Keywords:

☐ In stock

☐ Lead free

☐ RoHS Compliant

Search Again

Gold Through Hole Rectangular - Headers, Male Pin Connectors, Interconnect Molex Connector Corporation Header, Shrouded 87831-1220

CONN HEADER 12POS 2MM VERT GOLD - 87831-1220 - Connectors, Interconnects

Digi-Key Part Number	WM18561-ND
Manufacturer Part Number	**87831-1220**
Description	CONN HEADER 12POS 2MM VERT GOLD
Quantity Available	48

Price Break	Unit Price	Extended Price
1	1.61000	1.61
10	1.42400	14.24
100	.94950	94.95
500	.75960	379.80
1000	.63047	630.47

All prices are in USD

Purchase Gold Through Hole Rectangular - Headers, Male Pin Connectors, Interconnect Molex Connector Corporation Header, Shrouded 87831-1220

Quantity	Item Number	Customer Reference	
	WM18561-ND		Add to Order

When requested quantity exceeds displayed pricing table quantities, a lesser unit price may appear on your order. You may submit a request for quotation on quantities which are greater than those displayed in the pricing table.

Specifications for Gold Through Hole Rectangular - Headers, Male Pin Connectors, Interconnect Molex Connector Corporation Header, Shrouded 87831-1220

Technical/Catalog Information	87831-1220
Vendor	Molex Connector Corporation
Category	Connectors, Interconnects
Color	Black
Connector Type	Header, Shrouded
Contact Finish	Gold
Contact Mating Length	0.141" (3.60mm)
Features	-
Mounting Type	Through Hole
Number of Positions Loaded	All
Number of Rows	2
Pitch	0.079" (2.00mm)
Row Spacing	0.079" (2.00mm)
Packaging	Tube
Fastening Type	-
Termination	Solder
Contact Finish Thickness	15µin (0.38µm)
Number of Positions	12
Contact Type	Male Pin
Lead Free Status	Lead Free
RoHS Status	RoHS Compliant
Other Names	87831 1220 878311220 WM18561 ND

	WM18561ND WM18561

Photo Not Available

CONN HEADER 12POS 2MM VERT GOLD - 87831-1220 (WM18561-ND) - Connectors, Interconnects

Related parts Gold Through Hole Rectangular - Headers, Male Pin Connectors, Interconnect Molex Connector Corporation Header, Shrouded 87831-1220

Gold Through Hole Rectangular - Headers, Male Pin Connectors, Interconnect Molex Connector Corporation Header, Shrouded 87831-1220
Order 'conn header 12pos 2mm vert gold - 87831-1220' online from Digi-Key. Manufactured by Molex Connector Corporation. Digi-Key part number WM18561-ND.

Gold Through Hole Rectangular - Headers, Male Pin Connectors, Interconnect Molex Connector Corporation Header, Shrouded 87331-1220
Order 'conn header 12pos 2mm vert gold - 87331-1220' online from Digi-Key. Manufactured by Molex Connector Corporation. Digi-Key part number WM18062-ND.

Gold Through Hole Rectangular - Headers, Male Pin Connectors, Interconnect Molex Connector Corporation Header, Shrouded 87831-3420
Order 'conn header 34pos 2mm vert gold - 87831-3420' online from Digi-Key. Manufactured by Molex Connector Corporation. Digi-Key part number 87831-3420-ND.

Gold Through Hole Rectangular - Headers, Male Pin Connectors, Interconnect Molex Connector Corporation Header, Shrouded 87831-2420
Order 'conn header 24pos 2mm vert gold - 87831-2420' online from Digi-Key. Manufactured by Molex Connector Corporation. Digi-Key part number WM17470-ND.

Gold Surface Mount Rectangular - Headers, Male Pin Connectors, Interconnect Molex Connector Corporation Header, Shrouded 87332-1220
Order 'conn header 12pos 2mm vert smd - 87332-1220' online from Digi-Key. Manufactured by Molex Connector Corporation. Digi-Key part number WM18077-ND.

Gold Through Hole Rectangular - Headers, Male Pin Connectors, Interconnect Molex Connector Corporation Header, Shrouded 87831-0420
Order 'conn header 4pos 2mm vert gold - 87831-0420' online from Digi-Key. Manufactured by Molex Connector Corporation. Digi-Key part number WM18557-ND.

Gold Through Hole Rectangular - Headers, Male Pin Connectors, Interconnect Molex Connector Corporation Header, Shrouded 87831-0620
Order 'conn header 6pos 2mm vert gold - 87831-0620' online from Digi-Key. Manufactured by Molex Connector Corporation. Digi-Key part number WM18558-ND.

Gold Through Hole Rectangular - Headers, Male Pin Connectors, Interconnect Molex Connector Corporation Header, Shrouded 87831-0820
Order 'conn header 8pos 2mm vert gold - 87831-0820' online from Digi-Key. Manufactured by Molex Connector Corporation. Digi-Key part number WM18559-ND.

Gold Through Hole Rectangular - Headers, Male Pin Connectors, Interconnect Molex Connector Corporation Header, Shrouded 87831-1020
Order 'conn header 10pos 2mm vert gold - 87831-1020' online from Digi-Key. Manufactured by Molex Connector Corporation. Digi-Key part number WM18560-ND.

Gold Through Hole Rectangular - Headers, Male Pin Connectors, Interconnect Molex Connector Corporation Header, Shrouded 87831-1620
Order 'conn header 16pos 2mm vert gold - 87831-1620' online from Digi-Key. Manufactured by Molex Connector Corporation. Digi-Key part number WM18562-ND.

Gold Through Hole Rectangular - Headers, Male Pin Connectors, Interconnect Molex Connector Corporation Header, Shrouded 87831-1820

Order 'conn header 18pos 2mm vert gold - 87831-1820' online from Digi-Key. Manufactured by Molex Connector Corporation. Digi-Key part number WM18563-ND.

Gold Through Hole Rectangular - Headers, Male Pin Connectors, Interconnect Molex Connector Corporation Header, Shrouded 87831-2020

Order 'conn header 20pos 2mm vert gold - 87831-2020' online from Digi-Key. Manufactured by Molex Connector Corporation. Digi-Key part number WM18564-ND.

