ROMPS CRITICAL DESIGN REVIEW
Volume III—Furnace Module Design Documentation

(NASA-CR-191615) ROMPS CRITICAL DESIGN REVIEW. VOLUME 3: FURNACE MODULF DESIGN DOCUMENTATION (ERIM)

M.E. DOBBS
DECEMBER 1992

84 p

Prepared for:
NASA Goddard Space Flight Center
Space Technology Division
Greenbelt, MD 20771

Contract No. NAG 5-1517

SpARC
Space Automation & Robotics Center

ERIM
P.O. Box 134001
Ann Arbor, MI 48113-4001
FURNACE MODULE DESIGN DOCUMENTATION
EASYLAB PROGRAMS
DEFINITIONS
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>C: CLEAR PROFILE</td>
<td>27</td>
</tr>
<tr>
<td>C: COMM STATUS</td>
<td>32</td>
</tr>
<tr>
<td>C: CONTROL START STATUS</td>
<td>31</td>
</tr>
<tr>
<td>D: DGAIN</td>
<td>28</td>
</tr>
<tr>
<td>D: ENABLE STATUS</td>
<td>30</td>
</tr>
<tr>
<td>D: ERROR DESCRIPTION</td>
<td>33</td>
</tr>
<tr>
<td>D: EXECUTE TEMP PROFILE 1</td>
<td>25</td>
</tr>
<tr>
<td>D: EXECUTE TEMP PROFILE 2</td>
<td>26</td>
</tr>
<tr>
<td>D: FURNACE STATUS</td>
<td>31</td>
</tr>
<tr>
<td>D: IGAIN</td>
<td>28</td>
</tr>
<tr>
<td>D: ILIMIT</td>
<td>29</td>
</tr>
<tr>
<td>D: LOOP TIME</td>
<td>29</td>
</tr>
<tr>
<td>D: MODULE STATUS</td>
<td>32</td>
</tr>
<tr>
<td>D: OVT OVERRIDE</td>
<td>30</td>
</tr>
<tr>
<td>D: PGAIN</td>
<td>27</td>
</tr>
<tr>
<td>D: POWER</td>
<td>2</td>
</tr>
<tr>
<td>D: POWER PROFILE 1</td>
<td>4</td>
</tr>
<tr>
<td>D: POWER PROFILE 2</td>
<td>5</td>
</tr>
<tr>
<td>D: POWER PROFILE 3</td>
<td>6</td>
</tr>
<tr>
<td>D: POWER PROFILE 4</td>
<td>7</td>
</tr>
<tr>
<td>D: POWER PROFILE 5</td>
<td>8</td>
</tr>
<tr>
<td>D: POWER PROFILE 6</td>
<td>9</td>
</tr>
<tr>
<td>D: POWER PROFILE 7</td>
<td>10</td>
</tr>
<tr>
<td>D: TEMP PROFILE 1</td>
<td>11</td>
</tr>
<tr>
<td>D: TEMP PROFILE 2</td>
<td>12</td>
</tr>
<tr>
<td>D: TEMP PROFILE 3</td>
<td>13</td>
</tr>
<tr>
<td>D: TEMP PROFILE 4</td>
<td>14</td>
</tr>
<tr>
<td>D: TEMP PROFILE 5</td>
<td>15</td>
</tr>
<tr>
<td>D: TEMP PROFILE 6</td>
<td>16</td>
</tr>
<tr>
<td>D: TEMP PROFILE 7</td>
<td>17</td>
</tr>
<tr>
<td>D: TEMPERATURE</td>
<td>2</td>
</tr>
<tr>
<td>D: TIME PROFILE 1</td>
<td>18</td>
</tr>
<tr>
<td>D: TIME PROFILE 2</td>
<td>19</td>
</tr>
<tr>
<td>D: TIME PROFILE 3</td>
<td>20</td>
</tr>
<tr>
<td>D: TIME PROFILE 4</td>
<td>21</td>
</tr>
<tr>
<td>D: TIME PROFILE 5</td>
<td>22</td>
</tr>
<tr>
<td>D: TIME PROFILE 6</td>
<td>23</td>
</tr>
<tr>
<td>D: TIME PROFILE 7</td>
<td>24</td>
</tr>
</tbody>
</table>
NAME: **F:POWER**

SYNTAX:  
\[ F:POWER = x \quad \text{or} \quad ? F:POWER \]

\( x = \text{power level in watts} \)

\( 0 \leq x \leq 255 \)

DESCRIPTION: SET POWER LEVEL  
COMMANDCODE #1

Set the power level of the furnace or get the current power setting from the furnace controller. The power setting can be queried only if the last furnace setting was for power. If the last furnace setting was for temperature, this command returns an error.

EXAMPLE:  
\[ F:POWER = 50 \]

\(? F:POWER \]

50

NAME: **F:TEMPERATURE**

SYNTAX:  
\[ F:TEMP = x \quad \text{or} \quad ? F:TEMP \]

\( x = \text{temperature in degrees Celsius} \)

\( 0 \leq x \leq 1600 \)

DESCRIPTION: SET TEMPERATURE  
COMMANDCODE #2

Set the temperature of the furnace or get the current temperature setting from the furnace controller. The temperature setting can be queried only if the last furnace setting was for temperature. If the last furnace setting was for power, this command returns an error.

EXAMPLE:  
\[ F:TEMP = 1000 \]

\(? F:TEMP \]

1000
NAME: F:POWER.PROFILE.1

SYNTAX: F:POWER.PROFILE.1 = x  or  ? F:POWER.PROFILE.1

x = power level in watts
0 <= x <= 255

DESCRIPTION: SET POWER PROFILE STEP 1
COMMANDCODE #3

Set or return the power level for step 1 of a power profile. Power profiles consist of seven power/time steps. For each step, the power level of the furnace is set to a predefined level for a predefined amount of time. Once all seven steps have been executed, the power profile is finished.

EXAMPLE: F:POWER.PROFILE.1 = 50
F:TIME.PROFILE.1 = 60
F:POWER.PROFILE.2 = 100
F:TIME.PROFILE.2 = 120
F:POWER.PROFILE.3 = 200
F:TIME.PROFILE.3 = 120
F:POWER.PROFILE.4 = 250
F:TIME.PROFILE.4 = 300
F:POWER.PROFILE.5 = 200
F:TIME.PROFILE.5 = 300
F:POWER.PROFILE.6 = 100
F:TIME.PROFILE.6 = 120
F:POWER.PROFILE.7 = 0
F:TIME.PROFILE.7 = 60
F:EXECUTE.POWER.PROFILE

? F:POWER.PROFILE.1
50
NAME: F:POWER.PROFILE.2

SYNTAX: F:POWER.PROFILE.2 = x  or  ? F:POWER.PROFILE.2

x = power level in watts

0 <= x <= 255

DESCRIPTION: SET POWER PROFILE STEP 2
COMMANDCODE #4

Set or return the power level for step 2 of a power profile. Power profiles consist of seven power/time steps. For each step, the power level of the furnace is set to a predefined level for a predefined amount of time. Once all seven steps have been executed, the power profile is finished.

EXAMPLE:

F:POWER.PROFILE.1 = 50
F:TIME.PROFILE.1 = 60
F:POWER.PROFILE.2 = 100
F:TIME.PROFILE.2 = 120
F:POWER.PROFILE.3 = 200
F:TIME.PROFILE.3 = 120
F:POWER.PROFILE.4 = 250
F:TIME.PROFILE.4 = 300
F:POWER.PROFILE.5 = 200
F:TIME.PROFILE.5 = 300
F:POWER.PROFILE.6 = 100
F:TIME.PROFILE.6 = 120
F:POWER.PROFILE.7 = 0
F:TIME.PROFILE.7 = 60
F:EXECUTE.POWER.PROFILE

? F:POWER.PROFILE.2
100
NAME:  F:POWER.PROFILE.3

SYNTAX:  F:POWER.PROFILE.3 = x  or  ? F:POWER.PROFILE.3

x = power level in watts

0 <= x <= 255

DESCRIPTION:  SET POWER PROFILE STEP 3

COMMANDCODE #5

Set or return the power level for step 3 of a power profile. Power profiles consist of seven power/time steps. For each step, the power level of the furnace is set to a predefined level for a predefined amount of time. Once all seven steps have been executed, the power profile is finished.

