/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Ejemplo de Souliss con 3 leds, un led RGB, un pulsador,

entrada analógica y sensores de temperatura y humedad NODO MAESTRO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include "SoulissFramework.h"

#include <SimpleDHT.h>

// Configure the framework

#include "bconf/StandardArduino.h" // Use a standard Arduino

#include "conf/ethENC28J60.h" // Ethernet through Wiznet W5100

#include "conf/Gateway.h" // The main node is the Gateway, we have just one node

// Include framework code and libraries

#include <SPI.h>

// para usar sensor temperatura y humedad DHT11 en pin 18

#include <SimpleDHT.h>

int pinDHT11 = 18;

SimpleDHT11 dht11;

/\*\*\* All configuration includes should be above this line \*\*\*/

#include "Souliss.h"

// This identify the number of the LED logic

#define LED1 0

#define LED2 1

#define LED3 2

#define LEDCONTROL 3 // This is the memory slot for the logic that handle the light

#define LEDRED 4 // This is the memory slot for the logic that handle the light

#define LEDGREEN 5 // This is the memory slot for the logic that handle the light

#define LEDBLUE 6 // This is the memory slot for the logic that handle the light

#define ANALOGDAQ 7 // This is the memory slot used for the execution of the logic

#define TEMPERATURA 9 // memory slot para temperatura 9 y 10

#define HUMEDAD 11 // memory slot para humedad 11 y 12

// Define the network configuration according to your router settings

uint8\_t ip\_address[4] = {192, 168, 1, 44};

uint8\_t subnet\_mask[4] = {255, 255, 255, 0};

uint8\_t ip\_gateway[4] = {192, 168, 1, 1};

#define myvNet\_address ip\_address[3] // The last byte of the IP address (77) is also the vNet address

#define myvNet\_subnet 0xFF00

#define DEADBAND 0.05 // Deadband value 5%

void setup()

{

Initialize();

// Set network parameters

Souliss\_SetIPAddress(ip\_address, subnet\_mask, ip\_gateway);

SetAsGateway(myvNet\_address);

// Set this node as gateway for SoulissApp

SetAsPeerNode(83.44,1);

SetAsPeerNode(23.44,2);

SetAsPeerNode(28.44,3);

SetAsPeerNode(79.44,4);

SetAsPeerNode(24.44,5);

Set\_SimpleLight(LED1); // Define a simple LED light logic

Set\_SimpleLight(LED2); // Define a simple LED light logic

Set\_SimpleLight(LED3); // Define a simple LED light logic

Set\_LED\_Strip(LEDCONTROL); // Set a logic to control a LED strip

// Define inputs, outputs pins

pinMode(3, OUTPUT); // Power the LED

pinMode(5, OUTPUT); // Power the LED

pinMode(6, OUTPUT); // Power the LED

// We connect a pushbutton between 5V and pin4 with a pulldown resistor

// between pin4 and GND, the LED is connected to pin9 with a resistor to

// limit the current amount

pinMode(4, INPUT); // Hardware pulldown required

pinMode(9, OUTPUT); // Power the LED

pinMode(8, OUTPUT); // Power the LED

pinMode(7, OUTPUT); // Power the LED

Set\_AnalogIn(ANALOGDAQ); // Set an analog input value

Set\_T52(TEMPERATURA);

Set\_T53(HUMEDAD);

}

void loop()

{

// Here we start to play

EXECUTEFAST() {

UPDATEFAST();

FAST\_10ms() {

// Use Pin4 as command

DigIn(4, Souliss\_T1n\_ToggleCmd, LEDCONTROL);

// Execute the logic that handle the LED

Logic\_LED\_Strip(LEDCONTROL);

// Use the output values to control the PWM

analogWrite(5, 255 - mOutput(LEDRED)); // ex 5 es verde

analogWrite(3, 255 - mOutput(LEDGREEN)); // ex 6 es azul

analogWrite(6, 255 - mOutput(LEDBLUE)); // ex3 es rojo

// Just process communication as fast as the logics

ProcessCommunication();

}

FAST\_50ms() { // We process the logic and relevant input and output every 50 milliseconds

//DigIn(4, Souliss\_T1n\_ToggleCmd, LED1); // Use the pin4 as ON/OFF toggle command

Logic\_SimpleLight(LED1); // Drive the LED as per command

Logic\_SimpleLight(LED2); // Drive the LED as per command

Logic\_SimpleLight(LED3); // Drive the LED as per command

DigOut(9, Souliss\_T1n\_Coil, LED1); // Use the pin9 to give power to the LED according to the logic

DigOut(8, Souliss\_T1n\_Coil, LED2); // Use the pin8 to give power to the LED according to the logic

DigOut(7, Souliss\_T1n\_Coil, LED3); // Use the pin7 to give power to the LED according to the logic

}

FAST\_110ms()

{

// Compare the acquired input with the stored one, send the new value to the

// user interface if the difference is greater than the dead-band

Read\_AnalogIn(ANALOGDAQ);

}

FAST\_910ms()

{

// Acquire temperature from the microcontroller ADC

AnalogIn(A0, ANALOGDAQ, 0.097, 0); // The raw data is 0-1024, scaled as 0-100% without bias (100 / 1024 = 0.09)

byte temp = 0;

byte hum = 0;

float temperature;

float humidity;

dht11.read(pinDHT11, &temp, &hum, NULL);

temperature = temp;

humidity = hum;

Souliss\_ImportAnalog(memory\_map, TEMPERATURA, &temperature);

Souliss\_ImportAnalog(memory\_map, HUMEDAD, &humidity);

Logic\_T52(TEMPERATURA);

Logic\_T53(HUMEDAD);

}

// Process data communication

FAST\_GatewayComms();

}

}