



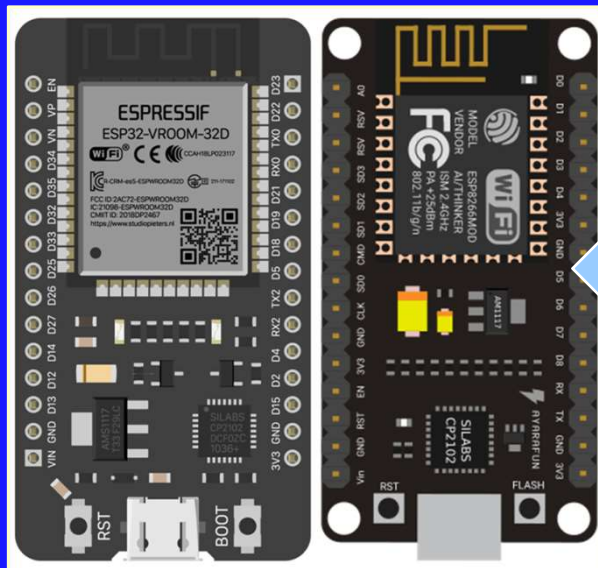
SISTEMAS
EMBEBIDOS
2021



Computadora Industrial
Abierta Argentina
Desarrollo colectivo



Interfaz Web para sistemas embebidos CL2, EDU-CIAA y PIC utilizando tecnologías ESP



Parte II – Aplicaciones - Ing. Esp. Néstor Cortez
Instituto de Tecnología Aplicada (ITA-UARG)

JORNADAS DE INFORMÁTICA 2021 – UNPA /UARG

29 de octubre 2021



SISTEMAS EMBEBIDOS 2021

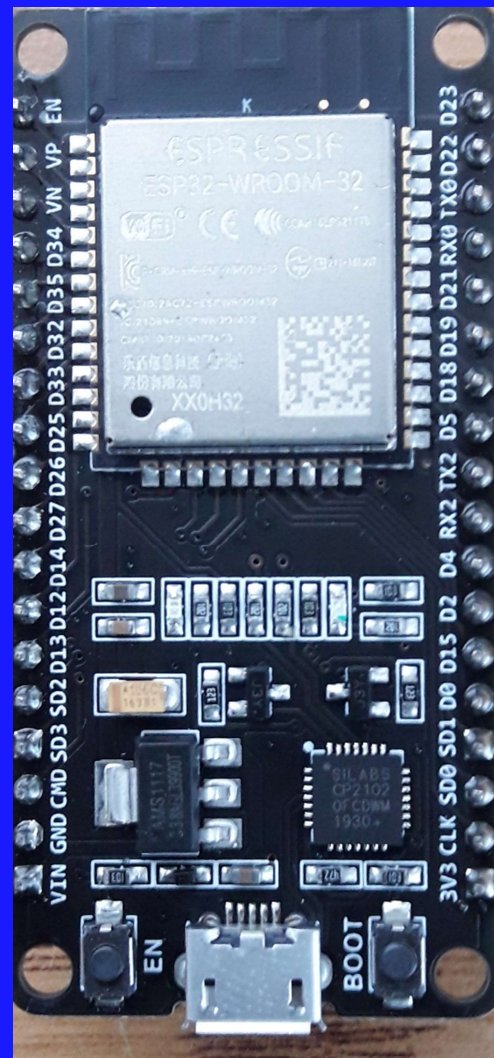
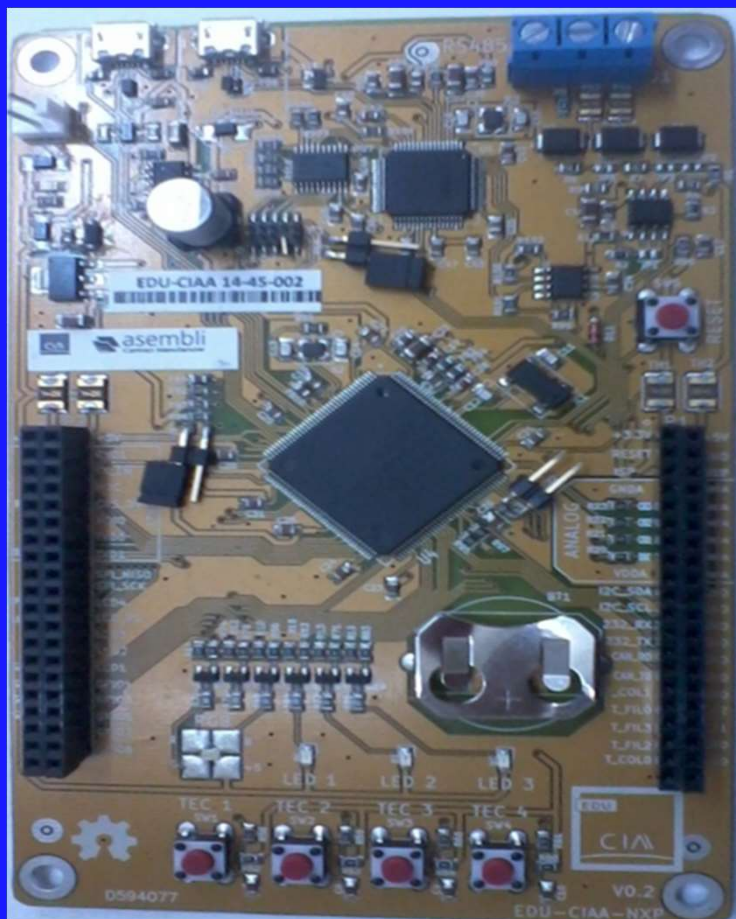


Computadora Industrial Abierta Argentina
Desarrollo colectivo



Nodemcu ESP32

EDUCIAA





SISTEMAS
EMBEBIDOS
2021



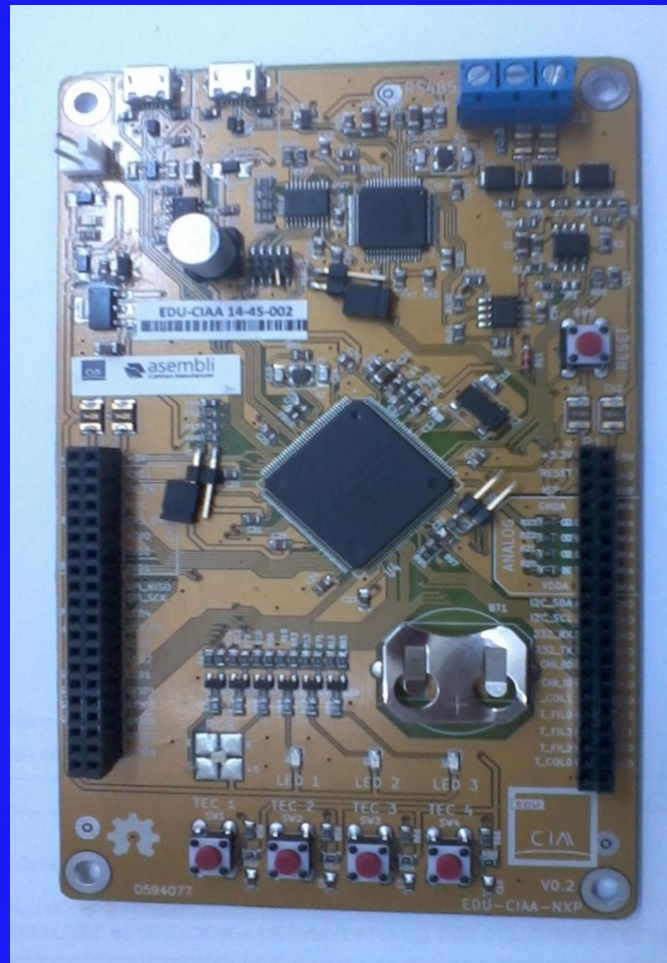
Computadora Industrial
Abierta Argentina
Desarrollo colectivo



CONTENIDOS :

- Comparación con PLC / Ladder, accionamientos comunes
- La conexión de CIAA y EDU.CIAA con el mundo exterior
- Adaptación de Drivers en C para ejemplos concretos
- Accionamiento de motor eléctrico EDU – CIAA + Display alfanumérico
- Prototipo de uso de poncho para accionamiento de motor eléctrico en inversión de giro.
- Demostración prototipo módulo de posicionamiento X-Y
- Aplicaciones con microcontroladores Microchip
- Interfaz Web - módulos ESP
- Conexión ESP - Sistema Embebido

EDU-CIAA-NXP



En la placa

- 2 puertos micro-USB (uno para aplicaciones y debugging, otro para alimentación),
- 6 salidas digitales, 3 mediante un led RGB (LED_R, LED_G y LED_B) y 3 leds con leds normales (LED_1 a LED_3),
- 4 entradas digitales con pulsadores (TEC_1 a TEC_4),
- 1 puerto de comunicaciones RS 485 con bornera.

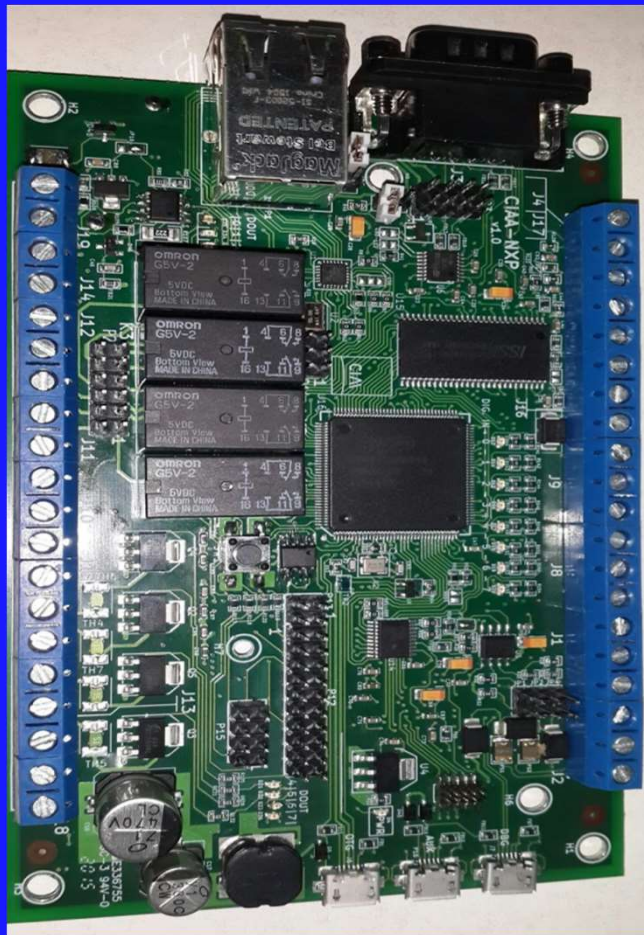
Conector de expansión CON0

- 3 entradas analógicas (ADC0 a ADC2),
- 1 salida analógica (DAC0),
- 1 conexión para un teclado de 3x4,
- 12 pines genéricos de I/O.

Conector de expansión CON1

- 1 puerto Ethernet,
- 1 puerto CAN,
- 1 puerto SPI,
- 1 puerto I2C,
- 12 pines genéricos de I/O.