EXAMPLE:

F:POWER.PROFILE.1 = 50
F:TIME.PROFILE.1 = 60
F:POWER.PROFILE.2 = 100
F:TIME.PROFILE.2 = 120
F:POWER.PROFILE.3 = 200
F:TIME.PROFILE.3 = 120
F:POWER.PROFILE.4 = 250
F:TIME.PROFILE.4 = 300
F:POWER.PROFILE.5 = 200
F:TIME.PROFILE.5 = 300
F:POWER.PROFILE.6 = 100
F:TIME.PROFILE.6 = 120
F:POWER.PROFILE.7 = 0
F:TIME.PROFILE.7 = 60
F:EXECUTE POWER PROFILE

? F:POWER.PROFILE.3
200
NAME: F:POWER.PROFILE.4

SYNTAX: F:POWER.PROFILE.4 = x or ? F:POWER.PROFILE.4

x = power level in watts

0 <= x <= 255

DESCRIPTION: SET POWER PROFILE STEP 4
COMMANDCODE #6

Set or return the power level for step 4 of a power profile. Power profiles consist of seven power/time steps. For each step, the power level of the furnace is set to a predefined level for a predefined amount of time. Once all seven steps have been executed, the power profile is finished.

EXAMPLE: F:POWER.PROFILE.1 = 50
F:TIME.PROFILE.1 = 60
F:POWER.PROFILE.2 = 100
F:TIME.PROFILE.2 = 120
F:POWER.PROFILE.3 = 200
F:TIME.PROFILE.3 = 120
F:POWER.PROFILE.4 = 250
F:TIME.PROFILE.4 = 300
F:POWER.PROFILE.5 = 200
F:TIME.PROFILE.5 = 300
F:POWER.PROFILE.6 = 100
F:TIME.PROFILE.6 = 120
F:POWER.PROFILE.7 = 0
F:TIME.PROFILE.7 = 60
F:EXECUTE.POWER.PROFILE

? F:POWER.PROFILE.4
250
NAME: F:POWER.PROFILE.5

SYNTAX: F:POWER.PROFILE.5 = x  or  ? F:POWER.PROFILE.5

x = power level in watts

0 <= x <= 255

DESCRIPTION: SET POWER PROFILE STEP 5

COMMANDCODE #7

Set or return the power level for step 5 of a power profile. Power profiles consist of seven power/time steps. For each step, the power level of the furnace is set to a predefined level for a predefined amount of time. Once all seven steps have been executed, the power profile is finished.

EXAMPLE:

F:POWER.PROFILE.1 = 50
F:TIME.PROFILE.1 = 60
F:POWER.PROFILE.2 = 100
F:TIME.PROFILE.2 = 120
F:POWER.PROFILE.3 = 200
F:TIME.PROFILE.3 = 120
F:POWER.PROFILE.4 = 250
F:TIME.PROFILE.4 = 300
F:POWER.PROFILE.5 = 200
F:TIME.PROFILE.5 = 300
F:POWER.PROFILE.6 = 100
F:TIME.PROFILE.6 = 120
F:POWER.PROFILE.7 = 0
F:TIME.PROFILE.7 = 60
F:EXECUTE POWER.PROFILE

? F:POWER.PROFILE.5
200
NAME: F:POWER.PROFILE.6

SYNTAX: F:POWER.PROFILE.6 = x or ? F:POWER.PROFILE.6

x = power level in watts

0 <= x <= 255

DESCRIPTION: SET POWER PROFILE STEP 6

COMMANDCODE #8

Set or return the power level for step 6 of a power profile. Power profiles consist of seven power/time steps. For each step, the power level of the furnace is set to a predefined level for a predefined amount of time. Once all seven steps have been executed, the power profile is finished.

EXAMPLE:

F:POWER.PROFILE.1 = 50
F:TIME.PROFILE.1 = 60
F:POWER.PROFILE.2 = 100
F:TIME.PROFILE.2 = 120
F:POWER.PROFILE.3 = 200
F:TIME.PROFILE.3 = 120
F:POWER.PROFILE.4 = 250
F:TIME.PROFILE.4 = 300
F:POWER.PROFILE.5 = 200
F:TIME.PROFILE.5 = 300
F:POWER.PROFILE.6 = 100
F:TIME.PROFILE.6 = 120
F:POWER.PROFILE.7 = 0
F:TIME.PROFILE.7 = 60
F:EXECUTE.PROFILE

? F:POWER.PROFILE.6
100
NAME: F:POWER.PROFILE.7

SYNTAX: F:POWER.PROFILE.7 = x  or  ? F:POWER.PROFILE.7

x  = power level in watts
0 <= x <= 255

DESCRIPTION: SET POWER PROFILE STEP 7
COMMANDCODE #9

Set or return the power level for step 7 of a power profile. Power profiles consist of seven power/time steps. For each step, the power level of the furnace is set to a predefined level for a predefined amount of time. Once all seven steps have been executed, the power profile is finished.

EXAMPLE:
F:POWER.PROFILE.1 = 50
F:TIME.PROFILE.1 = 60
F:POWER.PROFILE.2 = 100
F:TIME.PROFILE.2 = 120
F:POWER.PROFILE.3 = 200
F:TIME.PROFILE.3 = 120
F:POWER.PROFILE.4 = 250
F:TIME.PROFILE.4 = 300
F:POWER.PROFILE.5 = 200
F:TIME.PROFILE.5 = 300
F:POWER.PROFILE.6 = 100
F:TIME.PROFILE.6 = 120
F:POWER.PROFILE.7 = 0
F:TIME.PROFILE.7 = 60
F:EXECUTE.POWER.PROFILE

? F:POWER.PROFILE.7
0
NAME: F:TEMP.PROFILE.1

SYNTAX: F:TEMP.PROFILE.1 = x or ? F:TEMP.PROFILE.1

x = temperature in degrees Celsius

0 <= x <= 1600

DESCRIPTION: SET TEMPERATURE PROFILE STEP 1

COMMANDCODE #10

Set or return the temperature for step 1 of a temperature profile. Temperature profiles consist of seven temperature/time steps. For each step, the temperature of the furnace is set to a predefined level for a predefined amount of time. Once all seven steps have been executed, the temperature profile is finished.

EXAMPLE:

F:TEMPERATURE.PROFILE.1 = 100
F:TIME.PROFILE.1 = 60
F:TEMPERATURE.PROFILE.2 = 200
F:TIME.PROFILE.2 = 120
F:TEMPERATURE.PROFILE.3 = 1000
F:TIME.PROFILE.3 = 120
F:TEMPERATURE.PROFILE.4 = 1500
F:TIME.PROFILE.4 = 300
F:TEMPERATURE.PROFILE.5 = 800
F:TIME.PROFILE.5 = 300
F:TEMPERATURE.PROFILE.6 = 400
F:TIME.PROFILE.6 = 120
F:TEMPERATURE.PROFILE.7 = 0
F:TIME.PROFILE.7 = 60
F:EXECUTE.TEMPERATURE.PROFILE

? F:TEMPERATURE.PROFILE.1

100
NAME: F:TEMP.PROFILE.2

SYNTAX: F:TEMP.PROFILE.2 = x  or  ? F:TEMP.PROFILE.2

x = temperature in degrees Celsius

0 <= x <= 1600

DESCRIPTION: SET TEMPERATURE PROFILE STEP 2
COMMANDCODE #11

Set or return the temperature for step 2 of a temperature profile. Temperature profiles consist of seven temperature/time steps. For each step, the temperature of the furnace is set to a predefined level for a predefined amount of time. Once all seven steps have been executed, the temperature profile is finished.

EXAMPLE:

F:TEMPERATURE.PROFILE.1 = 100
F:TIME.PROFILE.1 = 60
F:TEMPERATURE.PROFILE.2 = 200
F:TIME.PROFILE.2 = 120
F:TEMPERATURE.PROFILE.3 = 1000
F:TIME.PROFILE.3 = 120
F:TEMPERATURE.PROFILE.4 = 1500
F:TIME.PROFILE.4 = 300
F:TEMPERATURE.PROFILE.5 = 800
F:TIME.PROFILE.5 = 300
F:TEMPERATURE.PROFILE.6 = 400
F:TIME.PROFILE.6 = 120
F:TEMPERATURE.PROFILE.7 = 0
F:TIME.PROFILE.7 = 60
F:EXECUTE.TEMPERATURE.PROFILE

? F:TEMPERATURE.PROFILE.2
200
NAME: F:TEMP.PROFILE.3

SYNTAX: F:TEMP.PROFILE.3 = x  or  ? F:TEMP.PROFILE.3

x = temperature in degrees Celsius
0 <= x <= 1600

DESCRIPTION: SET TEMPERATURE PROFILE STEP 3
COMMANDCODE #12

Set or return the temperature for step 3 of a temperature profile. Temperature profiles consist of seven temperature/time steps. For each step, the temperature of the furnace is set to a predefined level for a predefined amount of time. Once all seven steps have been executed, the temperature profile is finished.

