This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement **no 619912**

Deliverable D1.1.5
Integration of sensors and control strategies report

Project Acronym	**HI-LED**
Project title	**Human-centric Intelligent LED engines for the take up of SSL in Europe**
Website	www.hi-led.eu

Collaborative project

FP7-ICT-2013-11

Start date of project: 01.12.2013

Duration: 36 months

Project coordinator: Dr. Josep Carreras Molins

Project coordinator organization name: IREC

Tel: + 34 933 562 615.

Fax: + 34 933 563 802.

Mail: jcarreras@irec.cat

This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement **no 619912**

D1.1.5 – Integration of sensors and control strategies report

Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Document details	
Responsible beneficiary of this Deliverable	ALL
Work package	WP1
Authors	Oscar Motto, Wim Hertog, Dr. Perálvarez, Dr. Higuera, Dr. Josep Carreras
Other contributors	
Document ID	D 1 1 5 - Integration of sensors and control strategies report - HILED - 619912
Date	18/11/14

Revision history			
Version	Date	Author	Comments
1	18/11/2014	Oscar Motto	
2	15/12/2014	Jorge Higuera Portilla	
3	22/12/2014	Mariano Perálvarez	

This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement **no 619912**

CONTENTS

This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement **no 619912**

List of abbreviations

PCB:	Printed Circuit Board.
EEPROM:	Electrically Erasable Programmable Read-Only Memory.
ADC:	Analog to Digital Converter.
USART:	Universal Synchronous-Asynchronous Receiver-Transmitter.
SPI:	Serial Programming Interface.
I2C:	Inter-Integrated Circuit.
LED:	Light-Emitting Diode
IC:	Integrated Circuit.
PWM:	Pulse Width Modulation.
UART:	Universal Asynchronous Receiver-Transmitter.
GPIO:	General Purpose Input/Output.
PWR:	Power.
RX:	Reception.
TX:	Transmission.
GND:	Ground.
SMBus:	System Management Bus
LED:	Light-Emitting Diode
RPM:	Revolutions per minute
MOEMS:	Micro-Opto-Electro-Mechanical Systems
CMOS:	Complementary metal–oxide–semiconductor
SNR:	Signal to noise ratio
PID:	Proportional-integral-derivative controller
TEDS:	Transducer Electronic DataSheet

This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement **no 619912**

SECTION 1. Introduction

This deliverable focuses on providing an overview of the different methodologies in order to compensate variations in spectra (amplitude and wavelength) of the LED channels in basis of ageing and temperature measurements. How to detect and compensate these variations is the reason of this work.

First of all, we describe two different selected elements to detect variations on the LED performance: a temperature sensor and mini-spectrophotometer. The temperature sensor will help us to know the LED performance based on the temperature profile. On the other hand, the spectrophotometer will give us a full vision in real time of the generated synthetic spectrum allowing us to compare the response with the theoretical optimum output spectrum that has been selected.

Finally, in this document are explained the algorithms implemented in order to solve variations in amplitude and wavelength shifts. In addition, the document provides different strategies in order to compensate wavelength displacement in real time and how fix this problem on the hardware platform.

This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement **no 619912**

SECTION 2. Temperature sensor

The performance in a high-power LED is a direct function of its temperature. In a typical use situation, the forward voltage and the applied current diode produces a luminous flux. This light intensity varies with the current. However, the forward voltage-current relationship is also a function of the heat sink or board temperature. In this case, the luminous flux decreases when increases the temperature above operational limit. This data is usually found in the LED vendor datasheet in chart form. For that reason, it is important to measure the temperature of the LED PCB in order to compensate deviations in wavelength and amplitude of the LEDs in real time.

Spectral shifts due to temperature instabilities are managed by a microcontroller that reads temperature with an external digital temperature sensor. The MCP98242 from Microchip has been selected to monitor these temperature fluctuations in the LED front-end avoiding early faults related to overheating and to extend the LED lifetime. This sensor has a dynamic range from -40°C to +125°C to convert temperature in a digital word. It provides a resolution from 0.0625°C to 0.5°C (minimum/maximum). In addition, this sensor has an internal EEPROM memory containing 256 bytes which can be used to store metadata information related to selected market applications (horticulture, human centric lighting and artwork). The MCP98242 digital temperature sensor includes different user-programmable registers that provide flexibility for temperature-sensing applications. The registers allow user-selectable settings such as shutdown or low-power operational modes and the specification of different temperature events and critical output boundaries. When the temperature goes beyond a specified boundary limit, the MCP98242 triggers an event signal. The user has the option of setting the event output signal polarity as either an active-low or active-high comparator output for thermostat operation, or as a temperature event interrupt output for embedded systems. The event output can also be configured as a critical temperature output. This sensor has an industry standard 2-wire, I2C/SMBus compatible serial interface, allowing up to eight devices to be controlled in a single serial bus. To maintain interchangeability with the I2C/SMBus interface the electrical specifications are an operating voltage of 3.0V to 3.6V. In addition, a temperature conversion time between 30 ms to 260 ms is set depending on temperature resolution selection.

Apart from compensate the different channels bearing in mind temperature fluctuations; we can avoid overheating by turning off all the LED channels when temperature reaches certain limit; or in case to use a fan as active cooling method, we could manage the revolutions per minute (RPM) in a fan according temperature profile.

Figure 1 and Table 1 shows the experimental characterization of the LED engine based on three different channels (450 nm, 535 nm, 620 nm).

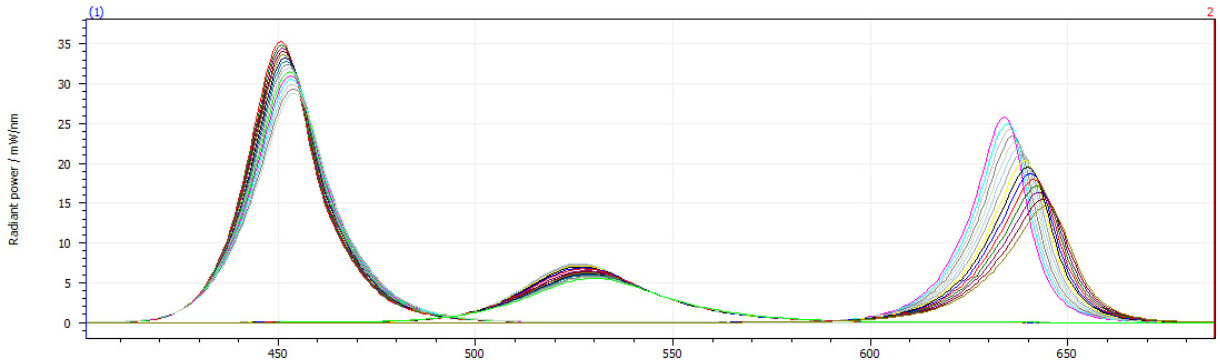

Figure 1. Spectral shifts due to the temperature instabilities

In figure 1 are shown amplitude deviations in radiant power (mW/nm) when temperature increases from 10 °C to 80 °C in the LED front end.