CIAA-NXP



Fuente de alimentación

Fuente de 12/24VDC a 5VDC y 3.3VDC

Interfaces de comunicación

- Ethernet con soporte PoE (requiere módulo de alimentación PoE) ([detalles técnicos](#))
- USB On-The-Go
- USB Device Auxiliar
- RS232 ([detalles técnicos](#))
- RS485

Entradas/Salidas

- 8 entradas digitales optoacopladas
- 4 entradas analógicas configurables por jumper 0-10V o 0-20mA ([detalles técnicos](#))
- 4 salidas open-drain de 24V, 1A
- 4 salidas a relé 24V, 2A ([detalles técnicos](#))
- 1 salida analógica configurable por jumper 0-10V o 0-20mA ([detalles técnicos](#))

¿QUÉ ES UN P.L.C?



ES UN SISTEMA ELECTRÓNICO PROGRAMABLE DISEÑADO PARA CONTROLAR DIFERENTES TIPOS DE MÁQUINAS O PROCESOS LÓGICOS O SECUENCIALES EN UN ENTORNO INDUSTRIAL

¿Dónde se utiliza?

Principalmente en industrias para automatización y control de procesos.



También en...



Domótica
Semáforos

Barreras y Señalización Automática

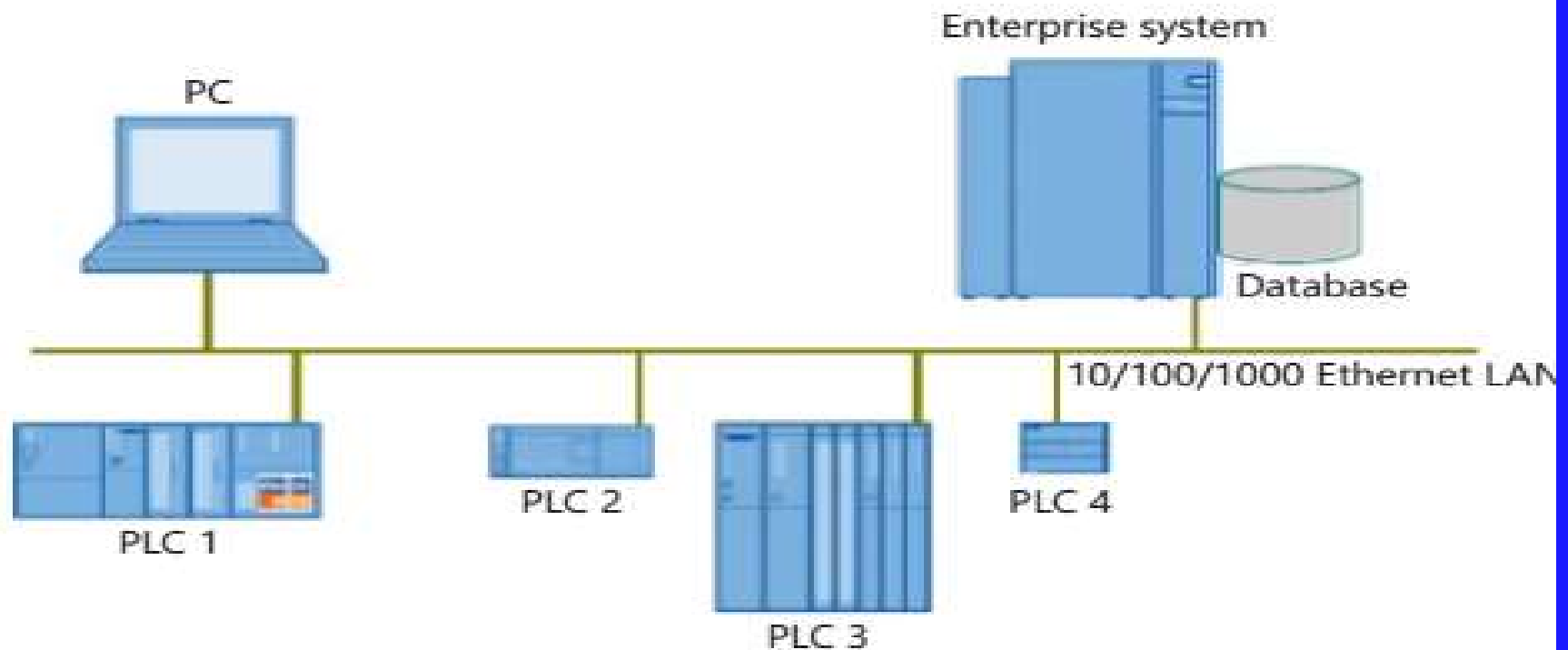
Adquirir datos del proceso por medio de sensores conectados eléctricamente a sus entradas digitales y analógicas.



Actuar sobre el proceso mediante sus salidas digitales y analógicas conectadas eléctricamente a actuadores.



Comunicarse con otros sistemas externos.



Lenguajes de programación

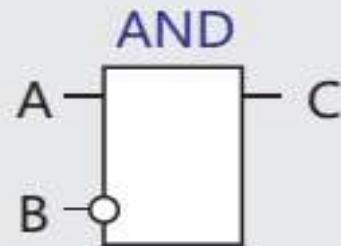
IL

```
LD    A
ANDN B
ST    C
```

ST

C = A AND NOT B

FBD



LD





SISTEMAS
EMBEBIDOS
2021



Computadora Industrial
Abierta Argentina
Desarrollo colectivo



Programación LADDER



para usuarios



SISTEMAS
EMBEBIDOS
2021



Computadora Industrial
Abierta Argentina
Desarrollo colectivo



Origen de IDE4PLC

Software desarrollado originalmente como implementación de referencia de la tesis de Ingeniería en Automatización y Control Industrial del Ing. Eric Pernía:

- “Diseño de software y hardware de un controlador lógico programable (PLC) y su entorno de programación”

Disponible en la web de IDE4PLC.



SISTEMAS
EMBEBIDOS
2021



Computadora Industrial
Abierta Argentina
Desarrollo colectivo



Características de IDE4PLC

Es un entorno de programación para equipos electrónicos PLC que permite realizar las siguientes tareas:

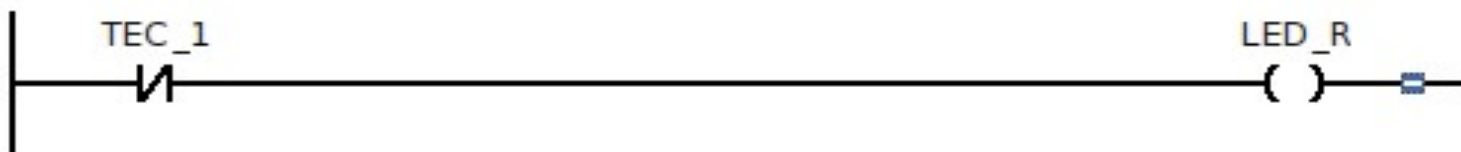
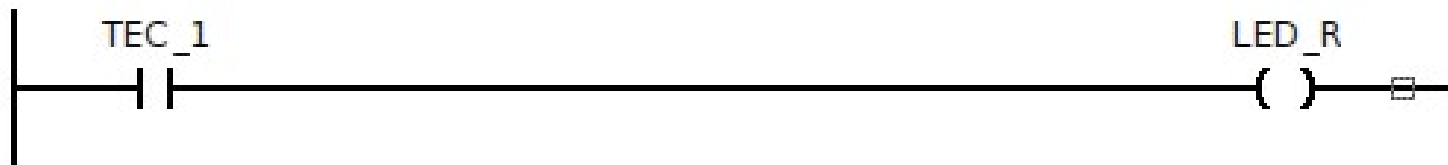
- Crear y editar un programa utilizando un lenguaje de programación de PLC.
- Compilar el programa.
- Descargar el programa a la memoria del Hardware PLC.



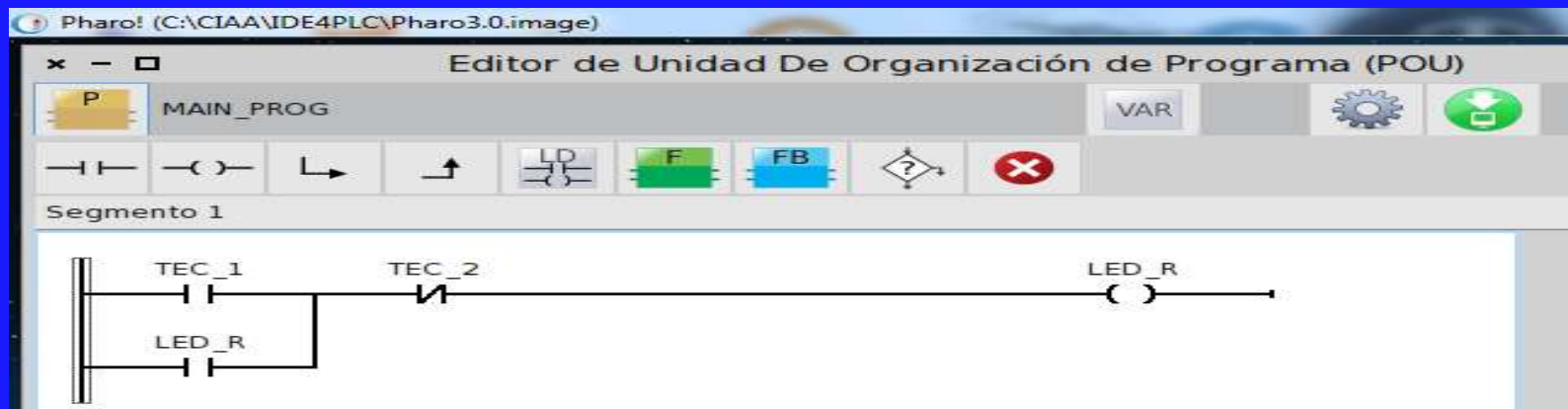
SISTEMAS
EMBEBIDOS
2021



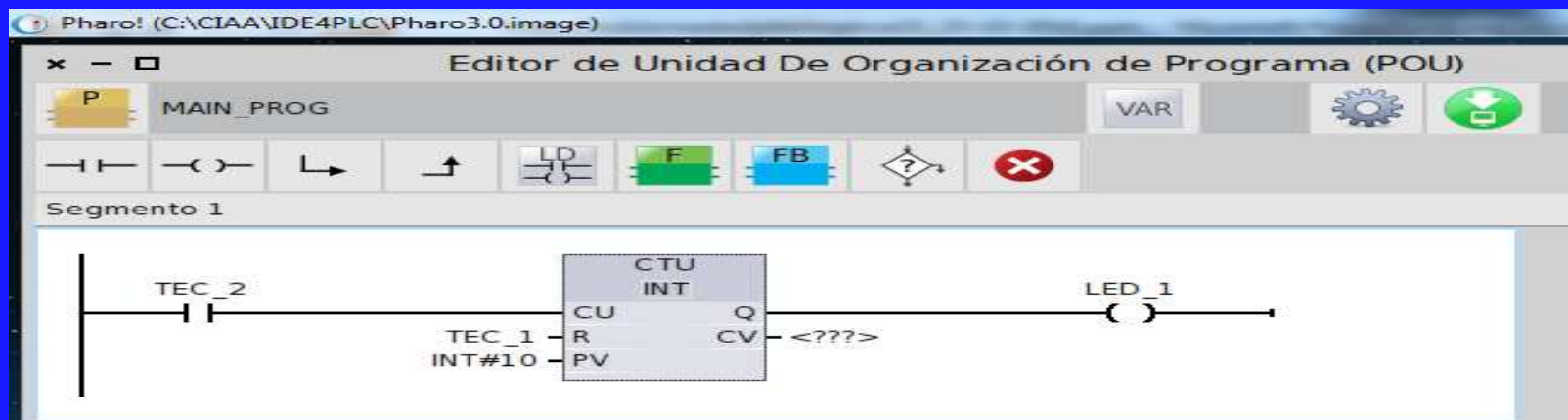
Computadora Industrial
Abierta Argentina
Desarrollo colectivo



