



**Escuela Universitaria
Politécnica** - La Almunia
Centro adscrito
Universidad Zaragoza

**ESCUELA UNIVERSITARIA POLITÉCNICA
DE LA ALMUNIA DE DOÑA GODINA (ZARAGOZA)**

ANEXOS

Automatización de una vivienda
unifamiliar

Automatization of a single-family house

424.19.51

Autor: Alba Continente Sisamón

Director: Dr. Diego Antolín Cañada

Fecha: 5 de Febrero de 2020

INDICE DE CONTENIDO

ANEXO 1. PLANIFICACIÓN	1
1.1. ESQUEMAS ELÉCTRICOS	1
1.1.1. Esquema completo (Node)	1
1.1.2. Node1	2
1.1.3. Node2	3
1.2. VISUALIZACIÓN DE LA VIVIENDA	4
ANEXO 2. PROGRAMACIÓN	5
2.1. ESQUEMA DE FUNCIONAMIENTO	5
2.2. CÓDIGO ARDUINO	6
2.3. PROGRAMA LABVIEW	8
ANEXO 3. (DATASHEETS)	9
3.1. NODEMCU V3	9
3.2. MF52	27
3.3. BH1750FVI	31
3.4. KS0052	35

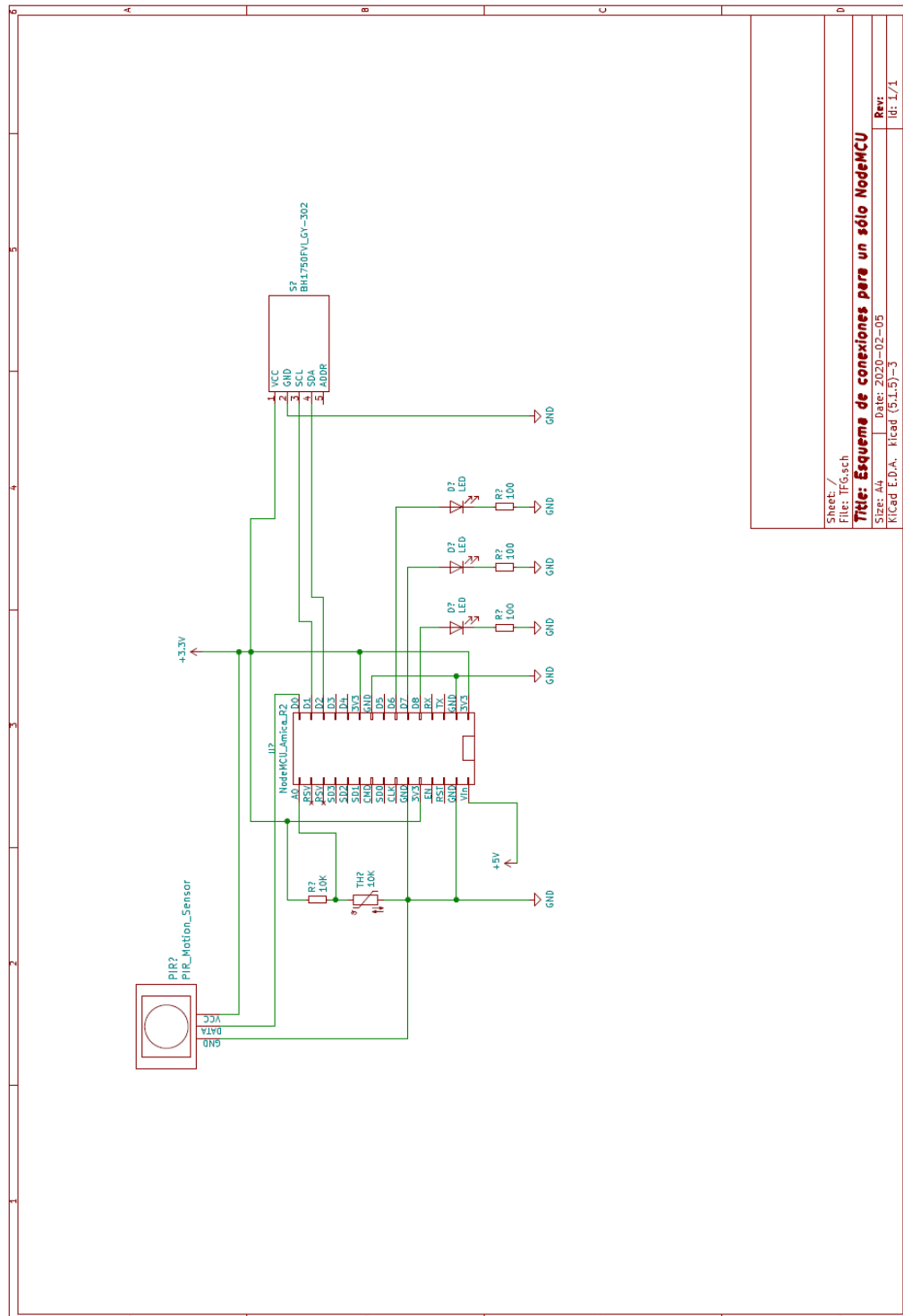
INDICE DE ILUSTRACIONES

Ilustración 1 Visualización de planta baja	4
Ilustración 2 Visualización de 1ª planta	4
Ilustración 3 Diagrama de funcionamiento global	5
Ilustración 4 Código para el Node1 (Persianas)	6
Ilustración 5 Código para el Node2 (Temperatura y Presencia)	7
Ilustración 6 Vista de bloques LabVIEW	8