EXAMPLE:
F:TEMPERATURE.PROFILE.1 = 100
F:TIME.PROFILE.1 = 60
F:TEMPERATURE.PROFILE.2 = 200
F:TIME.PROFILE.2 = 120
F:TEMPERATURE.PROFILE.3 = 1000
F:TIME.PROFILE.3 = 120
F:TEMPERATURE.PROFILE.4 = 1500
F:TIME.PROFILE.4 = 300
F:TEMPERATURE.PROFILE.5 = 800
F:TIME.PROFILE.5 = 300
F:TEMPERATURE.PROFILE.6 = 400
F:TIME.PROFILE.6 = 120
F:TEMPERATURE.PROFILE.7 = 0
F:TIME.PROFILE.7 = 60
F:EXECUTE.TEMPERATURE.PROFILE

? F:TEMPERATURE.PROFILE.3
1000
NAME: F:TEMP.PROFILE.4

SYNTAX: F:TEMP.PROFILE.4 = x or ? F:TEMP.PROFILE.4

x = temperature in degrees Celsius

0 <= x <= 1600

DESCRIPTION: SET TEMPERATURE PROFILE STEP 4
COMMANDCODE #13

Set or return the temperature for step 4 of a temperature profile. Temperature profiles consist of seven temperature/time steps. For each step, the temperature of the furnace is set to a predefined level for a predefined amount of time. Once all seven steps have been executed, the temperature profile is finished.

EXAMPLE: F:TEMPERATURE.PROFILE.1 = 100
F:TIME.PROFILE.1 = 60
F:TEMPERATURE.PROFILE.2 = 200
F:TIME.PROFILE.2 = 120
F:TEMPERATURE.PROFILE.3 = 1000
F:TIME.PROFILE.3 = 120
F:TEMPERATURE.PROFILE.4 = 1500
F:TIME.PROFILE.4 = 300
F:TEMPERATURE.PROFILE.5 = 800
F:TIME.PROFILE.5 = 300
F:TEMPERATURE.PROFILE.6 = 400
F:TIME.PROFILE.6 = 120
F:TEMPERATURE.PROFILE.7 = 0
F:TIME.PROFILE.7 = 60
F:EXECUTE.TEMPERATURE.PROFILE

? F:TEMPERATURE.PROFILE.4
1500
NAME: F:TEMP.PROFILE.5

SYNTAX: F:TEMP.PROFILE.5 = x or ? F:TEMP.PROFILE.5

x = temperature in degrees Celsius
0 <= x <= 1600

DESCRIPTION: SET TEMPERATURE PROFILE STEP 5
COMMAND CODE #14

Set or return the temperature for step 5 of a temperature profile. Temperature profiles consist of seven temperature/time steps. For each step, the temperature of the furnace is set to a predefined level for a predefined amount of time. Once all seven steps have been executed, the temperature profile is finished.

EXAMPLE:
F:TEMPERATURE.PROFILE.1 = 100
F:TIME.PROFILE.1 = 60
F:TEMPERATURE.PROFILE.2 = 200
F:TIME.PROFILE.2 = 120
F:TEMPERATURE.PROFILE.3 = 1000
F:TIME.PROFILE.3 = 120
F:TEMPERATURE.PROFILE.4 = 1500
F:TIME.PROFILE.4 = 300
F:TEMPERATURE.PROFILE.5 = 800
F:TIME.PROFILE.5 = 300
F:TEMPERATURE.PROFILE.6 = 400
F:TIME.PROFILE.6 = 120
F:TEMPERATURE.PROFILE.7 = 0
F:TIME.PROFILE.7 = 60
F:EXECUTE.TEMPERATURE.PROFILE

? F:TEMPERATURE.PROFILE.5
800
NAME: F:TEMP.PROFILE.6

SYNTAX: F:TEMP.PROFILE.6 = x  or  ? F:TEMP.PROFILE.6

x = temperature in degrees Celsius

0 <= x <= 1600

DESCRIPTION: SET TEMPERATURE PROFILE STEP 6
COMMANDCODE #15

Set or return the temperature for step 6 of a temperature profile. Temperature profiles consist of seven temperature/time steps. For each step, the temperature of the furnace is set to a predefined level for a predefined amount of time. Once all seven steps have been executed, the temperature profile is finished.

EXAMPLE:
F:TEMPERATURE.PROFILE.1 = 100
F:TIME.PROFILE.1 = 60
F:TEMPERATURE.PROFILE.2 = 200
F:TIME.PROFILE.2 = 120
F:TEMPERATURE.PROFILE.3 = 1000
F:TIME.PROFILE.3 = 120
F:TEMPERATURE.PROFILE.4 = 1500
F:TIME.PROFILE.4 = 300
F:TEMPERATURE.PROFILE.5 = 800
F:TIME.PROFILE.5 = 300
F:TEMPERATURE.PROFILE.6 = 400
F:TIME.PROFILE.6 = 120
F:TEMPERATURE.PROFILE.7 = 0
F:TIME.PROFILE.7 = 60
F:EXECUTE.TEMPERATURE.PROFILE

? F:TEMPERATURE.PROFILE.6
400
NAME: F:TEMP.PROFILE.7

SYNTAX: F:TEMP.PROFILE.7 = x or ? F:TEMP.PROFILE.7

x = temperature in degrees Celsius

0 <= x <= 1600

DESCRIPTION: SET TEMPERATURE PROFILE STEP 7
COMMANDCODE #16

Set or return the temperature for step 7 of a temperature profile. Temperature profiles consist of seven temperature/time steps. For each step, the temperature of the furnace is set to a predefined level for a predefined amount of time. Once all seven steps have been executed, the temperature profile is finished.

EXAMPLE:

F:TEMPERATURE.PROFILE.1 = 100
F:TIME.PROFILE.1 = 60
F:TEMPERATURE.PROFILE.2 = 200
F:TIME.PROFILE.2 = 120
F:TEMPERATURE.PROFILE.3 = 1000
F:TIME.PROFILE.3 = 120
F:TEMPERATURE.PROFILE.4 = 1500
F:TIME.PROFILE.4 = 300
F:TEMPERATURE.PROFILE.5 = 800
F:TIME.PROFILE.5 = 300
F:TEMPERATURE.PROFILE.6 = 400
F:TIME.PROFILE.6 = 120
F:TEMPERATURE.PROFILE.7 = 0
F:TIME.PROFILE.7 = 60
F:EXECUTE.TEMPERATURE.PROFILE

? F:TEMPERATURE.PROFILE.7
0
NAME: F:TIME.PROFILE.1

SYNTAX: F:TIME.PROFILE.1 = x  or  ? F:TIME.PROFILE.1

x = time in seconds

DESCRIPTION: SET TIME PROFILE STEP 1
COMMANDCODE #17

Set or return the time for step 1 of a power or temperature profile. Power and temperature profiles each consist of seven power/time or temperature/time steps. For each step, the power or temperature of the furnace is set to a predefined level for a predefined amount of time. Once all seven steps have been executed, the power or temperature profile is finished.

EXAMPLE:

F:TEMPERATURE.PROFILE.1 = 100
F:TIME.PROFILE.1 = 60
F:TEMPERATURE.PROFILE.2 = 400
F:TIME.PROFILE.2 = 120
F:TEMPERATURE.PROFILE.3 = 1000
F:TIME.PROFILE.3 = 120
F:TEMPERATURE.PROFILE.4 = 1200
F:TIME.PROFILE.4 = 300
F:TEMPERATURE.PROFILE.5 = 800
F:TIME.PROFILE.5 = 300
F:TEMPERATURE.PROFILE.6 = 200
F:TIME.PROFILE.6 = 120
F:TEMPERATURE.PROFILE.7 = 0
F:TIME.PROFILE.7 = 60
F:EXECUTE.TEMPERATURE.PROFILE

? F:TIME.PROFILE.1
60
NAME: F:TIME.PROFILE.2

SYNTAX: F:TIME.PROFILE.2 = x  or  ? F:TIME.PROFILE.2

x = time in seconds

DESCRIPTION: SET TIME PROFILE STEP 2
COMMANDCODE #18

Set or return the time for step 2 of a power or temperature profile. Power and temperature profiles each consist of seven power/time or temperature/time steps. For each step, the power or temperature of the furnace is set to a predefined level for a predefined amount of time. Once all seven steps have been executed, the power or temperature profile is finished.