Table 1 shown peak deviations of the LED front-end based on experimental measurements of temperature in Luxeon Rebel LEDs maintaining a current of 350mA.

Table 1. Peak deviation vs. temperature fluctuations in three different LED channels

Temperature (°C)	Ch 1 (450 nm) Peak deviation (nm)	Ch 5 (535 nm) Peak deviation (nm)	Ch 10 (620) Peak deviation (nm)
10	450.72	526.11	633.76
15	450.91	526.33	634.45
20	451.14	526.64	635.24
25	451.36	526.83	635.98
30	451.5	527.08	636.65
35	451.85	527.31	637.41
40	452.06	527.62	638.19
45	452.31	527.93	638.89
50	452.56	528.18	639.71
55	452.81	528.42	640.46
60	453.05	528.71	641.25
65	453.27	529.03	641.98
70	453.52	529.35	642.74
75	453.8	529.54	643.57
80	454.07	529.85	644.34

The peak wavelength deviation in the channel of 450 nm was near 5 nm when temperature rises to 80 °C. Also, peak deviation in LED channel of 535 nm was only 4nm but peak deviation rise to 10 nm in channel of 620 nm.

This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement **no 619912**

SECTION 3. Spectrophotometer

This section presents a feedback loop designed to detect and control spectral shifts due to LED aging and temperature instabilities based on a spectrophotometer adapted in the same LED front-end. An experimental setup evaluates the performance of two different types of mini-spectrophometers C12666MA-(X) as compared to C10988MA-01in the visible spectrum.

3.1 Mini-spectrophotometer

In order to detect small fluctuations in wavelength components, it is important to introduce a sensor element as feedback between the real time spectral reproduction and the LED response. In this case, a spectrophotometer is adapted in the LED array as a close loop system to monitor in real time the spectral reproduction. The evaluated mini-spectrophotomers incorporate two different technologies: Micro-Opto-Electro-Mechanical Systems (MOEMS)[1], and CMOS image sensor technologies [2]. The CMOS image sensor sub-module in Figure 2 is integrated with the light receiving slit. By other hand, the MOEMS sub-module consists of a convex lens on which the diffraction grating is formed by nano-imprint.

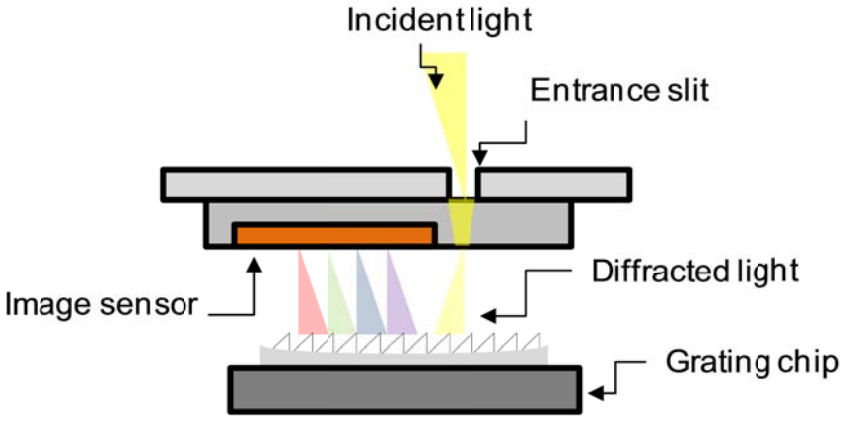

Figure 2. Mini-spectrophometer architecture

For the C10988MA-01, the theoretical spectral response range is between 340 nm to 750 nm with a spectral maximum resolution of 11 nm. Figure 3 shown C10988MA-01 spectral sensitivity responses in visible spectrum and figure 4 shown the spectral resolution @ air temperature of 25 °C.

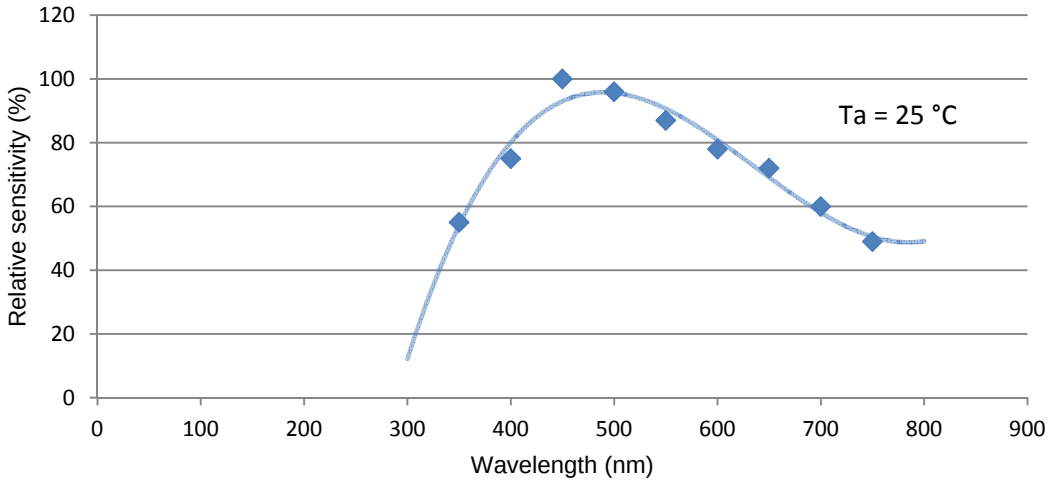

Figure 3. C10988MA-01 Relative sensitivity @ Ta= 25°C (340nm - 780nm)

In Figure 3 relative sensitivity changes with wavelength. For the range between 435 nm and 525 nm the sensitivity is above 90%.

Figure 4 shown spectral resolution of C10988MA-01 when air temperature is 25 °C. Spectral resolution in all of cases is between 9nm to 11 nm.

