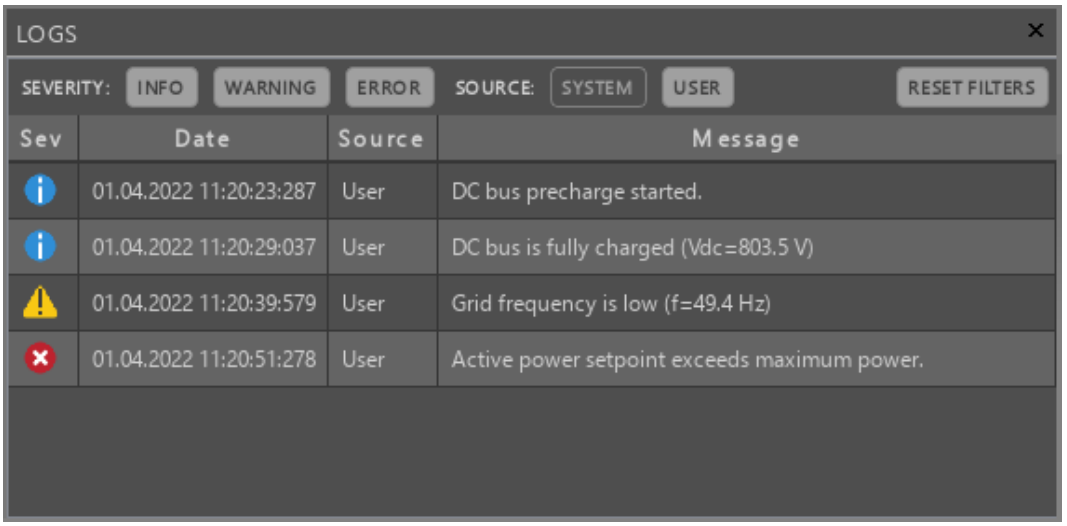
LOG - Write a log message

SD050 | Posted on April 6, 2022 | Updated on July 24, 2025

Julien ORSINGER
Power Applications Specialist
imperix

This block writes a user-defined message in the log module of Cockpit.



Log module of Cockpit

Numerical values can be inserted into the message using the conversion specifier “%f“, that will take to the value of the second input signal. This block only supports floating-point numbers. Optionally, the number of digits displayed after the decimal point can be specified using the precision modifier "`.`". Here are some examples:

Log message field	Input value	Message displayed in the log module
The value is: %f	3.141592653	The value is: 3.141593
The value is: %.3f	3.141592653	The value is: 3.142
The value is: %.0f	3.141592653	The value is: 3
The value is: %f	3	The value is: 3.000000
The value is: %.3f	3	The value is: 3.000
The value is: %.0f	3	The value is: 3

Note that the maximum length of a log message is 256 characters and the maximum amount of different log messages is 128.

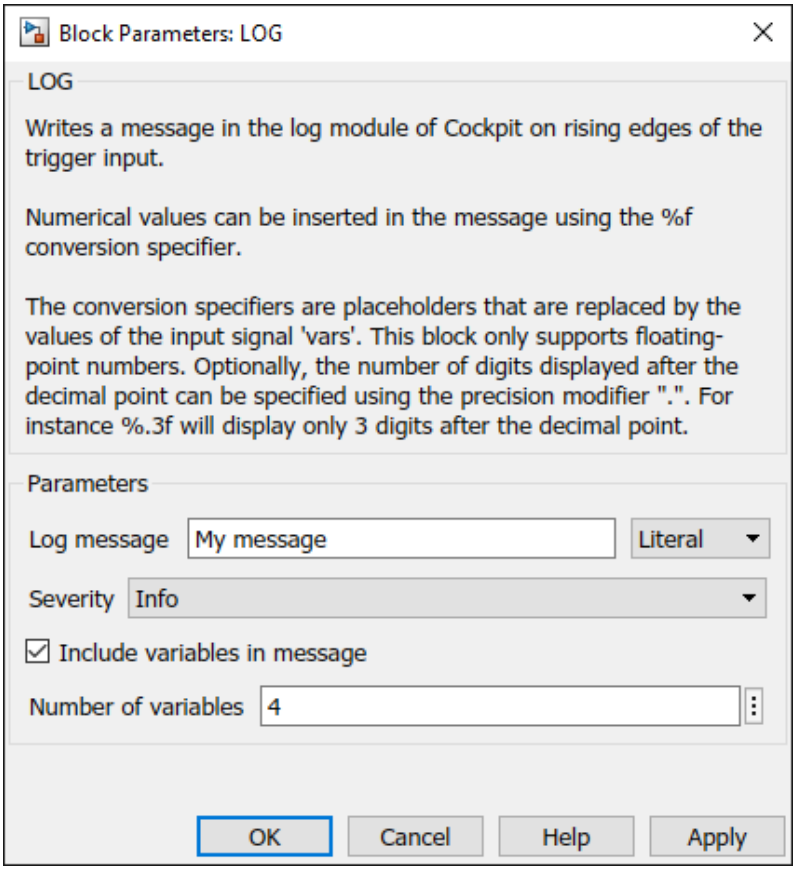
Simulink block

Signal specification

- The first input is a trigger port that generates a log message when its value rises from a negative or zero value to a positive value.
- The second input is only visible when `Include variables in message` is checked. It is used to input numerical value(s) into the message. The value(s) are sampled when the log message is triggered.
  Its width is defined by the parameter `Number of variables` and must also correspond to the number of conversion specifiers contained in the message.

