Configuring digital I/O with MATLAB® Data Acquisition Toolbox

These instructions explain how to set the direction of digital ports and bits from a Measurement Computing (MCC) data acquisition device using MATLAB Data Acquisition Toolbox*. The Measurement Computing brand miniLAB 1008 is used in this example.

**This example was tested with MATLAB version 7.*

1. Run *Insta*Cal and note the board number assigned to the miniLAB 1008.

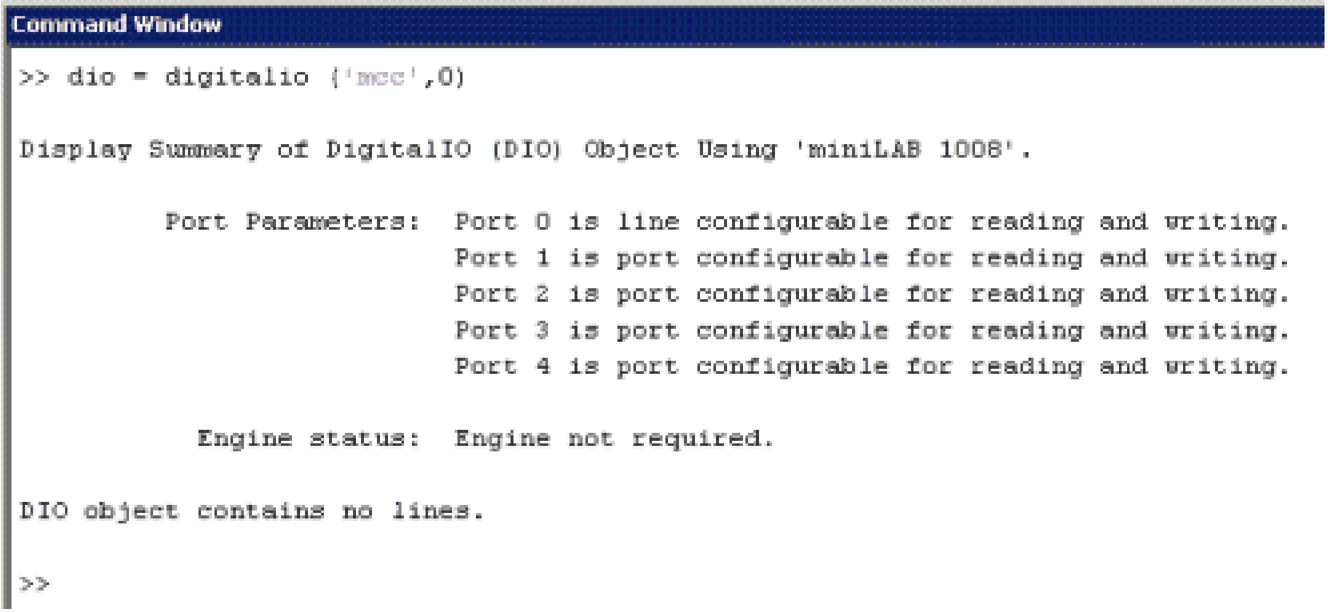
   In this example, the miniLAB 1008 is assigned Board# 0.

   Perform the following steps to control the miniLAB 1008 device's digital ports.

2. Run MATLAB and enter `dio = digitalio (‘mcc’,0)` at the >> prompt in the **Command Window**.

   This command creates a DIO object using board# 0 — the miniLAB 1008 — as an MCC-type device in MATLAB.