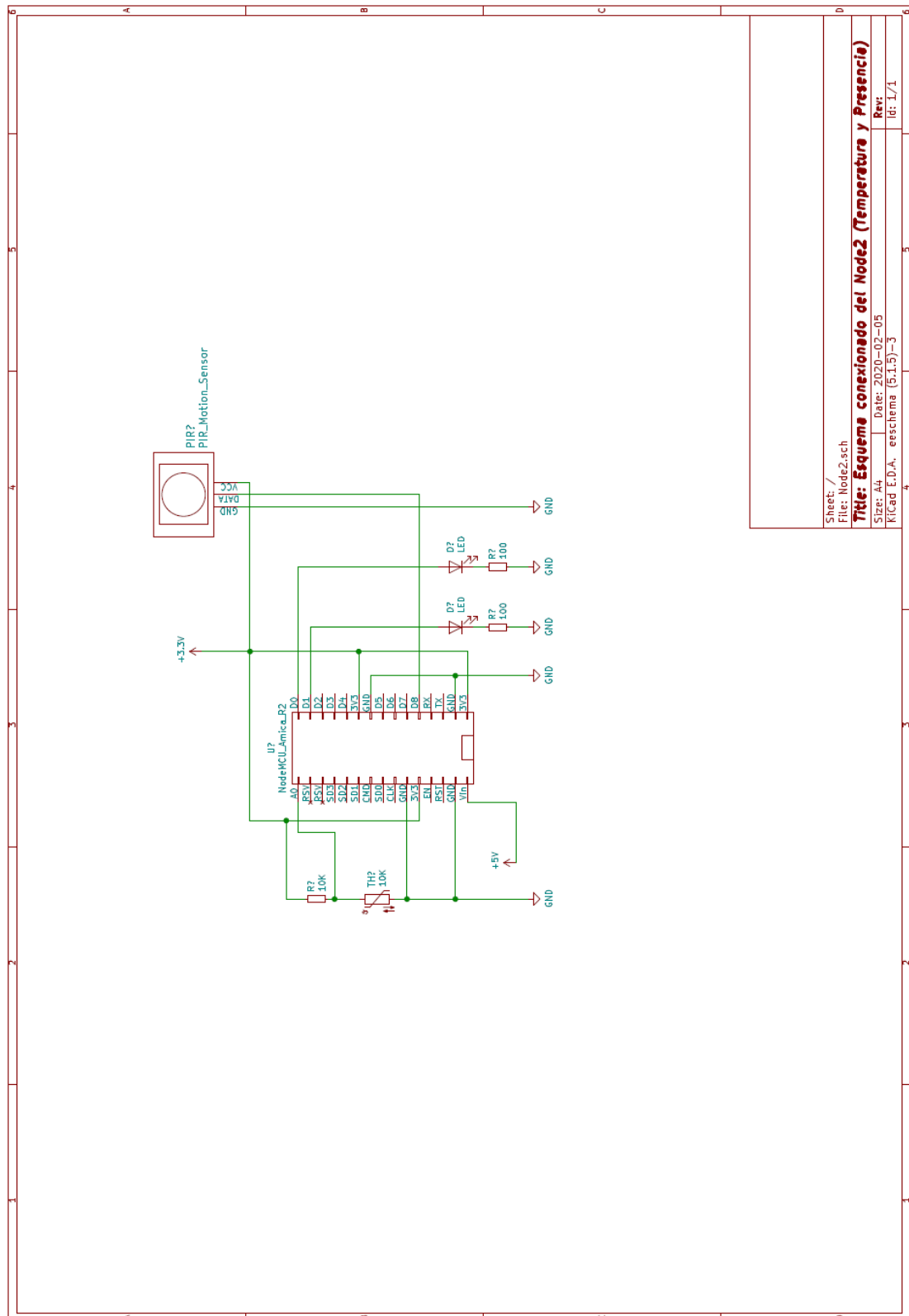
ANEXO 1. PLANIFICACIÓN

1.1. ESQUEMAS ELÉCTRICOS

1.1.1. Esquema completo (Node)



1.1.3. Node2



1.2. VISUALIZACIÓN DE LA VIVIENDA



Ilustración 1 Visualización de planta baja.



Ilustración 2 Visualización de 1ª planta.

