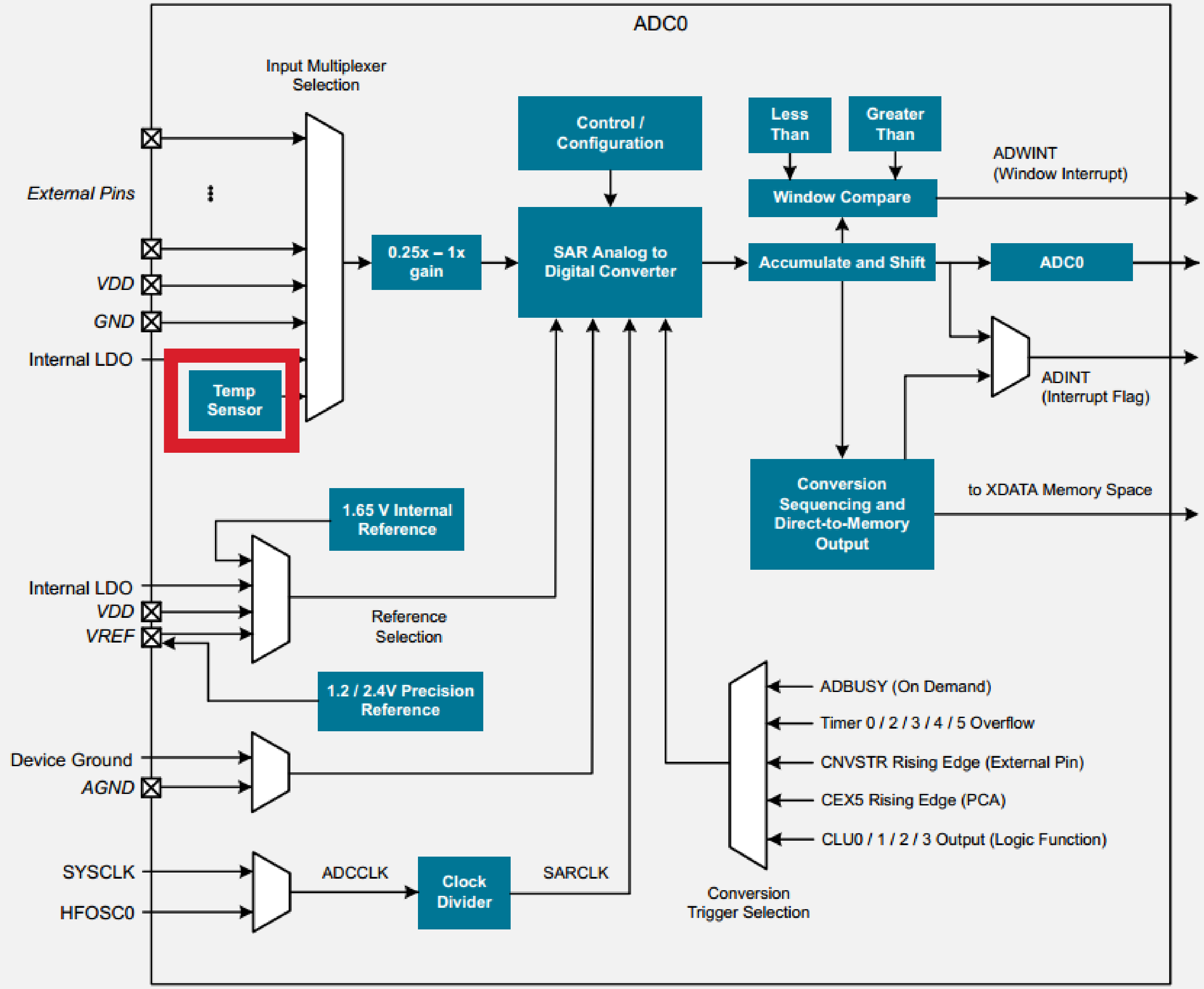
EFM8LB1 – Temp Sensor w/ Production Calibration

22 SEPTEMBER 2015

Agenda

- Temp Sensor Overview
- Production Calibration
- Temperature Calculation
- Numerical Example
- Software Example
  - EFM8LB1_TempSensor_WithCompensation
- Accuracy
- 2-Point calibration

Temp Sensor Overview



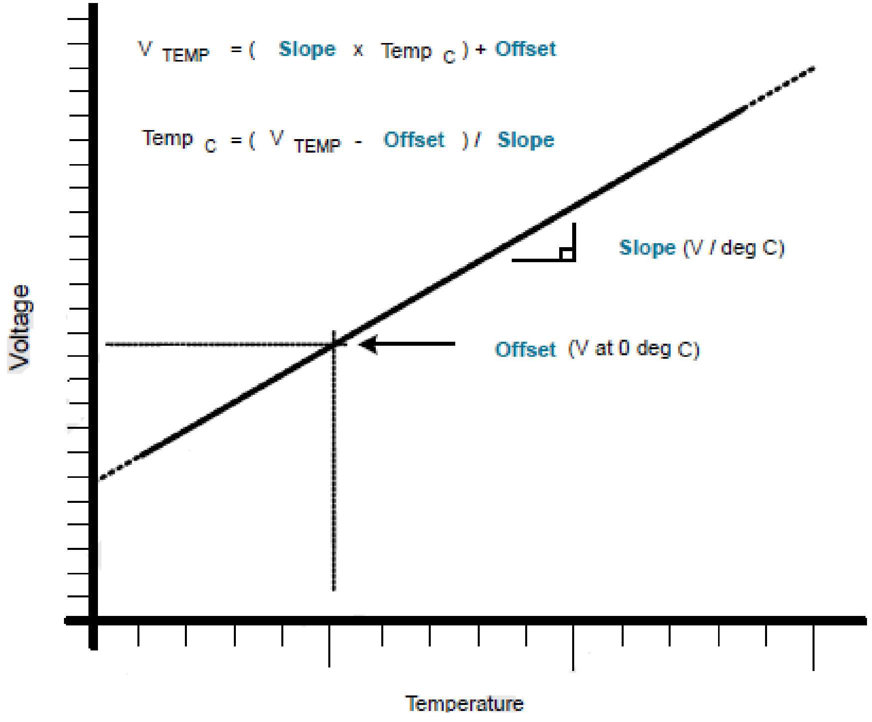
- Temp Sensor is available to the ADC input
- Selectable via input multiplexer

