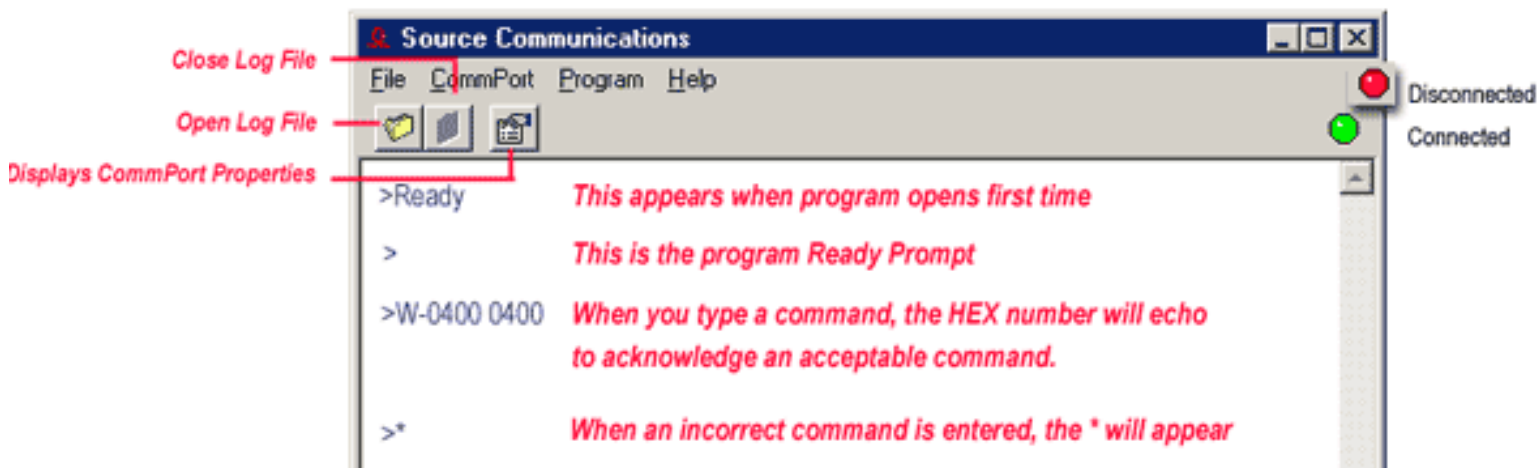


Using the **Source Communications** program (Version 2) for the SD-2 Programmable Source Driver board Source

Control Window



When the program is started, the ">" or "Ready>" prompt will appear within the Source Control Window and the Comm port settings match your connections.

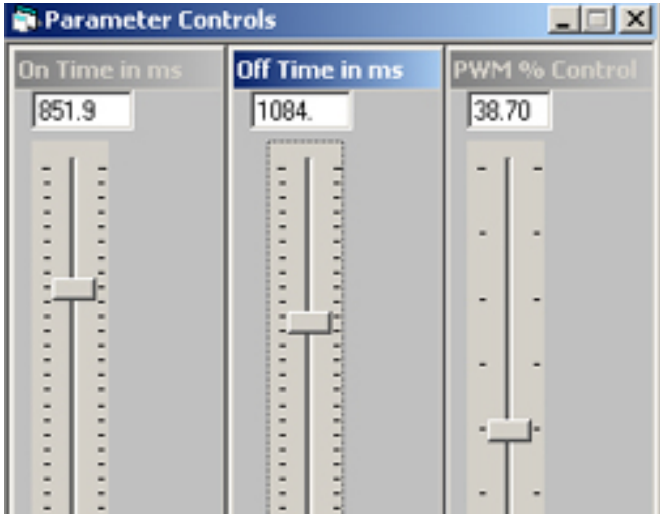
In the following documentation.

- Cautions and information transmitted from the board to the communications program are in **red**
- Links and Commands are in **blue**.
- The word **Chopper** refers to a low frequency interruption of the Source; this usage derives from times when the interruption was done mechanically.