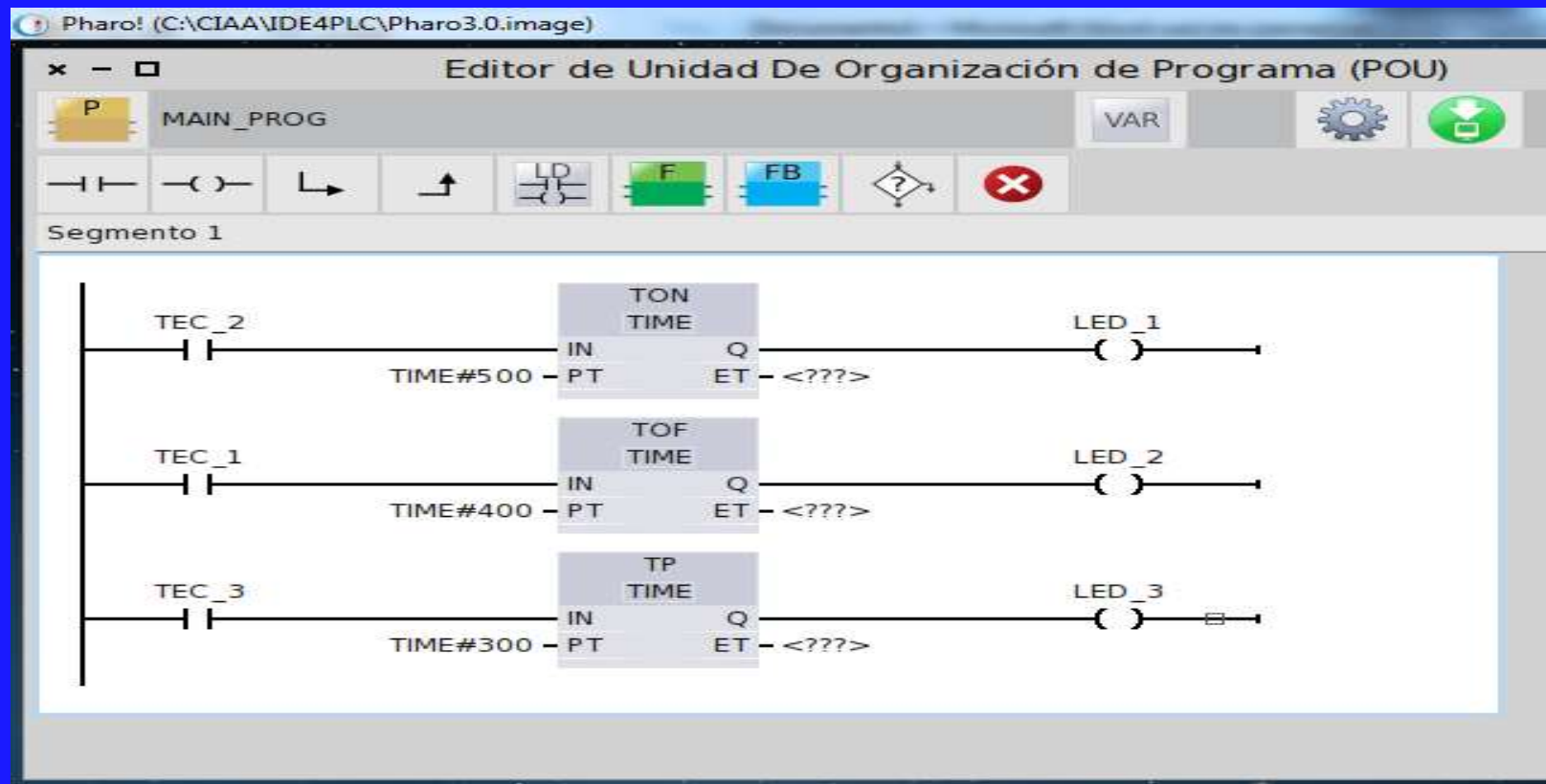
Ejemplos: Circuito con auto retención



Circuito contador



Ejemplos: Temporizadores





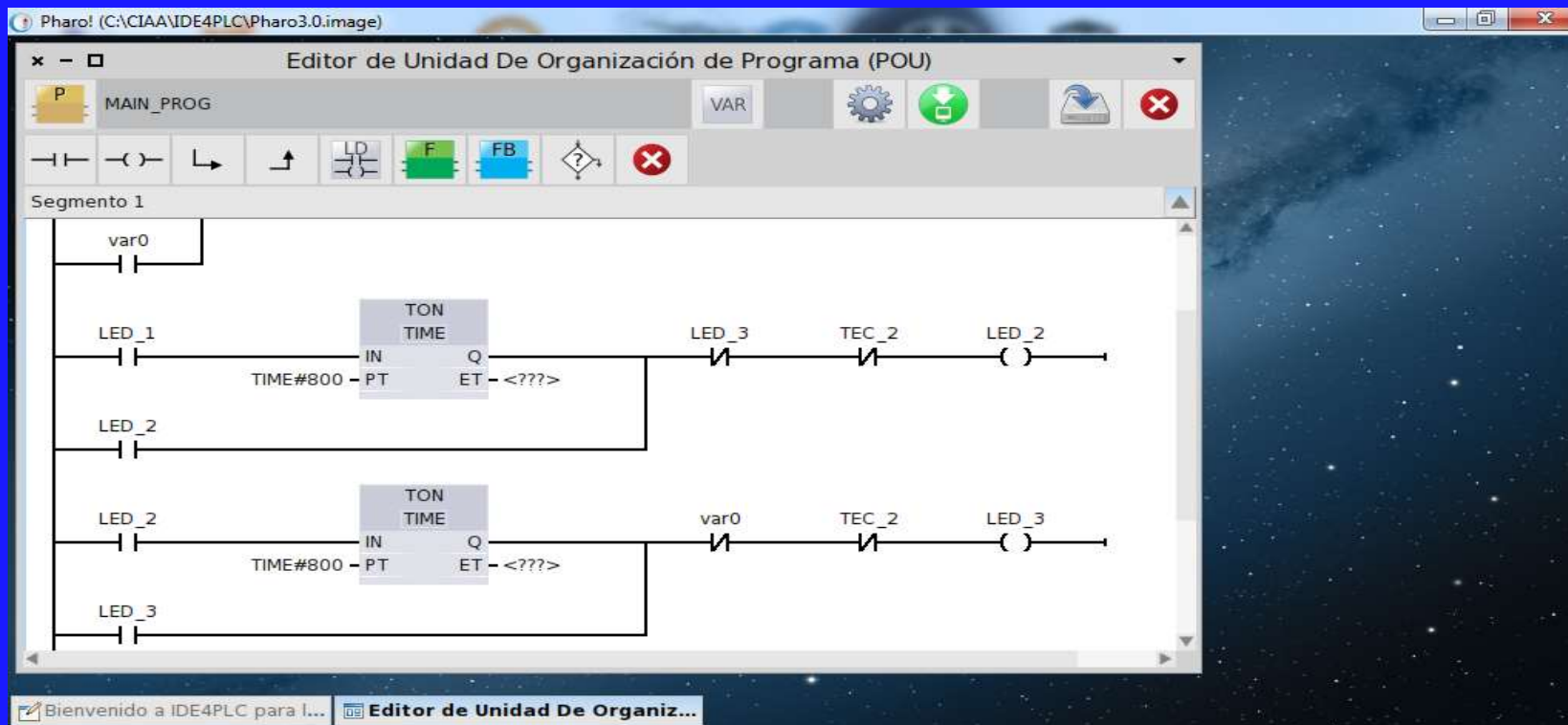
SISTEMAS
EMBEBIDOS
2021



Computadora Industrial
Abierta Argentina
Desarrollo colectivo



SOFTWARE DE SECUENCIA DE LUCES CONTENIDO EN EDU CIAA Y CIAA – NXP





SISTEMAS
EMBEBIDOS
2021



Computadora Industrial
Abierta Argentina
Desarrollo colectivo



PROGRAMACIÓN EN C DE LA EDU CIAA

- BARE METAL (sin sistema operativo)

Programación en C – Baremetal

Ejemplo de aplicación. Inversor de giro motor monofásico

Ubicación de archivos	SRC	INC	
App	<pre># include "Blinky.h" void main (void) { prendeLed (numero); while (TRUE); }</pre> <p>Blinky.c</p>	<pre># include "Led.h" </pre> <p>Blinky.h</p>	CAPAS
Driver	<pre># include "Led.h" void prendeLed (Led??) { LPC_GPIO; }</pre> <p>Led.c</p>	<pre># include "chip.h" void prendeLed (Led??);</pre> <p>Led.h</p>	
LPCOPEN		<pre>chip.h</pre>	



SISTEMAS
EMBEBIDOS
2021



Computadora Industrial
Abierta Argentina
Desarrollo colectivo



C/C++ - Firmware/projects/drivers_bm/src/ind.c - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer

- 14_Adquisicion_Transmision2
- 15_MotorLavarropa
- 16_Motor_Tecla
- 17_lcd
- 17_usolcd
- 18_MotorTecla_lcd
- 19_inout_lcd-poncho
- 20_lcd-poncho-sensores
- blinking
- drivers_bm
- EJ_1_BM
- EJ_2_BM
- EJ_7_BM
- ej01_sapi_switches_leds
- ej04_sapi_sequences
- ej08_sapi_adc_dac
- ej09_sapi_rtc
- ej12_sapi_hmc5883l
- lcd
- myproject
- Plantilla_1_blinking_baremetal
- sapi_bm
- sAPI_LCD_MAG
- 1_blinking_baremetal.rar
- 19_inout_lcd-poncho.rar
- drivers_bm.rar
- lcd.rar
- RemoteSystemsTempFiles
 - circle.yml
 - Makefile
 - Makefile.config
 - Makefile.mine
 - readme.md

```

70 #error CPU shall be defined
71 #endif
72 #if (lpc4337 == CPU)
73 #include "chip.h"
74 #elif (mk60fx512vlq15 == CPU)
75 #else
76 #endif
77 #include "ind.h"
78 #include "stdint.h"
79
80
81 /*-----[macros and definitions]-----*/
82 #define IND_1PAQUETE 1 /*puerto o paquete 1 para COL0 */
83 #define IND_4PAQUETE 4 /*puerto o paquete 4 para F0,F1,F2 y F3 */
84 #define IND_7PAQUETE 7 /*puerto o paquete 7 para COL1 Y COL2 */
85
86 #define IND0_PIN 4 /*PINES DEL PAQUETE 0 PUERTO 7 - COL1*/
87 #define IND1_PIN 5 /*PINES DEL PAQUETE 0 PUERTO 7 - COL2*/
88 #define IND2_PIN 0 /*PINES DEL PAQUETE 0 PUERTO 4 - F0*/
89 #define IND3_PIN 1 /*PINES DEL PAQUETE 0 PUERTO 4 - F1*/
90 #define IND4_PIN 5 /*PINES DEL PAQUETE 0 PUERTO 1 - COL0*/
91 #define IND5_PIN 2 /*PINES DEL PAQUETE 0 PUERTO 4 - F2*/
92 #define IND6_PIN 3 /*PINES DEL PAQUETE 0 PUERTO 4 - F3*/
93
94
95 #define IND0_BIT 12 /* NRO DE BIT Q ACOMPAÑA AL GPIO3*/
96 #define IND1_BIT 13 /* NRO DE BIT Q ACOMPAÑA AL GPIO3*/
97 #define IND2_BIT 0 /* NRO DE BIT Q ACOMPAÑA AL GPIO2*/

```

Out...