ANEXO 2. PROGRAMACIÓN

2.1. ESQUEMA DE FUNCIONAMIENTO

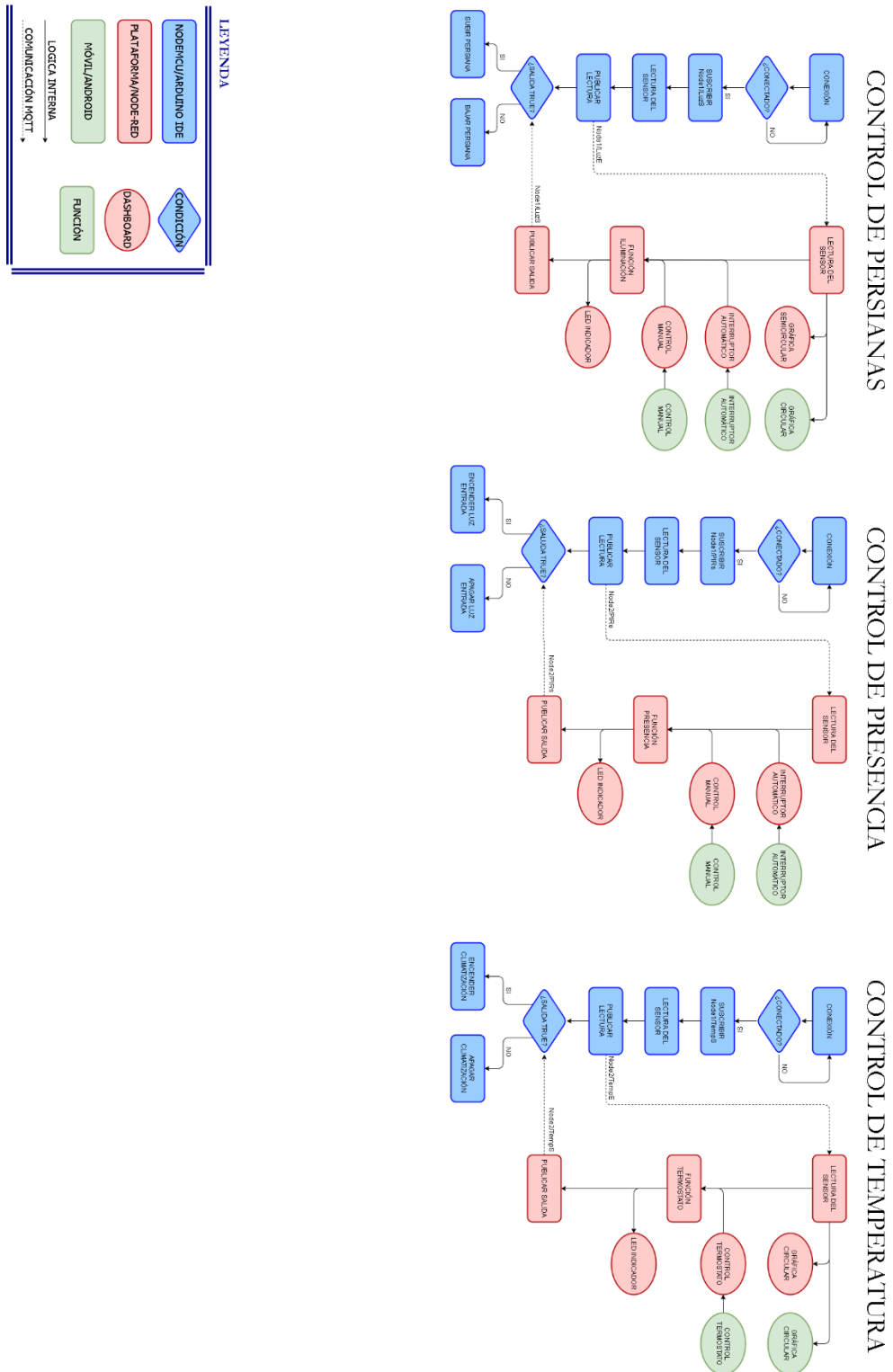


Ilustración 3 Diagrama de funcionamiento global

2.2. CÓDIGO ARDUINO

```

#include <BH1750FVI.h>

String LuzS;
float sensorValue = 0;
float outputValue = 0;
const int analogInPin = A0;
const int LED = D0;
long lastMsg;
int LuzE;
BH1750FVI SensorLuz(BH1750FVI::k_DevModeCo
ntLowRes);
String lux;

#include "EspMQTTClient.h"
#include "stdlib.h"

/*EspMQTTClient client(
  "Lowi06E2",
  "EWAKLZ7CHLYNPP",
  "m24.cloudmqtt.com",
  "vlavydww",
  "bqXdJpymzFhJ",
  "Node1",
  12869
);*/

EspMQTTClient client(
  "Aquaris X5 Plus_albacontinentesi",
  "976178820",
  "test.mosquitto.org",
  "Node1",
  1883
);
void setup()
{
  Serial.begin(115200);
  SensorLuz.begin();
  pinMode(LED,OUTPUT);
  client.enableDebuggingMessages();
  client.enableHTTPWebUpdater();
  client.enableLastWillMessage("Node1/lastwill",
  "I am going offline");
}

void Luz(const String &LuzS)
{
  if (LuzS == "true")
  {
    Serial.println(LuzS);
    digitalWrite(LED, HIGH);
  }
  if (LuzS == "false")
  {
    Serial.println(LuzS);
    digitalWrite(LED, LOW);
  }
}

void onConnectionEstablished()
{
  client.subscribe("Node1/LuzS", Luz);
}

void loop()
{
  client.loop();
  if(client.isConnected() == true)
  {
    lux = String(SensorLuz.GetLightIntensity());

    if (LuzS == "true")
    {
      digitalWrite(LED, HIGH);
    }
    long now = millis();
    if (now - lastMsg > 1000)
    {
      client.publish("Node1/LuzE", lux);
      lastMsg = now;
    }
  }
}

