DOM-Cal Users' Guide

John Kelley and Jim Braun

April 6, 2005

1 Purpose

DOM-Cal is stand-alone application which runs directly on the DOM mainboard and calibrates the DOM front-end electronics. The calibration constants obtained with DOM-Cal allow one to translate the DOM's digital output into voltage and time waveforms. Furthermore, DOM-Cal's high-voltage calibration measures the PMT gain at various voltage settings.

2 DOM-Cal

The DOM-Cal application resides on the DOM flash filesystem as the file $domcal. The application can be run directly from the iceboot command-line, using the command

```
> s" domcal" find if exec endif
```

The application will then ask the user for the date (the DOM has no concept of the date or time of day), and if a high-voltage gain calibration is to be performed (in which case the PMT high voltage will be enabled).

DOM-Cal can also be run through a Java interface; see section **??**.

2.1 Pulser Calibration

The DOM pulser produces SPE-like waveforms to test and calibrate mainboard functionality. Pulser amplitude is controlled through a DAC setting, DOM_HAL_DAC_INTERNAL_PULSER (DAC 12). The pulser calibration produces a linear relation between DAC setting and pulse amplitude in Volts.

2.2 ATWD Calibraton

The two Analog Transient Waveform Digitizers (ATWDs) function as the primary signal digitizers for the DOMs. The output of the ATWD consists of 128 samples of the input signal voltage, with each sample level being an integer from 0 to 1023.

The ATWD calibration produces a set of 768 (2x3x128) linear fits, one for each ATWD (0 and 1), channel (0, 1, and 2), and sample bin (0-127). These fits translate the ATWD level to an input voltage, before any channel-specific amplification. Because the fit parameters vary for each sample bin, the unique pedestal pattern of each ATWD is automatically taken care of.

The calibration uses the ATWD front-end bias voltage (DOM_HAL_DAC_PMT_FE_PEDESTAL) to vary the baseline voltage of the ATWD, independent of any channel amplification. CPU-triggered waveforms are recorded for a number of different bias voltages, and using a known relationship between the bias DAC setting and true bias voltage, fits for all of the ATWD bins are calculated.

2.3 Amplifier Calibration

Each of the three signal channels of the ATWDs has a given amplification of the input voltage: nominally 16x for channel 0, 2x for channel 1, and 0.25x for channel 2. The amplifier calibration determines the mean gain factors for each channel, along with the standard error of each gain.

The calibration uses the pulser to produce reasonably-sized waveforms for each channel. The maxima of a number of discriminator-triggered waveforms are recorded, and using the pulser calibration and the ATWD calibration previously obtained, translated to a voltage. The mean gain and the error on this mean are then calculated.

2.4 ATWD Frequency Calibration

ATWD frequency is controlled by DAC settings DOM_HAL_DAC_ATWD0_TRIGGER_BIAS and DOM_HAL_DAC_ATWD1_TRIGGER_BIAS (DACs 0 and 4, respectively). ATWD frequency calibration produces a linear relation between DAC setting and ratio between ATWD frequency and the mainboard 20MHz oscillator.

2.5 High Voltage Gain Calibration

As an option, DOM-Cal can calibrate the gain of the PMT as a function of the high voltage. DOM-Cal will return a linear fit of $\log(gain)$ versus $\log(voltage)$. Additionally, DOM-Cal returns the peak-to-valley ratio at each of the voltage settings at which a good charge histogram was achieved.

To find the gain at a particular high voltage, a number of discriminator-triggered waveforms are taken. Using the results of the previous calibrations, the waveform is integrated, and the charge in pC is recorded in a histogram.

The histogram is fit via nonlinear (Levenberg-Marquardt) regression to the functional form $A\ e^{-Bx} + C\ e^{-E(x-D)^2}$. The single photoelectron (SPE) peak (and thus the gain) is reasonably approximated by the parameter D, while the valley between the noise peak and the SPE peak is found via iterative (Newton-Raphson) search. A few heuristics are used to determine if the fit is sane, and only then is the extracted data used for the rest of the calibration.

After the gain is obtained for each high voltage setting, a linear regression is performed on $\log(HV)$ vs $\log(gain)$.

2.6 DOM-Cal Output

DOM-Cal writes calibration results to the DOM flash filesystem in a binary file named "calib_data". The format of this file is described in detail in table **??** and table **??**.