Parameters

- `Log message` is the message that will be displayed in the log module of Cockpit.
- `String type` indicates whether the `Log message` field contains a literal text string or an expression that should be evaluated.
- `Severity` defines the type of message (Info, Warning, or Error). The log module of Cockpit lets you filter the displayed message by severity.
- `Include variables in message` indicates if numerical values are inserted in the message using the `%f` placeholder. If checked, a second port appears to input the numerical values.
- `Number of variables` is the number of numerical values that are inserted into the message. This field is only visible if `Include variables in message` is checked.



PLECS block

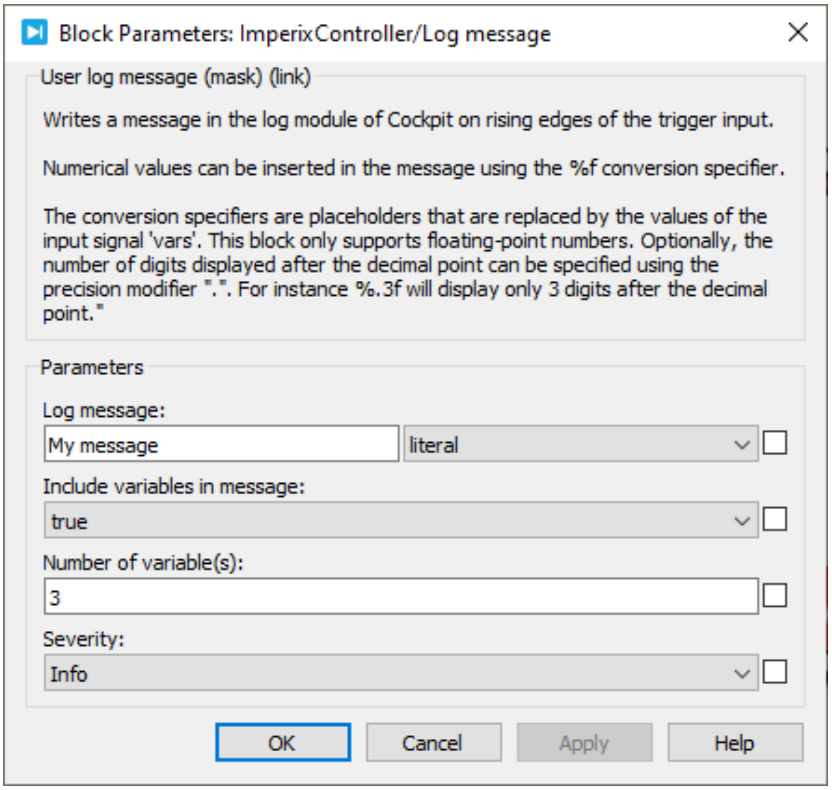
Signal specification

- The first input is a trigger port that generates a log message when its value rises from a negative or zero value to a positive value.
- The second input is only visible when `Include variables in message` is checked. It is used to input numerical value(s) into the message. The value(s) are sampled when the log message is triggered.
- Its width is defined by the parameter `Number of variables` and must also correspond to the number of conversion specifiers contained in the message.

Parameters

- `Log message` is the message that will be displayed in the log module of Cockpit.
- `String type` indicates whether the `Log message` field contains a literal text string or an expression that should be evaluated.
- `Severity` defines the type of message (Info, Warning, or Error). The log module of Cockpit lets you filter the displayed message by severity.

- `Include variables in message` indicates if numerical values are inserted in the message using the `%f` placeholder. If checked, a second port appears to input the numerical values.
- `Number of variables` is the number of numerical values that are inserted into the message. This field is only visible if `Include variables in message` is checked.



C++ functions

`Log_AddMsg` — Configure a log message

```
int  Log_AddMsg(int msg_id, int sev, const char *msg);Code language: C++ (cpp)
```

Configures a log message that can then be sent using `Log_SendMsg`.

It has to be called in `UserInit()`.

Parameters

- `msg_id`: a unique identifier for the message. It must begin at 0 and count up in order, ensuring no numbers are skipped.
- `sev`: the message severity (10: Info, 20: Warning, 30: Error)
- `msg`: the message that will be displayed in Cockpit

`Log_SendMsg` — Send a log message

```
Log_SendMsg(int msg_id, float vars[]=nullptr, int n_vars=0);Code language: C++ (cpp)
```

Displays the message in the Cockpit log module.

It has to be called in the interrupt.

Parameters

- `msg_id`: the message unique identifier
- `vars`: an array of floating-point values that are inserted into the message
- `n_vars`: the number of values to insert into the message

If the Log_SendMsg() function is called directly in the UserInterrupt it will spam thousands of messages per second! The user should make sure to implement a trigger mechanism, using a boolean for instance as shown in the example below.

Example of use

```
tUserSafe UserInit(void)
{
    //...
    // Configure a warning message with a unique id of 0
    Log_AddMsg(0, 20, "Operating limits exceeded (V=%.3fV / I=%.3fA)");
    //...
    return SAFE;
}

tUserSafe UserInterrupt(void)
{
    //...

    static bool warning_message_sent = false;
    if(V_meas > 850 || I_meas > 40){
        float log_values[2];
        log_values[0] = V_meas;
        log_values[1] = I_meas;
        // Display the message in Cockpit
        if(!warning_message_sent) {
            Log_SendMsg(0, log_values, 2);
            warning_message_sent = true;
        }
    } else {
        warning_message_sent= false;
    }

    //...

    return SAFE;
}
```

Code language: C++ (cpp)