Aug 29, 2002

APPLICATION NOTE 174

Monitor Calibration in Fiber Optic Applications

Paragraph Header

Fiber optic standards, such as the SFF-8472 and the now emerging XENPAK Standard, call for monitoring of key fiber optic signals such as bias current (mA), transmitted power (mW), and received power (mW). Transmitted and received powers are monitored by means of photodetectors whose currents serve as a measure of power level. Large variations in photodetector gains from part to part make it unreliable to use such measurements without calibration.

Need for Calibration

Figure 1 shows the path the measurement data follows during monitoring of transmitted power. An optical output is converted into current by virtue of the photodetector•fs responsivity (mA / mW). This current is converted into digital form through an A/D and then placed on the 2-Wire bus for easy access by a host. Responsivity of photodetectors can vary, from part to part, by as much as a factor of 10. Consequently, in a raw uncalibrated environment, the host will get significantly different data values for the same transmitted power, depending on the photodetector gain.

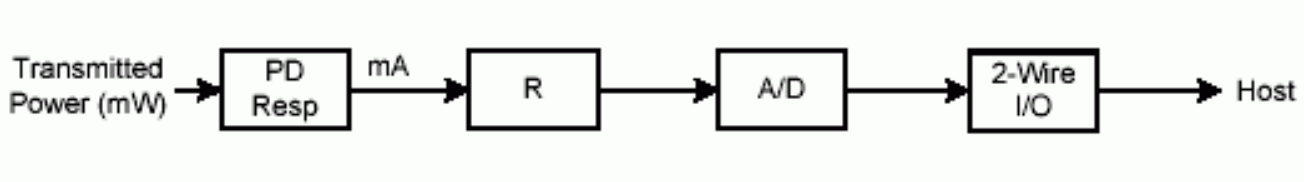

Figure 1.

Calibration ensures that there is a one-to-one correspondence from input to output. For example, 1mW of transmitted power will result in the same digital data value, regardless of the type or characteristic of photodetector. Furthermore, standards will impose a scaling requirement by clearly defining the full-scale read-out by the host. In short, the accuracy of the measurement is guaranteed by calibration and the scale is defined by the standard.

Calibration Implementation

In the DS1852 calibration is implemented either internally or externally.

Internal calibration can be compared to a programmable gain as shown in Figure 2. A single point calibration is executed on the user's factory test floor. This entails ramping up the signal preferably to a level around full-scale, then programming the internal calibration register until the digital value, Nad, reads as expected. Range and resolution values for internal calibration are specified in the SFF-8472 standard for each of the following signals: supply voltage, bias current, transmitted power, received power. Once this calibration is set, every incoming raw datapoint is scaled up accordingly without any intervention from the user, outputting a digital value or count consistent with the standard's scale. Hence, the term internal calibration is used.

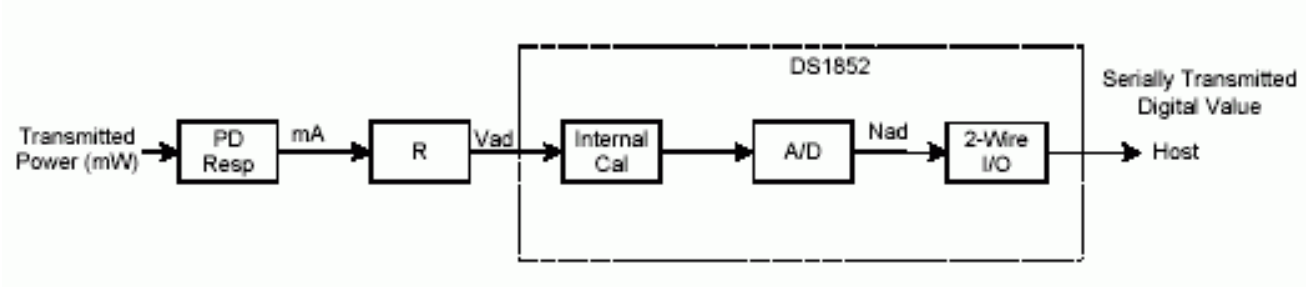

Figure 2.

External calibration is a feature of the DS1852, DS1854, DS1857, and DS1858. Figure 3 illustrates how it works. The IC's digital output Nad is uncalibrated. The user relies on a device with computing power, such as a microcontroller, to retrieve calibration constants from the DS185X IC and perform the computation. External calibration is explained in the SFF-8472 standard and falls under two general categories: linear and nonlinear. In the linear case, two constants are required, one for offset and one for gain; for every raw datapoint x, a calibrated value y = ax + b is calculated. The nonlinear case involves (n +1) coefficients, n being the highest power. For example, $y = ax^4 + bx^3 + cx^2 + dx + e$ is a suggested equation for received power.