- chip.h
- ind.h
- stdint.h
- # IND_1PAQUETE
- # IND_4PAQUETE
- # IND_7PAQUETE
- # IND0_PIN
- # IND1_PIN
- # IND2_PIN
- # IND3_PIN
- # IND4_PIN
- # IND5_PIN
- # IND6_PIN
- # IND0_BIT
- # IND1_BIT
- # IND2_BIT
- # IND3_BIT
- # IND4_BIT
- # IND5_BIT
- # IND6_BIT
- # NUMGPIO_IND4
- # NUMGPIO_IND235Y6
- # NUMGPIO_IND0Y1

Problems 4 errors, 11 warnings, 0 others

Description	Resource	Path	Location	Type
> Errors (4 items)				
> Warnings (11 items)				

Capa de aplicación

TECGIRO.C

TECGIRO.H

```

h tecgiro.h  c *tecgiro.c
4 la secuencia del modo uno. Si se acciona la Tec 2: gira en otro sentido
5 muestra secuencia en lcd y luego se apaga.
6 Si se selecciona la Tecla "TEC3" gira en un sentido, se para,
7 y luego gira en otro sentido
8 */
9
10 /*se agrega para el lcd*/
11
12 #include "puertos.h"
13 #include "lcd.h"
14 #include "lpc_types.h"
15
16 /*hasta aca*/
17
18 #include "tecgiro.h"
19 #include "tecla.h"
20 #include "rele1.h"
21 #include "triac.h"
22 #include "stdint.h"
23 #include "led.h"

```

Problems Tasks Console Properties
3 errors, 1 warning, 0 others

```

h tecgiro.h  c *tecgiro.c
1 /* Copyright 2016, XXXXXXXX
33
34 #ifndef TECGIRO_H
35 #define TECGIRO_H
36 /** \brief Bare Metal example header file
37 **
38 ** This is a mini example of the CIAA Firmware
39 **
40 **/
41
42 /** \addtogroup CIAA_Firmware CIAA Firmware
43 ** @{ */
44 /** \addtogroup Examples CIAA Firmware Examples
45 ** @{ */
46 /** \addtogroup Baremetal Bare Metal example header file
47 ** @{ */
48
49 /*
50 * Initials   Name
51 * -----

```

Problems Tasks Console Properties
3 errors, 1 warning, 0 others



capa driver relé.c (src) relé.h (inc)

```

77 #define RELE12 3 //Nro de bit GPIO3[3]pin
78 #define TRIAC 4 // Nro de bit GPIO3[4]pin
79 #define PORT_NUM_RELETRIAC 3 // Nro de GPIO port
80
81 #define OUT 1 //1:salida 0:entrada
82
83
84 void Iniciomotor (void)
85 {
86   Chip_GPIO_Init(LPC_GPIO_PORT);
87   Chip_SCU_PinMux(LED_PAQUETE, RELES, MD_PUP, FUNC0); /*mapea P6_1 en GPIO3[0], Rele 5v y habilita el pull up */
88   Chip_SCU_PinMux(LED_PAQUETE, RELE12, MD_PUP, FUNC0); /*mapea P6_4 en GPIO3[3], Rele 12 v y habilita el pull up */
89   Chip_SCU_PinMux(LED_PAQUETE, TRIAC, MD_PUP, FUNC0); /* mapea P6_5 en GPIO3[4], Triac y habilita el pull up */
90   Chip_GPIO_SetDir(LPC_GPIO_PORT, PORT_NUM_RELETRIAC, 1<<0|1<<3|1<<4, OUT); //Numero de GPIO=3 -numero de bit-out=1 salida
91
92 }
93 void arranquemotor(uint8_t paso)
94 {
95   switch(paso)
96   {
97     case P1_P_TRIAC:
98       Chip_GPIO_SetPinOutHigh(LPC_GPIO_PORT, PORT_NUM_RELETRIAC, TRIAC);
99       break;
100
101     case P2_P_RELE_12V:
102       //Chip_GPIO_SetPinOutHigh(LPC_GPIO_PORT, PORT_NUM_RELETRIAC, RELE12);

```

```

70 #define P4_P_RELE_5V 3
71 #define P5_P_RELE_12V 4
72 #define P6_A_RELE_12V 5
73 #define P7_A_RELE_5V 6
74 #define P8_A_TRIAC 7
75
76
77 //define TRIAC1 0
78
79
80
81 //define VERDE 1
82 //define AMARILLO 2
83 //define RGB_ROJO 3
84 //define RGB_VERDE 4
85 //define RGB_AMARILLO 5
86
87 /*=====typedef=====*/
88
89 /*=====external data declaration=====*/
90
91
92
93 void Iniciomotor (void);
94 void arranquemotor (uint8_t paso);
95 //void conectarRela(uint8_t rela);