   After you enter this command, the screen updates with miniLAB 1008 device information:

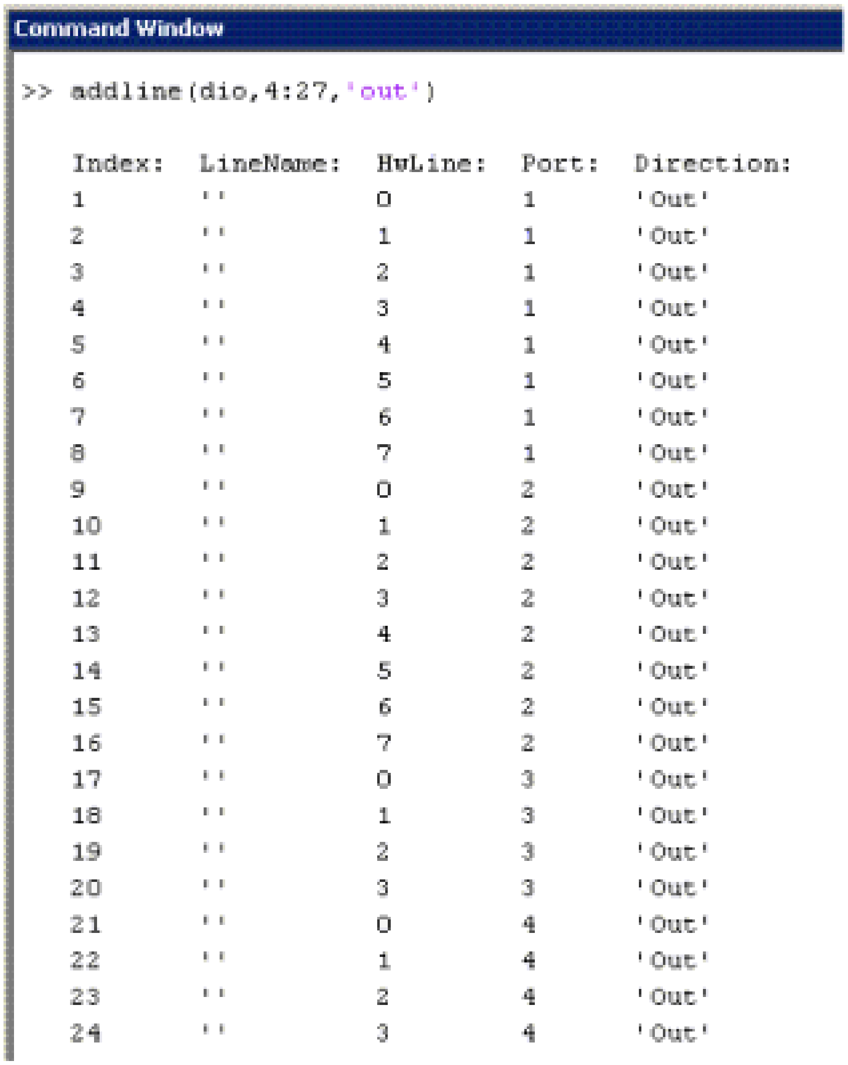


```
>> dio = digitalio ('mcc',0)

Display Summary of DigitalIO (DIO) Object Using 'miniLAB 1008'.

         Port Parameters:  Port 0 is line configurable for reading and writing.
                           Port 1 is port configurable for reading and writing.
                           Port 2 is port configurable for reading and writing.
                           Port 3 is port configurable for reading and writing.
                           Port 4 is port configurable for reading and writing.

           Engine status:  Engine not required.

DIO object contains no lines.

>>
```

3. At the >> command prompt, enter `addline(dio,4:27, 'out')`. The Command Window updates with the port information.

This command adds digital I/O lines and configures them for output (‘out’).

The `addline` method is zero-based — a device with 28 bits is written as 0 to 27. The miniLAB 1008 has 28 digital I/O lines: four DIO channels are on the top screw terminals (channels 0 to 3), and 24 DIO channels are on the 37-pin D connector (channels 4 to 27).



```
>> addline(dio,4:27,'out')

   Index:  LineName:  HwLine:  Port:  Direction:
   1       ''         0        1      'Out'
   2       ''         1        1      'Out'
   3       ''         2        1      'Out'
   4       ''         3        1      'Out'
   5       ''         4        1      'Out'
   6       ''         5        1      'Out'
   7       ''         6        1      'Out'
   8       ''         7        1      'Out'
   9       ''         0        2      'Out'
   10      ''         1        2      'Out'
   11      ''         2        2      'Out'
   12      ''         3        2      'Out'
   13      ''         4        2      'Out'
   14      ''         5        2      'Out'
   15      ''         6        2      'Out'
   16      ''         7        2      'Out'
   17      ''         0        3      'Out'
   18      ''         1        3      'Out'
   19      ''         2        3      'Out'
   20      ''         3        3      'Out'
   21      ''         0        4      'Out'
   22      ''         1        4      'Out'
   23      ''         2        4      'Out'
   24      ''         3        4      'Out'
```

