

```

/*Name: Darren Smith
/*URN: 6553517
/*Assignment 2: Aggregation Algorithm
/*Date: 3th December 2019
/*
/*
*/

#include "contiki.h"

#include <stdio.h>          /* For printf() */
#include "dev/light-sensor.h" //Light Sensor header file
#include "dev/sht11-sensor.h" // Temperature and Humidity Header File
#include <math.h>

/*-----*/
unsigned short d1(float f) // Integer part
{

    return((unsigned short)f);
}

unsigned short d2(float f) // Fractional part
{

return(1000*(f-d1(f)));
}

/*-----*/
PROCESS(sensor_reading_process, "Sensor Reading process");
AUTOSTART_PROCESSES(&sensor_reading_process);

/*-----*/
//Begin Proccess

PROCESS_THREAD(sensor_reading_process, ev, data)
{
/*-----*/
-----*/
/*Setting the array for Variance , Mean */

    static struct etimer timer; // Set a timer for sensor reading process

    int Variance(int var[], int n)
    {

```

```

        int v;

        int sum = 0, sum1 = 0, x_;

        for(v = 0; v<n; v++) sum += var[v];

        x_ = sum/n;

        for(v=0; v<n; v++)

            sum1 += (var[v]-x_)*(var[v]-x_);

        return(sum1);

    }

/*-----*/
-----*/
    PROCESS_BEGIN();

    etimer_set(&timer, 2 * CLOCK_SECOND); //Configuring Time

    SENSORS_ACTIVATE(light_sensor); //Activate the light sensor
    SENSORS_ACTIVATE(sht11_sensor);

    while(1) //start of the while loop
    {
        PROCESS_WAIT_EVENT_UNTIL(ev=PROCESS_EVENT_TIMER); // Wait4Time
/*-----*/
-----*/
/*Set Parameters for the looping of the buffer array , Variance and Standard
Deviation */
        int j, std ,buffer[12];

        for (j = 0; j < 12; j++)
        {

            float V_sensor =
1.5*light_sensor.value(LIGHT_SENSOR_PHOTOSYNTHETIC)/4096;
            float I = V_sensor/100000;
            float light_lx = 0.625*1e6*I*1000;

            buffer[j] = light_lx;
        }
        printf("\n");
        printf("B = { ");

/*-----*/
/*Variance*/
        for (j =0; j < 12; j++)
        {

```

```

        printf("%d, ",buffer[j]);
    }
    printf("}\n");

    printf("Variance = %d\n",Variance(buffer,j));
/*-----*/
/*Finding the Standard Deviation*/

    for(j = 0; j <12; j++)
    {

        std =  sqrtf(Variance(buffer,j));

    }
    printf("Std = %u.%03u\n" , d1(std) , d2(std));

    // Aggerate the values
/*-----*/
    int A;

    A = Variance(buffer,j);

    int set[6];
    if(A>45)
    {
        printf("Number of values to aggregate = 2\n");
        set[0]=(buffer[0] + buffer[1])/2;
        set[1]=(buffer[2] + buffer[3])/2;
        set[2]=(buffer[4] + buffer[5])/2;
        set[3]=(buffer[6] + buffer[7])/2;
        set[4]=(buffer[7] + buffer[9])/2;
        set[5]=(buffer[10] + buffer[11])/2;

        printf("X = {%d, %d, %d, %d, %d, %d}\n",set[0],set[1],set[2],set[3],set[4],set[5]);
        printf("\n");

    }
    else
    {
        printf("Number of values to aggregate = 4\n");

        set[0]=(buffer[0]+buffer[1]+buffer[2]+buffer[3])/4;

        set[1]=(buffer[4]+buffer[5]+buffer[6]+buffer[7])/4;

        set[2]=(buffer[8]+buffer[9]+buffer[10]+buffer[11])/4;
        printf("X = {%d, %d, %d}\n",set[0],set[1],set[2]);
        printf("\n");

    }

/*-----*/

```

```
        etimer_reset(&timer); //Reset the Timer
    }
    PROCESS_END();
}
```