```



SISTEMAS
EMBEBIDOS
2021



**Computadora Industrial
Abierta Argentina**
Desarrollo colectivo



APLICACIONES



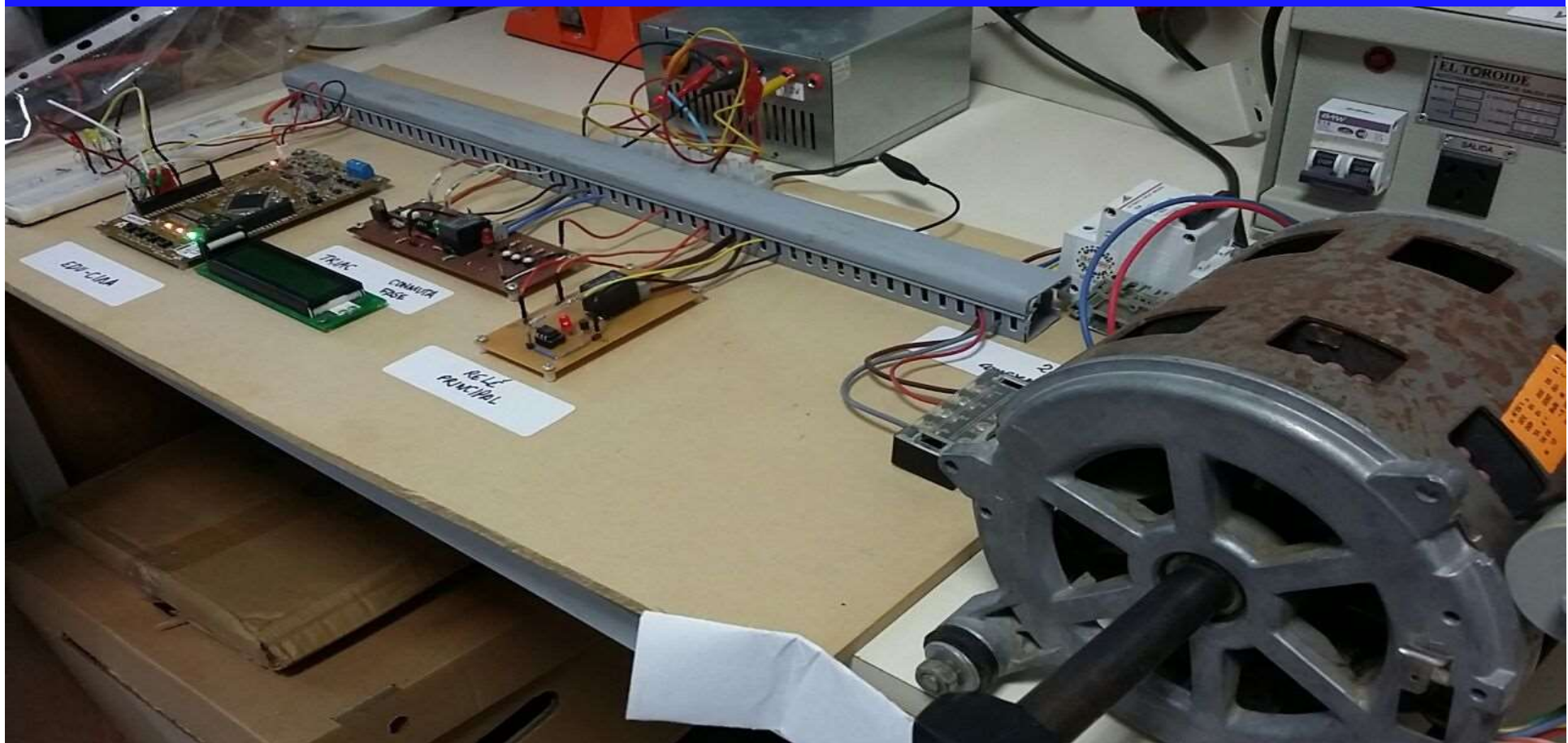
SISTEMAS
EMBEBIDOS
2021

CIA

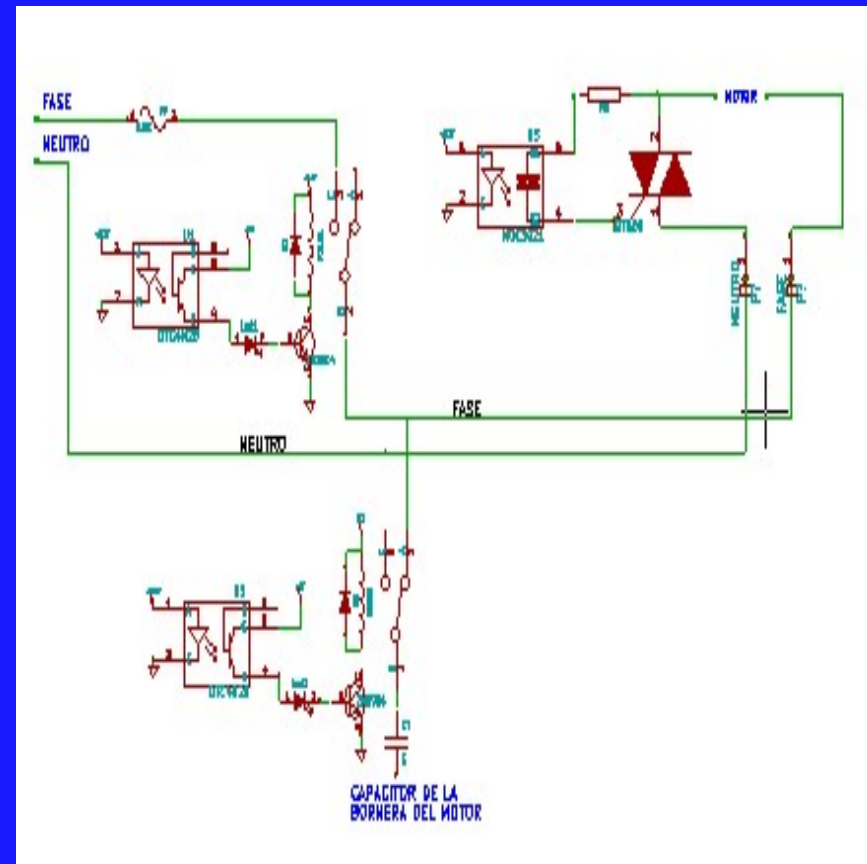
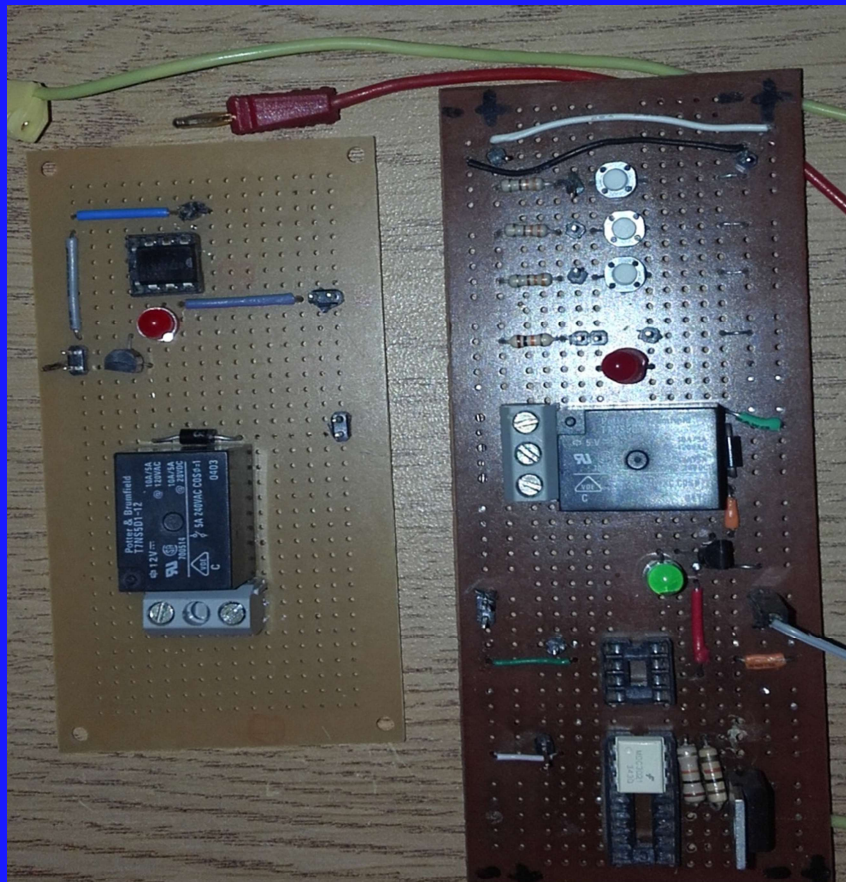
Computadora Industrial
Abierta Argentina
Desarrollo colectivo



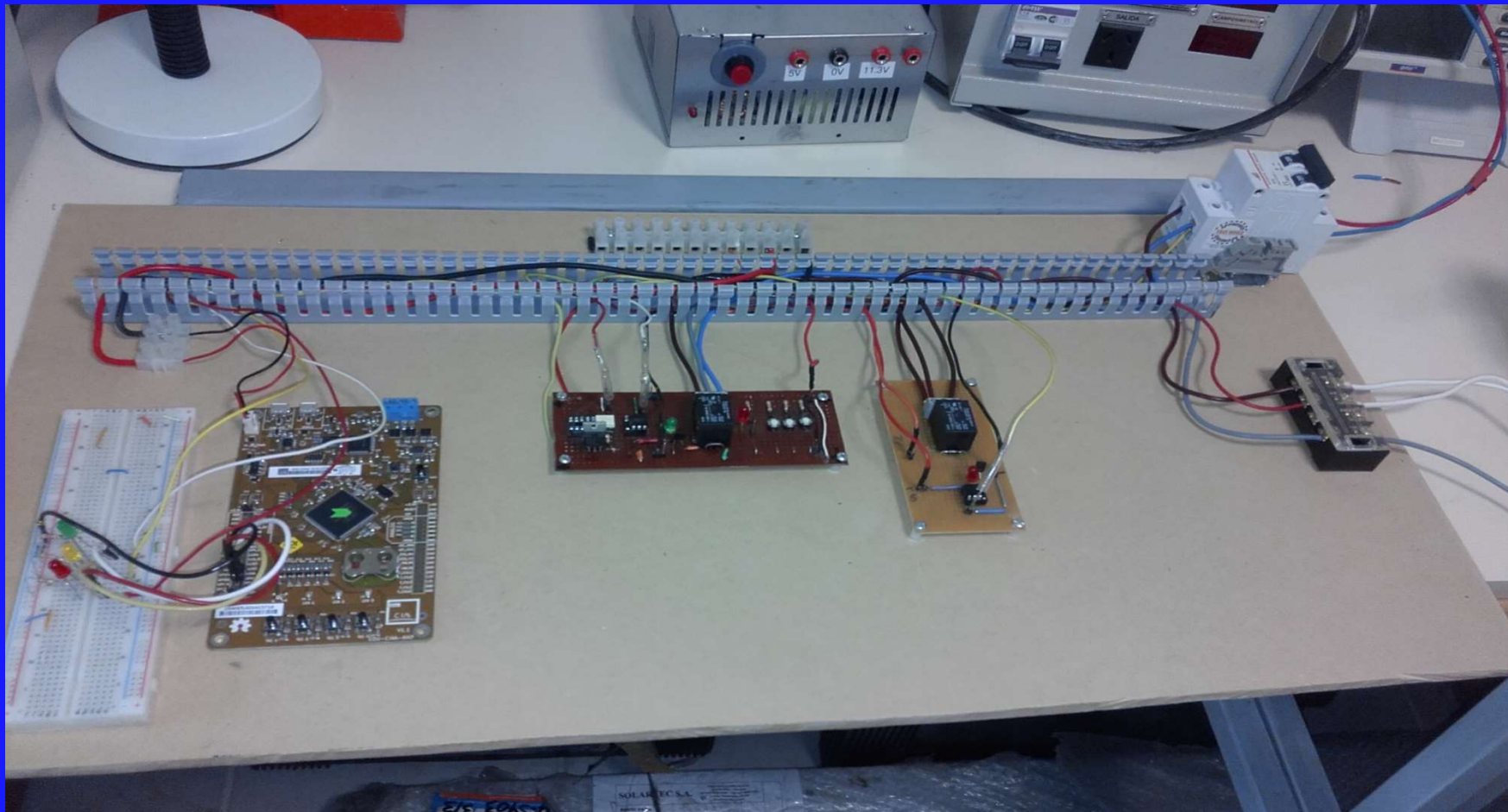
Ejemplo de aplicación. Inversor de giro motor monofásico