On this window, indices 1 to 8 are FirstPortA, indices 9 to 16 are FirstPortB, indices 17 to 20 are FirstPortCL, and indices 21 to 24 are FirstPortCH.

4. Use the `putvalue` command to set the logic state of a bit, port, or any number of bits you choose to control.

   The example commands below control FirstPortA, bit 0.

   The `putvalue` command is one-based — a device with 28 bits is written as 1 to 28.

- To turn FirstPortA bit 0 *on*, enter, `putvalue(dio.line(1),1)`.

  FirstPortA bit 0 is index number 1.

- To turn FirstPortA bit 0 *off*, enter `putvalue(dio.line(1),0)`.
- To turn all 8 bits contained in FirstPortA *on*, enter `putvalue(dio.line(1:8), 255)`.
- To turn all 8 bits contained in FirstPortA *off*, enter `putvalue(dio.line(1:8), 0)`.

You control the other ports in the same manner.

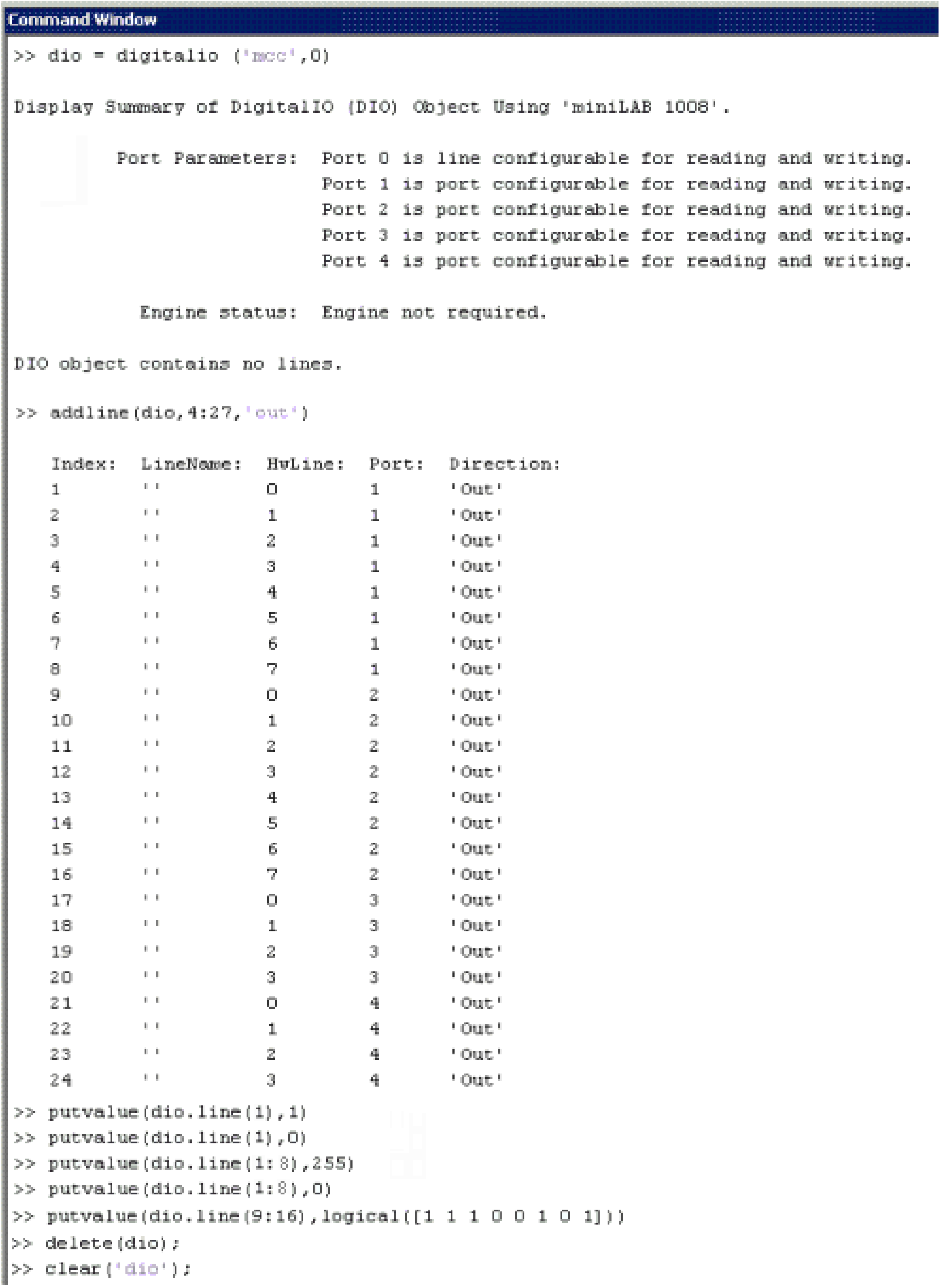
- To control the individual bits contained in FirstPortB, but still update the entire port at once, enter `putvalue(dio.line(9:16),logical([111000101]))`

