systems. This is another standard technique this is widely used.

**Usage and Example**

The program will be called from the terminal as follows:

mmltomidi [-o output\_path] [file]

The “-o” switch and output file name is optional: if it is not included then the output of the program will be stored in a file called “output.midi” in the folder in which the program is run.