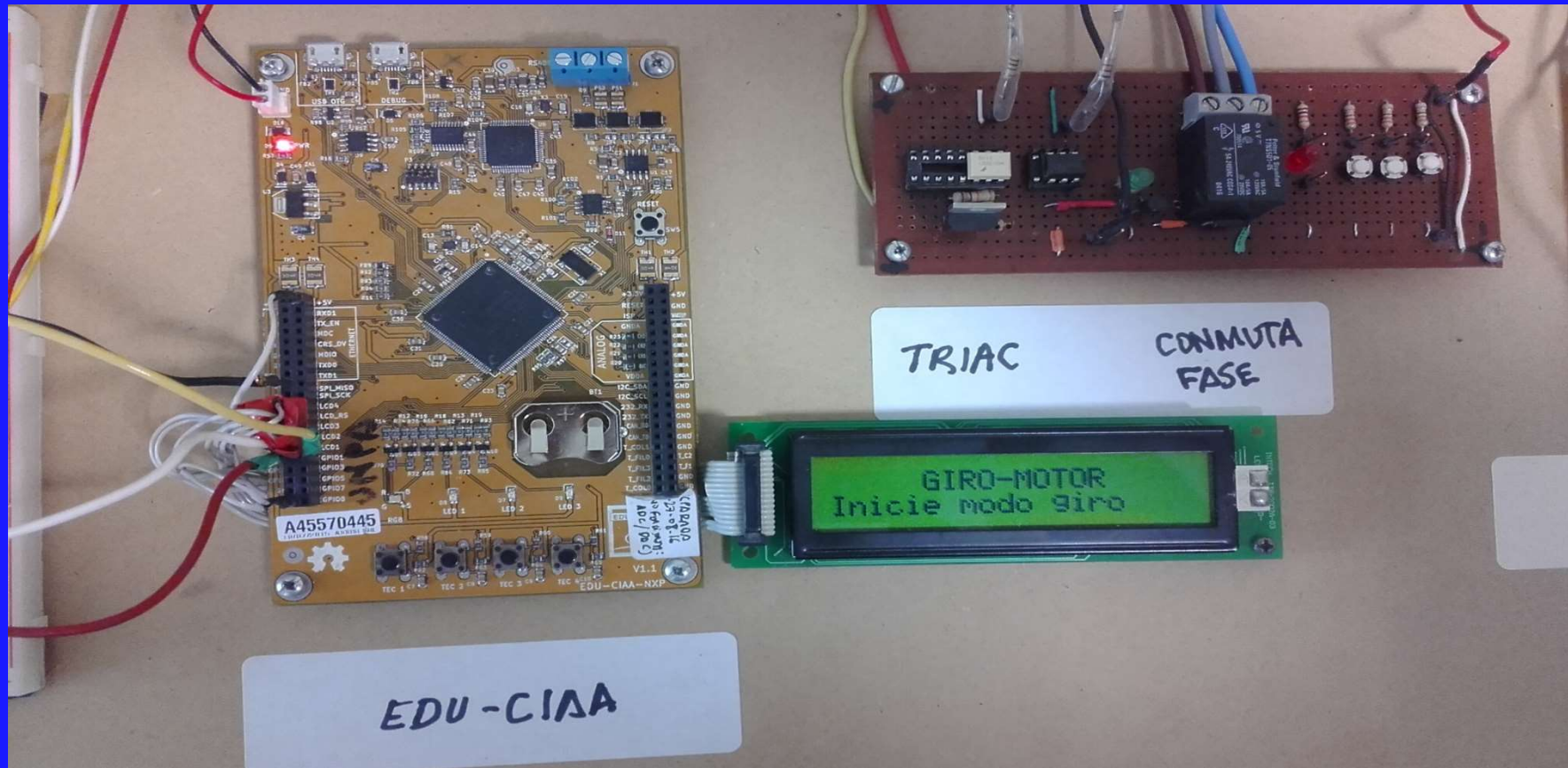
Drivers de inversión de giro



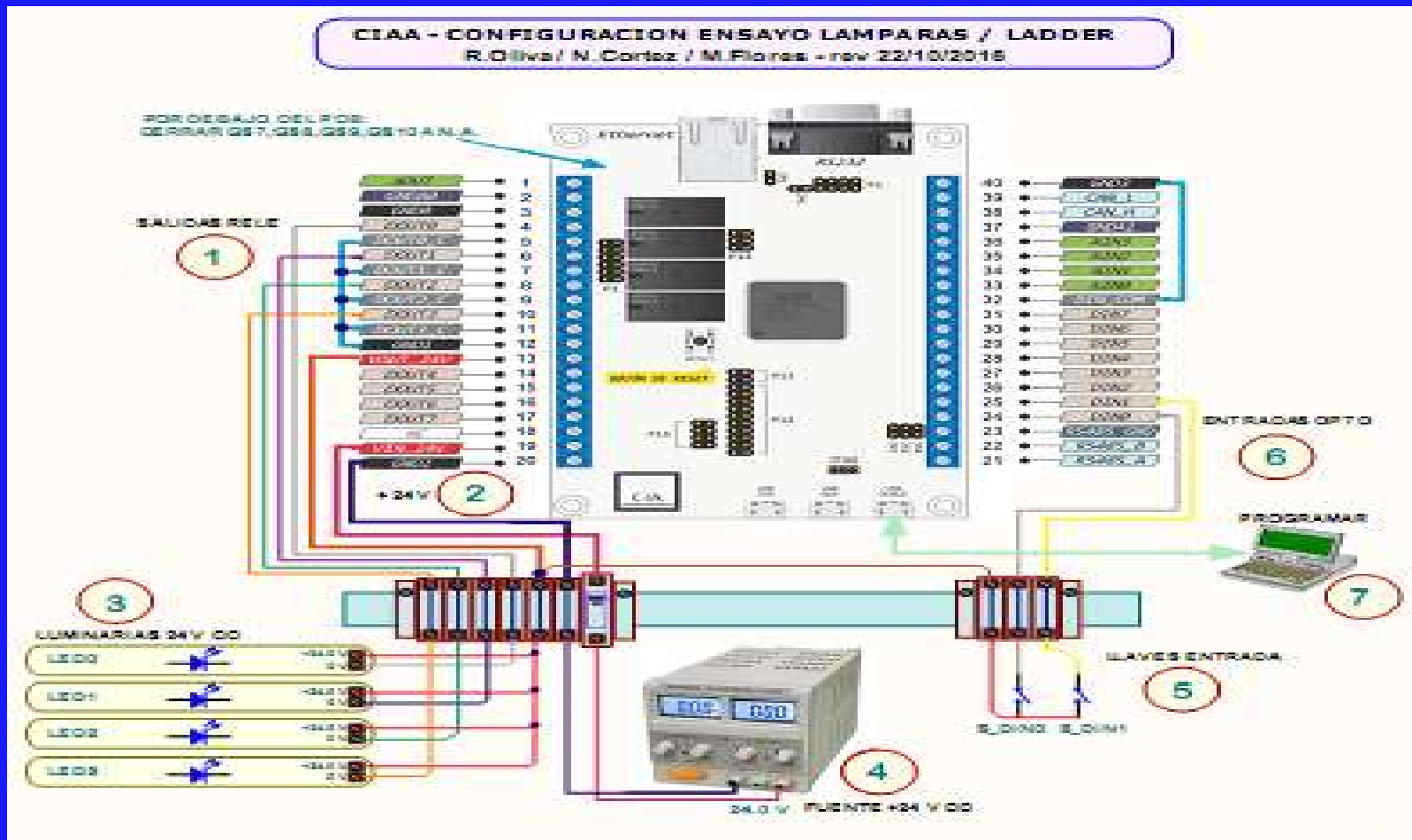
Tablero del inversor de giro



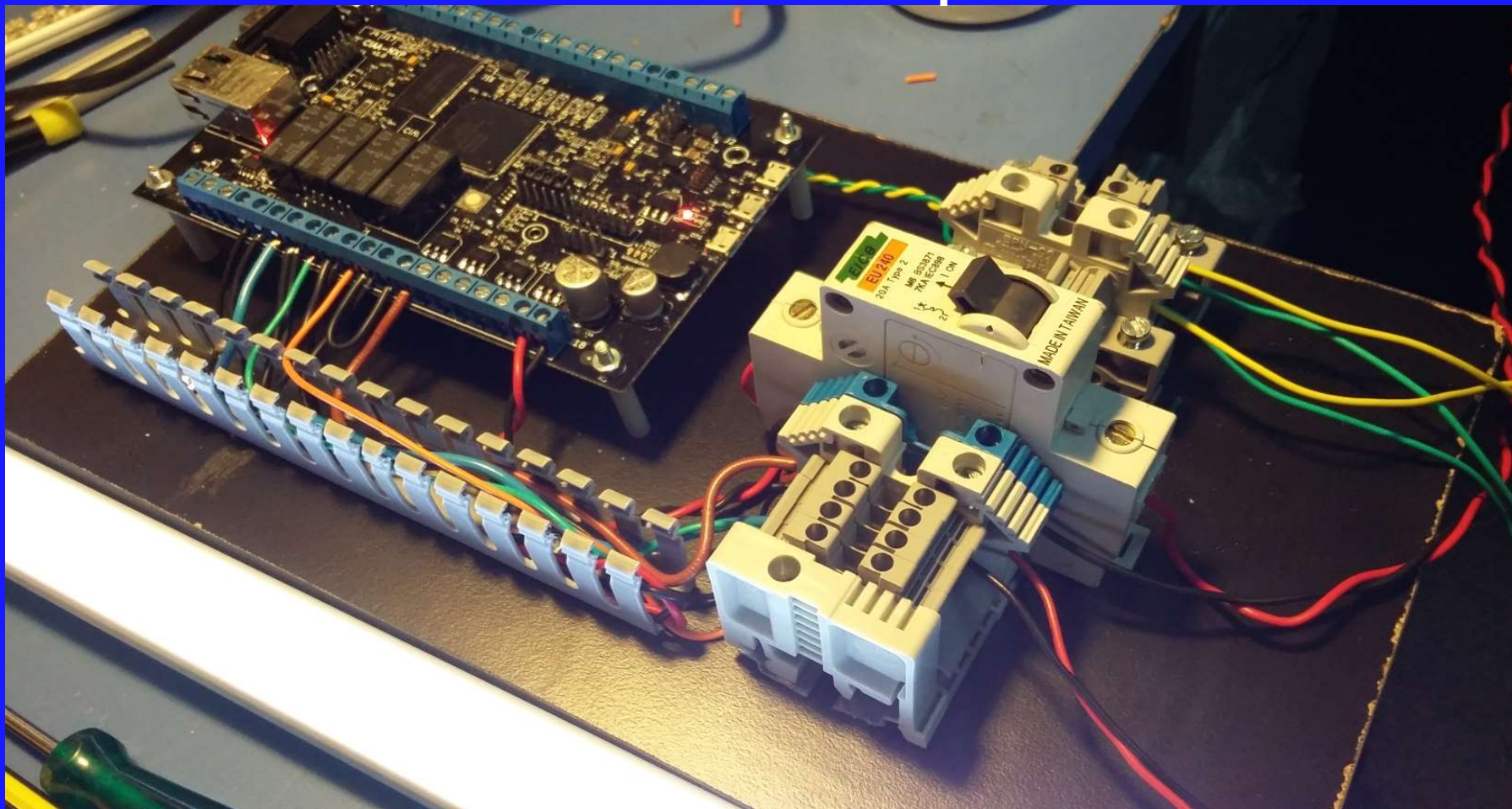
Tablero del inversor de giro - LCD



Esquema de conexión de la CIAA - Secuencia



Conexión de la CIAA para control





SISTEMAS
EMBEBIDOS
2021



Computadora Industrial
Abierta Argentina
Desarrollo colectivo



MÓDULOS CONECTABLES A LA EDU CIAA

-PONCHOS-



SISTEMAS
EMBEBIDOS
2021

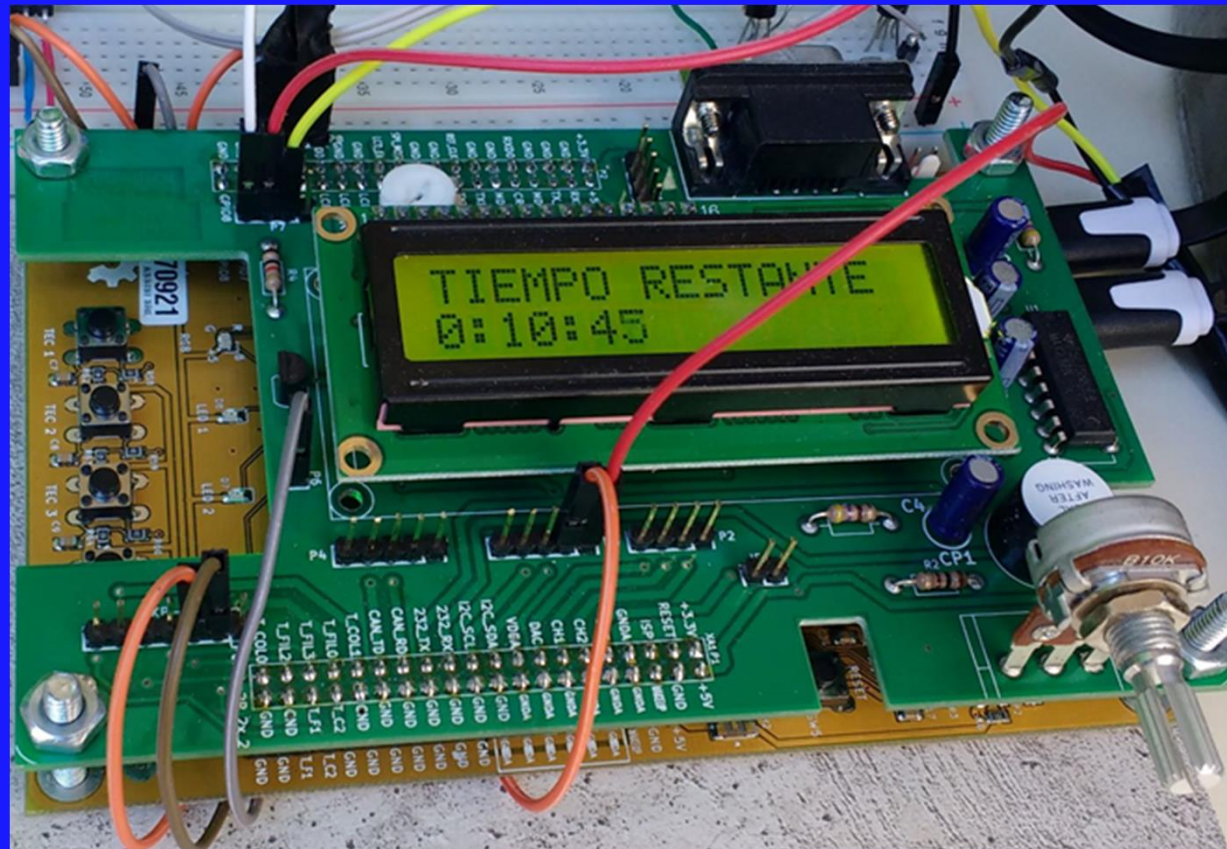


Computadora Industrial
Abierta Argentina
Desarrollo colectivo



PONCHOS PARA LA EDU CIAA

Llamamos “Ponchos” a los módulos conectables sobre alguna CIAA que extienden la funcionalidad de la misma agregando el hardware necesario para distintas aplicaciones.



