IO Shield Library : Temperature Module

Revision: August 3, 2011

1300 NE Henley Court, Suite 3
Pullman, WA 99163
(509) 334 6306 Voice | (509) 334 6300 Fax

Introduction

This Document describes the functions written to interface with Microchips TCN75A temperature sensor. The TCN75A has a range of -40ºC - 125ºC with precision ranging from .5ºC - .0625ºC depending user configurable resolution. All temperature data is in ºC. This document only goes over the basic information to properly use the functions. Therefore if more information on the device is needed please refer to the products document at the following location:
http://ww1.microchip.com/downloads/en/DeviceDoc/21935D.pdf

This library makes use of the existing Wire library to communicate with the EEPROM on the IOShield. As such, when you write sketches that use the IOShieldEEPROM library, make sure that you also include Wire.h in your sketch. Note that while many of the data types used in this library are not common Arduino/MPIDE datatypes, using an int or byte in their place will be converted to the correct type when the sketch is built.

Note: In order to use the chipKit Max user must manually connect SDA and SCL pins(20 and 21) to pins A4 and A5 of IO Shield

Note: For chipKit Uno Users you must have Jumpers JP6 and Jp8 set in the RG3 and Rg2 positions

The following API functions make up the Temp Module interface.

void config(uint8_t configuration)

Parameters
configuration - Value to be written to config register

This function writes the configuration register with the given value. There are a number of defined values as described below that can be or'd together to set multiple parameters. For example if one wished to put the device in one shot mode and use 12 bit resolution the following call could be made.

```
Config(ONESHOT | RES12)
```

```
IOSHIELDTEMP_ONESHOT        0x80  //One Shot mode
IOSHIELDTEMP_RES9           0x00  //9-bit resolution
IOSHIELDTEMP_RES10          0x20  //10-bit resolution
IOSHIELDTEMP_RES11          0x40  //11-bit resolution
IOSHIELDTEMP_RES12          0x60  //12-bit resolution
IOSHIELDTEMP_FAULT1         0x00  //1 fault queue bits
IOSHIELDTEMP_FAULT2         0x08  //2 fault queue bits
IOSHIELDTEMP_FAULT4         0x10  //4 fault queue bits
IOSHIELDTEMP_FAULT6         0x18  //6 fault queue bits
IOSHIELDTEMP_ALERTLOW       0x00  //Alert bit active-low
IOSHIELDTEMP_ALERTHIGH      0x04  //Alert bit active-high
IOSHIELDTEMP_CMPMODE        0x00  ///comparator mode
```

```
IOSHIELDTEMP_INTMODE        0x02  //interrupt mode
IOSHIELDTEMP_STARTUP        0x00  //Shutdown disabled
IOSHIELDTEMP_SHUTDOWN       0x01  //Shutdown enabled
IOSHEIDLTEMP_CONF_DEFAULT         //Power up initial configuration
```

For descriptions on all of the modes please refer to page 24 of the data sheet were the modes are described in detail.

float getTemp()

Parameters
None

Return
float - current temperature measured in Celsius.

This function retrieves the current temp from the temp sensor and converts the returned value into a signed floating point value.

void setTempHyst(float tMin)

Parameters
tMin - Alert reset temperature in Celsius

This function sets the hysteresis value on the temperature module. This is a temperature (rounded to the nearest half degree Celsius) representing the point that resets the alert pin. When the temperature passes the value defined with setTempLimit the alert pin is asserted, and it remains asserted until the temperature falls below the value tMin defined here.

The limits of the range supported by this routine are defined in IOSHIELDTEMP_MAX and IOSHIELDTEMP_MIN.

void setTempLimit(float_t tMax)

Parameters
tMax - Alert assertion temperature in Celsius

This function sets the limit temperature on the temperature module. This is a temperature (rounded to the nearest half degree Celsius) representing the point that asserts the alert pin. When the temperature passes this point the alert pin will be asserted, and it will remain asserted until the temperature falls below the value tMin defined in setTempHyst.

The limits of the range supported by this routine are defined in IOSHIELDTEMP_MAX and IOSHIELDTEMP_MIN.

float convCtoF(float tempC)

Parameters
tempC - an arbitrary temperature in degrees Celsius.

Returns
float - the representation of the input temperature in Fahrenheit.

This routine takes in a temperature in degrees Celsius and converts it to Fahrenheit.

float convFtoC(float tempF)

Parameters
tempF - an arbitrary temperature in degrees Fahrenheit.

Returns
float - the representation of the input temperature in Celsius.

This routine takes in a temperature in degrees Fahrenheit and converts it to Celsius.