Gold Through Hole Rectangular - Headers, Male Pin Connectors, Interconnect Molex Connector Corporation Header, Shrouded 87831-3020

Order 'conn header 30pos 2mm vert gold - 87831-3020' online from Digi-Key. Manufactured by Molex Connector Corporation. Digi-Key part number WM18565-ND.

Gold Through Hole Rectangular - Headers, Male Pin Connectors, Interconnect Molex Connector Corporation Header, Shrouded 87831-4020

Order 'conn header 40pos 2mm vert gold - 87831-4020' online from Digi-Key. Manufactured by Molex Connector Corporation. Digi-Key part number WM18566-ND.

Gold Through Hole Rectangular - Headers, Male Pin Connectors, Interconnect Molex Connector Corporation Header, Shrouded 87831-4420

Order 'conn header 44pos 2mm vert gold - 87831-4420' online from Digi-Key. Manufactured by Molex Connector Corporation. Digi-Key part number WM18567-ND.

Options for Gold Through Hole Rectangular - Headers, Male Pin Connectors, Interconnect Molex Connector Corporation Header, Shrouded 87831-1220

Send this link via E-Mail
Help With This Screen

CONN HEADER 12POS 2MM VERT GOLD - 87831-1220

Electronic parts by: **Component Title Manufacturer Category Manufacturer Part No.**
Part categories by: **Name Manufacturer**
Manufacturers by: **Name Number of Parts**
Products by: **Category Title**
Grouped parts by: **Title Model Number**
Popular products by: **Category Title**
All parts by: **Title Manufacturer Category Manufacturer Part No.**
Previous: CONN HEADER 12POS 2MM VERT GOLD - 87758-1216 Next: CONN HEADER 12POS 2MM VERT GOLD - DF3-12P-2DSA(01)
CONN HEADER 12POS 2MM VERT GOLD - 87831-1220

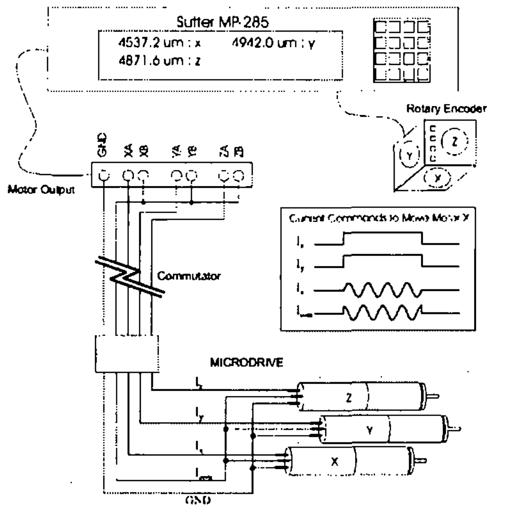


Figure A.3. Schematic of motor connections to the modified Sutter MP-285 controller. Each motor has three connections. These connections are normally driven by three sinusoidal voltage inputs with 120° phase shift. To reduce the number of connections required to drive the three motors, all motors were provided with one common ground and a common sinusoidal input current (I_{com}). The third input was controlled independently for each motor. The Sutter manipulator controller was modified to apply a sinusoidal input current (with a 90° phase shift) only to the motor being driven (e.g., I_x), and a constant bias current to the other inputs (e.g., I_y and I_z) to prevent uncommanded movement of the other motors.

addition to those required for recording neural signals: analog ground, + Vcc for the headstage preamplifier, and the four neural signals (three electrodes and a differential ground reference). In the alternate approach, the motors are driven by two sinusoidal current inputs, one at 0° and one at 90°, as a stepper motor is usually driven. The third connection on each motor is connected to analog ground. The wiring is reduced because all motors share one of these sinusoidal signals (e.g., the 0° signal), referred to as I_{com}, so that when any one motor is 'on', the 'off' motors also have one winding energized (Figure A.3). The second (i.e., 90°) current input is applied only to the motor selected to be 'on'. The 'off' motors are unlikely to turn with only one energized winding, but to prevent any possible spurious rotation, a constant DC current is applied to non-energized winding of the two 'off' motors, locking them in place.

The motor control was implemented using a modified commercial manipulator controller (Sutter Instruments, MP-285). As originally designed, the MP-285 is used to control a stepper-motor-driven, three-axis manipulator. Manipulator movements are computer-controlled in response to commands from a cluster of three rotary-encoded wheels. The embedded computer also keeps track of the current position of the manipulator axes and turns off power to the motors after some delay (t_i) during periods of inactivity. A serial port output allows the depth of the electrodes to be logged by custom-designed computer control software used to record the neural data. These features make the MP-285 well suited to the control of the three-motor microdrive, with each axis controlling one electrode and motor.

Several modifications of the MP-285 were required. Modifications of the controller firmware were kindly provided by Sutter Instruments (Joe Immel, personal communication). One modification was a re-calibration of position display to reflect the electrode displacement per motor cycle of the motorized microdrive (which is different from the original manipulator). Another modification allows t_i to be user programmed. This delay is set short (<0.5 sec) so that high values of drive current may be used for transient motor movements without thermally overloading the motor.

A simple circuit was added internally to the MP-285 controller to detect command input from the rotary encoder and then perform two functions. First, since analog ground is used also for motor ground, the circuit connects the analog ground line to the controller power supply ground when any motor is activated. When the motors are not in use, the ground is automatically disconnected from the controller power supply to eliminate noise on the electrode signals. Second, the circuit applies the bias 'locking' current to the motors that are not in use. For example, if command input to the x-axis is detected, bias current is applied to the y- and z-axis motors. This design results in the constraint that only one electrode may be moved at a time. Circuit details are available from Sutter Instruments.

Microdrive reproducibility was limited on occasions in which the motors would briefly stall; this problem was minimized by careful attention to the construction process. However, it should be noted that since the microdrive control via the MP-285 is open-loop, any stalling of the motors will produce errors in the estimate of the electrodes depth. The motorized microdrive was found to be quite robust. Across all implants done to date, none of the motors appeared to suffer damage inflicted by the bird. In addition, because the motor subassembly detaches from the microdrive body, the process of reusing the microdrive after an experiment is straightforward. The motor subassembly is detached, the microdrive body is cleaned of acrylic, the motor unit is reattached and the drive is reloaded with electrodes to prepare for the next experiment. The motor or gearboxes occasionally fail for unknown reasons and must be replaced during the reconstruction process.