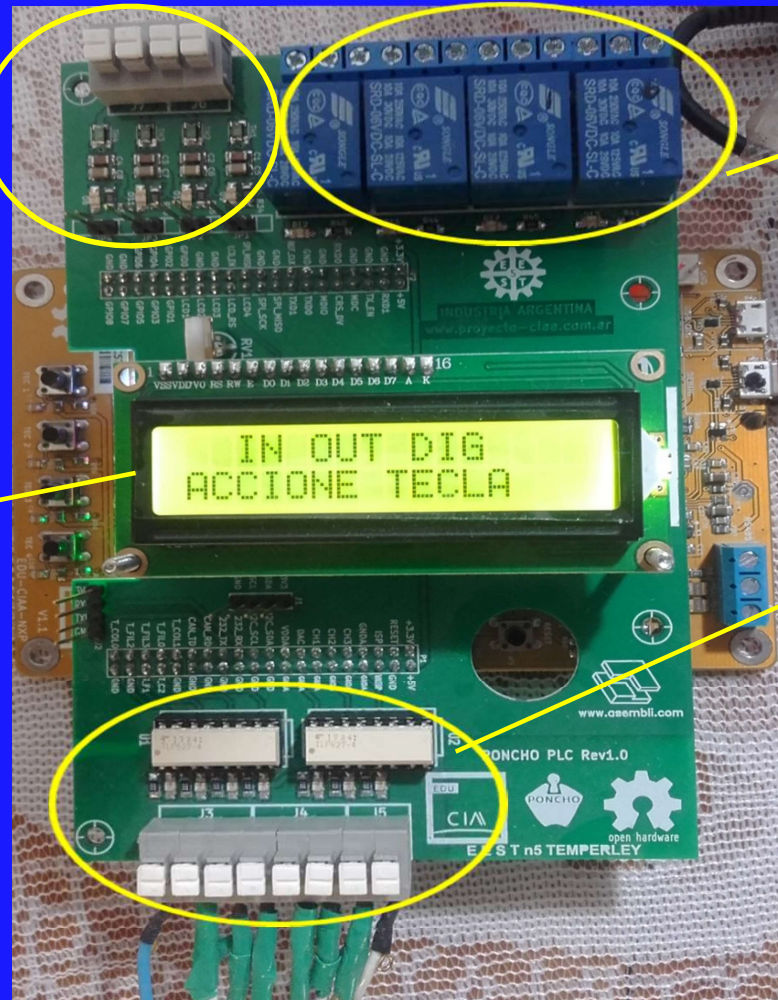
PONCHO P.L.C

SALIDAS DIGITALES
TRANSISTOR

SALIDAS DIGITALES
RELÉ

DISPLAY LCD

ENTRADAS DIGITALES
OPTOACOPLADAS

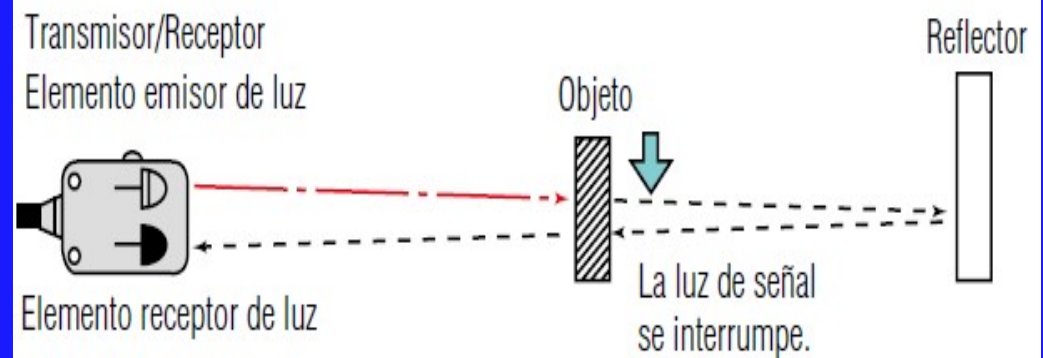


ENTRADAS - SALIDAS - DIGITALES

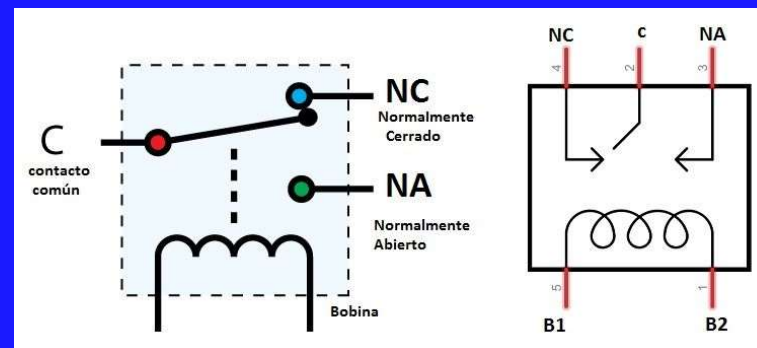
SENSORES CAPACITIVOS - INDUCTIVOS



SENSOR ÓPTICO RETROREFLECTIVO



SALIDA RELÉ

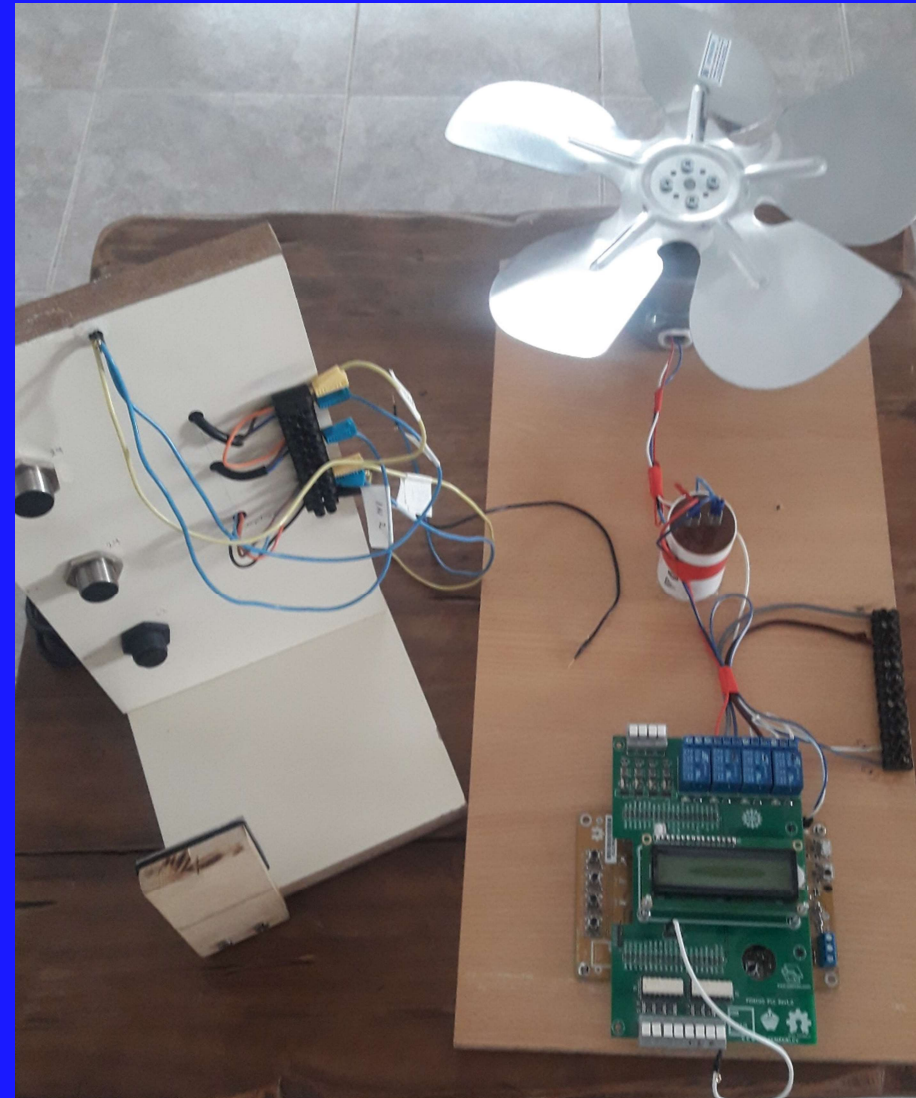
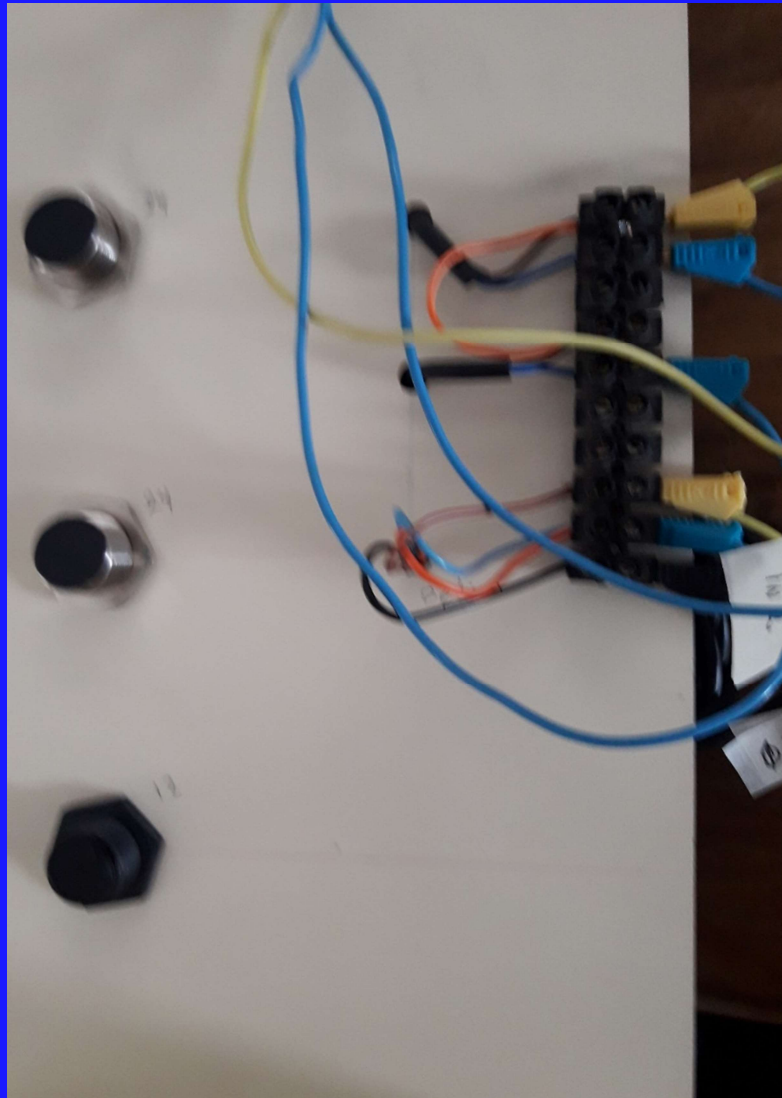




