

Section 3. Software Configuration

3-1. Section Overview

This section describes the software configuration of the LC/Mark4 interface.

3-2. Required Software

The following software is required to implement the LC/Mark4 interface:

- **DOS** operating system version 5.0 installed on the LC module.
 - DOS is loaded during module testing, however, a copy should be available in case the LC module RAM memory becomes corrupted (if necessary, refer to [Section 4](#) for instructions on installing DOS).
- A **configuration file** must be created (described in [Section 3-3](#)) and installed on the LC module (as described in [Section 4](#)).
- An **AUTOEXEC.BAT** file must be created (described in [Section 3-4](#)) and installed on the LC module (described in [Section 4](#)) so that the interface can be started automatically.
- LC Module utilities
 - **RLCEXTPC.EXE** (disk RLC10A)
 - **RLCFLASH.EXE** (disk RLC20A)
- The executable program **MARK4.EXE** is provided with the distribution diskette and must be installed on the LC module (described in [Section 4](#)).

3-3. Creating the Configuration File

The first step is to create the configuration file in a text editor. Once the configuration file has been created and verified, it must be loaded along with the driver executable file (MARK4.EXE) to the LC module.

The LC/Mark4 interface uses a configuration file to specify communication parameters and correspondences between Mark IV data fields and Ovation LC module registers. Ovation algorithms are used in the Ovation controller to map Ovation LC module registers to Ovation process points.

3-3.4. Channel/Offset Parameters

The data items passed by the Mark IV are specified in the configuration file by channel number, byte offset within the channel, and type of data at that offset. [Table 3-2](#) describes the parameters used in the configuration file to specify the Channel/Offset definitions.

The channel number specified by the Channel parameter is in effect until a subsequent channel number specification.

Table 3-2. Channel/Offset Parameter List

Parameter (alias)	Description	Value/Range	Example
channel	Channel number	1 through 8	channel 1
offset	Byte offset within channel message	0 through 253	offset 122
bit	Bit number for digitals	0 through 7	bit 7
type	Type of data at specified offset	int8, uint8, int16, uint16, int32, uint32, float, digital. See Table 3-3 .	type float
point	Ovation LC module style pseudo-point name	See Table 3-4 .	point S0100

Mark IV Type Specifiers

The type specifier for each Channel/Offset tells the interface software which of several Mark IV data types is associated with that Channel/Offset. The Mark IV data types supported by the LC/Mark4 interface are described in [Table 3-3](#).

Table 3-3. Mark IV Type Specifiers

Specifier	Description
int8	Treat a single byte as an eight-bit signed integer in the range -128 through +127.
uint8	Treat a single byte as an eight-bit unsigned integer in the range 0 through 255.
int16	Treat two bytes as a sixteen-bit signed integer in the range -32768 through +32767.
uint16	Treat two bytes as a sixteen-bit unsigned integer in the range 0 through 65535.
int32	Treat four bytes as a 32-bit signed integer in the range -2147483648 through +2147483647.
uint32	Treat four bytes as a 32-bit unsigned integer in the range 0 through 4294967295.
float	Treat four bytes as an IEEE format floating point number in the range $3.4E\pm38$.
digital	Treat an individual bit of the byte as a digital state (used in conjunction with <i>bit</i> parameter).

Ovation LC Module Point Name

The point parameter for each Channel/Offset is entered as though it was an Ovation point name (a pseudo point name). Analog values and digital states are exchanged between the Ovation controller and the Ovation LC module through a shared memory region which is thought of as consisting of 16-bit registers.

The point reference consists of an initial letter which indicates the type of reference, and a four digit number which specifies an LC register address, or offset into the shared memory region. This is described in [Table 3-5](#).

Table 3-4. LC Module Point Names

Point Name	Point Type	# LC Regs	Value Type	Read Algorithm and Format ¹	Write Algorithm and Format ¹
D0000 to D2047	Digital	1		SLCDIN ²	SLCDOUT ²
I0000 to I2047	Analog	1	integer	SLCAIN - 0 format ³	SLCAOUT - 0 format ³
F0000 to F2046	Analog	2	float (IEEE)	SLCAIN - 1 format ⁴	SLCAOUT - 1 format ⁴
S0000 to S2045	Analog	3	float (IEEE)	SLCAIN - 2 or 3 format ⁵	SLCAOUT - 2 or 3 format ⁵

¹ Refer to “[Ovation Algorithm Reference Manual](#)” (R3-1100) for more information about the SLC algorithms and their formats.

² The interface software and algorithm use a single 16 bit word to represent the digital status word as described in “[Ovation Record Types](#)” (R3-1140). The state of the point is represented as the least significant bit of the word. When reading a digital using the SLCDIN algorithm, some of the remaining bits of the digital status word are used to set the quality of the point.