MP-285 Interface Improvements

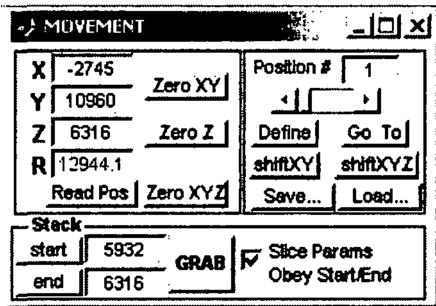
Tools

Added by Vijay Iyer, last edited by Vijay Iyer on Dec 15, 2008 ()

ScanImage control of X/Y & Z position, of either the objective or specimen, requires use of a Sutter Instruments MP-285 controller. Several improvements to the MP-285 interface have been added in ScanImage 3.5:

- The 'calibration' of the X/Y & Z motions has been improved. When custom stages and/or stepper motors are used, the user modifies physically intuitive *umPerStepX/Y/Z* variables which reside in the INI File Rig Configuration section.
- The *# Slices* and *Z step/slice* values in the Standard Controls window can now be constrained to the stack start/end-points set in the **Movement** window (if both are in factset). This new option is active by default.
- MP-285 error conditions are now uniformly handled. A simple procedure allows recovery from most errors.

Movement



New Controls

- **Stack Start/End**: These read-only display controls allow Z positions set via *start* and *end* buttons to be seen, and stored even as the current Z position may be updated.
- **Slice Params Obey Start/End**: If selected, when either one of the *# slices* or *Z step/slice* is adjusted in the Standard Controls window, the other parameter will auto-adjust, so the start/endpoints set here remain in effect. For example, the user can vary the *Z step/slice* and the *# slices* will auto-adjust based on the distance between the stack start/end. If the checkbox is deselected, any change to the *Z step/slice* or *# slices* will clear the stack start/endpoints.

Modified Controls

- **GRAB**: This control, which initiates a GRAB acquisition consisting of a Z stack between the displayed stack start/endpoints, is now inactive unless both a stack start and end have been entered.

INI File *Rig Configuration* Changes

View *Rig Configuration* Section of Model INI File

New Variables

- **umPerStepX/Y/Z**: Specifies the distance traveled per motor *step*, which also comprises the effective position resolution of the stage/motor system. The default values (2.0) match the Sutter stages which come together with the MP-285. If a custom stage and/or stepper motor is used in any dimension, then this value should be adjusted to match the stage/stepper system for that dimension.
  See Supported Devices (in Appendix 2) for more details about *umPerStep* values for typical ScanImage hardware configurations and how, generally, to determine this value.
  The MP-285 firmware encodes an assumed *um/step* value in each dimension, which should ideally match the stage/stepper to ensure the controller displays accurate values.
  See Supported Devices (in Appendix 2) for more details.
- **posnResolution**: ScanImage validates the motor end position following all move operations, except for those in the midst of a stack acquisition. Infrequently, the end position will not exactly match the specified position. This value specifies, in microns, the smallest difference that will be tolerated without alerting the user.

Modified Variables

These variables are not generally modified by typical users.

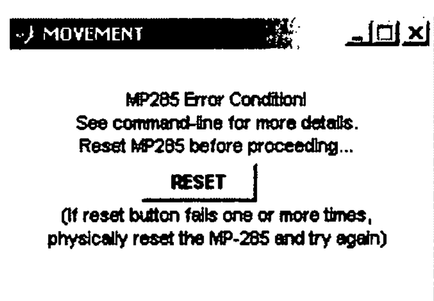
- **velocityFast/Slow**: (Previously below Rig Configuration section) These values specify the 'velocity' of the motor movement employed for long moves (fast) and short moves (slow). Movements during stack acquisitions always follow *velocitySlow*. Other movements (e.g. entering a new position in one of the X/Y/Z controls) use *velocityFast* for moves larger than 100um to get within 100um of the final position. The movement always finish using *velocitySlow* to reach the final position, which helps ensure the correct position is reached. The *velocityFast* value is roughly in um/second, whereas *velocitySlow* is in units about half of this.
  The precise calibration of these 'velocities' is not fully understood and is known to be somewhat nonlinear. The default values in *standard_model.INI* (2000 and 400, respectively) have been found empirically to be satisfactory.
  Values for *velocityFast* and *velocitySlow* must not exceed 6500 and 1300, respectively.

Removed Variables

- **calibrationFactorX/Y/Z**: (Previously below Rig Configuration section) These values are replaced by the new *umPerStepX/Y/Z* variables described above.

Error Handling

If an error condition is encountered during MP-285 operation, including timeout errors, the **Movement** window will clearly indicate the error condition:



Hitting the *RESET* button will attempt to restore communication with the MP-285 using serial commands sent to the MP-285 (the 'Interrupt' command is sent multiple times). Following that, the X,Y, and Z positions are read and updated. This ensures that future motor commands pertain to the correct actual positions.

If the *RESET* operation is unsuccessful, the user is advised to press the physical Reset button on the MP-285 controller. After this, pressing the software *RESET* button in the **Movement** window should operate correctly.

When the MP-285 is in an error condition, ScanImage operation can continue, but motor operations will not be possible.

Labels

(None)

Subject: Re: Old MP-285 manuals
From: "Rick Ayer" <rick@sutter.com>
Date: Fri, March 5, 2010 9:14 am
To: "Diana Jeong" <diajeong@physics.ucsd.edu>
Cc: ali@sutter.com
"Rick Ayer" <rick@sutter.com>
"Yungui Tang" <yungui@sutter.com>

Diana

Were you aware of this "oldness" when you were asking about the extra characters before? I don't have any actual knowledge of whether or not there were such extra characters in the older version of the microprocessor which would be in this unit, ***but it would not surprise me if there was a slightly different response, or extra characters.*** This should say version 2.762 if you go to the info screen (press PGRM, tab to Setup, press ENTR, tab to Utilities, press ENTR, tab to Info, press ENTR and read the version number and date. The newer controllers in the lab will say 3.04, this one will say 2.762 or could even be older in which case we will know nothing.