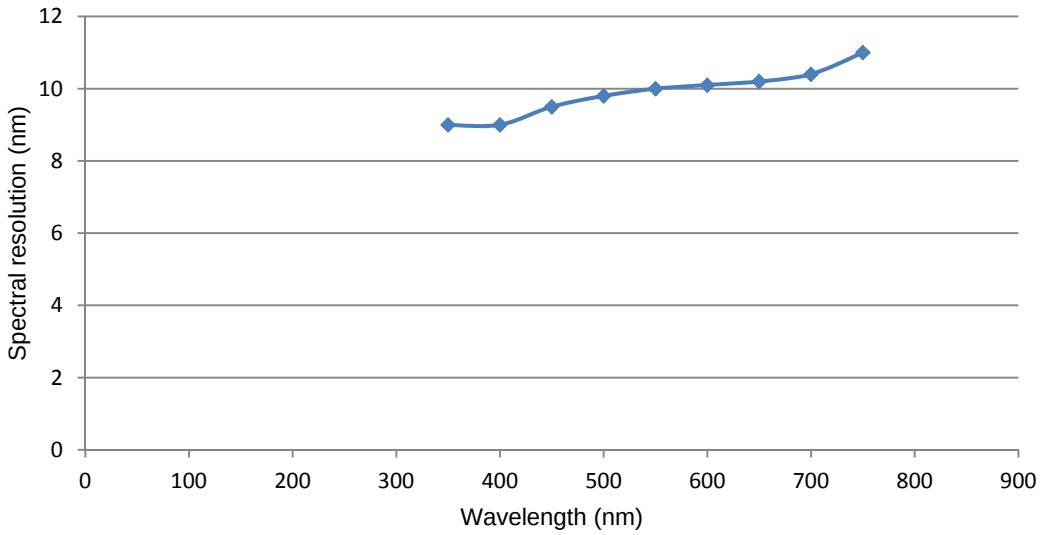

Figure 4. C10988MA-01 Spectral resolution @ Ta= 25°C (340nm - 780nm)

3.2 Hardware interface

The interface used to communicate with the analog sub-module of the spectrometer is showed in Figure 5.

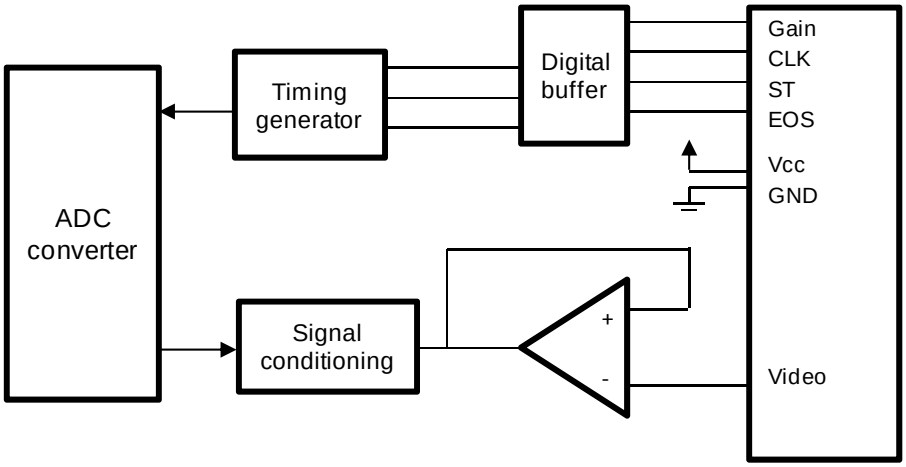

Figure 5. Hardware interface used to communicate with the analog mini-spectrometer module

The input video signal is amplified by 1.5 with an adaptation circuit as shown in Figure 6.

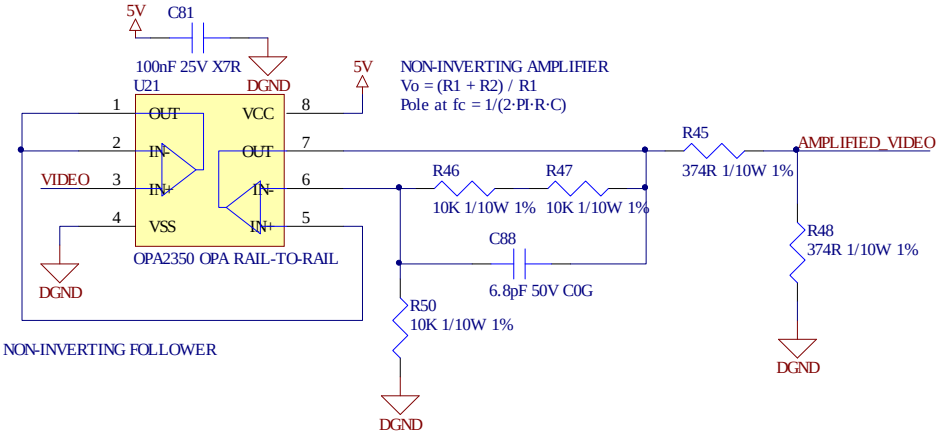

Figure 6. Signal conditioning amplifier module

Once we get the amplified signal, we have used an analog to digital converter (ADC) with 12 bit resolution ADC in order to get a digital value between (1ms to 100 ms). The Gain and clock frequency parameters are configured in basis of the previously received spectrum in order to get all dynamic range of the input spectrum. All digital I/O signals are buffered through voltage level translator from 3v3 to 5v as shown in Figure 7 before to be managed by the microcontroller unit.

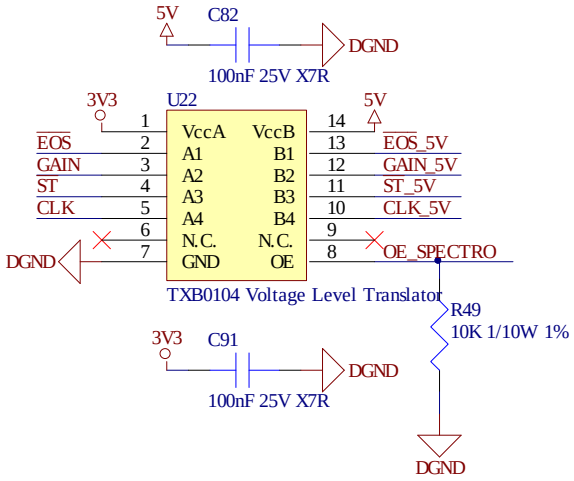

Figure 7. Analog to Digital Converter module

3.3 Mini-spectrophotometer: experimental characterization

The figure 8 shows an experimental setup to evaluate both C10988MA-01 and C12666MA-(X) mini-spectrophometers. Three different wavelengths (450nm, 525 nm and 660 nm) and 3 different currents (100 mA, 286 mA, and 1050 mA) were choices to study spectral range, spectral sensitivity (SNR) and dynamic range.