EXAMPLE:
F:TEMPERATURE.PROFILE.1 = 250
F:TIME.PROFILE.1 = 60
F:TEMPERATURE.PROFILE.2 = 500
F:TIME.PROFILE.2 = 120
F:TEMPERATURE.PROFILE.3 = 750
F:TIME.PROFILE.3 = 120
F:TEMPERATURE.PROFILE.4 = 1500
F:TIME.PROFILE.4 = 300
F:TEMPERATURE.PROFILE.5 = 1000
F:TIME.PROFILE.5 = 300
F:TEMPERATURE.PROFILE.6 = 500
F:TIME.PROFILE.6 = 120
F:TEMPERATURE.PROFILE.7 = 0
F:TIME.PROFILE.7 = 60
F:EXECUTE.TEMPERATURE.PROFILE

? F:TIME.PROFILE.2
120
NAME: F:TIME.PROFILE.3

SYNTAX:  
F:TIME.PROFILE.3 = x  or  ? F:TIME.PROFILE.3

x = time in seconds

DESCRIPTION: SET TIME PROFILE STEP 3
COMMANDCODE #19

Set or return the time for step 1 of a power or temperature profile. Power and temperature profiles each consist of seven power/time or temperature/time steps. For each step, the power or temperature of the furnace is set to a predefined level for a predefined amount of time. Once all seven steps have been executed, the power or temperature profile is finished.

EXAMPLE:

F:TEMPERATURE.PROFILE.1 = 350
F:TIME.PROFILE.1 = 60
F:TEMPERATURE.PROFILE.2 = 450
F:TIME.PROFILE.2 = 120
F:TEMPERATURE.PROFILE.3 = 770
F:TIME.PROFILE.3 = 120
F:TEMPERATURE.PROFILE.4 = 900
F:TIME.PROFILE.4 = 300
F:TEMPERATURE.PROFILE.5 = 1200
F:TIME.PROFILE.5 = 300
F:TEMPERATURE.PROFILE.6 = 670
F:TIME.PROFILE.6 = 120
F:TEMPERATURE.PROFILE.7 = 200
F:TIME.PROFILE.7 = 60
F:EXECUTE.TEMPERATURE.PROFILE

? F:TIME.PROFILE.3
120
NAME: F:TIME.PROFILE.4

SYNTAX: F:TIME.PROFILE.4 = x or ? F:TIME.PROFILE.4

x = time in seconds

DESCRIPTION: SET TIME PROFILE STEP 4
COMMANDCODE #20

Set or return the time for step 4 of a power or temperature profile. Power and temperature profiles each consist of seven power/time or temperature/time steps. For each step, the power or temperature of the furnace is set to a predefined level for a predefined amount of time. Once all seven steps have been executed, the power or temperature profile is finished.

EXAMPLE:
F:POWER.PROFILE.1 = 50
F:TIME.PROFILE.1 = 60
F:POWER.PROFILE.2 = 100
F:TIME.PROFILE.2 = 120
F:POWER.PROFILE.3 = 200
F:TIME.PROFILE.3 = 120
F:POWER.PROFILE.4 = 230
F:TIME.PROFILE.4 = 300
F:POWER.PROFILE.5 = 220
F:TIME.PROFILE.5 = 300
F:POWER.PROFILE.6 = 170
F:TIME.PROFILE.6 = 120
F:POWER.PROFILE.7 = 0
F:TIME.PROFILE.7 = 60
F:EXECUTE POWER PROFILE

? F:TIME.PROFILE.4
300
NAME: F:TIME.PROFILE.5

SYNTAX: F:TIME.PROFILE.5 = x or ? F:TIME.PROFILE.5

x = time in seconds

DESCRIPTION: SET TIME PROFILE STEP 5
COMMANDCODE #21

Set or return the time for step 5 of a power or temperature profile. Power and temperature profiles each consist of seven power/time or temperature/time steps. For each step, the power or temperature of the furnace is set to a predefined level for a predefined amount of time. Once all seven steps have been executed, the power or temperature profile is finished.

EXAMPLE: F:TEMPERATURE.PROFILE.1 = 440
F:TIME.PROFILE.1 = 60
F:TEMPERATURE.PROFILE.2 = 460
F:TIME.PROFILE.2 = 120
F:TEMPERATURE.PROFILE.3 = 500
F:TIME.PROFILE.3 = 120
F:TEMPERATURE.PROFILE.4 = 530
F:TIME.PROFILE.4 = 300
F:TEMPERATURE.PROFILE.5 = 550
F:TIME.PROFILE.5 = 300
F:TEMPERATURE.PROFILE.6 = 300
F:TIME.PROFILE.6 = 120
F:TEMPERATURE.PROFILE.7 = 120
F:TIME.PROFILE.7 = 60
F:EXECUTE.TEMPERATURE.PROFILE

? F:TIME.PROFILE.5
300
NAME: F:TIME.PROFILE.6

SYNTAX: F:TIME.PROFILE.6 = x or ? F:TIME.PROFILE.6

x = time in seconds

DESCRIPTION: SET TIME PROFILE STEP 6
COMMANDCODE #22

Set or return the time for step 6 of a power or temperature profile. Power and temperature profiles each consist of seven power/time or temperature/time steps. For each step, the power or temperature of the furnace is set to a predefined level for a predefined amount of time. Once all seven steps have been executed, the power or temperature profile is finished.

EXAMPLE:
F:POWER.PROFILE.1 = 20
F:TIME.PROFILE.1 = 60
F:POWER.PROFILE.2 = 30
F:TIME.PROFILE.2 = 120
F:POWER.PROFILE.3 = 40
F:TIME.PROFILE.3 = 120
F:POWER.PROFILE.4 = 50
F:TIME.PROFILE.4 = 300
F:POWER.PROFILE.5 = 70
F:TIME.PROFILE.5 = 300
F:POWER.PROFILE.6 = 30
F:TIME.PROFILE.6 = 120
F:POWER.PROFILE.7 = 10
F:TIME.PROFILE.7 = 60
F:EXECUTE.POWER.PROFILE

? F:TIME.PROFILE.6
120
NAME: F:TIME.PROFILE.7

SYNTAX: F:TIME.PROFILE.7 = \( x \) or \( ? \) F:TIME.PROFILE.7

\( x \) = time in seconds

DESCRIPTION: SET TIME PROFILE STEP 7

COMMAND CODE #23

Set or return the time for step 1 of a power or temperature profile. Power and temperature profiles each consist of seven power/time or temperature/time steps. For each step, the power or temperature of the furnace is set to a predefined level for a predefined amount of time. Once all seven steps have been executed, the power or temperature profile is finished.

EXAMPLE:

F:POWER.PROFILE.1 = 35
F:TIME.PROFILE.1 = 60
F:POWER.PROFILE.2 = 60
F:TIME.PROFILE.2 = 120
F:POWER.PROFILE.3 = 120
F:TIME.PROFILE.3 = 120
F:POWER.PROFILE.4 = 150
F:TIME.PROFILE.4 = 300
F:POWER.PROFILE.5 = 255
F:TIME.PROFILE.5 = 300
F:POWER.PROFILE.6 = 100
F:TIME.PROFILE.6 = 120
F:POWER.PROFILE.7 = 0
F:TIME.PROFILE.7 = 60

? F:TIME.PROFILE.7
60
NAME: F:EXECUTE.TEMP.PROFILE

SYNTAX: F:EXECUTE.TEMP.PROFILE

DESCRIPTION: EXECUTE TEMPERATURE PROFILE
COMMAND CODE #24

Execute a temperature profile. Temperature profiles consist of seven temperature/time steps. For each step, the temperature of the furnace is set to a predefined level for a predefined amount of time. Once all seven steps have been executed, the temperature profile is finished. If any temperature/time steps are undefined, they are simply not executed, and the temperature profile continues on to the next step.

EXAMPLE:

F:TEMPERATURE.PROFILE.1 = 200
F:TIME.PROFILE.1 = 60
F:TEMPERATURE.PROFILE.2 = 500
F:TIME.PROFILE.2 = 120
F:TEMPERATURE.PROFILE.3 = 400
F:TIME.PROFILE.3 = 120
F:TEMPERATURE.PROFILE.4 = 700
F:TIME.PROFILE.4 = 300
F:TEMPERATURE.PROFILE.5 = 1300
F:TIME.PROFILE.5 = 300
F:TEMPERATURE.PROFILE.6 = 900
F:TIME.PROFILE.6 = 120
F:TEMPERATURE.PROFILE.7 = 100
F:TIME.PROFILE.7 = 60
F:EXECUTE.TEMP.PROFILE
NAME: F:EXECUTE.POWER.PROFILE

SYNTAX: F:EXECUTE.POWER.PROFILE

DESCRIPTION: EXECUTE POWER PROFILE
COMMANDCODE #25

Execute a power profile. Power profiles consist of seven Power/time steps. For each step, the power of the furnace is set to a predefined level for a predefined amount of time. Once all seven steps have been executed, the power profile is finished. If any power/time steps are undefined, they are simply not executed, and the power profile continues on to the next step.