I can possibly dig out the old info on the interface, ***I know that it used a different velocity command, small v instead of capital V.*** There was no resolution bit in the v command. this controller will probably choke if you give it a V instead of a v

It would be far easier to send the new microprocessor and you can see if that makes the problems go away.

Where did you get this controller, I was unaware that the Kleinfeld lab had any that old....6969 was sold in 1996 or 97. The mechanical you have did not ship with it. That is considerably newer.

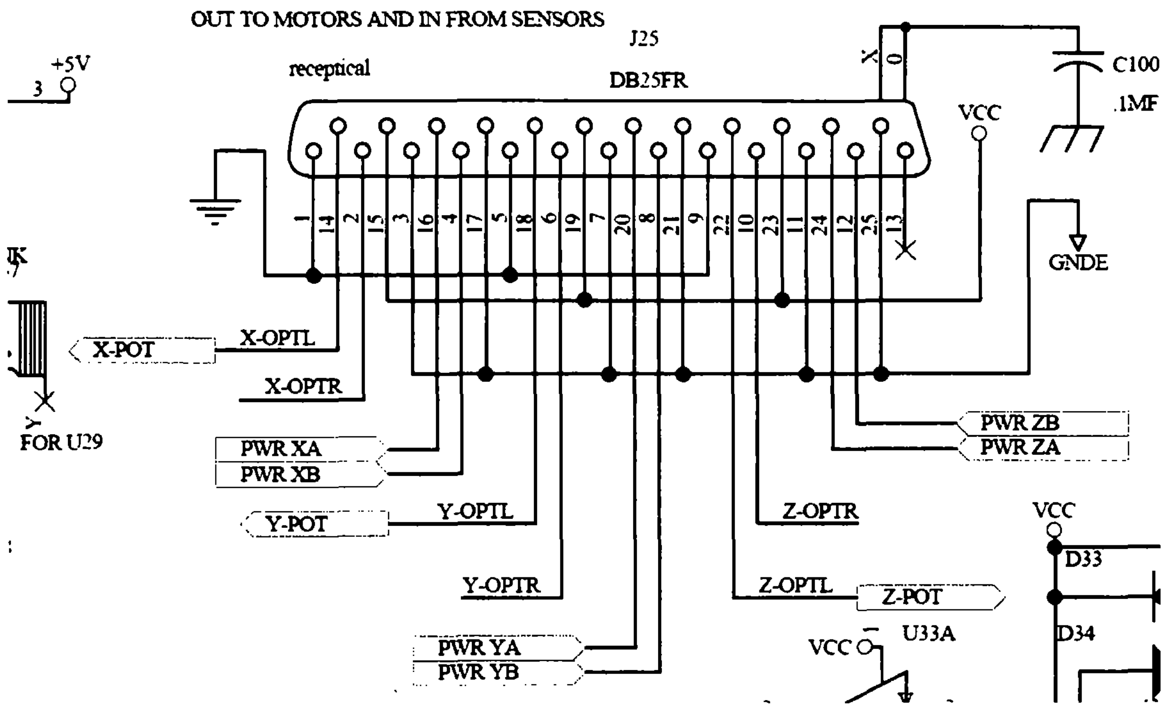
The electronics, save the microprocessor code are functionally identical in what you have compared to what we sell now. The pin outs on the manipulator connector are the same. Presumably this controller works fine when you try to move the manipulator using the ROE? So I am not sure why you would need the pin outs on the connector, but the are in the attachment

Rick

Diana Jeong wrote:
> Hello,
>
> I currently have "old" MP-285 controller, probably from back in mid-90's.
> I see that the new version of manuals do not suit my controller well (in
> particular, in serial communication, I get 0x00 instead of carriage return
> as a terminating character). Could you send me the old manual, if

> possible? The serial number is 6969.
>
> And the manipulator I have has serial number of Mp-285 M-19625. Could you
> identify the pin-out and motor type for the manipulator?
>
> Thank you,
> Diana Jeong

LOOKING INTO DB-25(S)



ANDY'S RIG: SUTTER MP285 / M-1962S (WORM DRIVE)

Z MOTOR ID: T960705

1-5 N.C.	19-23 N.C.
1-9 N.C.	17-7 N.C.
	17-21 N.C.
~~4-16~~	7-21 N.C.

DOES <u>NOT</u> AGREE WITH DWGS!

To Allan
(Sutter Stages P/N out)

X-axis

Pin 3 ground

Pin 4 Power XB

Pin 16 Power XA

Pin 17 ground

Y-Axis

Pin 7 ground

Pin 8 Power YB

Pin 20 Power YA

Pin 21 ground

Z-Axis

Pin 11 ground

Pin 12 Power ZB

Pin 24 Power ZA

Pin 25 ground

End of travel switch data in is called OPTL and OPTR for optical left and right respectively. It is not clear without trial and error, which switch will be the right and left end of travel for your particular design. There are X Y and Z inputs for OPT, for eg, Pins 14 and 2 are XOPTL and XOPTR respectively. Also provided for the EOT switches is +5VDC at pins 15,19 and 23 and digital ground at pins 1,5,9.

Hi Allan,

I have TA section until noon - should be in the lab by 1pm tomorrow. We can call Rick and find out what is going on, if it is OK with you. I am trying to setup fiber optics alignment right now, and I didn't have much time to look at the sutter code. See you tomorrow.

Date: Tue, March 2, 2010 9:22 pm

rick@sutter.com nd yungui@sutter.com (Rick Ayer)

Regards to the MP285. First, the MP285 programmer has not worked at Sutter for close to 10 years, so exact info may be hard to come by.

Yungui Tang, cc'd above wrote the PC-GUI, so he can help a bit

>>1. I am wondering if the 255 (0xFF) stands for any error message by
>any chance.

I think the only error codes sent out are those listed in the computer inteface document and I think they are also sent to the screen of the MP285, so that is why I asked if you were getting errors on the MP285 display as well

>>2. Also when sending out "move" command through my computer, when
>does CR given out by the Sutter- is it right after it receives the command
>or after it finishes the move?

