# SMBus Temperature Sensing and Software Visualization

### Objective

Learn to use the IR sensor for measuring object “temperature”[[1]](#footnote-1) as well as more experience with Percepio Tracealyzer and Doxygen.

### Specifics

* IR Sensor[[2]](#footnote-2) – Attach to J7 (*I2C1*) such that the green wire is attached to *SCL1* and the red wire is attached to 3V3. (See p. 17 of the Cerebot RM.) ***Do not write to the IR Sensor!!!*** Instead, follow Figure 8 from the IR datasheet to read the temperature, and then follow p. 27 to determine the object (TOBJ1) temperature in degrees Fahrenheit. (Beware of integer arithmetic and sign-extension.)
* Project Requirements.
	+ Implement a CN task that generates a CN interrupt by setting the CN interrupt flag every 6 ms, using TaskDelayUntil(). Task priority 1.
	+ Implement a CN ISR that unblocks a handler task using FreeRTOS direct task notification. PIC32 interrupt priority 2.
	+ Implement a handler task that is unblocked by the CN ISR, reads the object temperature from the IR sensor, forms a string for the LCD to display the temperature in Fahrenheit (“Temp=XXX.X”), and loads a pointer to the local string buffer into a FreeRTOS message buffer. Task priority 2.
	+ Implement an LCD task that reads the pointer from the message buffer and displays the message on the LCD. Task priority 1.
	+ Blink task: toggle LEDA every 3 ms. Task priority 3.
* Follow the Barr Group’s advice on “[Bug-Killing](https://barrgroup.com/Embedded-Systems/How-To/Bug-Killing-Standards-for-Embedded-C)” standards, as well as their file templates[[3]](#footnote-3) for headers and source code, but use @author instead of @par[[4]](#footnote-4). As you develop your program, use the following Doxygen features on every function that **you** write: @brief, @param (include direction), and @return. Add additional C comments (//) as needed. Fill in the “Project Information” in the Doxygen GUI prior to generating the Doxygen PDF output. Project logo optional. ☺
* Add [Tracealyzer](https://percepio.com/tz/freertostrace/) capability to your project and capture activity to include in your report. Save the BIN files.
* Please submit a “brief” report. Rather than describing the project functionality or “design process” focus on how you used Tracealyzer to verify and characterize your projects performance.
* The MPLAB project folder, Tracealyzer BIN file, and Doxygen PDF output should be packaged into a single zip archive and shared via OneDrive by the due date on the web.
* Please upload to Canvas a “brief” report. Rather than describing the project functionality or “design process” focus on how you used Tracealyzer to verify and characterize your project’s performance.

Questions? Just ask!

1. Technically, Infrared radiation, not temperature [↑](#footnote-ref-1)
2. A program for testing your sensor is provided under [Handouts/SMBus/IR Sensor Test Program](http://www.mrc.uidaho.edu/mrc/people/jff/443/Handouts/SMBus/IR%20Sensor%20Test%20Program/) [↑](#footnote-ref-2)
3. C:\Users\jfrenzel\OneDrive - University of Idaho\web\443\Handouts\Programming\Coding Standards [↑](#footnote-ref-3)
4. There is a special Doxygen command if you want to copyright your code. ☺ [↑](#footnote-ref-4)