EXAMPLE:

F:POWER.PROFILE.1 = 20
F:TIME.PROFILE.1 = 60
F:POWER.PROFILE.2 = 70
F:TIME.PROFILE.2 = 120
F:POWER.PROFILE.3 = 50
F:TIME.PROFILE.3 = 120
F:POWER.PROFILE.4 = 100
F:TIME.PROFILE.4 = 300
F:POWER.PROFILE.5 = 50
F:TIME.PROFILE.5 = 300
F:POWER.PROFILE.6 = 90
F:TIME.PROFILE.6 = 120
F:POWER.PROFILE.7 = 0
F:TIME.PROFILE.7 = 60
F:EXECUTE.POWER.PROFILE
NAME: F:CLEAR.PROFILE
SYNTAX: F:CLEAR.PROFILE

DESCRIPTION: CLEAR PROFILES
COMMANDCODE #26

Clear all seven power/temperature/time profile steps.

EXAMPLE: F:CLEAR.PROFILE

NAME: F:PGAIN
SYNTAX: F:.PGAIN = x or ? F:PGAIN

x = proportional gain term (KP) for servo calculations.

0 <= x <= 255

DESCRIPTION: PROPORTIONAL GAIN COMMAND
COMMANDCODE #27

Define the proportional gain term (KP) used in the servo calculations for the furnace controller or get the current proportional gain term from the furnace controller.

EXAMPLE: F:PGAIN = 0
? F:PGAIN
0
NAME: F:IGAIN
SYNTAX: F:IGAIN = x  or  ? F:IGAIN

x = integral gain term (KI) for servo calculations.

0 <= x <= 255

DESCRIPTION: INTEGRAL GAIN COMMAND
COMMANDCODE #28

Define the integral gain term (KI) used in the servo calculations for the furnace controller or get the current integral gain term from the furnace controller.

EXAMPLE: F:IGAIN = 0
? F:IGAIN
0

NAME: F:DGAIN
SYNTAX: F:DGAIN = x  or  ? F:DGAIN

x = derivative gain term (KD) for servo calculations.

0 <= x <= 255

DESCRIPTION: DERIVATIVE GAIN COMMAND
COMMANDCODE #29

Define the proportional gain term (KP) used in the servo calculations for the furnace controller or get the current derivative gain term from the furnace controller.

EXAMPLE: F:PGAIN = 0
? F:PGAIN
0
NAME:   F:ILIMIT

SYNTAX:  F:ILIMIT = x or ? F:ILIMIT

x = integrator limit for servo calculations.

0 <= x <= 255

DESCRIPTION: INTEGRATOR LIMIT COMMAND
COMMANDCODE #30

Define the integrator limit used in the servo calculations for the furnace controller get the current integrator limit from the furnace controller.

EXAMPLE:  F:ILIMIT = 0
? F:ILIMIT
0

NAME:   F:LOOPTIME

SYNTAX:  F:LOOPTIME = x or ? F:LOOPTIME

x = looptime for servo calculations.

0 <= x <= 255

DESCRIPTION: LOOPTIME COMMAND
COMMANDCODE #31

Define the looptime used in the servo calculations for the furnace controller or get the current looptime from the furnace controller.

EXAMPLE:  F:LOOPTIME = 0
? F:LOOPTIME
0
NAME: F:OVT.OVERRIDE

SYNTAX: F:OVT.OVERRIDE = 0/1  or  ? F:OVT.OVERRIDE

0 = don’t override
1 = override

DESCRIPTION: OVERTEMP OVERRIDE COMMAND
COMMANDCODE #32

Override/don’t override overtemp fault condition or return the last overtemp override setting. If an overtemp fault condition is overridden, it is important to clear the override after the fault condition is removed. If the override is not cleared, overtemp be detected.

EXAMPLE: ? F:OVT.OVERRIDE
1
F:OVT.OVERRIDE = 0

NAME: F:OVEN.SELECT.STATUS

SYNTAX: ? F:OVEN.SELECT.STATUS

DESCRIPTION: OVEN SELECT STATUS COMMAND
COMMANDCODE #33

Get the status from the furnace controller and return a status byte containing the select status for oven A and oven B:

0 = B Oven on
1 = A Oven on

EXAMPLE: ? F:SELECT.STATUS
0
NAME: F:CONTROL.START.STATUS

SYNTAX: ? F:CONTROL.START.STATUS

DESCRIPTION: CONTROL START STATUS COMMAND
COMMANDCODE #34

Get the status from the furnace controller and return a status byte containing the control start status:

0 = Control start is disabled
1 = Control start is enabled

EXAMPLE: ? F:CONTROL.START.STATUS
0

NAME: F:FURNACE.STATUS

SYNTAX: ? F:FURNACE.STATUS

DESCRIPTION: FURNACE CONTROLLER STATUS COMMAND
COMMANDCODE #35

Get the status from the furnace controller and return a bitmapped status byte:

Bit 0 Invalid command ID or invalid byte count
Bit 1 28 volt bus too low to achieve command setpoint
Bit 2 Overtemp occurred
Bit 3 Invalid checksum
Bit 4 Last power or temperature setpoint out of range
Bit 5 Watchdog activated
Bit 6 Not used
Bit 7 Not used

EXAMPLE: ? F:FURNACE.STATUS
0
NAME: F:COMM.STATUS

SYNTAX: ? F:COMM.STATUS

DESCRIPTION: COMMUNICATION STATUS COMMAND
COMMANDCODE #36

Return a bitmapped status byte containing the communication status of the last furnace command:

Bit 0 Not used
Bit 1 Not used
Bit 2 Not used
Bit 3 Not used
Bit 4 Invalid checksum
Bit 5 Invalid command code
Bit 6 Invalid byte count
Bit 7 Interbyte timeout

EXAMPLE: ? F:COMM.STATUS
0

NAME: F:MODULE.STATUS

SYNTAX: ? F:MODULE.STATUS

DESCRIPTION: FURNACE MODULE STATUS COMMAND
COMMANDCODE #37

Return the status of the last EasyLab command:

1 = Hard abort
2 = User stop
3 = Furnace Controller communication error
4 = Furnace fault
9 = Furnace module cannot sign on
10 = Furnace version is not available
11 = Invalid furnace command
12 = Command is not for this furnace
13 = Memory request denied (insufficient memory)
14 = Dictionary entry does not exist
15 = Dictionary entry already exists
16 = Illegal furnace index

EXAMPLE: ? F:MODULE.STATUS
0
NAME: F:ERROR.DESCRIPTION

SYNTAX: ? F:ERROR.DESCRIPTION

DESCRIPTION: ERROR DESCRIPTION COMMAND
COMMANDCODE #38

Return a description of the last error.

EXAMPLE: ? F:ERROR.DESCRIPTION
FURNACE/CONTROLLER COMMUNICATION ERROR
FURNACE
EASYLAB COMMANDS
FLOW CHARTS
SET POWER LEVEL
COMMANDCODE #1

command message exchange

input command?

NO

last control setpoint
POWER?

YES

return

get power level

calculate watts

return watts

command message exchange

NO

return -1

set power level

calculate power level

YES

int LENGTH
char TYPE
int FORMAT
char DISK
char USAGE
char ABBREV
char NAMELENGTH
char *NAME
char *COMMAND

int MODULEID
char COMMANDCODE
char FORMATCODE
float VALUE
SET TEMPERATURE
COMMANDCODE #2

input command?

last control setpoint TEMPERATURE?

set temp. level

calculate temperature counts

return degrees Centigrade

calculate degrees Centigrade

get temp. level

serial message exchange

command message exchange

command message exchange

int MODULEID
char COMMANDCODE
char FORMATCODE
float VALUE

int LENGTH
char TYPE
int FORMAT
char DISK
char USAGE
char ABBREV
char NAMELENGTH
char *NAME
char *COMMAND

int MODULEID
char COMMANDCODE
char FORMATCODE
float VALUE

int LENGTH
char TYPE
int FORMAT
char DISK
char USAGE
char ABBREV
char NAMELENGTH
char *NAME
char *COMMAND
SET POWER PROFILE STEP 1
COMMANDCODE #3

int LENGTH
char TYPE
int FORMAT
char DISK
char USAGE
char ABBREV
char NAMELENGTH
char *NAME
char *COMMAND

command message exchange

NO
input command?