I am pretty sure it is after the end of the move

>3. Say my stage is at (0,0,0), if I give command to move to (10, 10, 0)
>>would it move in diagonal fashion or would it move in sequential
>fashion i.e., y move after x move.

Full 3D point to point straight line move. That is one of the unique features of the MP285 controller.

>>Also, when is a convenient time to call you tomorrow afternoon? Let

>me know then I will give you a call, if possible.

I will be around all day tomorrow. I am in a meeting from 9:30 to

10:30 pacific time, but should be free otherwise

I assume that 285.exe is working correctly for you?

Rick

>>Thanks,

>>Diana

Getting Started with Serial I/O

On this page...

Example: Getting Started

This example illustrates some basic serial port commands.

If you have a device connected to the serial port COM1 and configured for a baud rate of 4800, execute the following example.

```
s = serial('COM1');
set(s,'BaudRate',4800);
fopen(s);
fprintf(s,'*IDN?')
out = fscanf(s);
fclose(s)
delete(s)
clear s
```

The `*IDN?` command queries the device for identification information, which is returned to `out`. If your device does not support this command, or if it is connected to a different serial port, modify the previous example accordingly.

> **Note** `*IDN?` is one of the commands supported by the Standard Commands for Programmable Instruments (SCPI) language, which is used by many modern devices. Refer to your device documentation to see if it supports the SCPI language.

Back to Top

The Serial Port Session

This example describes the steps you use to perform any serial port task from beginning to end.

The serial port *session* comprises all the steps you are likely to take when communicating with a device connected to a serial port. These steps are:

1. Create a serial port object — Create a serial port object for a specific serial port using the `serial` creation function.

   Configure properties during object creation if necessary. In particular, you might want to configure properties associated with serial port communications such as the baud rate, the number of data bits, and so on.
2. Connect to the device — Connect the serial port object to the device using the `fopen` function.

   After the object is connected, alter the necessary device settings by configuring property values, read data, and write data.
3. Configure properties — To establish the desired serial port object behavior, assign values to properties using the `set` function or dot notation.

   In practice, you can configure many of the properties at any time including during, or just after, object creation. Conversely, depending on your device settings and the requirements of your serial port application, you might be able to accept the default property values and skip this step.

4. Write and read data — Write data to the device using the `fprintf` or `fwrite` function, and read data from the device using the `fgetl`, `fgets`, `fread`, `fscanf`, or `readasync` function.

   The serial port object behaves according to the previously configured or default property values.
5. Disconnect and clean up — When you no longer need the serial port object, disconnect it from the device using the `fclose` function, remove it from memory using the `delete` function, and remove it from the MATLAB® workspace using the `clear` command.

The serial port session is reinforced in many of the serial port documentation examples. To see a basic example that uses the steps shown above, see Example: Getting Started.

Back to Top

Configuring and Returning Properties

This example describes how you display serial port property names and property values, and how you assign values to properties.

You establish the desired serial port object behavior by configuring property values. You can display or configure property values using the `set` function, the `get` function, or dot notation.

Displaying Property Names and Property Values

After you create the serial port object, use the `set` function to display all the configurable properties to the command line. Additionally, if a property has a finite set of string values, `set` also displays these values.

```
s = serial('COM1');
set(s)
    ByteOrder: [ {littleEndian} | bigEndian ]
    BytesAvailableFcn
    BytesAvailableFcnCount
    BytesAvailableFcnMode: [ {terminator} | byte ]
    ErrorFcn
    InputBufferSize
    Name
    OutputBufferSize
    OutputEmptyFcn
    RecordDetail: [ {compact} | verbose ]
    RecordMode: [ {overwrite} | append | index ]
    RecordName
    Tag
    Timeout
    TimerFcn
    TimerPeriod
    UserData

    SERIAL specific properties:
    BaudRate
    BreakInterruptFcn
    DataBits
    DataTerminalReady: [ {on} | off ]
    FlowControl: [ {none} | hardware | software ]
    Parity: [ {none} | odd | even | mark | space ]
    PinStatusFcn
    Port
    ReadAsyncMode: [ {continuous} | manual ]
    RequestToSend: [ {on} | off ]
    StopBits
    Terminator
```

Use the `get` function to display one or more properties and their current values to the command line. To display all properties and their current values:

```
get(s)
    ByteOrder = littleEndian
    BytesAvailable = 0
```

```
        BytesAvailableFcn =
        BytesAvailableFcnCount = 48
        BytesAvailableFcnMode = terminator
        BytesToOutput = 0
        ErrorFcn =
        InputBufferSize = 512
        Name = Serial-COM1
        OutputBufferSize = 512
        OutputEmptyFcn =
        RecordDetail = compact
        RecordMode = overwrite
        RecordName = record.txt
        RecordStatus = off
        Status = closed
        Tag =
        Timeout = 10
        TimerFcn =
        TimerPeriod = 1
        TransferStatus = idle
        Type = serial
        UserData = []
        ValuesReceived = 0
        ValuesSent = 0

        SERIAL specific properties:
        BaudRate = 9600
        BreakInterruptFcn =
        DataBits = 8
        DataTerminalReady = on
        FlowControl = none
        Parity = none
        PinStatus = [1x1 struct]
        PinStatusFcn =
        Port = COM1
        ReadAsyncMode = continuous
        RequestToSend = on
        StopBits = 1
        Terminator = LF
```

To display the current value for one property, supply the property name to `get`.

```
get(s,'OutputBufferSize')
ans =
   512
```

To display the current values for multiple properties, include the property names as elements of a cell array.

```
get(s,{'Parity','TransferStatus'})
ans =
    'none'    'idle'
```

Use the dot notation to display a single property value.

```
s.Parity
ans =
none
```

Configuring Property Values

You can configure property values using the `set` function:

```
set(s,'BaudRate',4800);
```

or the dot notation:

```
s.BaudRate = 4800;
```

To configure values for multiple properties, supply multiple property name/property value pairs to `set`.

```
set(s,'DataBits',7,'Name','Test1-serial')
```

Note that you can configure only one property value at a time using the dot notation.

In practice, you can configure many of the properties at any time while the serial port object exists — including during object creation. However, some properties are not configurable while the object is connected to the device or when recording information to disk. For information about when a property is configurable, see Property Reference.

