

Temperature Calculation

The calculation of the CCD temperature is based upon the CY7 Series Silicon Diodes Standard Curve #10 from Omega Engineering. This curve is fit by a polynomial expansion based upon the Chebychev polynomials. This curve has four different fits based upon the temperature range required. Each fit uses eleven coefficients and differing values for the upper and lower voltage limits. The default values in CCDtool are set for the range of temperatures from 100 °K to 475 °K.

Recall that the Chebychev equation has the basic form:

$$T(x) = \sum_{n=0}^X a_n t_n(x)$$

where $T(x)$ is the temperature in Kelvin, a_n is the n^{th} coefficient, $t_n(x)$ is the n^{th} Chebychev polynomial, and x is the dimensionless voltage defined by

$$x = \frac{(V - V_L) + (V_U - V)}{(V_U - V_L)}$$

where V_U and V_L are the upper and lower voltage limits, respectively. The Chebychev polynomials were calculated using the recursion relation:

$$t_{n+1}(x) = 2x t_n(x) - t_{n-1}(x)$$

where $t_0(x) = 1$ and $t_1(x) = x$.

The DSP software determines the ADU count, not the voltage through the diode. It is therefore necessary to convert from ADUs to volts in order to make use of the above equations. The ADU count varies linearly with the voltage, so it is only necessary to determine the conversion factor from ADUs to volts (`ccdtool.setup.aduPerVolt` in the setup file) and the ADU offset (`ccdtool.setup.aduOffset` in the setup file). The default value used for `aduPerVolt` is 1366.98 and the value for `aduOffset` is 2045. These values were calibrated from the 2048x2048 CCD at Mount Laguna Observatory.

$$\text{Volts} = \frac{\text{adu} - \text{aduOffset}}{\text{aduPerVolt}}$$

Changing the Variables Used in Calculating the Temperature

The following variables can be changed within the setup file using your favorite text editor:

Variables associated with the Calculation of the CCD Temperature	
number of temperature coefficients	n = .setup.numberTemperatureCoefs
the temperature coefficients	a0 = .setup.temperatureCoeff0 a1 = .setup.temperatureCoeff1 a2 = .setup.temperatureCoeff2 a3 = .setup.temperatureCoeff3 a4 = .setup.temperatureCoeff4 a5 = .setup.temperatureCoeff5 a6 = .setup.temperatureCoeff6 a7 = .setup.temperatureCoeff7 a8 = .setup.temperatureCoeff8 a9 = .setup.temperatureCoeff9 a10 = .setup.temperatureCoef10
upper voltage limit	VU = .setup.Vupper
lower voltage limit	VL = .setup.Vlower
ADU per Volt	adu_per_volt = .setup.aduPerVolt
ADU offset	adu_offset = .setup.aduOffset

Currently, this is the only way to modify these values. The maximum number of coefficients that CCDtool is capable of handling is eleven (0 -- 10).

It is possible to implement a linear equation with the current temperature formula. The number of coefficients in this instance would be equal to 2 and only the a0 and a1 coefficients would be used to calculate the temperature.

Changing the Temperature Equation

Should there ever be a need to change the actual equations used to calculate the temperature, this is relatively easy to do. There are two source code files specific to calculating the temperature: include/temperature.h and tool/temperature.c. As long as the same basic format for the temperature function is followed and the setup variables above can be used, these are the *only* two files which need to be changed.

There are two functions associated with calculating the temperature: calc_temp_adu() and ccd_temperature(). The function calc_temp_adu() (calculate from temperature to adus) calculates the number of ADUs for a specific temperature. The basic procedure used in this function is to calculate that temperature in the middle of the upper and lower voltage limits and continue bisecting the segment which is closer to the target temperature within some tolerance defined to be 0.5°C. This function will generally not need to be changed since it uses ccd_temperature() to calculate the temperature.

The function `ccd_temperature()` calculates the temperature in degrees Celsius. This is the only function which needs to be changed for a new temperature formula. Some things to note in this function are 1) that the temperature variable is initialized to -273.15°C , which takes care of the conversion from Kelvin to centigrade; and 2) that the tag `DEBUG_TEMPERATURE` is not a normal compile switch -- it is used for debugging the temperature equation code.