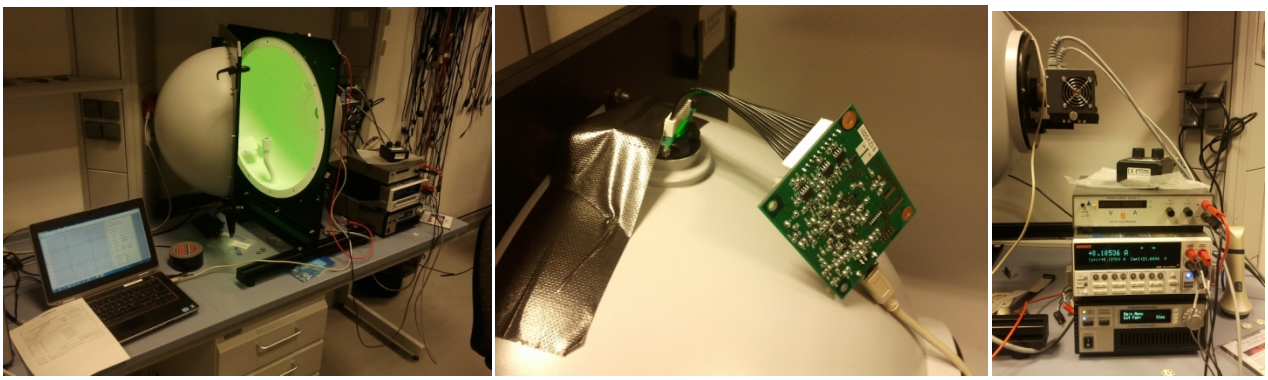

Figure 8. (a) General overview of the setup. (b) minispectrometer attached to a secondary port of the sphere. (c) current sources and measurement equipment

We have used a calibrated (with junction temperature control) integrating sphere. The LEDs were placed on the input port whereas the mini-spectrometers were located on a secondary port of the sphere. Both Figure 9 and Figure 10, shows experimental values of SNR for two different mini-spectrophotometers in three different wavelengths and currents.

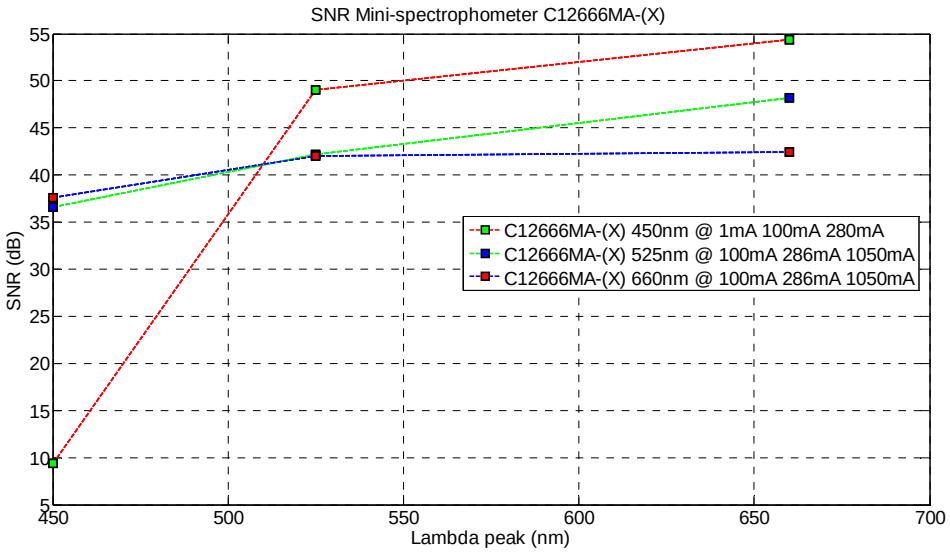

Figure 9. C12666MA-(X) Signal to noise response (SNR)

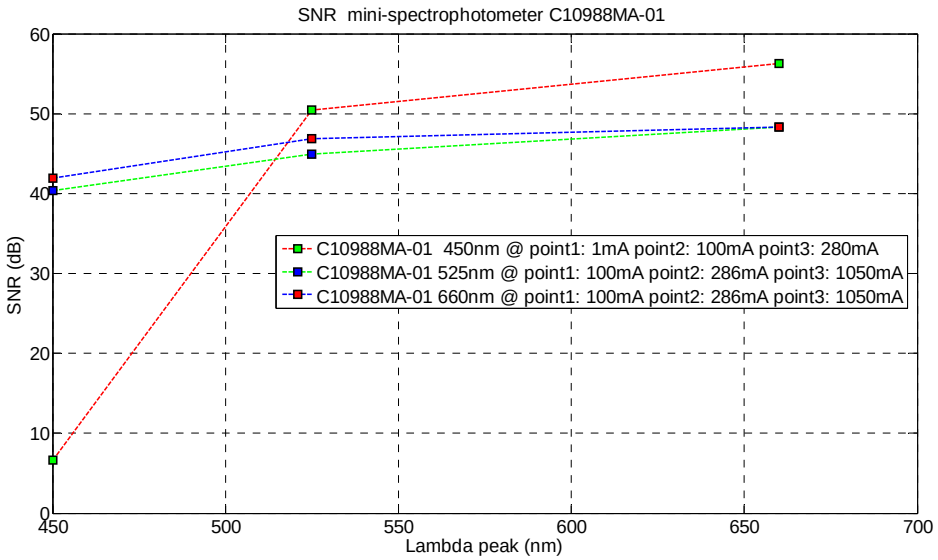

Figure 10. C10988MA-01 Signal to noise response (SNR)

Experimental results in Figure 9 and 10 shows that the C12666MA-(X) is a little less sensitive in the wavelengths studied (450nm -700nm).

Spectral range: Both spectrometers C10988MA-01 and C12666MA-(X) have demonstrated to have similar spectral range characteristics. Noteworthy, the C12666MA-(X) has an increased wavelength

range [311 nm to 801 nm] as compared to [330 nm to 769 nm], (about 12% more) of the C10988MA-01 model, however both sensors have the same number of pixels (254).

Dynamic range: Both models have considerable DC offset added to the spectral output. This, in practice, limits the dynamic range because of the dark signal that depends on the integration time. The C10988MA-01 model saturates earlier for the same conditions (light levels, gain and integration time). However, from a signal-to-noise ratio perspective, this not represents a disadvantage because both preserve similar SNR response.

