**Overview**

The pmodTC1 library provides an interface to a MAX31855 Cold-Junction Compensated

Thermocouple-to-Digital Converter. The library interfaces with the MAX31855 to supply an easy to use instruction set. The data sheet for the MAX31855 can be found here.

http://datasheets.maximintegrated.com/en/ds/MAX31855.pdf

**Library Operation**

**Library Interface**

The header file TC1.h contains the TC1 class. This includes the class variables and function calls. To instantiate a TC1 object, include the TC1 library and instantiate an TC1 object.

**TC1 Initialization**

After power is given to the TC1, the device will immediately start outputting data in the form of 32 bit words. Below is the memory map of the output. (Taken from the MAX31855 data sheet)

|  |  |  |
| --- | --- | --- |
| **Bit** | **Name** | **Description** |
| D[31:18] | 14-Bit Thermocouple  Temperature Data | These bits contain the signed 14-bit thermocouple temperature value. |
| D17 | Reserved | This bit always reads 0. |
| D16 | Fault | This bit reads at 1 when any of the SCV, SCG, or OC faults are active. Default value  is 0. |
| D[15:4] | 12-Bit Internal Temperature  Data | These bits contain the signed 12-bit value of the reference junction temperature |
| D3 | Reserved | This bit always reads 0 |
| D2 | SCV Fault | This bit is a 1 when the thermocouple is short-circuited to VCC. Default value is 0. |
| D1 | SCG Fault | This bit is a 1 when the thermocouple is short-circuited to GND. Default value is 0. |
| D0 | OC Fault | This bit is a 1 when the thermocouple is open (no connections). Default value is 0. |

**TC1 Library Functions**

**TC1 Class**

**Public Functions**

**ACL2()**

Parameters:

None

Return Value:

None

Constructor for class TC1.

**begin(**int CS**)**

Parameters:

int CS chip select pin for SPI communications

Return Value:

None

This function starts the SPI communication and stores the desired chip select.

**double getTemp()**

Parameters:

None

Return Value:

temp thermocouple temperature value in Celsius

This function extracts the 14 bit thermocouple temperature data and converts it into Celsius to return to the user.

**double getAMBTemp()**

Parameters:

None

Return Value:

temp on chip temperature value in Celsius

This function extracts the 12 bit internal temperature data and converts it into Celsius to return to the user.

**readData()**

Parameters:

None

Return Value:

none

This function reads the 32-bit value from the MAX31855 and stores it into the data class member “data”.

**Uint16\_t getFault()**

Parameters:

None

Return Value:

uint16\_t fault fault code from the MAX31855

This function will extract the fault data from the data word and return to fault code.

**double celToFar(double celsius)**

Parameters:

celsius value to convert to farenheit

Return Value:

double farenheit converted farenheit value

This function converts a temperature in celsius to farenheit

**Class members**

|  |  |
| --- | --- |
| **Name** | **Description** |
| uint32\_t data | Data word from MAX31855 |
| static int chipSelect | Chips select pin to be used by SPI operations |
| int fault | High if a fault exists |
| int SCV | This bit is a 1 when the thermocouple is short-circuited to VCC. Default value is 0. |
| int SCG | This bit is a 1 when the thermocouple is short-circuited to GND. Default value is 0. |
| int OC | This bit is a 1 when the thermocouple is open (no connections). Default value is 0. |