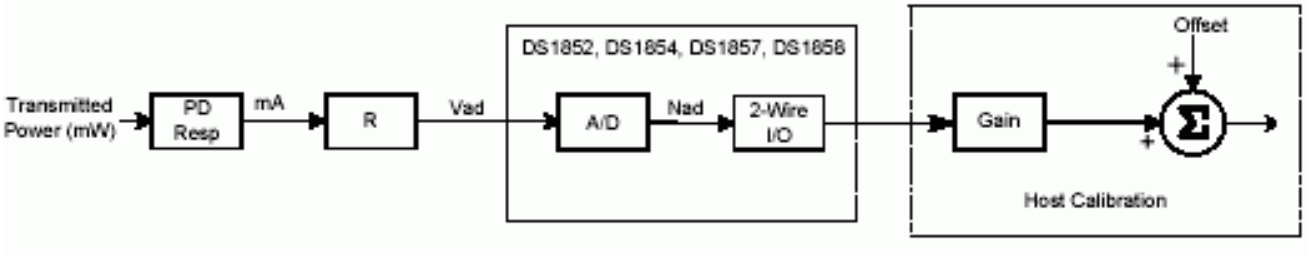

Figure 3.

Transfer Functions

Regardless of the approach taken the result from the A/D is the product of two transfer functions, the first is the relationship between the parameter to be measured and the voltage applied to the A/D. In the case of transmitted power, this is the photodetector responsivity.

Vad = f1 * Param, where Param is the parameter to be measured, f1 is the transfer function of the conversion process, and Vad is the voltage input to the A/D.

f1 is simply a function of the circuitry within the module, used to derive a voltage indicative of the measured parameter.

The second is the relationship between the voltage input to the A/D and the calibrated result.

Nad = Gain * Vad + Offset, where Nad is the numerical output of the A/D, and Gain is the scaling factor.

So overall, the transfer function is Nad = f1 * Gain * Param + Offset. Internal calibration as implemented with the DS1852 has no provision for offset. In this case, Nad = f1 * Gain * Param.

Internal Calibration

The standard imposes an additional constraint on Nad for internal calibration. It requires that Nad = f2 * Param, where f2 is a scaling constraint defined by the standard so that one universal scale is used for all applications.

For example, TX output power is defined as Param (mW) = Nad * 0.0001, or f2 = 10,000.

This, in turn, constrains the product of f1 * Gain to also be 10,000.

Since f2 = f1 * Gain, the value of Gain, given that f2 is a universal value set by the standard, is typically dependent on f1, that is the circuitry within the module. For example, a resistor, converting current to voltage, is commonly used in series with a laser monitor photodiode to implement an APC function. Therefore, the voltage across the resistor (the feedback voltage to the laser driver) is also the input to the A/D, Vad, in the case of transmitted power monitoring. Changing the resistor value (a component of f1) would entail changing the gain value, so that in the end their product is constant and the standard's scale is observed.

For the DS1852, Gain is set by a combination of four coarse-scaling bits and 12 fine-scaling bits. The procedure to set these bits is described in Appendix A.

External Calibration

The big difference here is that there is no constraint on the product of f1 and Gain, other than noise if the signal level is very low. The value of Nad is somewhat arbitrary because the final calibrated result is not subject to a precision format imposed by the standard. This is the case for internal calibration, where the precision format is 16-bits fixed. The user has to perform an arithmetic operation on this result to convert it to calibrated units (counts). The necessary constants to perform this operation are stored in memory. The use of calibration constants is detailed in the SFF-8472 document. For a linear relationship, calibrated counts = Nad * calconstant1 + calconstant2.

An additional advantage of this approach, at least if the DS1852 is used, is that the A/D can be scaled such that a full-scale count corresponds to the maximum analog signal value for this particular product, not just a standard defined value. For example, if a module was designed for 1mW maximum TX power, the A/D could be scaled such that 1mW produced an output of FFFFh. In this case the best achievable resolution would be 0.02µW (1/65536 of 1mW), compared to 0.1µW in the "Internal Calibration" mode as per the SFF-8472 standard.

The user needs to make sure his computing engine is not limited by a 16-bit fixed precision for external calibration. If this is the case, the resolution advantage outlined in this section is foregone. Either a 32-bit fixed, or a floating-point precision is necessary to maintain this advantage.

Calibration constants are derived during production, for each module under test and for each channel being monitored. These constants are then stored in EEPROM memory in the DS185X IC to be used later for calibration. An (n + 1) point calibration is necessary to extract these coefficients, where n is the order of the calibrating equation. For a linear fit, two coefficients are needed, one for gain and one for offset: y = ax +b. The user ramps his setup until signal point 1 (i.e., transmitted power) is achieved. X1 (the A/D raw count) is read; y1 is a value the standard recommends for a digital readout corresponding to signal point1. Similarly, x2 and y2 are determined for signal point 2. The tester derives a and b and stores them in memory in the DS185X IC.