This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement **no 619912**

SECTION 4. Close-loop algorithms

This section is devoted to explain three different algorithms to control spectral reproduction stability in order to compensate wavelength displacement in real time and how fix this problem on the hardware platform. The first method is an algorithm based on PID controllers [3], mainly because of their effectiveness and simple structure. In this case, a proportional-integral-derivative controller (PID) will be tuned for each LED channel. The second method includes an electronic transducer datasheet (TEDS) syntactic structure organized as a look-up table to store all calibration values according an auto-configuration sequence. Finally, the third method is an algorithm based on the wavelength displacement according a self-tuning peak detector engine.

4.1 PID Controller

The PID controller includes a close loop feedback network in order to calculate error values as the difference between a measured spectrum and a reference wavelength sequence. The PID controller contains three different parameters defined as proportional, integral and derivative values. Each proportional value actuates in present error, however, the integral value acts only on accumulated past errors and finally, the derivative value is an influence on future error prediction that depend on the current signal rate of change. The tasks related to tuning the different constants is an experimental activity that involves adjust every output response in order to maintain an specific reference spectrum with a probability of error less than 1%.

4.1.2 PID parameter calculation

The first approach of the algorithm is a PID designed for each LED channel independently. We have implemented twelve PID controllers in order to adjust every channel peak value at its certain wavelength amplitude.

A PID controller follows the equation (1):

$$u(t) = K_p e(t) + \frac{K_p}{T_i}\int_0^t e(t)\,dt + K_p T_d \frac{de(t)}{dt} \qquad (1)$$

Where *e(t)* is the current error and *u(t)* is the input signal. K_p is the controller path gain, T_i is the integrator time constant and T_d is the derivative time constant.

In frequency domain the equation (1) is rewritten as the equation (2):

$$U(S) = K_p\left(1 + \frac{1}{T_i S} + T_d S\right)E(s) \tag{2}$$

Then, the transfer function for a digital PID becomes to the equation (3). The parameter T is the sampling interval of the converter.

$$U(Z) = K_p\left[1 + \frac{T}{T_i(1 - Z^{-1})} + T_d\frac{(1 - Z^{-1})}{T}\right]E(Z) \tag{3}$$

The transfer function can be also written as in (4):

$$\frac{U(Z)}{E(Z)} = a + \frac{b}{1 - Z^{-1}} + c(1 - Z^{-1}) \tag{4}$$

Where:

$$a = K_p \quad b = \frac{K_p T}{T_i} \quad c = \frac{K_p T_d}{T} \tag{5}$$

There are some different implementations of PID controllers but the most used is the parallel one as shown in Figure 11 where the input is E(kT) and the output is M(kT) to trigger the PWM driver LED channel. In addition, a,b and c are PID constants.

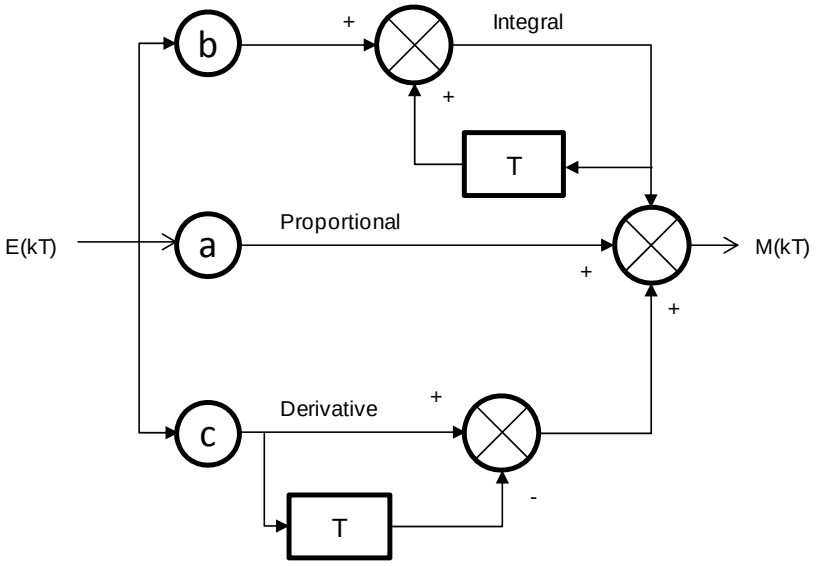

Figure 11. PID controller parallel implementation

4.1.3 Tuning a PID controller using Ziegler-Nichols rule

The Ziegler-Nichols rule [4], is a heuristic PID tuning method that attempts to produce optimized values for the three PID gain parameters:

K_P = Proportional gain

K_d = Derivative gain

K_I = integral gain

T_I = reset time

T_D = rate time or derivative time

Three steps allow us to determine PID controller parameters:

a. Reduce the integrator and derivative gains to 0.
b. Increase Kp from 0 to some critical value Kp=Kc at which sustained oscillations occur
c. Note the value Kc and the corresponding period of sustained oscillation, Tc
d. The controller gains are now specified as follows:

Ziegler-Nichols rule assumes that the system has a transfer function of the following form (6) as shown Figure 12:

