

Hi,

I would like someone to confirm that the temperature I am getting is somewhat in the ballpark to be expected. If it isn't, could someone check my values that I am using from the datasheet?

I am currently getting 11 C which seems low for ambient room temperature. I know that the ADC internal temperature measures junction temperature. I expected +/- 5 C swing, not -14 C.

I can also confirm my ADC is working with DMA as I am using it to read internal voltage and various other analog measurements.

For this particular microcontroller, the values are taken off the datasheet and are designed to spec:

6.3.16 Temperature sensor characteristics

Table 57. Temperature sensor calibration values

| Calibration value name | Description | Memory address |
|------------------------|--|---------------------------|
| TS_CAL2 | TS ADC raw data acquired at temperature of 130 °C ± 5 °C, V _{DDA} = 3 V ± 10 mV | 0x1FF8 007E - 0x1FF8 007F |

Table 58. Temperature sensor characteristics

| Symbol | Parameter | Min | Typ | Max | Unit |
|---------------------------------------|--|------|------|------|-------|
| T _L ⁽¹⁾ | V _{SENSE} linearity with temperature | - | ±1 | ±2 | °C |
| Avg_Slope ⁽¹⁾ | Average slope | 1.48 | 1.61 | 1.75 | mV/°C |
| V ₁₃₀ | Voltage at 130°C ± 5°C ⁽²⁾ | 640 | 670 | 700 | mV |
| I _{DDA(TEMP)} ⁽³⁾ | Current consumption | - | 3.4 | 6 | µA |
| t _{START} ⁽³⁾ | Startup time | - | - | 10 | µs |
| T _{S_temp} ⁽⁴⁾⁽³⁾ | ADC sampling time when reading the temperature | 10 | - | - | |

1. Guaranteed by characterization results, not tested in production.
2. Measured at V_{DD} = 3 V ± 10 mV. V30 ADC conversion result is stored in the TS_CAL1 byte.
3. Guaranteed by design, not tested in production.
4. Shortest sampling time can be determined in the application by multiple iterations.

I am using the low level driver function to determine temperature:

```
/** 
 * @brief Helper macro to calculate the temperature (unit: degree Celsius)
 * from ADC conversion data of internal temperature sensor.
 * @note Computation is using temperature sensor typical values
 * (refer to device datasheet).
 * @note Calculation formula:
 *     Temperature = (TS_TYP_CALx_VOLT(uV) - TS_ADC_DATA * Conversion_uV)
 *                 / Avg_Slope + CALx_TEMP
 *     with TS_ADC_DATA      = temperature sensor raw data measured by ADC
 *                       (unit: digital value)
 *     Avg_Slope       = temperature sensor slope
 *                       (unit: uV/Degree Celsius)
 *     TS_TYP_CALx_VOLT = temperature sensor digital value at
 *                       temperature CALx_TEMP (unit: mV)
 * Caution: Calculation relevancy under reserve the temperature sensor
 * of the current device has characteristics in line with
 * datasheet typical values.
 * If temperature sensor calibration values are available on
 * this device (presence of macro __LL_ADC_CALC_TEMPERATURE()),
 * temperature calculation will be more accurate using
 * helper macro @ref __LL_ADC_CALC_TEMPERATURE().
 * @note As calculation input, the analog reference voltage (Vref+) must be
 * defined as it impacts the ADC LSB equivalent voltage.
 * @note Analog reference voltage (Vref+) must be either known from
 * user board environment or can be calculated using ADC measurement
 * and ADC helper macro @ref __LL_ADC_CALC_VREFANALOG_VOLTAGE().
 * @note ADC measurement data must correspond to a resolution of 12bits
 * (full scale digital value 4095). If not the case, the data must be
 * preliminarily rescaled to an equivalent resolution of 12 bits.
 * @param __TEMPSENSOR_TYP_AVGSLOPE__ Device datasheet data: Temperature sensor slope typical value (unit: uV/DegCelsius).
 *                                     On STM32L0, refer to device datasheet parameter "Avg_Slope".
 * @param __TEMPSENSOR_TYP_CALX_V__ Device datasheet data: Temperature sensor voltage typical value (at temperature and Vref+ defined in parameter
 *                                     On STM32L0, refer to device datasheet parameter "V130" (corresponding to TS_CAL2).
 * @param __TEMPSENSOR_CALX_TEMP__ Device datasheet data: Temperature at which temperature sensor voltage (see parameter above) is correspondir
 * @param __VREFANALOG_VOLTAGE__ Analog voltage reference (Vref+) voltage (unit: mV)
 * @param __TEMPSENSOR_ADC_DATA__ ADC conversion data of internal temperature sensor (unit: digital value).
 * @param __ADC_RESOLUTION__ ADC resolution at which internal temperature sensor voltage has been measured.
 * This parameter can be one of the following values:
 * @arg @ref LL_ADC_RESOLUTION_12B
 * @arg @ref LL_ADC_RESOLUTION_10B
 * @arg @ref LL_ADC_RESOLUTION_8B
 * @arg @ref LL_ADC_RESOLUTION_6B
 * @retval Temperature (unit: degree Celsius)
 */
#define __LL_ADC_CALC_TEMPERATURE_TYP_PARAMS(__TEMPSENSOR_TYP_AVGSLOPE__, \
                                             __TEMPSENSOR_TYP_CALX_V__, \
                                             __TEMPSENSOR_CALX_TEMP__, \
                                             __VREFANALOG_VOLTAGE__, \
                                             __TEMPSENSOR_ADC_DATA__, \
                                             __ADC_RESOLUTION__)
((( \
    (int32_t)((__TEMPSENSOR_ADC_DATA__) * (__VREFANALOG_VOLTAGE__)) \
    / __LL_ADC_DIGITAL_SCALE(__ADC_RESOLUTION__)) \
    * 1000) \
    \
```

```

        (int32_t)(((_TEMPSENSOR_TYP_CALX_V__)
                    * 1000)
                  )
                ) / (_TEMPSENSOR_TYP_AVGSLOPE__)
                ) + (_TEMPSENSOR_CALX_TEMP__)
            )
}

```

My Code:

```

// Values derived from datasheet STM32L011x4
#define ADC_TEMP_AVG_SLOPE      1610
#define ADC_TEMP_TYP_CALX_V     670
#define ADC_TEMP_TYP_CALX_TEMP   130
#define ADC_TEMP_REF_VOLTAGE    3300

int16_t adc_get_temperature_internal(void)
{
    // Get Active Buffer (Opposite End of What the DMA is Filling)
    // Covert Values Into a Voltage.
    if(adc_active_buffer)
    {
        return __LL_ADC_CALC_TEMPERATURE_TYP_PARAMS(ADC_TEMP_AVG_SLOPE,
                                                      ADC_TEMP_TYP_CALX_V,
                                                      ADC_TEMP_TYP_CALX_TEMP,
                                                      ADC_TEMP_REF_VOLTAGE,
                                                      data[ADC_DMA_LOWER_TEMPERATURE_INTERNAL],
                                                      LL_ADC_RESOLUTION_12B);
    }
    else
    {
        return __LL_ADC_CALC_TEMPERATURE_TYP_PARAMS(ADC_TEMP_AVG_SLOPE,
                                                      ADC_TEMP_TYP_CALX_V,
                                                      ADC_TEMP_TYP_CALX_TEMP,
                                                      ADC_TEMP_REF_VOLTAGE,
                                                      data[ADC_DMA_UPPER_TEMPERATURE_INTERNAL],
                                                      LL_ADC_RESOLUTION_12B);
    }
}
?????????????????????????????????
```

Thanks!