TITLE:	The Maestro System	Library No. 34
		Version No. 1
		Date: 1969

Author	Further Information from: Q.E.C.
S. Price	Location of Program Tapes: Q.E.C.

SIR	X	Program	X	Source code	
ALGOL		Procedure		Relocatable binary	
FORTRAN		Subprogram		Absolute binary	X
		Subroutine			

Approximate store used:

Peripherals used	Program suitable for 903 only?
Paper tape reader	
Paper tape punch	Modifications planned?
On-line teleprinter	Users spec. available?

Abstract:

The Meastro system consists of a series of inter-related programs:-

1. Maestro

2. Tuneup

3. Keyboard

4. Bin-out

1 to 3 are supplied as one binary tape which is loaded at 8181.

1. MAESTRO

Maestro is the master program responsible for converting tunes, input as a series of notes, into a program which plays the tune. It is analogous to an assembler and like an assembler it can output assembled tunes. Maestro occupies locations 8 to 3200 and can use the rest of the store for storing tunes.

Maestro has a range of 25 notes (2 octaves) supplied ready tuned (stored in locations 1700 to

S. Price				Location of Program Tapes: Q.E.C.	
SIR	X	Program	X	Source code	
ALGOL		Procedure		Relocatable binary	
FORTRAN		Subprogram		Absolute binary	X
		Subroutine			

Approximate store used:

Peripherals used		Program suitable for 903 only?	
Paper tape reader			
Paper tape punch		Modifications planned?	
On-line teleprinter		Users spec. available?	

Abstract:

The Meastro system consists of a series of inter-related programs:-

1. Maestro

2. Tuneup

3. Keyboard

4. Bin-out

1 to 3 are supplied as one binary tape which is loaded at 8181.

1.   MAESTRO

Maestro is the master program responsible for converting tunes, input as a series of notes, into a program which plays the tune. It is analogous to an assembler and like an assembler it can output assembled tunes. Maestro occupies locations 8 to 3200 and can use the rest of the store for storing tunes.

Maestro has a range of 25 notes (2 octaves) supplied ready tuned (stored in locations 1700 to 3000); these notes can be given any names desired (and any one note may have more than one name). By entering a new dictionary of notes, consisting

of names for the 25 notes separated by ';`. If more
than one name for the same note is required separate
them by ',` and terminate with a ';`, end the list
with a halt code.

The standard dictionary consits of:-

C;CS;DF;D;DS,EF;E;F;FS,GF;G;GS,AF;A;AS,BF;B;CC;CCS,
DDF;DD;DDS,EEF;EE;FF;FFS,GGF;GG;GGS,AAF;AA;AAS,BBF;
BB;CCC;

This is the dictionary with which maestro is
supplied, the names have their obvious meanings
with f signifying flat and s sharp.

To write a tune for maestro the normal music
notation must be converted into maestro's language;
each note must have a number after it signifying
the length of than note.

Each note has a built in rest, a single length
note is half note and half rest, a double length note
is three quarters note and one quarter rest - twice
as long, and so on for greater lengths.

The rest can be suppressed by typing '$\frac{1}{2}$` after
the number.

An individual note can be repeated by typing '<`
after it for each repetition.

Additional rests can be introduced by typing
'-`, the rest being as long as a single length note.

Sections of tunes which are repeated can be
written as procedures, enclosed in round brackets,
the first 3 characters being the name of the procedure.
The procedure is called by this name followed by
a number which is the number of times the procedure
is to be played.

Nested or recursive procedures are not allowed.

A whole tune has a block structure defined by
square brackets everything except a halt code is
ignored before the first '[` further '[` may have
a repeat count preceding them. Every '[` must have
a matching ']`; procedures may contain blocks.

MAESTRO ENTRY POINTS

8	Load and assemble music tapes
9	Continue (after parity error etc.).
10	Play currently loaded tune.
11	Load new dictionary
12	Dump currently loaded tune.
13	Load dumped tune.