- LB1 ADC is 14-bit
- BB3 ADC is 12-bit

Production Calibration

$$Temp_C = (V_{temp} - Offset) / Slope$$

$$= (Value_{ADC} - Offset_{ADC}) / Slope_{ADC}$$



Temperature Sensor Transfer Function

- Only available on LB1
- The temp sensor is calibrated in production at 0 C
- Calibration setup
  - ADC in 14-bit mode
  - Right justified
  - 1.65V internal reference
  - Gain = 1x
- 14-bit value is stored in flash at 0xFFD4
  - For most devices, the value stored in Flash will have a decimal (base 10) value between 6800 (685mV) and 7900(796mV).
- With calibration, accuracy is about +/- 3 degrees C

Die Temperature Calculation

- The die temperature in Celsius is calculated by using the following formula:

$$T(C) = (ADC\ Sample - Temp\ Sensor\ 0C\ value) / 28$$

- Temperature change of 1 degree C will change ADC code by 28
- 28 corresponds to a temp sensor slope of ~2.82 mV per degree C
- 2^14*2.82/1650=28.001745
- (7900-6800)/28 = 39.3 degree C

Numerical Example

- Decimal notation used below

1. Calibration value stored in Flash = 7408
2. Average ADC output when measuring the temperature sensor = 8255
3. Die temperature = (8255 – 7408) / 28 = 30.25 degrees C.

Software Example

- EFM8LB1_TempSensor_WithCompensation example can be found in Simplicity Studio > Software Examples
- Configuration
  - ADC 14-bit mode
  - Accumulate 32 samples (2^5)
  - Right shift 3 to fit 16 bit ADC result register.
  - (hardware shift for accumulator mode, 2^14*32>>3 = 2^16)
  - The result is 16 bit at this stage.
- Run Time
  - Accumulated value is divided by 4 (2^2) in firmware to get the real 14bit result.
  - Final temperature result is 14-bit value
- The temperature output to PC through UART interface.

Software Example

- Die temperature is scaled by 100 for precision of 2 decimal places
  - e.g. 3212 represents 32.12 C
- Therefore, die temperature calculation is:

$$T(C*100) = (ADC\ Sample - Temp\ Sensor\ 0C\ value)*100/28$$

The type that save the value should larger than 16bit.

```
// Calculate the scaled temperature using the formula provided in the datasheet
temp_scaled = ((ADC_AVG - TEMPSENSOR_0C)*SCALE) / 28;

// Calculate the temp's whole number portion by unscaling
temp_whole = temp_scaled / SCALE;

// The temp_frac value is the unscaled decimal portion of temperature
temp_frac = temp_scaled % SCALE;

printf(" T = %ld.%d(C) \n", temp_whole, temp_frac);
```

Accuracy

- After offset calibration in production line, the accuracy is decided by slope error
- According the datasheet:, slope error is 25uV/C (1 delta of standard deviations)

Then the error will be 25uV*105*3=7.875mV for 105C, this corresponding to 6.375/2.835=2.778C

2-Points calibration

- If customer need to more accuracy, he could run 2 point calibration.
- He could get ADC data under another well controlled temperature point. Then combine this with the data in flash for 0C and get the slope.
- Assume the ADC value is Point2@TempC. Then the slope should be:
- If possible customer even could run multi point calibration. Collect lot of data at different temperature then use the least square method to get the offset/slope.

$$Slope = \left(Point2 - ADC\ \ Value\ \ 0C\right)/TempC$$

2-Points calibration

- Basically the calculation/theory is simple. But consideration the implementation on production line it is very hard to get highly reliable calibration result:
  - Usually the ADC reference for production calibration made by Silabs is based on well controlled external VREF. The internal VREF is not good reference candidate because it will change when temperature and power supply voltage changed.
  - The die temperature is also well controlled by putting the chip into oil bath.
  - It is difficult for customer to control the die temperature in production line. Customer could only control the ambient temperature.
  - There are many other factors that could contribute to the difference between ambient and die temperature, eg. Board layout, casing, heat dissipation, device power consumption, etc.

2-Points calibration

- I am not sure how the optical module user will utilize the die temperature. As far as I know some customer
  - need to use this temperature to estimate the temperature of laser emitter for power compensation purpose.
  - optical module is mounted on some field station, the module report temperature to operation/maintenance center through network for remote monitoring.
- If this is the case, I think it make more sense to get the slope and offset directly (between ADC value versus laser temperature)
- Then customer could use some other method to compensate the error cause by other factor (for example power consumption, power supply voltage, etc).

$$\mathrm{Temp}_{laser} = \left(V_{\mathrm{temp}} - Offset\right) / \, Slope$$

Thank you!