YES

int MODULEID
char COMMANDCODE
char FORMATCODE
float VALUE

return previously stored value

store power value for step 1 execution of a power profile

int MODULEID
char COMMANDCODE
char FORMATCODE
float VALUE

command message exchange

int LENGTH
char TYPE
int FORMAT
char DISK
char USAGE
char ABBREV
char NAMELENGTH
char *NAME
char *COMMAND
SET POWER PROFILE STEP 2
COMMANDCODE #4

int LENGTH
cchar TYPE
tint FORMAT
cchar DISK
cchar USAGE
cchar ABBREV
cchar NAMELENGTH
cchar *NAME
cchar *COMMAND

command message exchange

input command?

NO

return previously stored value

store power value for step 2 execution of a power profile

YES

input command?

command message exchange

int MODULEID
cchar COMMANDCODE
cchar FORMATCODE
cfloat VALUE

int LENGTH
cchar TYPE
tint FORMAT
cchar DISK
cchar USAGE
cchar ABBREV
cchar NAMELENGTH
cchar *NAME
cchar *COMMAND

int MODULEID
cchar COMMANDCODE
cchar FORMATCODE
cfloat VALUE
SET POWER PROFILE STEP 3
COMMANDCODE #5

int LENGTH
char TYPE
int FORMAT
char DISK
char USAGE
char ABBREV
char NAMELENGTH
char *NAME
char *COMMAND

command message exchange

input command?

YES
store power value for step 3 execution of a power profile

NO
return previously stored value

command message exchange

int MODULEID
char COMMANDCODE
char FORMATCODE
float VALUE

int LENGTH
char TYPE
int FORMAT
char DISK
char USAGE
char ABBREV
char NAMELENGTH
char *NAME
char *COMMAND

int MODULEID
char COMMANDCODE
char FORMATCODE
float VALUE
SET POWER PROFILE STEP 4
COMMANDCODE #6

int LENGTH
char TYPE
int FORMAT
char DISK
char USAGE
char ABBREV
char NAMELENGTH
char *NAME
char *COMMAND

command message exchange

input command?

NO

command message exchange

YES

return previously stored value

store power value for step 4 execution of a power profile

int MODULEID
char COMMANDCODE
char FORMATCODE
float VALUE

int LENGTH
char TYPE
int FORMAT
char DISK
char USAGE
char ABBREV
char NAMELENGTH
char *NAME
char *COMMAND

int MODULEID
char COMMANDCODE
char FORMATCODE
float VALUE
SET POWER PROFILE STEP 5
COMMANDCODE #7

int LENGTH
char TYPE
int FORMAT
char DISK
char USAGE
char ABBREV
char NAMELENGTH
char *NAME
char *COMMAND

command message exchange

input command?
NO
return previously stored value
store power value for step 5 execution of a power profile

YES

int MODULEID
char COMMANDCODE
char FORMATCODE
float VALUE

command message exchange

int LENGTH
char TYPE
int FORMAT
char DISK
char USAGE
char ABBREV
char NAMELENGTH
char *NAME
char *COMMAND

int MODULEID
char COMMANDCODE
char FORMATCODE
float VALUE
SET POWER PROFILE STEP 6
COMMANDCODE #8

command message exchange

input command?

store power value for step 6 execution of a power profile

return previously stored value

command message exchange

int LENGTH
char TYPE
int FORMAT
char DISK
char USAGE
char ABBREV
char NAMELENGTH
char *NAME
char *COMMAND

int MODULEID
char COMMANDCODE
char FORMATCODE
float VALUE

int MODULEID
char COMMANDCODE
char FORMATCODE
float VALUE
SET POWER PROFILE STEP 7
COMMANDCODE #9

input command?

command message exchange

NO

YES

command message exchange

input command?

return previously stored value

store power value for step 7 execution of a power profile

command message exchange

command message exchange

int LENGTH
char TYPE
int FORMAT
char DISK
char USAGE
char ABBREV
char NAMELENGTH
char *NAME
char *COMMAND

int MODULEID
char COMMANDCODE
char FORMATCODE
float VALUE

int MODULEID
char COMMANDCODE
char FORMATCODE
float VALUE

int LENGTH
char TYPE
int FORMAT
char DISK
char USAGE
char ABBREV
char NAMELENGTH
char *NAME
char *COMMAND

int MODULEID
char COMMANDCODE
char FORMATCODE
float VALUE

int LENGTH
char TYPE
int FORMAT
char DISK
char USAGE
char ABBREV
char NAMELENGTH
char *NAME
char *COMMAND

int MODULEID
char COMMANDCODE
char FORMATCODE
float VALUE

int LENGTH
char TYPE
int FORMAT
char DISK
char USAGE
char ABBREV
char NAMELENGTH
char *NAME
char *COMMAND

int MODULEID
char COMMANDCODE
char FORMATCODE
float VALUE
SET TEMPERATURE PROFILE STEP 1
COMMANDCODE #10

command message exchange

command message exchange

NO input command?
YES

store temperature value for step 1 execution of a temperature profile

return previously stored value

int LENGTH
char TYPE
int FORMAT
char DISK
char USAGE
char ABBREV
char NAMELENGTH
char *NAME
char *COMMAND

int MODULEID
char COMMANDCODE
char FORMATCODE
float VALUE

int MODULEID
char COMMANDCODE
char FORMATCODE
float VALUE
SET TEMPERATURE PROFILE STEP 2
COMMANDCODE #11

int LENGTH
char TYPE
int FORMAT
char DISK
char USAGE
char ABBREV
char NAMELENGTH
char *NAME
char *COMMAND

int MODULEID
char COMMANDCODE
char FORMATCODE
float VALUE

command message exchange

NO
input command?

YES

return previously stored value

store temperature value for step 2 execution of a temperature profile

int LENGTH
char TYPE
int FORMAT
char DISK
char USAGE
char ABBREV
char NAMELENGTH
char *NAME
char *COMMAND

int MODULEID
char COMMANDCODE
char FORMATCODE
float VALUE

command message exchange
SET TEMPERATURE PROFILE STEP 3
COMMANDCODE #12

int LENGTH
char TYPE
int FORMAT
char DISK
char USAGE
char ABBREV
char NAMELENGTH
char *NAME
char *COMMAND

command message exchange

input command?

YES

store temperature value for step 3 execution of a temperature profile

NO

return previously stored value

command message exchange

int MODULEID
char COMMANDCODE
char FORMATCODE
float VALUE

int LENGTH
char TYPE
int FORMAT
char DISK
char USAGE
char ABBREV
char NAMELENGTH
char *NAME
char *COMMAND

int MODULEID
char COMMANDCODE
char FORMATCODE
float VALUE
SET TEMPERATURE PROFILE STEP 4
COMMANDCODE #13

int LENGTH
char TYPE
int FORMAT
char DISK
char USAGE
char ABBREV
char NAMELENGTH
char *NAME
char *COMMAND

command message
exchange

NO
input
command?

YES

return
previously
stored value

store temperature
value for step 4
execution of a
temperature profile

int MODULEID
char COMMANDCODE
char FORMATCODE
float VALUE

command message
exchange

int LENGTH
char TYPE
int FORMAT
char DISK
char USAGE
char ABBREV
char NAMELENGTH
char *NAME
char *COMMAND

int MODULEID
char COMMANDCODE
char FORMATCODE
float VALUE
SET TEMPERATURE PROFILE STEP 5
COMMANDCODE #14

int LENGTH
char TYPE
int FORMAT
char DISK
char USAGE
char ABBREV
char NAMELENGTH
char *NAME
char *COMMAND

command message
exchange

NO
input command?

YES

return
previously
stored value

store temperature
value for step 5
execution of a
temperature profile

int MODULEID
char COMMANDCODE
char FORMATCODE
float VALUE

int LENGTH
char TYPE
int FORMAT
char DISK
char USAGE
char ABBREV
char NAMELENGTH
char *NAME
char *COMMAND

command message
exchange

int MODULEID
char COMMANDCODE
char FORMATCODE
float VALUE
SET TEMPERATURE PROFILE STEP 6
COMMANDCODE #15

command message exchange

NO
input command?