$$G(S) = \frac{K_o}{1 + ST} e^{-sL} \quad (6)$$

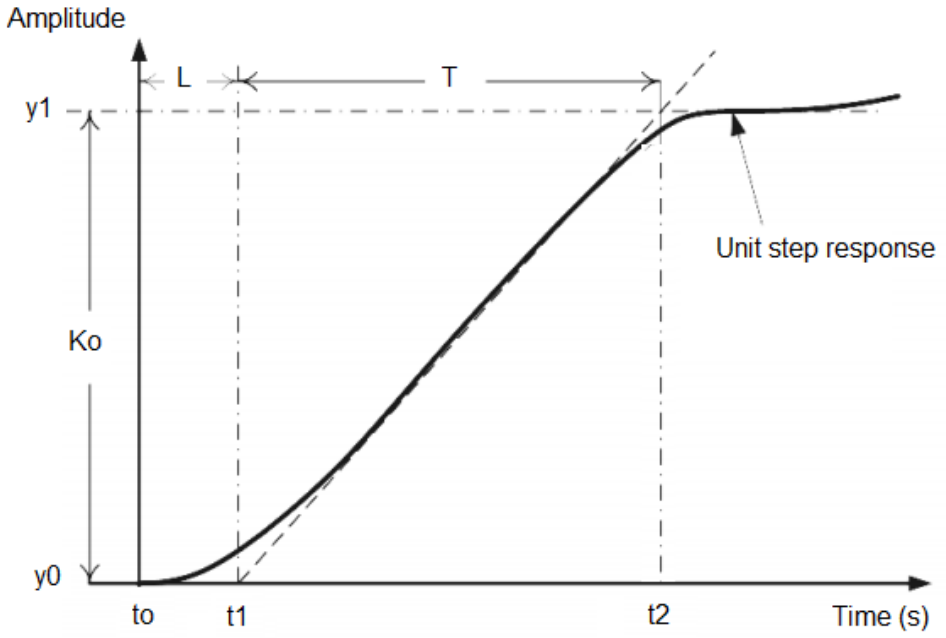

Figure 12. Parameters of interest in Ziegler Nichols open loop tuning

Where: K_0 is the static gain, T the time constant, and L the time delay. Given the magnitude and phase open-loop response curves of the system, we can obtain the coefficients (K_0), (T), (L) and (m) as:

$$L = t_1 - t_0 \quad (7)$$

$$T = t_2 - t_1 \quad (8)$$

$$m = \frac{y_1 - y_0}{t_2 - t_1} \quad (9)$$

The parameters in equations (7), (8) and (9) can be obtained from a simple step response obtained with the mini-spectrophotometer. If the actual system is linear, monotonic, and sluggish, results are good enough.

4.1.4 Firmware Implementation

The algorithm used to implement the PID controller in the microcontroller is showed in the Figure 13:

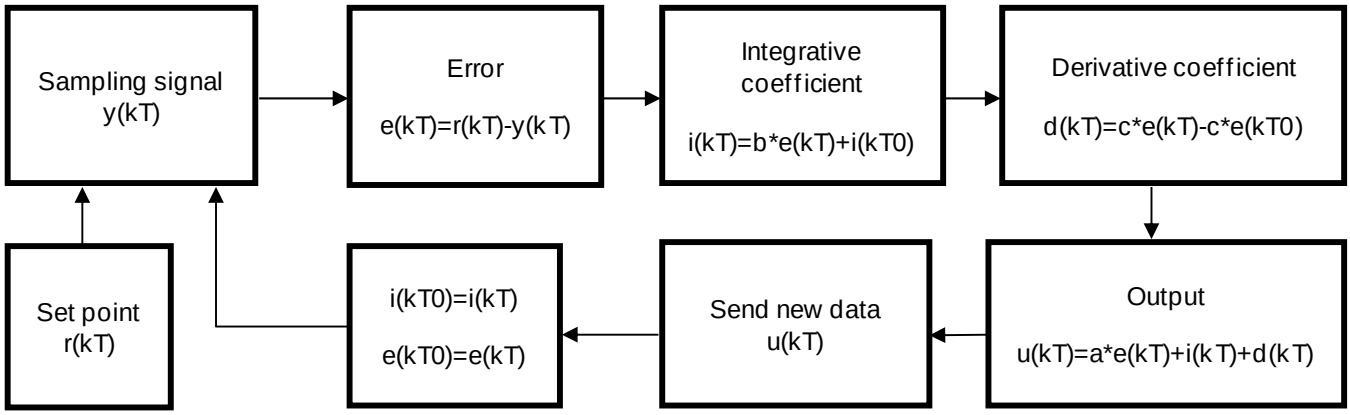

Figure 13. Algorithm used to implement the PID controller in each LED channel

The sampling time must be less than 10 times the open-loop system response that is near 1 ms to improve the accuracy of the system. In Ziegler-Nichols model $T < T_o/ 4$. Figure 13 shows the block diagram of the implemented LED driver controller. The closed-loop function can be written as H(s) = C(s)G(s). The gain in the mini-spectrophotometer is configured between 1 ms to 10 ms. The closed-loop function can be written as H(s)=C(s)G(s).

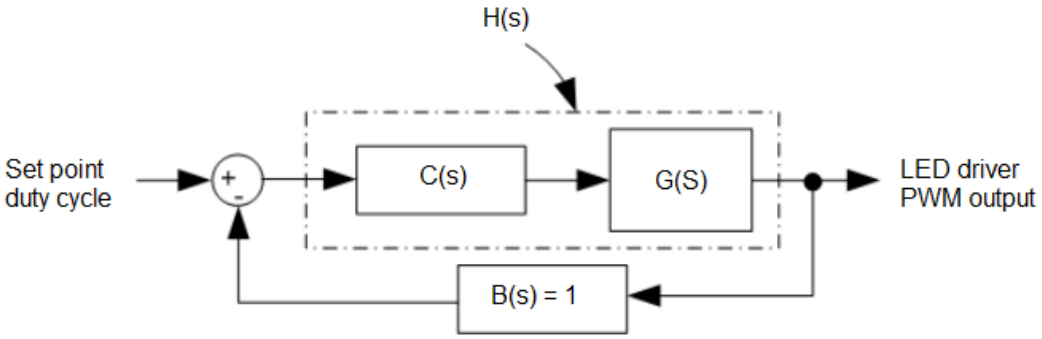

Figure 13. Block diagram of the implemented LED driver controller.

Figure 14 (left) shown spectral reproduction according the Ziegler-Nichols rule algoritm (red line) and the theoretical syntetic spectra (black line) with peak amplitudes [500, 0, 0, 230, 0, 0, 800, 0, 0, 100, 0, 300]. While Figure 14 (right) shown peak amplitudes [500, 100, 650, 230, 600, 700, 800, 600, 500, 100, 200, 300]