Appendix A - Setting the Scaling Bits

The scaling factor for each input (V_{CC}, Vbin, Vpin, and Vrin) are 16 bits wide. They are located in Table 3 address C8h to CFh, respectively. The 16 bits are a combination of two trims. The lower 12 bits are binary weighted and give the high-resolution trim for scaling the input to output relationship. The upper four bits are a course adjust of the lower 12 bits. In other words, the upper four bits scale the LSB value of the binary weighted lower 12 bits.

Let's assume that your module has a maximum output power of 5mW. Adjust the laser until it is running at 5mW. The desired output reading is therefore 50,000 decimal (C350h).

1) Set the scale trim to 0FFFh (the upper four bits to all zeros and the lower 12 bits to all ones).

2) Use a SAR approach on the upper four bits, starting with 1000b, to find the smallest 4-bit trim necessary to cause the output reading to be above the desired value (in this case, greater than or equal to C350h). If the value exceeds C350, that is okay. That means 0000b is the needed value for the upper four bits. This step has now adjusted the LSB equivalent value in the lower 12 bits so that the best possible trim is acquired with the lower 12 bits. If too high a value is used in the four bits, then the resolution of the 12 bits is too high and absolute accuracy is sacrificed. If too low of a value is used in the four bits, then the resolution of the 12 bits is too small and a max reading is not possible and a large gain error is present through the entire range.

3) With the value found in step two (let's say it was 0010b), use a SAR approach on the lower 12 bits to finish trimming the parts.

Appendix B - Example Code

```
procedure writeTrimValue(trim :integer); (* write trim to part *)
procedure forceReference(Vin: real); (* applies reference voltage to input pin*)
procedure waitForNewConversion(); (* writes update byte to zero and waits for
        update of new conversion with present trim *)
function readValue():integer; (* reads converted digital answer *)
procedure ProgramTrim(trim :integer);(*enables EE for table3 & programs trim*)


Procedure calVad(maxInput :real)

var
      Bit           :integer; (* counter - represents bininary bit 15-0*)
      trim          :integer; (* present trim values *)
      lsb           :real; (* the lsb value of the voltage reading *)
      Vin           :real; (* the reference input voltage *)
      Dec_in        :integer; (* decimal representation of voltage input *)
      Dec_out       :integer; (* voltage reading in decimal *)
      delta         :integer; (* present error of voltage reading *)
      bestDelta     :integer; (* the error closest to zero *)
      bestTrim      :integer; (* the trim that gave the best delta *)

Begin
      bestDelta := 1000000;
      bestTrim := 0;
      trim := 4095; (* 0FFFh *)

      lsb := maxInput / 65535;

      Dec_in := 63888; (* input is ~97.5% of full scale *)

      Vin := lsb * Dec_in;

      forceReference(Vin);

(*Use SAR approach to trim course adjust on VAD scale bits 15-12*)
      for Bit := 15 downto 11 do (* must go 1 bit too far so that zero is
               possible solution *)
      begin
             if (Bit > 11) then
             begin
                    trim := trim + lshft(1,Bit);
```

```
            end;

            writeTrimValue(trim);

            waitForNewConversion;


            Dec_out := readVoltage;

            delta := Dec_out - Dec_in;

            if ( delta >= 0 ) then
            begin
                  bestTrim := trim;

                  if (Bit > 11) then
                  Begin
                        trim := trim - lshft(1,trimBit);
                  end;
            end;
      end;

(* The upper 4 bits of bestTrim contains the answer for the course adjustment.*)
(* It is the smallest value that allows full scale trim with the lower 12 bits*)

(* zero out the lower 12 bits of bestTrim *)
      trim := land(bestTrim,61440(*F000h*)); (*land() is logical AND function*)
(*Use SAR approach to trim resolution on VAD scale bits 11-0*)
      for Bit := 11 downto -1 do (* must go 1 bit too far so that zero is
            possible solution *)
      begin
            if (Bit > 0) then
            begin
                  trim := trim + lshft(1,Bit);
            end;

            writeTrimValue(trim);

            waitForNewConversion;

            Dec_out := readVoltage;

            delta := Dec_out - Dec_in;

            if ( abs(delta) <= abs(bestDelta) ) then
            begin
                  bestTrim := trim;
                  bestDelta := delta;
            end;

            if ( delta > 0 ) then
            begin
                  if (Bit > 0) then
                  Begin
                        trim := trim - lshft(1,trimBit);
                  end;
            end;
      end;
```

```
(* bestTrim now contains the trim that gives the lowest delta*)
      ProgramTrim(bestTrim);
end;
```

More Information

DS1852:	QuickView	-- Full (PDF) Data Sheet	-- Free Samples
DS1854:	QuickView	-- Full (PDF) Data Sheet	-- Free Samples
DS1857:	QuickView	-- Full (PDF) Data Sheet	-- Free Samples
DS1858:	QuickView	-- Full (PDF) Data Sheet	-- Free Samples