Specifying Property Names

Serial port property names are presented using mixed case. While this makes property names easier to read, use any case you want when specifying property names. Additionally, you need use only enough letters to identify the property name uniquely, so you can abbreviate most property names. For example, you can configure the `BaudRate` property any of these ways:

```
set(s,'BaudRate',4800)
set(s,'baudrate',4800)
set(s,'BAUD',4800)
```

When you include property names in an M-file, you should use the full property name. This practice can prevent problems with future releases of MATLAB software if a shortened name is no longer unique because of the addition of new properties.

Default Property Values

Whenever you do not explicitly define a value for a property, the default value is used. All configurable properties have default values.

> **Note** Your operating system provides default values for all serial port settings such as the baud rate. However, these settings are overridden by your MATLAB code and have no effect on your serial port application.

If a property has a finite set of string values, the default value is enclosed by { }. For example, the default value for the `Parity` property is `none`.

```
set(s,'Parity')
[ {none} | odd | even | mark | space ]
```

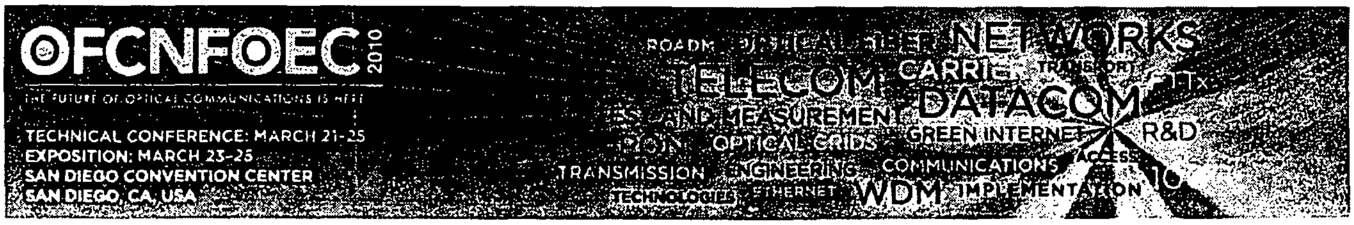
You can find the default value for any property in the property reference pages.

Back to Top

Provide feedback about this page

Overview of the Serial Port Creating a Serial Port Object

Search



Short Course Schedule by Topic Category

OFC Categories

Category A. Fibers and Optical Propagation Effects
Category C. Optical Devices for Switching, Filtering and Signal Compensation
Category D. Optoelectronic Devices
Category E. Digital Transmission Systems
Category F. Transmission Subsystems and Network Elements
Category G. Optical Processing and Analog Subsystems
Category H. Core Networks
Category I. Access Networks
Category J. Network Experiments and Non-Telecom Applications

NFOEC Categories

NFOEC 1: Optical Networks and Services
NFOEC 2: Network Technologies
Additional Short Course Category: Industry Best Practices

View Schedule by Time

Category A. Fibers and Optical Propagation Effects

SC186 **Hands-on Specialty Fiber Splicing**, *Clyde J. Troutman; 3SAE Technologies, USA*
SC208 **Optical Fiber Design for Telecommunications and Specialty Applications**, *David J. DiGiovanni; OFS Labs, USA*
SC210 **Hands-on Polarization-Related Measurements Workshop**, *Danny Peterson[1], Tasshi Dennis[2], Ivan T. Lima Jr.[3], Paul Williams[2]; [1]Verizon Business, USA, [2]NIST, USA, [3]North Dakota State Univ., USA*
SC288 **Fundamentals of Polarization, PMD and PDL in Lightwave Systems**, *Robert Jopson; Bell Labs, Alcatel-Lucent, USA*

Category C. Optical Devices for Switching, Filtering and Signal Compensation

SC265 **Passive Optical Components and Filtering Technologies**, *Bruce Nyman[1], Christi Madsen[2]; [1]Consultant, USA, [2]Texas A&M Univ., USA*

Category D. Optoelectronic Devices

Watch Chairs Discuss Photonic Integration

Click here to download plugin.

SC175 **Packaging of Optoelectronic, Photonic and MEMS Components**, *Tolga Tekin[1,2]; [1]Technische Univ. Berlin, Germany, [2]Fraunhofer-Inst. fuer Zuverlaessigkeit und Mikrointegration (IZM) System Integration and Interconnection Technologies, Germany*
SC177 **High-Speed Semiconductor Lasers and Modulators**, *John Bowers; Univ. of California at Santa Barbara, USA*
SC178 **Test and Measurement of High-Speed Communications Signals**, *Greg LeCheminant; Agilent Technologies, USA*
SC182 **Biomedical Optical Diagnostics and Sensing**, *Thomas Huser; Univ. of California at Davis, USA*
SC267 **Silicon Microphotonics: Technology Elements and the Roadmap to Implementation**, *Lionel Kimerling; MIT, USA*
SC325 **Semiconductor Photonic Integrated Circuits**, *Chris Doerr; Bell Labs, Alcatel-Lucent, USA*

Category E. Digital Transmission Systems

SC102 **WDM in Long-Haul Transmission Systems**, *Neal S. Bergano; Tyco Telecommunications, USA*
SC203 **40/100 Gb/s Transmission Systems, Design and Design Trade-offs**, *Martin Birk[1], Benny Mikkelsen[2]; [1]AT&T Labs, Res., USA, [2]Acacia Communications, USA*
SC205 **Integrated Electronic Circuits for Fiber Optics**, *Y. K. Chen, Noriaki Kaneda; Bell Labs, Alcatel-Lucent, USA*
SC259 **Electronic and Optical Impairment Mitigation**, *Chris Fludger[1], Seb Savory[2]; [1]CoreOptics GmbH, Germany, [2]Univ. College London, UK*
SC289 **Basics of Optical Communication Systems and WDM**, *Gerd Keiser[1,2]; [1]PhotonicsComm Solutions Inc., USA, [2]Natl. Taiwan Univ. of Science and Technology, Taiwan*
SC314 **Hands-on Fiber Characterization for the Engineering of Long Haul and Metro Deployments**, *Daniel Peterson[1], Christine Tremblay[2]; [1]Verizon, USA, [2]École de Technologie Supérieure, Univ. du Québec, Canada*
SC327 **Modeling and Design of Fiber-Optic Communication Systems**, *Rene-Jean Essiambre; Bell Labs, Alcatel-Lucent, USA*
NEW! SC342 **Digital Coherent Optical Communications**, *Maurice O'Sullivan; Nortel, Canada*