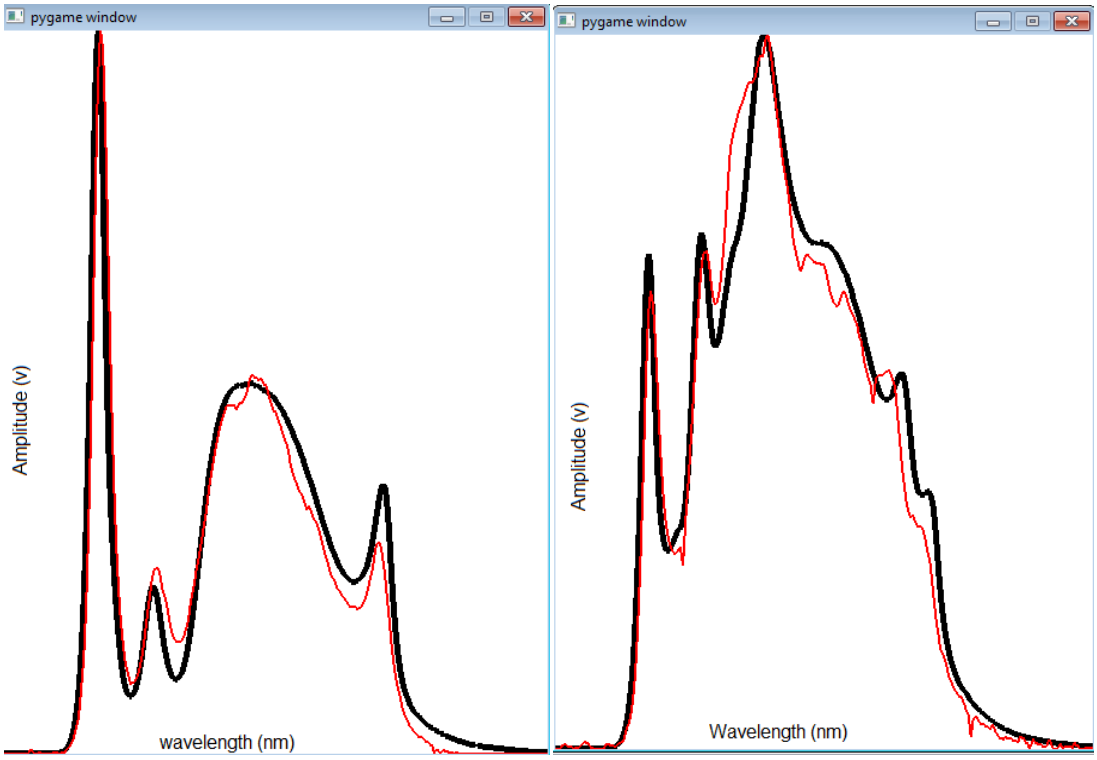

Figure 14. Experimental spectral reproduction according Ziegler-Nichols rule algoritm

Experimental results with a python application have shown close performances for the different spectral reproduction in terms of color, peak wavelength and distribution obtained with the embedded mini-spectrometer (red line).

4.2 Transducer electronic datasheet (TEDS)

Each LED engine for HI-LED includes a look-up table to store calibration values for each LED channel as a transducer electronic datasheet (TEDS) [5]. Overall TEDS metadata structure is defined in a non volatile Flash memory containing a set of three different sections: global configuration, sensor calibration section (temperature and mini-spectrometer) and a spectra database section. Table2 shown TEDS global section includes default values of peak wavelengths, voltage, current, output power and efficacy in different operational points.

Table 2. Transducer Electronic DataSheet (TEDS). Global section

TEDS Global section	
Vendor	HI-LED
Model	HI-LED100
Serial	HI-LED100-00000000001
No Channels	12
No LEDs	[2, 2, 2, 2, 4, 4, 10, 10, 10, 10, 2, 2]
Peak wavelengths (nm)	[428, 450, 471, 473, 497, 523, 547, 547, 600, 600, 641, 665]
LED voltages (v)	[3.03, 3.11, 3.13, 3.23, 2.99, 2.98, 2.94, 3.13, 3.30, 2.19, 2.29, 2.30],
LED current (mA)	[750, 750, 750, 750, 750e-3, 750, 350, 350, 750, 750, 750, 750]
Output power (W)	[0.4822, 0.4743, 0.3737, 0.3791, 0.4895, 0.2036, 0.3466, 1.2888, 0.5427, 0.6403, 0.3122, 0.3521, 0.3761]
Maximum power (W)	0.8
Efficacy	Matrix [10 x 12] values

Table 3 shown sensor configuration and wireless interface default values. The more important values for any LED engine with an onboard spectrometer configuration are: model, number of pixels, wavelength coefficients and the matrix of values containing each spectral reproduction.

Table 3. TEDS Sensing and communication interface

Sensing and communication interface section	
Spectrometer on board	True
Spectrometer model	001
Spectrometer nPixels	256
Spectrometer ADC resolution	12
Spectrometer wavelenght Coefficients	[1e-4, 1e-4, 1e-3, 1e-2, 0.5e-2, 1e-1],
Spectrometer Out_function	[Matrix 1x 256]
Wireless physical interface	WiFi, Bluetooth, 6loWPAN
Network discover method	Auto-discover
Temperature sensor on board	True
Temperature sensor wavelenght shifts deviation per Kelvin (K)	[-0.12,0.22,-0.33, 0.10,0.21,0.33,0.43,0.03,-0.009,0.21,-0.01,0.054,-0.28]
Temperature sensor power shifts deviation	[-0.12,0.22,-0.33, 0.10,0.21,0.33,0.43,0.03,-0.009,0.21,-0.01,0.054,-0.28]
Energy measurement in power supply	False
Energy measurement in driver electronics	False
Energy measurements in LED channels	False

Each LED engine defines a WiFi wireless interface with auto-discover capabilities by default. Also, the temperature sensor configuration in each module is defined as a matrix containing wavelength shifts deviation and the power shift deviation associated with temperature fluctuations.

Compensation values according temperature fluctuations involve a matrix containing values for each 5 degrees in a dynamic range from 5 C to 80 C degrees. This table is composed by 12 rows per 10 columns. Each row defines the channels and each column defines an operational point at a certain temperature. Once we know the performance of each LED channel, the fitting algorithm is applied again in order to compensate the output bearing in mind this new scenario, it means to apply the new operational configuration.

Table 4 stores a database containing a set of registers to store different natural and artificial lighting spectra. Each stored spectra is organized as a matrix of 81 values in the range of 310 to 800 nm.

Table 4. TEDS Spectra database section

Spectra database	
Spectra1	[Matrix 1x81]
Spectra2	[Matrix 1x81]
Spectra3	[Matrix 1x81]
Spectra4	[Matrix 1x81]
Spectra5	[Matrix 1x81]
Spectra6	[Matrix 1x81]
Spectra (n)	[Matrix 1x81]

4.3 Wavelength displacement

In order to detect wavelength displacements, a peak detection algorithm [10], [11] in series has been implemented in order to helps us to find the LED channel peaks related to the captured spectrum. Once we know the position of every LED channel we can compute the wavelength displacement of each channel and compensate it.