```

Ilustración 4 Código para el Node1 (Persianas)

```

#include "EspMQTTClient.h"
#include "stdlib.h"
#include "math.h"

String TempS;
String PIRs;
float sensorValue = 0;
float outputValue = 0;
const int analogInPin = A0;
const int PIR = D7;
const int LEDPIR = D0;
const int LED = D1;
long lastMsg;

const int Rc = 10000; //valor de la resistencia
const int Vcc = 5;
float A = 1.105430345e-3;
float B = 2.386828139e-4;
float C = 6.429677074e-8;
float K = 2.5; //factor de disipacion en mW/C

/*EspMQTTClient client(
  "Lowi06E2", // WiFi SSID
  "EWAKLZ7CHLYNPP",
  "m24.cloudmqtt.com",
  "vlavydww",
  "bqXdJpymzFhJ",
  "Node2",
  12869
);*/

EspMQTTClient client(
  "Aquaris X5 Plus_albacontinentesi",
  "976178820",
  "test.mosquitto.org",
  "Node2",
  1883
);

void setup()
{
  Serial.begin(115200);
  pinMode(LED,OUTPUT);
  pinMode(LEDPIR,OUTPUT);
  pinMode(PIR,INPUT);
  client.enableDebuggingMessages();
  client.enableLastWillMessage("Node2/lastwill", "I am
going offline");
}
void Temperatura(const String &TempS)
{
  if (TempS == "true")
  {
    Serial.println(TempS);
    digitalWrite(LED, HIGH);
  }
  if (TempS == "false")
  {
    Serial.println(TempS);
    digitalWrite(LED, LOW);
  }
}

void fPIR(const String &PIRs)
{
  if (PIRs == "true")
  {
    Serial.println(PIRs);
    digitalWrite(LEDPIR, HIGH);
  }
  if (PIRs == "false")
  {
    Serial.println(PIRs);
    digitalWrite(LEDPIR, LOW);
  }
}

void onConnectionEstablished()
{
  client.subscribe("Node2/TempS", Temperatura);
  client.subscribe("Node2/PIRs", fPIR);
}

void loop()
{
  client.loop();
  if(client.isConnected() == true)
  {
    sensorValue = analogRead(analogInPin);
    float V = sensorValue / 1024 * Vcc;
    float R = (Rc * V) / (Vcc - V);
    float logR = log(R);
    float R_th = 1.0 / (A + B * logR + C * logR * logR *
logR );
    float kelvin = R_th - V*V/(K * R)*1000;
    float outputValue = kelvin - 273.15;
    int PIRe = digitalRead(PIR);
    long now = millis();
    if (now - lastMsg > 1000)
    {
      client.publish("Node2/TempE",
String(outputValue,2));
      client.publish("Node2/PIRe", String(PIRe,2));
      lastMsg = now;
    }
  }
}

```

Ilustración 5 Código para el Node2 (Temperatura y Presencia)

