

Thermal Marker on openC⁴D
Serine Protocol Communication Manual

Version 1

by
Claudimir Lucio do Lago
Department of Fundamental Chemistry
University of São Paulo
Brazil

December 2016

Contents

Introduction.....	2
Commands.....	3
Identification.....	3
Program.....	4
Sync	4
Run.....	4
Halt	4
Test	5
Status.....	5

Introduction

The hardware was conceived using a Teensy 2.0 allowing implementing a thermal marker system synchronized to an openC⁴D. The communication is allowed through a serial port over USB.

According to the Serine Protocol, the ID may be changed by the command “Ix”. In this case, the new ID must be used instead of the default ID ‘t’. For sake of simplicity, in this manual, the other virtual device exchanging messages is always ‘m’. However, in real communication this character should be replaced by the valid ID of the virtual device that is communicating with the detector.

The “\0” at the end of each message is just to remember that an additional position with a null character (ASCII code 0) at the end of the array is needed for most of the program languages (such as C, C++, Java, etc.). However, it is not part of the Serine Protocol.

Commands

Identification

This command allows get the ID of the virtual device as well as to change it.

Get – command string:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
t	m	I	;	\0																												

Return string:

0	1	2	3	4																												
m	t	i	y	s	...	s	;	\0																								

y – ‘t’ means that the following string (*s...s*) is a temporary identification; ‘P’ means that the following string (*s...s*) is a proprietary identification; ‘S’ means that the following string (*s...s*) is a valid ID provided by Serine Identification Service (SIS). Other character may be used by other public providers of identification service.

Change – command string:

0	1	2	3	4	5																											
t	m	I	x	y	s	...	s	;	\0																							

y – is the new ID to be adopted by the marker if the following string (*s...s*) is equal to its identification string. If the ID is changed, all the other messages exchanged with this device must use it instead of the default ID ‘t’.

Examples:

Sender	This device	Comment
tmI;	mtit_just_a_test;	The device has a temporary identification “t_just_a_test”
tmI;	mtiSdL021042;	The device is the SIS virtual device dL02, version 1, serial number 042.
tmIxwSdL021042;		The device identified with “SdL021042” has now the ID w instead of t;

Program

This command allows programing the thermal marker.

Command string:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
t	m	P	w	w	w	w	p	p	p	d	d	d	d	d	d	d	c	c	c	c	c	n	n	;	\0							

w – pulse width (ms).

p – power (0 to 100).

d – dwell time (ms) before the first pulse.

c – period (ms) of each cycle.

n – number of cycles.

Sync

This command allows enabling and disabling the synchronization of the beginning of the thermal marker program with the beginning of an electropherogram.

Command string:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
t	m	W	y	;	\0																											

y – 'N' turns synchronization on, 'F' turns synchronization off.

Run

This command starts the program of the thermal marker. The synchronization is lost whether it was previously programmed by the Sync command. A test of the filament and transistor is performed before starting the program. The result may be getting with Status command.

Command string:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
t	m	R	;	\0																												

Halt

This command aborts the program and sync mode.

Command string:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
t	m	H	;	\0																												

Test

This command executes an integrity test of filament and power transistor. Sync mode is automatically lost, as well as the program is aborted whether it is running. The result may be getting with Status command.

Command string:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
t	m	T	;	\0																												

Status

This command allows getting the state of the filament and transistor according with the last either Test or Run commands.

Command string:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
t	m	S	;	\0																												

Return string:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
m	t	s	f	t	p	w	x	x	n	n	;	\0																				

f – '1' if the filament is broken, '0' otherwise.

t – '1' if the transistor is broken, '0' otherwise.

p – '1' if the program is running, '0' otherwise.

w – '1' if the marker is synced with the detector, '0' otherwise.

xx – number of cycles to go.

nn – total number of cycles to be executed.