3 Java Interface to DOM-Cal

DOM-Cal may be easily accessed through the icecube.daq.domcal java package found in the dom-cal project. Implimentations to read binary calibration data from a DOM or initiate a calibration cycle and read back calibration data are available. DOM servers, such as dtsx and domserv, are required to read calibration data with the java client. To read calibration data, type

```
java -cp {$BFD_HOME}/lib/dom-cal.jar icecube.daq.domcal.DOMCal [host] [port]
[output directory]
```

Quantity	Datum Size	Comments
Version	Short (16 bits)	Version number – use to determine endianness (version ¡ 256)
Record Length	Short	Length of domcal binary record
Date – Day	Short	Day calibration was performed
Date – Month	Short	Month calibration was performed
Date – Year	Short	Year calibration was performed
Unused	Short	Not used
DOM ID hi bits	Int (32 bits)	DOM ID in true hex format (not ASCII)
DOM ID low bits	Int	DOM ID in true hex format (not ASCII)
Temperature	Float (32 bits)	DOM temperature in Kelvin
Initial DAC values	Short[16]	Initial values of DAC 0-15
Initial ADC values	Short[24]	Initial values of ADC 0-23
FADC calibration pedestal	Short	
FADC calibration gain	Short	
FE pulser calibration	LinearFit	Calibration of internal pulser
ATWD gain calibration	LinearFit[2][3][128]	Calibration of ATWD 0-1, channels 0-2, bins 0-127, such that member 128 is atwd 0, channel 0, bin 127, and member 129 is atwd 0, channel 1, bin 0, etc.
FE amplifier calibration	Float[6]	Channel 0 amplification and error, channel 1 amplification and error, channel 2 amplification and error.
ATWD frequency calibration	LinearFit[3]	Frequency calibration for atwd 0, atwd 1.

Table 1: DOM-Cal binary output format

Quantity	Datum Size	Comments
Slope	Float	Slope parameter of linear fit
Y-intercept	Float	Y-intercept parameter of linear fit
R-squared	Float	Quality of fit

Table 2: Binary format of linear fit

to download calibration data from a DOM on $host at $port and store xml-formatted output in $output directory. The xml calibration file is named by domId. In many cases, it is advantageous to read calibration from many DOMs ant once from a sequence of ports. To read calibration data from many ports, type

```
java -cp {$BFD_HOME}/lib/dom-cal.jar icecube.daq.domcal.DOMCal [host] [port]
[num ports] [output directory]
```

where $num ports is the number of sequential ports to scan above $port. The ports need not be sequential, i.e. if three DOMs are available on ports 5000, 5003, and 5008, setting $port = 5000 and $num ports = 9 would access all three DOMs. To initiate calibration, add the string "calibrate dom" at the end of your command line arguments:

```
java -cp {$BFD_HOME}/lib/dom-cal.jar icecube.daq.domcal.DOMCal [host] [port]
[num ports] [output directory] calibrate dom
```

would attempt to calibrate and read out $num ports DOMs, starting from $port.

3.1 XML Calibration Output

The XML calibration output is much more user-friendly then the DOM-resident binary file. Figure **??** is an example output file with the ATWD calibration, DAC, and ADC values truncated. An actual output file contains 2x3x128 = 768 ATWD calibration linear fits.

```
<domcal>
    <date>6-11-2004</date>
    <domid>f771bb4dce28</domid>
    <temperature format="Kelvin">238.9725</temperature>
    <dac channel="0">850</dac>
    .
    .
    <dac channel="15">0</dac>
    <adc channel="0">95</adc>
    .
    .
    <adc channel="23">0</adc>
    <pulser>
        <fit model="linear">
            <param name="slope">9.181886E-5</param>
            <param name="intercept">3.7924567E-4</param>
            <regression-coeff>0.9999254</regression-coeff>
        </fit>
    </pulser>
    <atwd id="0" channel="0" bin="0">
        <fit model="linear">
            <param name="slope">-0.0021576127</param>
            <param name="intercept">2.6414666</param>
            <regression-coeff>0.9999896</regression-coeff>
        </fit>
    </atwd>
    .
    .
    <atwd id="1" channel="2" bin="127">
        <fit model="linear">
            <param name="slope">-0.0022831585</param>
            <param name="intercept">2.6237197</param>
            <regression-coeff>0.9999343</regression-coeff>
        </fit>
    </atwd>
    <fadc parname="pedestal" value="2112"/>
    <fadc parname="gain" value="2184"/>
    <amplifier channel="0">
        <gain error="0.008047878">-19.214722</gain>
    </amplifier>
    <amplifier channel="1">
        <gain error="0.0013110796">-2.8445616</gain>
    </amplifier>
    <amplifier channel="2">
        <gain error="2.8432324E-4">-0.1962693</gain>
    </amplifier>
    <atwdfreq chip="0">
        <fit model="linear">
            <param name="slope">0.012672497</param>
            <param name="intercept">3.5881329</param>
            <regression-coeff>0.9979777</regression-coeff>
        </fit>
    </atwdfreq>
    <atwdfreq chip="1">
        <fit model="linear">
            <param name="slope">0.012642916</param>
            <param name="intercept">3.7248025</param>
            <regression-coeff>0.9978187</regression-coeff>
        </fit>
    </atwdfreq>
</domcal>
```


Figure 1: Sample XML calibration output