YES
store temperature value for step 6
execution of a temperature profile

command message exchange

return previously stored value

int LENGTH
char TYPE
int FORMAT
char DISK
char USAGE
char ABBREV
char NAMELENGTH
char *NAME
char *COMMAND

int MODULEID
char COMMANDCODE
char FORMATCODE
float VALUE

int MODULEID
char COMMANDCODE
char FORMATCODE
float VALUE

int LENGTH
char TYPE
int FORMAT
char DISK
char USAGE
char ABBREV
char NAMELENGTH
char *NAME
char *COMMAND

int MODULEID
char COMMANDCODE
char FORMATCODE
float VALUE
SET TEMPERATURE PROFILE STEP 7
COMMANDCODE #16

int LENGTH
char TYPE
int FORMAT
char DISK
char USAGE
char ABBREV
char NAMELENGTH
char *NAME
char *COMMAND

command message exchange

input command?

NO

return previously stored value

store temperature value for step 7 execution of a temperature profile

YES

int MODULEID
char COMMANDCODE
char FORMATCODE
float VALUE

command message exchange

int LENGTH
char TYPE
int FORMAT
char DISK
char USAGE
char ABBREV
char NAMELENGTH
char *NAME
char *COMMAND

int MODULEID
char COMMANDCODE
char FORMATCODE
float VALUE
SET TIME PROFILE STEP 1
COMMANDCODE #17

int LENGTH
char TYPE
int FORMAT
char DISK
char USAGE
char ABBREV
char NAMELENGTH
char *NAME
char *COMMAND

command message exchange

input command?

NO
return previously stored value

YES
keep time value for step 1 execution of a time profile

int MODULEID
char COMMANDCODE
char FORMATCODE
float VALUE

int MODULEID
char COMMANDCODE
char FORMATCODE
float VALUE

command message exchange

int LENGTH
char TYPE
int FORMAT
char DISK
char USAGE
char ABBREV
char NAMELENGTH
char *NAME
char *COMMAND
SET TIME PROFILE STEP 2
COMMANDCODE #18

command message
exchange

input command?

return previously stored value

keep time value for step 2 execution of a time profile

command message
exchange

int LENGTH
char TYPE
int FORMAT
cchar DISK
cchar USAGE
cchar ABBREV
cchar NAMELENGTH
cchar *NAME
cchar *COMMAND

int MODULEID
char COMMANDCODE
char FORMATCODE
float VALUE

int MODULEID
char COMMANDCODE
char FORMATCODE
float VALUE

int LENGTH
char TYPE
int FORMAT
cchar DISK
cchar USAGE
cchar ABBREV
cchar NAMELENGTH
cchar *NAME
cchar *COMMAND
SET TIME PROFILE STEP 3
COMMANDCODE #19

command message exchange

input command?

return previously stored value
keep time value for step 3 execution of a time profile

command message exchange

int LENGTH
char TYPE
int FORMAT
cchar DISK
cchar USAGE
cchar ABBREV
cchar NAMELENGTH
cchar *NAME
cchar *COMMAND

int MODULEID
cchar COMMANDCODE
cchar FORMATCODE
cfloat VALUE

int MODULEID
cchar COMMANDCODE
cchar FORMATCODE
cfloat VALUE
SET TIME PROFILE STEP 4
COMMANDCODE #20

command message exchange

input command?

YES

keep time value for step 4 execution of a time profile

return previously stored value

command message exchange

int LENGTH
char TYPE
int FORMAT
char DISK
char USAGE
char ABBREV
char NAMELENGTH
char *NAME
char *COMMAND

int MODULEID
char COMMANDCODE
char FORMATCODE
float VALUE

int LENGTH
char TYPE
int FORMAT
char DISK
char USAGE
char ABBREV
char NAMELENGTH
char *NAME
char *COMMAND

int MODULEID
char COMMANDCODE
char FORMATCODE
float VALUE

int MODULEID
char COMMANDCODE
char FORMATCODE
float VALUE
SET TIME PROFILE STEP 6
COMMANDCODE #22

input command? 

command message exchange

return previously stored value

keep time value for step 6 execution of a time profile

command message exchange

int LENGTH
char TYPE
int FORMAT
char DISK
char USAGE
char ABBREV
char NAMELENGTH
char *NAME
char *COMMAND

int MODULEID
char COMMANDCODE
char FORMATCODE
float VALUE

int MODULEID
char COMMANDCODE
char FORMATCODE
float VALUE
SET TIME PROFILE STEP 7
COMMAND CODE #23

int LENGTH
cchar TYPE
int FORMAT
cchar DISK
cchar USAGE
cchar ABBREV
cchar NAMELENGTH
cchar *NAME
cchar *COMMAND

int MODULEID
cchar COMMAND CODE
cchar FORMAT CODE
float VALUE

command message exchange

NO
input command?

YES

return previously stored value

keep time value for step 7 execution of a time profile

int MODULEID
cchar COMMAND CODE
cchar FORMAT CODE
float VALUE

command message exchange

int LENGTH
cchar TYPE
int FORMAT
cchar DISK
cchar USAGE
cchar ABBREV
cchar NAME LENGTH
cchar *NAME
cchar *COMMAND

int MODULEID
cchar COMMAND CODE
cchar FORMAT CODE
float VALUE
EXECUTE TEMPERATURE PROFILE
COMMANDCODE #24

int LENGTH
char TYPE
int FORMAT
char DISK
char USAGE
char ABBREV
char NAMELENGTH
char *NAME
char *COMMAND

command message exchange

current step = 1

set temperature for current step
serial message exchange

wait pre-defined # of seconds for current step

increment current step

NO

current step > 7 ?

YES

int LENGTH
char TYPE
int FORMAT
char DISK
char USAGE
char ABBREV
char NAMELENGTH
char *NAME
char *COMMAND

command message exchange
EXECUTE POWER PROFILE
COMMANDCODE #25

```
int LENGTH
char TYPE
int FORMAT
char DISK
char USAGE
char ABBREV
char NAMELENGTH
char *NAME
char *COMMAND
```

```
command message exchange
```

```
current step = 1
```

```
set power level for current step
```

```
serial message exchange
```

```
wait pre-defined # of seconds for current step
```

```
increment current step
```

```
command message exchange
```

```
current step > 7 ?
```

```
NO
```

```
set power level for current step
```

```
serial message exchange
```

```
increment current step
```

```
command message exchange
```

```
int LENGTH
char TYPE
int FORMAT
char DISK
char USAGE
char ABBREV
char NAMELENGTH
char *NAME
char *COMMAND
```

```
current step > 7 ?
```

```
YES
```

```
set power level for current step
```

```
serial message exchange
```

```
increment current step
```

```
command message exchange
```

```
int LENGTH
char TYPE
int FORMAT
char DISK
char USAGE
char ABBREV
char NAMELENGTH
char *NAME
char *COMMAND
```
CLEAR PROFILES
COMMANDCODE #26

command message exchange

command message exchange

CLEAR PROFILES
COMMANDCODE #26

int LENGTH
c char TYPE
int FORMAT
c char DISK
c char USAGE
c char ABBREV
c char NAMELENGTH
c char *NAME
c char *COMMAND

clear power profile steps
clear temperature profile steps
clear time profile steps

int LENGTH
c char TYPE
int FORMAT
c char DISK
c char USAGE
c char ABBREV
c char NAMELENGTH
c char *NAME
c char *COMMAND
PROPORTIONAL GAIN COMMAND
COMMANDCODE #27

- int LENGTH
  - char TYPE
  - int FORMAT
  - char DISK
  - char USAGE
  - char ABBREV
  - char NAMELENGTH
  - char *NAME
  - char *COMMAND

- command message exchange

- input command?
  - NO
  - serial message exchange
  - return proportional gain term KP
  - update proportional gain term KP
  - set PID factors
  - get status
  - NO

- YES
  - get status
  - serial message exchange
  - update proportional gain term KP
  - set PID factors

- command message exchange

- int MODULEID
  - char COMMANDCODE
  - char FORMATCODE
  - float VALUE

- int MODULEID
  - char COMMANDCODE
  - char FORMATCODE
  - float VALUE
INTEGRAL GAIN COMMAND
COMMANDCODE #28

INPUT COMMAND

command message exchange

command message exchange

input command?

NO

get status

serial message exchange

update integral gain term KI

set PID factors

return integral gain term KI

int MODULEID
char COMMANDCODE
char FORMATCODE
float VALUE

int LENGTH
char TYPE
int FORMAT
char DISK
char USAGE
char ABBREV
char NAMELENGTH
char *NAME
char *COMMAND

YES

get status

int MODULEID
char COMMANDCODE
char FORMATCODE
float VALUE

int LENGTH
char TYPE
int FORMAT
char DISK
char USAGE
char ABBREV
char NAMELENGTH
char *NAME
char *COMMAND
DERIVATIVE GAIN COMMAND
COMMANDCODE #29

int LENGTH
char TYPE
int FORMAT
char DISK
char USAGE
char ABBREV
char NAMELENGTH
char *NAME
char *COMMAND

command message exchange

input command?

