/\*

\* Copyright (c) 2007, Swedish Institute of Computer Science.

\* All rights reserved.

\*

\* Redistribution and use in source and binary forms, with or without

\* modification, are permitted provided that the following conditions

\* are met:

\* 1. Redistributions of source code must retain the above copyright

\* notice, this list of conditions and the following disclaimer.

\* 2. Redistributions in binary form must reproduce the above copyright

\* notice, this list of conditions and the following disclaimer in the

\* documentation and/or other materials provided with the distribution.

\* 3. Neither the name of the Institute nor the names of its contributors

\* may be used to endorse or promote products derived from this software

\* without specific prior written permission.

\*

\* THIS SOFTWARE IS PROVIDED BY THE INSTITUTE AND CONTRIBUTORS ``AS IS'' AND

\* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE

\* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE

\* ARE DISCLAIMED. IN NO EVENT SHALL THE INSTITUTE OR CONTRIBUTORS BE LIABLE

\* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL

\* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS

\* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)

\* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT

\* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY

\* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF

\* SUCH DAMAGE.

\*

\* This file is part of the Contiki operating system.

\*

\*/

/\*\*

\* \file

\* Sensing temperature, humidity and light intensity in the vicinity of Tmote-sky

\* \author

\* Adam Dunkels <adam@sics.se>

\*/

#include "contiki.h"

#include "dev/light-sensor.h"

#include "dev/sht11/sht11-sensor.h"

#include <stdio.h>

#include <math.h>

/\*---------------------------------------------------------------------------\*/

PROCESS(sensor\_acq\_process,"Sensor Acquisition");

AUTOSTART\_PROCESSES(&sensor\_acq\_process);

PROCESS\_THREAD(sensor\_acq\_process,ev,data)

{

static struct etimer et;

static int val;

static float s = 0;

static int dec;

static float frac;

PROCESS\_BEGIN();

printf("Starting Sensor Example.\n");

while(1)

{

etimer\_set(&et, CLOCK\_SECOND \* 2);

SENSORS\_ACTIVATE(light\_sensor);

SENSORS\_ACTIVATE(sht11\_sensor);

PROCESS\_WAIT\_EVENT\_UNTIL(etimer\_expired(&et));

val = sht11\_sensor.value(SHT11\_SENSOR\_TEMP);

if(val != -1)

{

s= ((0.01\*val) - 39.60);

dec = s;

frac = s - dec;

printf("\nTemperature=%d.%02u C (%d)\n", dec, (unsigned int)(frac \* 100),val);

}

val=sht11\_sensor.value(SHT11\_SENSOR\_HUMIDITY);

if(val != -1)

{

s= (((0.0405\*val) - 4) + ((-2.8 \* 0.000001)\*(pow(val,2))));

dec = s;

frac = s - dec;

printf("Humidity=%d.%02u %% (%d)\n", dec, (unsigned int)(frac \* 100),val);

}

val = light\_sensor.value(LIGHT\_SENSOR\_TOTAL\_SOLAR);

if(val != -1)

{

s = (float)(val \* 0.4071);

dec = s;

frac = s - dec;

printf("Light=%d.%02u lux (%d)\n", dec, (unsigned int)(frac \* 100),val);

}

etimer\_reset(&et);

SENSORS\_DEACTIVATE(light\_sensor);

SENSORS\_DEACTIVATE(sht11\_sensor);

} //end of while

PROCESS\_END();

}