Category F. Transmission Subsystems and Network Elements

SC101 **Hands-on Workshop on Fiber Optic Measurements and Component Testing**, *Lorenz Cartellieri[1], Roger Rutz[1], Caroline Connolly[2], Richard Buerli[2]; [1]Experior Photonics, USA, [2]OptoTest, USA*
SC103 **Dynamic ROADMs, Switches and Integrated Technologies and Techniques for Dynamically Reconfigurable and Packet Switched Optical Networks**, *Daniel Blumenthal; Univ. of California at Santa Barbara, USA*
SC105 **Modulation Formats and Receiver Concepts for Optical Transmission Systems**, *Peter Winzer, Chandrasekhar Sethumadhavan; Bell Labs, Alcatel-Lucent, USA*
SC141 **Combating and Monitoring Data-Degrading Effects in Non-Static WDM Systems**, *Alan Willner; Univ. of Southern California, USA*
SC239 **Short-Reach Optical Interconnects**, *Brian E. Lemoff; West Virginia High Tech Foundation, USA*
NEW! SC341 **OFDM for Optical Communications**, *Sander L. Jansen; Nokia Siemens Networks GmbH & Co. KG, Germany*

Category G. Optical Processing and Analog Subsystems

SC160 **Microwave Photonics**, *Keith Williams; NRL, USA*
SC217 **Hybrid Fiber Radio: The Application of Photonic Links in Wireless Communications**, *Dalma Novak; Pharad, USA*

Category H. Core Networks

SC216 **An Introduction to Optical Network Design and Planning**, *Jane M. Simmons; Monarch Network Architects, USA*
SC243 **Next Generation Transport Networks: The Evolution from Circuits to Packets**, *Ori A. Gerstel; Cisco Systems, USA*
SC261 **ROADM Technologies and Network Applications**, *Thomas Strasser; Nistica Inc., USA*

Category I. Access Networks

SC114 **Passive Optical Networks (PONs)**, *Frank J. Effenberger; Huawei Technologies, USA*

Category J. Network Experiments and Non-Telecom Applications

SC187 **Hands-on Basic Fiber Optics for the Absolute Beginner**, *Dennis Horwitz; Micronor Inc., USA*
SC291 **Hands-on Fiber Optics for Engineers Designing for Military, Aerospace, Shipboard and Industrial Harsh Environmental Applications**, *Dennis Horwitz; Micronor Inc., USA*

Back to Top

NFOEC 1: Optical Networks and Services

SC171 **Introduction to Optical Control Plane Concepts, Technologies and Practices**, *Greg Bernstein; Grotto Networking, USA*
SC176 **Metro Network: The Transition to Ethernet**, *Loudon Blair; Ciena Corp., USA*
SC328 **New Developments in Optical Transport Networking (OTN)**, *Stephen Trowbridge; Alcatel-Lucent, USA*

NFOEC 2: Network Technologies

SC185 **Hands-on Polishing, Inspection and Testing of Connectors**, *Cameron Karch[1], Steve Baldo[2], Neal Wagman[3]; [1]Light Brigade Inc., USA, [2]Seikoh Giken Co. Ltd., USA, [3]Norland Products, USA*
SC264 **Introduction to Ethernet Technologies**, *Jeffrey Cox; BT, USA*
SC266 **Quantum Cryptography and Quantum Information**, *Richard Hughes[1], Thomas Chapuran[2]; [1]Los Alamos Natl. Lab, USA, [2]Telcordia, USA*

Additional Short Course Category: Industry Best Practices

NEW! SC347 **Reliability and Qualification of Fiber-Optic Components**, *David Maack; Corning, USA*

Back to Top

SPIE/40 years of optical manipulation
As part of our year-long series celebrating LaserFest, our feature on "40 Years of Optical Manipulation" traces the evolution of this exciting field.

And for your monthly fix of laser history, this month we're forgetting Bell Labs and *Physical Review Letters* and turning to different sources: comic books, movies, and television. In his "Popular History of the Laser," Stephen Wilk traces the evolution of the laser in science fiction and pop culture—from Buck Rogers to James Bond.

Finally, our March issue also contains an excellent profile from OSA Fellow Barry Masters. If any life exemplifies the value of interdisciplinary education and good mentoring, it is that of Hermann Ludwig Ferdinand von Helmholtz. Helmholtz was a physicist, teacher, medical doctor, aesthete and more. He invented the ophthalmoscope—a device that revolutionized ophthalmology—when he was just 29 years old, and went on to conduct many more critical investigations into nerve conduction and physiological optics.

Nature Physics March 2010

New 2010 issue Nature Photonics Technology
Things of Beauty: Aharanov-Bohm effect, ..
Len Fisher, "The Perfect Swarm"
Ultrafast science: one-femtosecond film
Nature Physics 6, 159–160 (1 March 2010) | doi:10.1038/nphys1601

Ultrafast science: Towards a one-femtosecond film

Olga Smirnova & Misha Ivanov

Abstract

The dynamics of a 'hole' — that is, the space left vacant by an absent electron — created in an atom or a molecule by ionization can be extremely fast, with the early response lasting about 50 as (ref. 1).

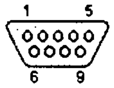
Eckehard Scholl, "Chaos control"
Durbin & Collela, "X-ray optics"
Adilson Motter, "spontaneous sychrony breaking"
"Back action evading moeasurement of nanomechanical mothion"
Stephen Quake, "pressure gain valves"
self organized adaptation of simple neural circuit

PINOUTS.RU

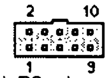
PC serial port (RS-232 DE9) pinout

bookmark this page

serial interface



9 pin D-SUB male connector at the computer



10 pin IDC male connector at the inside, at motherboard

This RS232C DE-9 (usually miscalled DB-9) port is very common and available at almost any PC, some Sun (at least Ultra 5/ 10, Blade 100/150) and many other computers. Document includes description of how PC serial mouse works.