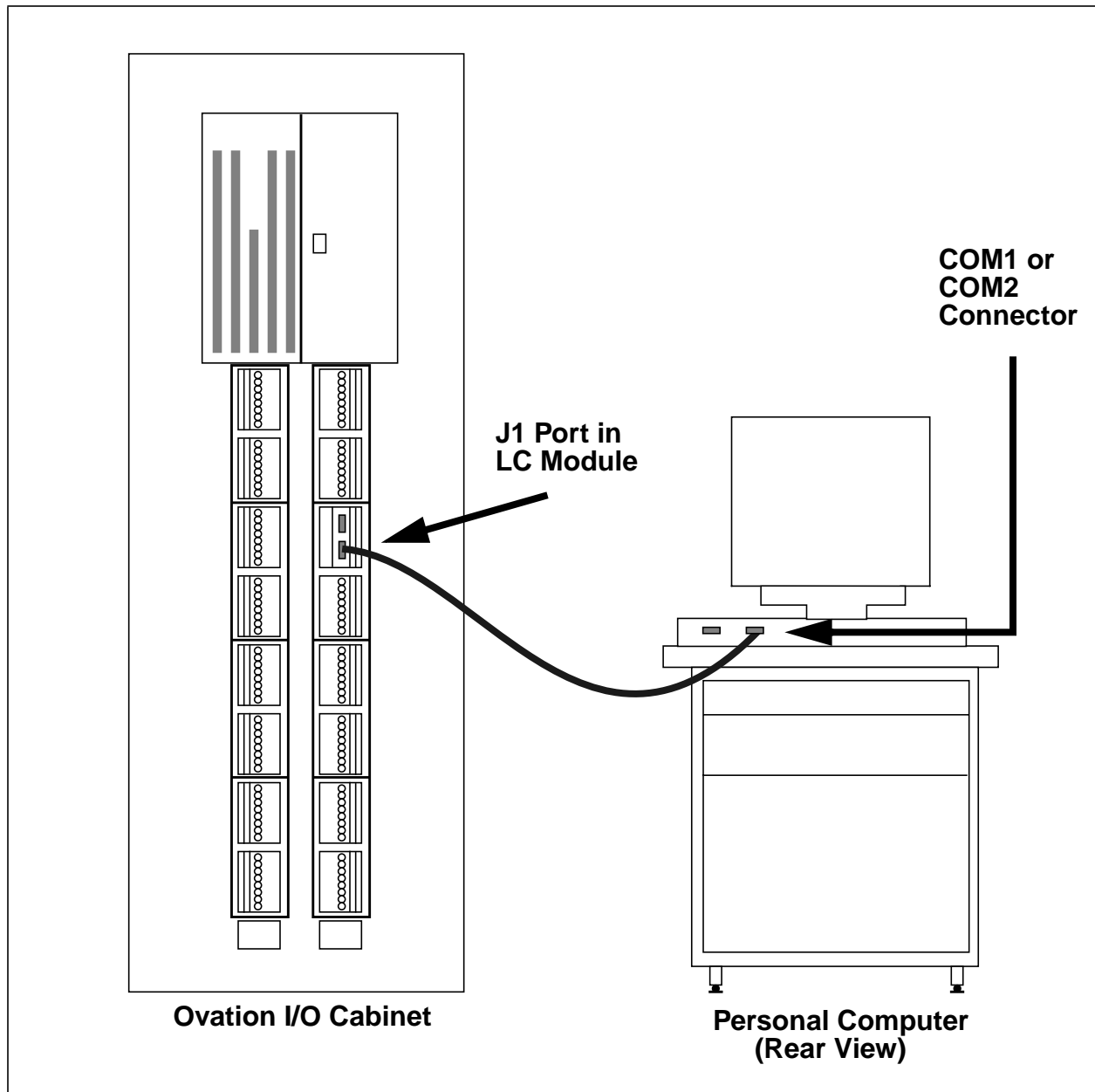
³ When using the SLCAIN or SLCAOUT algorithm with format 0 and an I0000 style point designator, the interface software and algorithm pass a single 16 bit word treated as a signed integer.

⁴ When using the SLCAIN or SLCAOUT algorithm with format 1 and an F0000 style point designator, the interface software and algorithm pass a four byte (two word or two register) IEEE format floating point value.

⁵ The interface software and algorithm pass a 16 bit word which represents the Analog Status Word followed by a four byte (two word or two register) IEEE format floating point value. A single S0000 style point uses three LC registers, this must be taken into account when laying out the data.

4-3. Link Controller Initialization

To perform the initial programming (or any later action requiring keyboard/CRT I/O), a serial port (J1) on the LC is linked to the external personal computer's COM1 or COM2 port. In this configuration, code generated on the external personal computer can be loaded into the LC. Use Procedure 1 or 2, as applicable.



Section 6. GE Speedtronic Mark IV Simulation

6-1. Section Overview

The GE Speedtronic Mark IV interface LC/Mark4 can be used to simulate the Mark IV turbine controller. This provides a way to confirm operation of the interface with the project configuration before actually connecting to the Mark IV controller.

This section describes the setup and operation of the Mark IV simulation.

6-2. Mark IV Simulation

The Mark IV simulation is accomplished by running the LC/Mark4 software on a PC compatible computer and using the `-t` command line parameter to tell the PC to enable the simulation menu. The PC serial port is then connected to the application port of the target Ovation LC module using a standard PC to PC 9-pin female to 9-pin female null-modem cable.

Using the same configuration file on both the target LC module and the simulator PC allows the simulating software to prompt you for the correct data types. For example, if the LC module command line is:

```
Mark4 -f stmtrb.cfg
```

then, on the simulator PC, use the command line:

```
Mark4 -f stmtrb.cfg -t
```

6-3. Simulation Menu

When the LC/Mark4 software is started with the `-t` command line parameter, a simulation menu becomes available for driving the simulation. The simulation menu is activated by pressing the “1” key. The user is then prompted to either enter values for channel/offsets or send a message to the target LC module. [Figure 6-1](#) shows a sample Mark IV simulation session. An analysis of the session is provided after the figure. User keyboard entries are indicated by **bold face type**.