  Remember, indices 9 to 16 are FirstPortB.

5. When you are done controlling the digital channels, enter the following commands at the >> prompt to remove the DIO object from memory and from the MATLAB workspace.

   - `delete(dio);`
   - `clear('dio');`

The full MATLAB **Command Window** display is shown here.



```
>> dio = digitalio ('mcc',0)

Display Summary of DigitalIO (DIO) Object Using 'miniLAB 1008'.

         Port Parameters:  Port 0 is line configurable for reading and writing.
                           Port 1 is port configurable for reading and writing.
                           Port 2 is port configurable for reading and writing.
                           Port 3 is port configurable for reading and writing.
                           Port 4 is port configurable for reading and writing.

           Engine status:  Engine not required.

DIO object contains no lines.

>> addline(dio,4:27,'out')

   Index:  LineName:  HwLine:  Port:  Direction:
   1       ''         0        1      'Out'
   2       ''         1        1      'Out'
   3       ''         2        1      'Out'
   4       ''         3        1      'Out'
   5       ''         4        1      'Out'
   6       ''         5        1      'Out'
   7       ''         6        1      'Out'
   8       ''         7        1      'Out'
   9       ''         0        2      'Out'
   10      ''         1        2      'Out'
   11      ''         2        2      'Out'
   12      ''         3        2      'Out'
   13      ''         4        2      'Out'
   14      ''         5        2      'Out'
   15      ''         6        2      'Out'
   16      ''         7        2      'Out'
   17      ''         0        3      'Out'
   18      ''         1        3      'Out'
   19      ''         2        3      'Out'
   20      ''         3        3      'Out'
   21      ''         0        4      'Out'
   22      ''         1        4      'Out'
   23      ''         2        4      'Out'
   24      ''         3        4      'Out'
>> putvalue(dio.line(1),1)
>> putvalue(dio.line(1),0)
>> putvalue(dio.line(1:8),255)
>> putvalue(dio.line(1:8),0)
>> putvalue(dio.line(9:16),logical([1 1 1 0 0 1 0 1]))
>> delete(dio);
>> clear('dio');
```

For more information on digital I/O operations with the MATLAB Data Acquisition Toolbox, refer to Chapter 7 of the *Data Acquisition Toolbox User's Guide* (*Version 2*) at www.mathworks.com/access/helpdesk/help/pdf_doc/daq/daqug.pdf.