Communications window commands

"W" High-Frequency Pulse Width	Source Intensity control
" L" Limit setting for the High-Frequency Pulse Width	Protection against accidental excessive power settings
"H" Chopper on time	Duration of Pulsing On Time in .104 Milliseconds
"O" Chopper off time	Duration of Off time in .104 Milliseconds

"E" Memory examination command	Access the driver board memory locations
"N" Disable/Enable automatic NVRAM writing	Disable the on-chip NVRAM



Sliders:*In a separate window*

Provide alternate controls for On time, Off time, and Pulse width

- [PWM % Control](#) Provides control of High Frequency pulse width
- [Chopper on Time](#)
- [Chopper off Time](#)

Menubar functions

File

- **Start Logging:** Starts a logging file for the communications window
- **Stop Logging:** Stops the logging for the communications window
- **Clear Window:** Clears the communications window
- **Exit:** Exits the Source Communications window

CommPort

- **Port Open:** Toggles the program link to the communications port
- **Properties:** Allows setting/reading the Serial port number

Program

- Starts the independent program to load the driver board Program

Help

- **Help:** Displays this page
- **About:** Displays Version Number, Copyright information and disclaimer

Important Notes

Communication startup: If you plug in the communications cable from the host computer to the driver board **before** starting this program (at least once), the driver board will be forced to the reset condition until this program is started. No damage is done.

Number Formatting: All numeric input and output is in **16-bit HEX**, thus 000C is 12 decimal. During entry the input is right justified; thus entering "**C**" **<enter>** is the same as 000C An **<enter>** is automatically assumed after you type a fourth number).

If you make a mistake: *If you type in an unrecognized character (say "T") or a non-Hex number:* The line will be aborted and echo an asterisk "*". The Backspace is not programmed, but will produce the same result--the line will be aborted and an * will appear. This can also be used to abort a mistaken entry.

Controlling the high-frequency pulse width (intensity)

In the following, user-typed characters are in **black** and extra data generated by the driver board is in **red**.

W Hex number: sets the PWM pulse width in .25usec units. The limit permitting you can go to 88uSec (84.6% of $10^6/9600$); actual entry is 160, the Hex symbol for 4*88 decimal.

Caution: Up to a 84.6% duty cycle is allowed, **HOWEVER**

THIS WILL BURN OUT THE SOURCE if this setting is higher than the maximum allowable setting for the source being used. See [LIMIT setting below](#). There are instructions for checking this on our web site; if in doubt please follow them.

To set the width.

At the prompt, type W 00XX (XX is .25uSec) and press **<enter>**
If the setting is successful the driver will return the following.

Example

```
>W-0040 0040  
>E-232=0040
```

The user has typed **W** followed by hex **0040** (decimal 64). The driver has acknowledged by echoing the new setting. The high frequency pulse width has been set to: $64 \cdot .25$ or 16 microseconds, out of 104 microseconds ($10^6/9600$); a duty cycle of about 16%.

Illegal or unrecognized commands and arguments are echoed as *****; for instance.

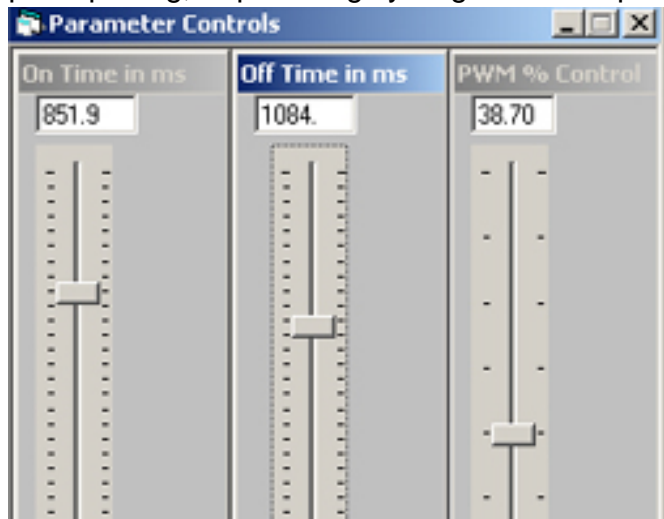
```
> W-004T * where T is the non-recognized character.
```

The original "0040" is the echoing from the Source Driver board communications section. The E-232 command is generated by the Source communications program, to verify the actual value stored and used. This is also used to update the "PWM % Control". A value that is in excess of the PW limit produces a "*" instead of the first "0040" response, and the E-232 command will show the present value. Memory location "232" is where the Pulse Width is stored in RAM.