MAESTRO ERRORS

1	Store full	Continuation not possible
2	Too many blocks	Continuation not possible
3	Block looped zero, or too many times.	Once substituted

4	♭ Misplaced or ♯ misplaced	Ignored
5	Number misplaced	Ignored
6	Illegal character	Ignored
7	unrecognised note	Rest substituted
8	Note zero length or too long	Shortest note substituted
9	Note repeated zero or too many times	Once substituted
10	Dictionary full	Continuation not possible
11	Name already in use	Name not stored in Dict.
12	Recursion not allowed	Rest substituted
13	No procedure name, or illegal name	Name not stored in Dict.
14	No nested procedure declarations allowed (round brackets displaced)	Ignored
15	Number of [not equal to number of] inside procedure	
16	♭ follows procedure call	Ignored
17	OT dumped music tape (or mispunched or misread tape)	
18	Sum check failure on input of binary tape (i.e. mispunched or misread)	

GENERAL NOTES

1. Parity errors cause a halt and the playing of a few notes, continue at 9.

2. Blanks, spaces, carriage returns, line feeds and erases are always ignored.

3. A halt code causes a stop, continue at 9.

4. Entry at 9 causes a dynamic stop unless a halt code or a parity error has occured.

5. All characters before the first open square brackets are ignored (except parity errors and halt codes).

    No modifications are planned.

2.   TUNEUP

     Tuneup is a program used to tune the notes used by maestro. It occupies locations 4059 to 4520 and is entered at 4096. Tuneup can be overwritten by maestro when using long tunes.

     To understand tuneup, some knowledge of the structure of the notes is required.

Each note is a 50 instruction subroutine incorporating a loop 40 instructions long, the number of times this loop is executed determines the length of the note. The loop consists of 6 1 and 14 47 instructions in a proportion which determines the frequency of the note and the range over which it can be tuned, more 6 1 make for a higher note.

The note is tuned by adding or subtracting from the address part of the 14 47 instructions, when all 14 47 instructions have been reduced to 14 0 and 'L` is output signifying the lowest note possible, similarly and 'H` signifys the highest note.

After each change to the loop tuneup recalculates the number of times the loop must be executed in order to maintain its original length.

The actual use of tuneup is not easy since the different control characters are only recognised at certain times and just when is not defined.

For the purpose of tuneup the notes are numbered 1 to 25. On entry to tuneup a note is selected by typing its number. The note can now be raised by typing ' followed by a number representing the degree of the increase (typically 50 to 200), similarily a decrease is produced by first typing \ and then a number. The proportions of 6 1 to 14 47 instructions can be changed by typing ← space followed by two numbers adding up to 40. After each of these processes the length of the note is recalculated and the note played; 0 replay the tuned note type &.

The overall length of the note can be changed by typing ´. When T= is output, then new length is then typed (the set length is 1500).

A new note can be chosen by typing * followed by its number.

It has been found that upon entry to tuneup, after choosing the first note the note length needs to be specified as say 1500 and the note rechosen.

The present version of tuneup tends to be unpredictable and practice in its use is recommended before serious tuning is contemplated. A new version of tuneup is planned which will overcome these difficulties.

3. KEYBOARD

Keyboard enables the on-line teleprinter to be used as a keyboard. Keyboard occupies locations 4521 to 4582 and is entered at 15. The letters A to Z correspond to the notes in order (low to high).

The length of the notes can be increased or decreased in a range of 1 to 10 units (initially 2). The length is increased by one unit for every space typed, and decreased for every '−` all other characters are ignored.

Keyboard does not produce notes as clear as
maestro and too fast an input can cause unpredictable
results (often entry to other parts of the system),
improvement to overcome these problems are planned.

4.   BIN-OUT

Bin-out is a Binary tape loaded at 8181 which
occupies the top end of the store, it is entered
at 3175.

Bin-out outputs a sumchecked binary tape of any
areas of store which can be reloaded at 8181, it
can be used to output tuned notes (1700 to 3000) or
separate copies of the constituents of the maestro
system.

The areas of store to be output are specified
as a pair of integers and the sumcheck is added at
the end by typing % . Any number of disjoint areas
of store can be output on one tape.

With the Maestro system are supplied two compiled
tunes:-

1.   Cockles and Mussels.

2.   When the Saints Come Marching In.

The Maestro system is supplied by

        Queen Elizabeth Computing Society

Computing Society tape library references LCSD12
(Maestro System) and LCST12 (Bin-out)