2.3. PROGRAMA LABVIEW

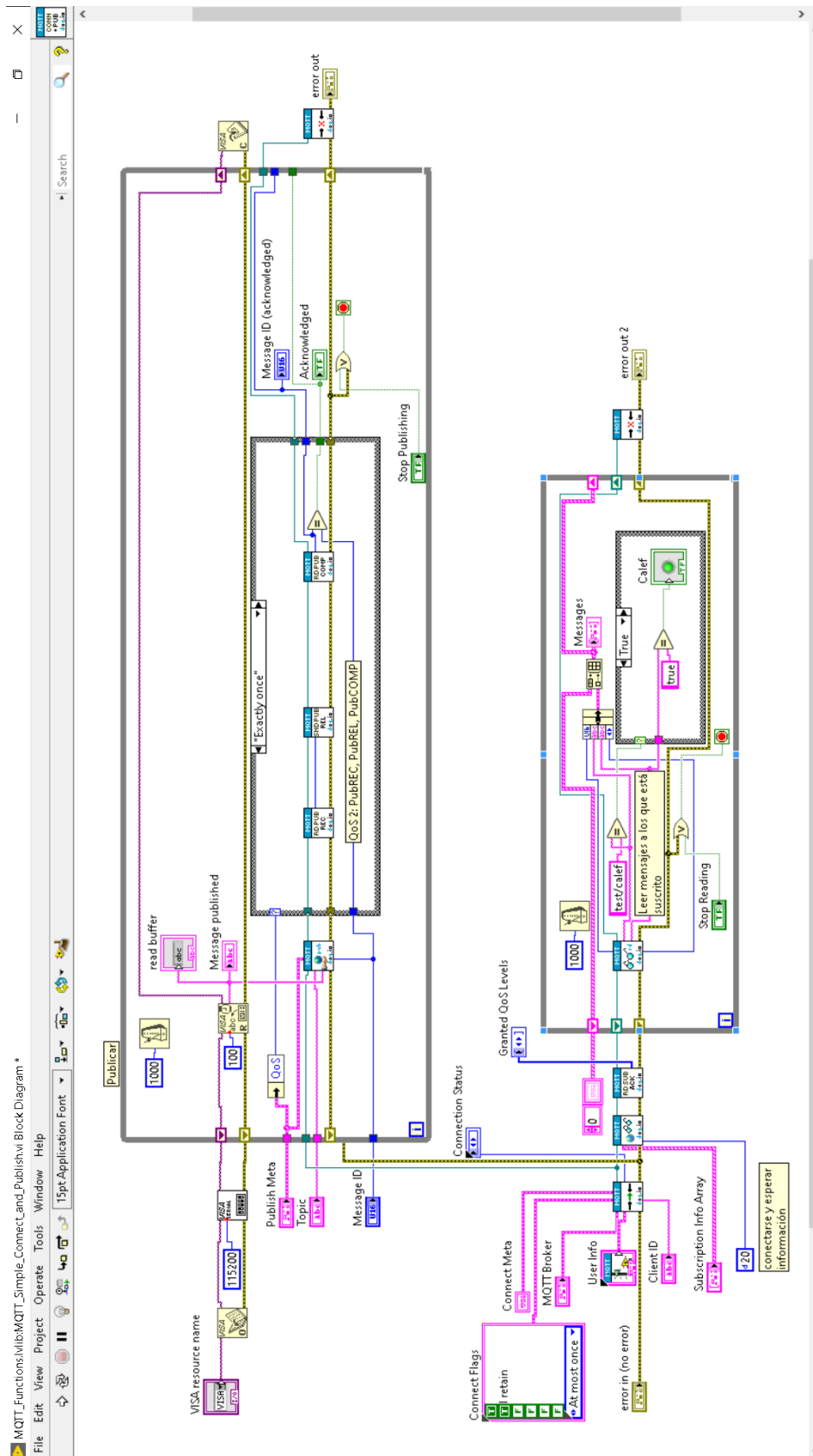


Ilustración 6 Vista de bloques LabVIEW



ANEXO 3. (DATASHEETS)

3.1. NODEMCU V3





3.2. MF52



3.3. BH1750FVI



3.4. KS0052

Introduction:

Pyroelectric infrared motion sensor can detect infrared signals from a moving person or moving animal, and output switching signals. It can be applied to a variety of occasions to detect the movement of human body. Conventional pyroelectric infrared sensors require body pyroelectric infrared detector, professional chip, complex peripheral circuit, so the size is bigger, with complex circuit, and lower reliability. Now we launch this new pyroelectric infrared motion sensor, specially designed for Arduino. It uses an integrated digital body pyroelectric infrared sensor, has smaller size, higher reliability, lower power consumption and simpler peripheral circuit.

Specification:

Input Voltage: 3.3 ~ 5V, 6V Maximum

Working Current: 15uA

Working Temperature: -20 ~ 85

Output Voltage: High 3V, low 0V

Output Delay Time (High Level): About 2.3 to 3 Seconds

Detection angle: 100 °

Detection distance: 7 meters

Output Indicator LED (When output HIGH, it will be ON)

Pin limit current: 100mA

Size: 30*20mm

Weight: 4g



Relación de documentos

<input type="checkbox"/> Memoria	80	páginas
<input checked="" type="checkbox"/> Anexos	35	páginas

La Almunia, a 05 de 02 de 2020

Firmado: Alba Continente Sisamón