PWM % Slider Control

This slider displays and controls the present Pulse width for the PWM operation. Attempts to move this slider beyond the limit will result in the Pulse Width being unchanged.

The windowed display "PWM %" gives the duty cycle of the present pulse width modulation. The popup windows show the hex value of the present pulse width. Initiation of the communications program, successful port opening, or powering cycling the board power; will attempt to update this slider to the true value.



The text window reflects the commands sent to and received from the board.

Limit setting for the High-Frequency Pulse Width

L Hex number: sets the limit for the High frequency pulse width.

The allowed pulse widths are actually values below the limit. Thus, if the limit is "44", then the last allowed value is "43".

The **maximum for the limit setting is Hex 160 or 84.6%** (as mentioned under "W"). Setting the limit below the present pulse width value resets the pulse width to the highest new allowed value; the limit minus one.

Example

>L-44 004

The user has typed L, followed by hex 44, followed by a <enter>, and the driver has acknowledged by echoing the new setting back. The limit has been set to decimal 17 microseconds.

Illegal or unrecognized commands and arguments are echoed as *; for instance.

>L-004T * where T is the non-recognized character.

Chopper On Time

The On slider allows smooth adjustment of the On time. Time shown in the on window is in msec; the popup value is the corresponding command that will be sent to the board. Please note that the board time is in .104 ms increments. The communications text window also reflects the value sent to the board.

H Hex number: sets the Chopper on time in .104ms units.

The limit is 963D in Hex; this is equivalent to 3.999 sec.
Setting H to 0 turns off chopping and leaves the driver in the off state.

Example

>H-1000 1000

The user has typed **H** followed by hex **1000** followed by a <enter>, and the driver has acknowledged by echoing the new setting. The low frequency on time is now 425.984 msec.

Illegal or unrecognized commands and arguments are echoed as *****; for instance. **>H-004T *** where **T** is the non-recognized character.

Chopper Off Time

The Off slider allows smooth adjustment of the Off time. Time shown in the Off window is in msec; the popup value is the corresponding command that will be sent to the board. Please note that the board time is in .104 ms increments. The communications text window also reflects the value sent to the board.

O Hex number: sets the Chopper off time in .104ms units.

The limit is 963D in Hex; this is equivalent to 3.999 sec.
Setting O to 0 turns off chopping and leaves the pulsing driver in the on state.

Example

>O-1000 1000 The user has typed **O** followed by hex **1000** followed by <enter>, and the driver has acknowledged by echoing the new setting. The

low frequency off time is now 425.984msec.

Illegal or unrecognized commands and arguments are echoed as *****; for instance. **>O-004T *** where **T** is the non-recognized character.

E General communication command to examine driver board memory.

E Hex Number: Examines ram locations on the board.

Useful and supported numbers are:

- 232** Pulse width in .25 microseconds
- 234** Chopping frequency in .25 seconds
- 236** Pulse width limit value in .25 microseconds.

Example

>E-0232 0044 The operator has asked for the value stored in memory location **232** (hex) and the driver board has responded with hex word **44** (the present pulse width in .25 seconds).

N Disabling the NVram writing

This command enables and disables NVRAM writing by the sliders OR the communications window.

This will stay in effect until the board power is cycled or N is changed.

The "**N 0**" command allows the user to do many **F**, **W** commands (active real time control) (>10,000).

Examples

>N-0 0000

N 0 turns off NVRAM write. This is disabled until Power Up or the **N 1** command is entered.

>N-1 0001

N 1 turns on NVRAM write. This is the default after power up. Illegal or

unrecognized commands and arguments are echoed as **"*"**; for instance. **>N-T *** where **T** is the non-recognized argument.

To download new programs into the source driver board

Menubar Program button: If enabled this brings up a file selector window to allow you to select a binary file for the driver. Then the programming program is executed and downloaded to the driver. Selecting programs that are not program files will erase the MSP430 and then it **MUST** be reprogrammed with a good file.

After programming with no restoring, the board will default to an **off time of 1 second**, **an on time of 1 second** and **Width=8 counts (1.9%)** . At this point you can type **"P 4"** and **"W 44"**; this should produce a visible response.

When a new **Hotime board code.msi** is downloaded and activated the code resides in **My Documents\HOTime board code** as **source_time.txt**. This is where you will find any new code to be downloaded to the driver board.