NO

get status

serial message exchange

get status

YES

update derivative gain term KD

set PID factors

return derivative gain term KD

command message exchange

int MODULEID
char COMMANDCODE
char FORMATCODE
float VALUE

int MODULEID
char COMMANDCODE
char FORMATCODE
float VALUE
LOOPTIME COMMAND
COMMANDCODE #31

int LENGTH
char TYPE
int FORMAT
char DISK
char USAGE
char ABBREV
char NAMELENGTH
char *NAME
char *COMMAND

command message exchange

input command?

get status

serial message exchange

update looptime

set PID factors

return looptime

get status

command message exchange

int MODULEID
char COMMANDCODE
char FORMATCODE
float VALUE

int LENGTH
char TYPE
int FORMAT
char DISK
char USAGE
char ABBREV
char NAMELENGTH
char *NAME
char *COMMAND

int MODULEID
char COMMANDCODE
char FORMATCODE
float VALUE
OVERTEMP OVERRIDE COMMAND
COMMANDCODE #32

int LENGTH
c char TYPEint FORMAT
c char DISK
c char USAGE
c char ABBREV
c char NAMELENGTH
c char *NAME
c char *COMMAND

command message exchange

NO

input command?

YES

return overtemp override flag
serial message exchange

set overtemp override flag

int MODULEID
c char COMMANDCODE
c char FORMATCODE
float VALUE

command message exchange

int MODULEID
c char COMMANDCODE
c char FORMATCODE
float VALUE

int LENGTH
c char TYPEint FORMAT
c char DISK
c char USAGE
c char ABBREV
c char NAMELENGTH
c char *NAME
c char *COMMAND

command message exchange
SELECT STATUS COMMAND
COMMANDCODE #33

int LENGTH
char TYPE
int FORMAT
char DISK
char USAGE
char ABBREV
char NAMELENGTH
char *NAME
char *COMMAND

command message exchange

get status

serial message exchange

int MODULEID
char COMMANDCODE
float VALUE

return oven
select status

int LENGTH
char TYPE
int FORMAT
char DISK
char USAGE
char ABBREV
char NAMELENGTH
char *NAME
char *COMMAND

command message exchange

int MODULEID
char COMMANDCODE
float VALUE
FURNACE STATUS COMMAND
COMMANDCODE #35

int LENGTH
char TYPE
int FORMAT
char DISK
char USAGE
char ABBREV
char NAMELENGTH
char *NAME
char *COMMAND

command message
exchange

get status

int MODULEID
char COMMANDCODE
float VALUE

serial message
exchange

return
furnace status

int LENGTH
char TYPE
int FORMAT
char DISK
char USAGE
char ABBREV
char NAMELENGTH
char *NAME
char *COMMAND

command message
exchange

int MODULEID
char COMMANDCODE
float VALUE
COMMUNICATION STATUS COMMAND
COMMANDCODE #36

int LENGTH
char TYPE
int FORMAT
char DISK
char USAGE
char ABBREV
char NAMELENGTH
char *NAME
char *COMMAND

command message exchange

return communication status from last furnace module/controller message exchange

int MODULEID
char COMMANDCODE
float VALUE

int LENGTH
char TYPE
int FORMAT
char DISK
char USAGE
char ABBREV
char NAMELENGTH
char *NAME
char *COMMAND

int MODULEID
char COMMANDCODE
float VALUE
MODULE STATUS COMMAND
COMMANDCODE #37

int LENGTH
char TYPE
int FORMAT
char DISK
char USAGE
char ABBREV
char NAMELENGTH
char *NAME
char *COMMAND

command message exchange

return error code:
0 = no error;
>0 = error occurred

int MODULEID
char COMMANDCODE
float VALUE

int LENGTH
char TYPE
int FORMAT
char DISK
char USAGE
char ABBREV
char NAMELENGTH
char *NAME
char *COMMAND

int MODULEID
char COMMANDCODE
float VALUE

command message exchange
ERROR DESCRIPTION COMMAND
COMMANDCODE #38

int LENGTH
cchar TYPE
ing FORMAT
cchar DISK
cchar USAGE
cchar ABBREV
cchar NAMELENGTH
cchar *NAME
cchar COMMAND

int MODULEID
cchar COMMANDCODE
cchar *DESCRIPTION

return a description of the last non-status request command

int LENGTH
cchar TYPE
ing FORMAT
cchar DISK
cchar USAGE
cchar ABBREV
cchar NAMELENGTH
cchar *NAME
cchar COMMAND

int MODULEID
cchar COMMANDCODE
cchar *DESCRIPTION

command message exchange
EASYLAB PROGRAMS
DEFINITIONS
FURNACE MODULE
EASYLAB PROGRAMS

Space Automated Research Center (SpARC)

December 3, 1992
TABLE OF CONTENTS

GET.FROM.FURNACE ................................................................. 2
PUT.INTO.FURNACE ............................................................... 2
NAME: GET.FROM.FURNACE

SYNTAX: GET.FROM.FURNACE

DESCRIPTION: Remove a sample from a furnace. A sample must be in the furnace.

RETURNS: OK - successful return
         NOTOK - error return

In addition to a NOTOK error return, a message is printed on the terminal and F:MODULE.STATUS is set to indicate the error.

EXAMPLE: RACK.NUMBER = 1
         SAMPLE.NUMBER = 5
         GET.FROM.RACK
         FURNACE.NUMBER = A
         PUT.INTO.FURNACE
         GET.FROM.FURNACE

NAME: PUT.INTO.FURNACE

SYNTAX: PUT.INTO.FURNACE

DESCRIPTION: Put a sample into a furnace. A sample must be in the robot hand and the furnace number must be defined before this command is executed.

RETURNS: OK - successful return
         NOTOK - error return

In addition to a NOTOK error return, a message is printed on the terminal and F:MODULE.STATUS is set to indicate the error.

EXAMPLE: RACK.NUMBER = 1
         SAMPLE.NUMBER = 5
         GET.FROM.RACK
         FURNACE.NUMBER = B
         PUT.INTO.FURNACE
FURNACE
EASYLAB PROGRAMS
FLOW CHARTS
Use FURNACE_NUMBER to build furnace variables

Move to last clear position

Move to furnace clear position; Update last clear position

NO

Hand full?

Furnace empty?

NO

Move up into furnace

Update processing parameters: furnace is full

Return NOT OK to interpreter

YES

Return OK to interpreter
EASYLAB PROGRAM: GET.FROM.FURNACE
PROCESSING FLOW CHART

Use FURNACE.NUMBER to build furnace variables

NO

Furnace empty?

YES

Move to last clear position

Update processing parameters: furnace is empty

Return NOT OK to interpreter

Return OK to interpreter
FURNACE MODULE
FAULT CONDITIONS
# Furnace Module Software
## Fault Handling Summary

<table>
<thead>
<tr>
<th>Fault Condition</th>
<th>Fault Detection</th>
<th>Fault Response</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>STOP EZC Processing</strong></td>
<td>User presses STOP key OR System ISR updates EZC Processing status monitored by Stop Task.</td>
<td>Stop Task sends “SET POWER = 0” and “SET TEMPERATURE = 0” commands to Furnace Controller. Furnace Task updates Error Status and terminates command.</td>
</tr>
<tr>
<td><strong>Furnace/Furnace Controller Communication Error</strong></td>
<td>Furnace Task sends a message to the Furnace Controller; Furnace Controller sends a one byte error code in response.</td>
<td>Furnace Task attempts to send the message until the retries are exhausted, then updates Error Status and terminates command.</td>
</tr>
<tr>
<td><strong>Invalid controller command, 28V bus too low to achieve setpoint, overtemp occurred, invalid checksum, setpoint out of range, watchdog timeout</strong></td>
<td>Furnace Task sends a “READ STATUS” message to the Furnace Controller; Furnace Controller sends ten status bytes in response.</td>
<td>Furnace Task updates Error Status and terminates command.</td>
</tr>
<tr>
<td><strong>Invalid Command</strong></td>
<td>Furnace Task compares Command Code to valid Command Codes.</td>
<td>Furnace Task updates Error Status and terminates command.</td>
</tr>
<tr>
<td><strong>Command Is Not For This Furnace</strong></td>
<td>Furnace Task compares Command Module ID to it's own Module ID</td>
<td>Furnace Task updates Error Status and terminates command.</td>
</tr>
</tbody>
</table>