The entropy of any sequence of M values A = <a_1, a_2,…,a_M> is defined as follows in equation (10):

$$H_w(A) = \sum_{i=1}^{M} \left(-P_w(a_i) log\left(P_w(a_i)\right)\right) \quad (10)$$

Where $P_w(a_i)$ is an estimate of the probability density at (a_i). The kernel density technique also called Parzen window [6], [7] that can be used to estimate the probability density $p(a_i)$ at I^{th} value (a_i) in the given sequence as shown in equation (11):

$$P_w(a_i) = \frac{1}{M|a_i - a_{i+w}|} \sum_{j=1}^{M} K\left[\frac{a_i - a_j}{a_i - a_{i+w}}\right] \quad (11)$$

Where ***K*** is a suitable kernel function and ***w*** > 0 is a given integer. The subscript ***w*** in equation (10) for H_w and equation (11) for P_w indicates the width parameter used in kernel density estimation. Epanechnikov [8] and Gaussian [9] are two well-known kernel functions defined in equation (12) and (13):

$$K(x) = \frac{3}{4}(1 - x^2) \qquad \text{If } |x| < 1 \qquad (12)$$

$$K(x) = 0 \qquad \text{otherwise}$$

$$K(x) = \frac{1}{\sqrt{2\pi}} e^{\frac{-1}{2}x^2} \qquad (13)$$

The function S(z) in equation (14) computes the difference in the entropy of the two sequences ***N(k,i,T)*** and ***N'(k,i,T)***, which gives an idea of how significant x_i is in this window. The Gaussian kernel according equation (12) or (13) are used to compute the density estimate as shown equation (14).

$$S(k, w, i, x_i, T) = H_w\big(N(k, i, T)\big) - H_w\big(N'(k, i, T)\big) \qquad (14)$$

The pseudo code shows the entropy based in a peak function that has been used in order to detect all peaks of interest in a captured spectra:

```
algorithm peak1 // one peak detection algorithms that uses peak function S1
        input T = x1, x2, …, xN, N // input time-series of N points
        input k // window size around the peak
        input h // typically 1 <= h <= 3
        output O // set of peaks detected in T
begin
        O = 0 // initially empty
        for (i = 1; i < n; i++) do
                a[i] = S1(k,i,xi,T); // compute peak function value for each of the N
        points in T
        end for
        Compute the mean m' and standard deviation s' of all positive values in array a;
        for (i = 1; i < n; i++) do // remove local peaks which are "small" in global
context
                if (a[i] > 0 && (a[i] - m') >( h * s')) then O = O U {xi}; end if
        end for
        Order peaks in O in terms of increasing index in T
        // retain only one peak out of any set of peaks within distance k of each other
        for every adjacent pair of peaks xi and xj in O do
                if |j - i| <= k then remove the smaller value of {xi, xj} from O end if
        end for
end
```

In order to compensate the displacement we have to re-calculate each LED channel coefficient bearing in mind the new LED channel characterization.

This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement **no 619912**

SECTION 5. Conclusions

This deliverable provides a description of three different methods for thermal management and aging LED compensation based on temperature sensors and the spectral response according an embedded mini-spectrometer.

The first approach uses a temperature sensor to control the temperature at PCB level. The temperature sensor may be configured as an on/off detector which activates an alarm when temperature exceeds a specific threshold or execute temperature measurements to adjust a PID algorithm to update LED intensities and minimize peak deviations.

The second approach is based on a look-up table is embedded in a TEDS metadata structure. We have defined the syntactic structure for temperature matrices with different set-points. The different values were settled in order to compensate peak deviations errors according an initial calibration.

The third method uses an embedded mini-spectrometer in order to compare the present spectrum with a previous synthetic spectrum reproduced in the luminaire. In this case a fitting algorithm in the microcontroller finds the different LED coefficients in order to reduce the multidimensional distance between colour coordinates x, y and wavelength amplitudes with the expected one.

In all cases, we have developed these algorithms for real-time operating, despite that in fact we can improve the response by adding more inputs in the process but this task involve more processing load to process the information. In the future, we will replace the Flash memory by another model with more capacity to store more information.

References

[1] Antila, J., Tuohiniemi, M., Rissanen, A., Kantojärvi, U., Lahti, M., Viherkanto, K., Kaarre, M. and Malinen, J. 2014. MEMS- and MOEMS-Based Near-Infrared Spectrometers. Encyclopedia of Analytical Chemistry. 1–36.

[2] Jouko Malinen ; Anna Rissanen ; Heikki Saari ; Pentti Karioja ; Mikko Karppinen, et al.Advances in miniature spectrometer and sensor development, Proc. SPIE 9101, Next-Generation Spectroscopic Technologies VII, 91010C (May 21, 2014)

[3] Garcia, D.; Karimi, A.; Longchamp, R., "Robust proportional integral derivative controller tuning with specifications on the infinity-norm of sensitivity functions," Control Theory & Applications, IET , vol.1, no.1, pp.263,272, January 2007.

[4] Tomislav B. Šekara, Miroslav R. Mataušek, Revisiting the Ziegler–Nichols process dynamics characterization, Journal of Process Control, Volume 20, Issue 3, March 2010, Pages 360-363

[5] K. Lee. Transducer Electronic Data Sheet (TEDS). NIST. Available online (2014): http://www.nist.gov/el/isd/ieee/teds.cfm

[6] Babich, G.A.; Camps, O.I., "Weighted Parzen windows for pattern classification," Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.18, no.5, pp.567,570, May 1996.

[7] Archambeau, C.; Valle, M.; Assenza, A.; Verleysen, M., "Assessment of probability density estimation methods: Parzen window and finite Gaussian mixtures," Circuits and Systems, 2006. ISCAS 2006. Proceedings. 2006 IEEE International Symposium on , vol., no., pp.4 pp.,, 21-24 May 2006

[8] Qingchang Guo; Xiaojuan Chang; Hongxia Chu. Mean-Shift of Variable Window Based on the Epanechnikov Kernel. International Conference on Mechatronics and Automation, 2007. pp.2314,2319, 5-8 Aug. 2007

[9] Varewyck, M.; Martens, J.-P. A Practical Approach to Model Selection for Support Vector Machines With a Gaussian Kernel. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, vol.41, no.2, pp.330-340, April 2011.

[10] Azzini I., Dell'Anna R., Ciocchetta F., Demichelis F., Sboner A., Blanzieri E., Malossini A. Simple Methods for Peak Detection in Time Series Microarray Data- Proceedings of the CAMDA'04. Critical Assessment of Microarray Data conference 2004.

[11] Jordanov V.T., Hall D.L., Kastner M. Digital Peak Detector with Noise Threshold", Proc. IEEE Nuclear Science Symposium Conference, vol. 1, pp. 140 – 142. 2002.