Almost each PC nowdays equiped with one/two/four serial interface (RS232C). This PC serial port interface is single ended (connects only two devices with each other), the data rate is less than 20 kbps. It"s a voltage loop serial interface with full-duplex communication represented by voltage levels with respect to system ground. A common ground between the PC and the associated device is necessary.

DB-9 Pin	IDC internal pin name *	Name	Dir	Description
1	1	CD	←	Carrier Detect
2	3	RXD	←	Receive Data
3	5	TXD	→	Transmit Data
4	7	DTR	→	Data Terminal Ready
5	9	GND	—	System Ground
6	2	DSR	←	Data Set Ready
7	4	RTS	→	Request to Send
8	6	CTS	←	Clear to Send
9	8	RI	←	Ring Indicator

Note: Direction is DTE (Computer) relative DCE (Modem)

* Pin assignment of internal connector may be different for different motherboard models. Pin 10 removed in connector. Internal IDC connector wired to external port with a simple flat ribbon cable.

PC serial port pinout signals explanations

Since PC serial port is based on RS-232 standard, you may find signal details in the RS-232 interface pinout document

Standard RS232 data packet

RS232 data usually is sent as a packet with 7 or 8 bit words, start, stop, parity bits (may be varied). Sample transmission shown on picture: Start bit (active low, usually between +3v and +15v) followed by data bits, parity bit (depends on protocol used) and finished by stop bit (used to bring logic high, usually between -3v and -15v).

Sample serial port device. How serial mouse works

Typical PC mouse controlling system has the following parts: sensors -> mouse controller -> communication link -> data interface -> driver -> software. Sensors are the movement detectors (typically optomechanical) which sense the mouse movement and button swiches which sense the button states. Mouse controller reads the state of those sensors and takes acount of current mouse position. When this information changes the mouse controller sends a packet of data to the computer serial data interface controller. The mouse driver in the computer received that data packet and decodes the information from it and does actions based on the information.

PC serial mouse voltage levels:

Mouse takes standard RS-232C output signals (+-12V) as its input signals. Those outputs are in +12V when mouse is operated. Mouse takes some current from each of the RS-232C port output lines it is connected (about 10mA). Mouse send data to computer in levels that RS-232C receiver chip in the computer can uderstand as RS-232C input levels. Mouse outputs are normally something like +-5V, 0..5V or sometimes +-12V. Mouse electronics normally use +5V voltage.

Related pinouts:

PC serial (25 pin) pinout

RS232 interface pinout

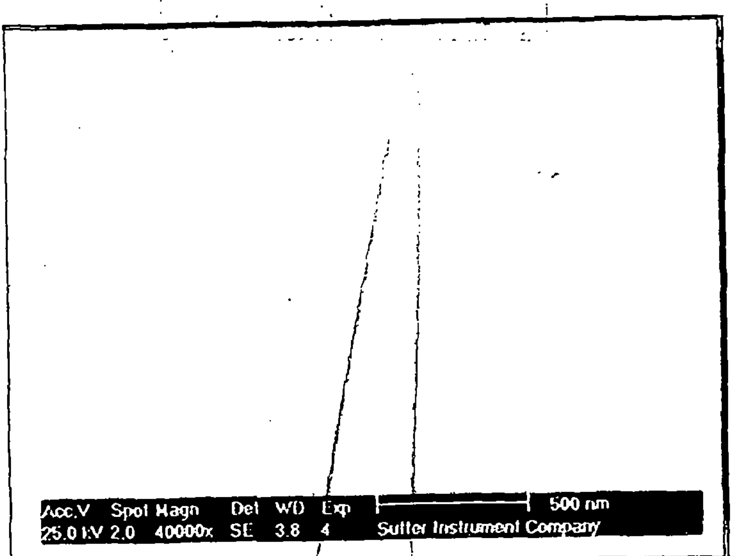
Serial device hardware implementation

Dear Investigator,

Prior to shipping this Model P-2000 Puller, we pulled a series of micropipettes using the program #0, and measured the resultant pipettes using our scanning electron microscope. All tip diameters were in the range of .017 microns at a taper length of 9-10mm. The photo below is of a representative from this series and measures .017 microns after correction for the gold coating (100 Angstroms). We also pulled a series of *patch-type* pipetes using program #2, and obtained consistent results as measured with and optical scope outfitted with reticle.

Thank you.

Sutter Instrument Company



The MP-285 you have just received is equipped with a new "swinging-gate" headstage mount. This document describes how to use the swinging gate, as this information has not yet been incorporated into the current manuals. For those who have used the old version, the most important change relates to the brake screw used to adjust and secure the rotating dovetail. In the old version, this was a thumbscrew as the intention was that the dovetail would be continuously rotated to aid in the removal of pipettes. In the swinging gate, pipette removal is facilitated by opening the gate and a thumbscrew is used to keep the gate closed. Because the rotary dovetail on the swinging gate is intended to be set to a particular angle and left alone, the brake screw used for adjustment is now a set screw rather than a thumb screw.

Operation of the swinging gate:

1) **The brake screw on the swinging gate is a setscrew that must be turned with a hex key to adjust the angle of inclination of your pipette. Select the approximate angle by looking at the indexing ring and tighten the brake screw with the 1.5mm hex key provided. It is not intended that this screw be repeatedly loosened and tightened during normal operation. Pipette removal should be accomplished opening the swinging gate.**
2) **Open the swinging gate by loosening the black-capped clasping screw. The screw need only be turned about 3 turns to allow the gate to swing open.**
3) **Close the gate by holding it firmly in the closed position and retightening the clasping screw. The design is such that the screw will draw the gate into its fully closed position. The screw should be securely finger tightened while holding the gate closed to achieve accurate and secure repositioning. Do not use pliers or another tool to tighten the clasping screw!**
4) **When the gate is open, you have access to the four Phillips-head screws that mount the new headstage mount to the front of your manipulator. If necessary, the gate can be moved to other positions on the face of the z-axis. The mounting screws must be tightened sufficiently to keep the headstage mount from moving on the front of the manipulator.**

Sutter Instrument Company
51 digital drive, novato ca 94949
tel: (415) 883-0128 fax: (415) 883-0572 email info@sutter.com