SISTEMAS
EMBEBIDOS
2021



Computadora Industrial
Abierta Argentina
Desarrollo colectivo

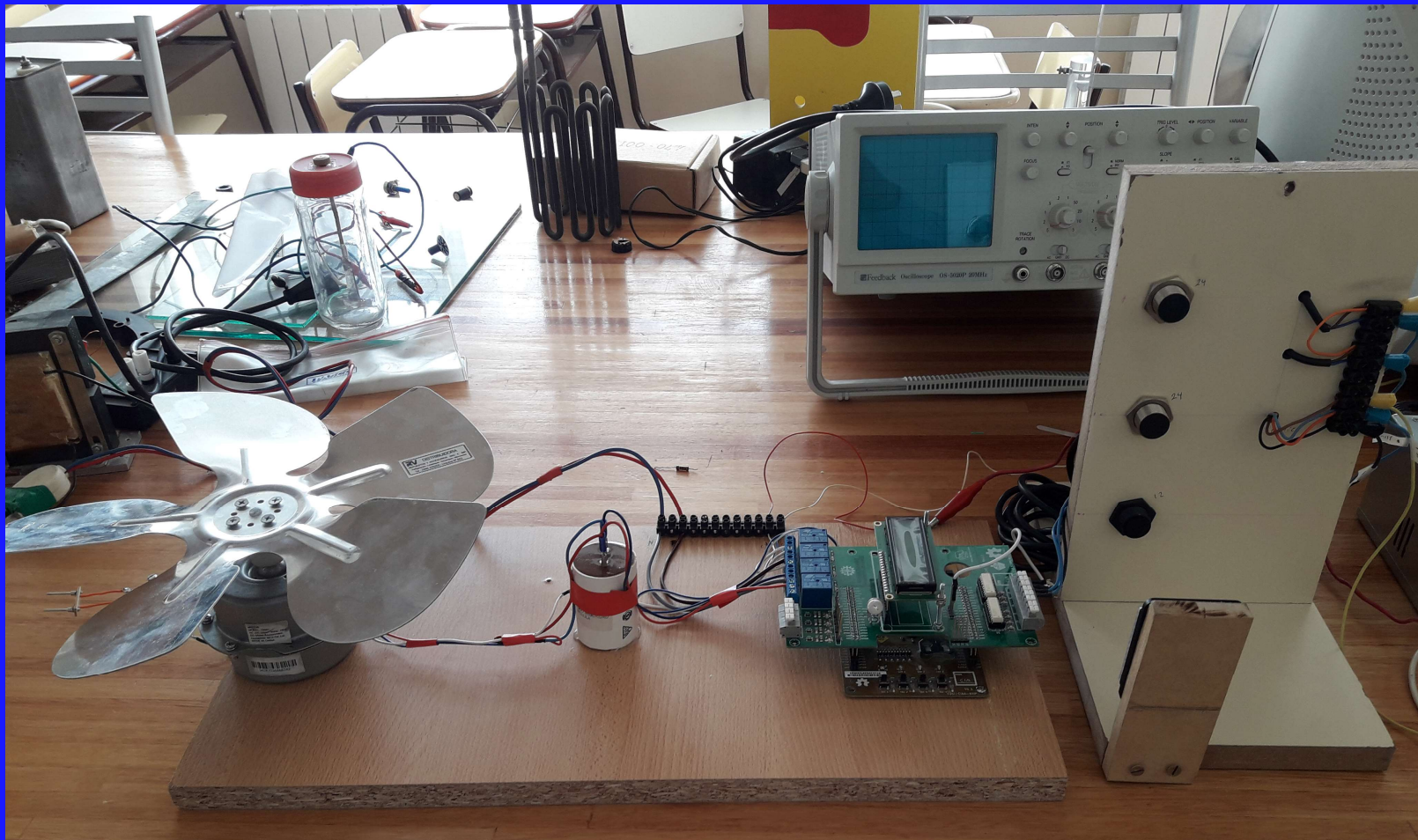




SISTEMAS
EMBEBIDOS
2021



Computadora Industrial
Abierta Argentina
Desarrollo colectivo





<https://www.youtube.com/watch?v=Jr6z9elGYnc>



SISTEMAS
EMBEBIDOS
2021

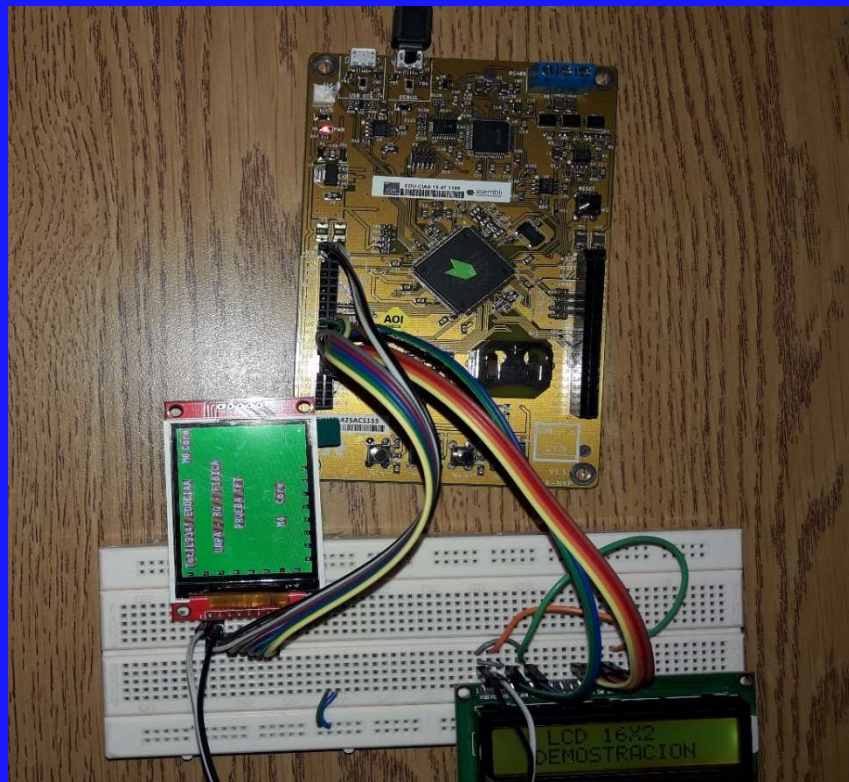


Computadora Industrial
Abierta Argentina
Desarrollo colectivo



EDU CIAA - LCD

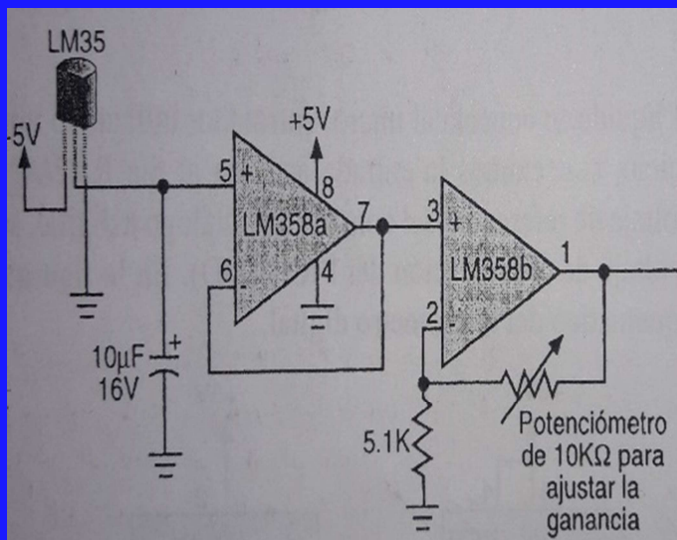
LABORATORIO VIRTUAL REMOTO - EDU CIAA



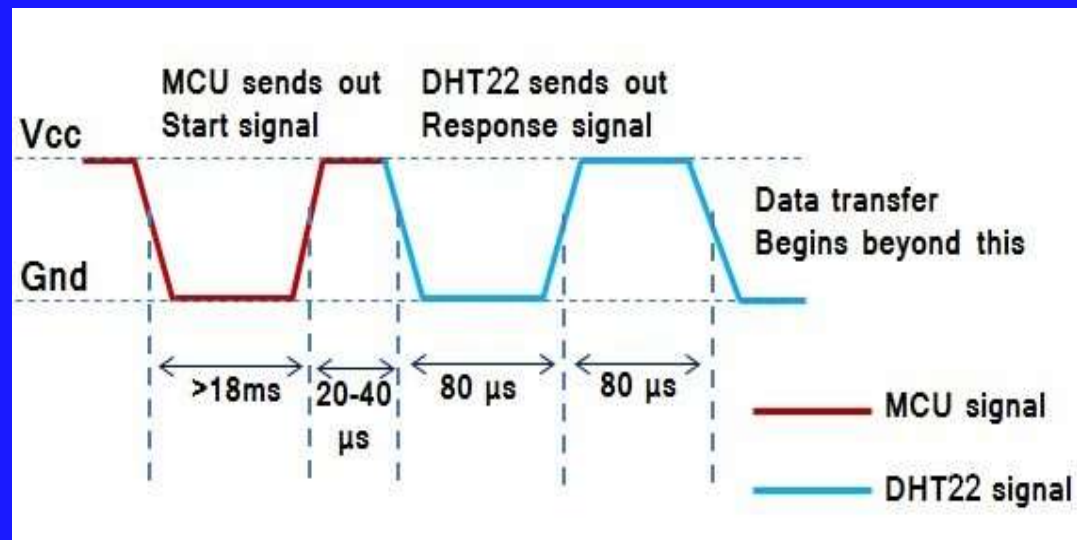
<https://youtu.be/hSuYMOONWlo>

https://www.youtube.com/watch?v=S5G6ET4t_yA&feature=emb_logo

SENSOR LM35



COMUNICACIÓN SENSOR DHT22

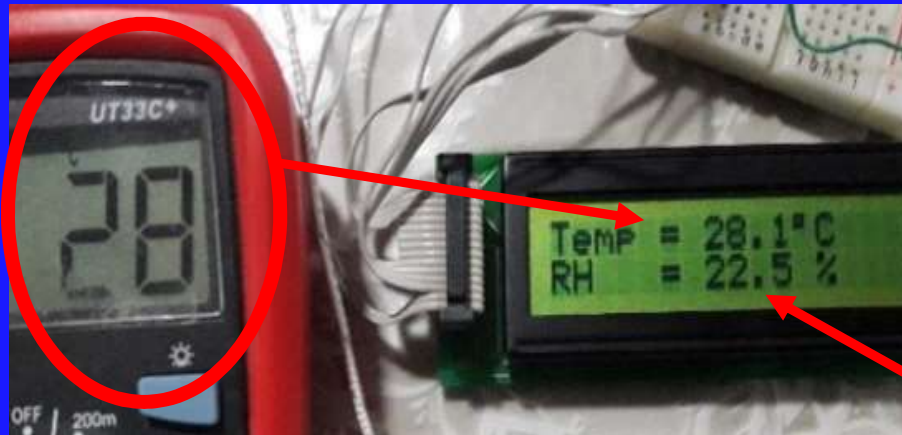




SISTEMAS EMBEBIDOS 2021



Computadora Industrial Abierta Argentina
Desarrollo colectivo



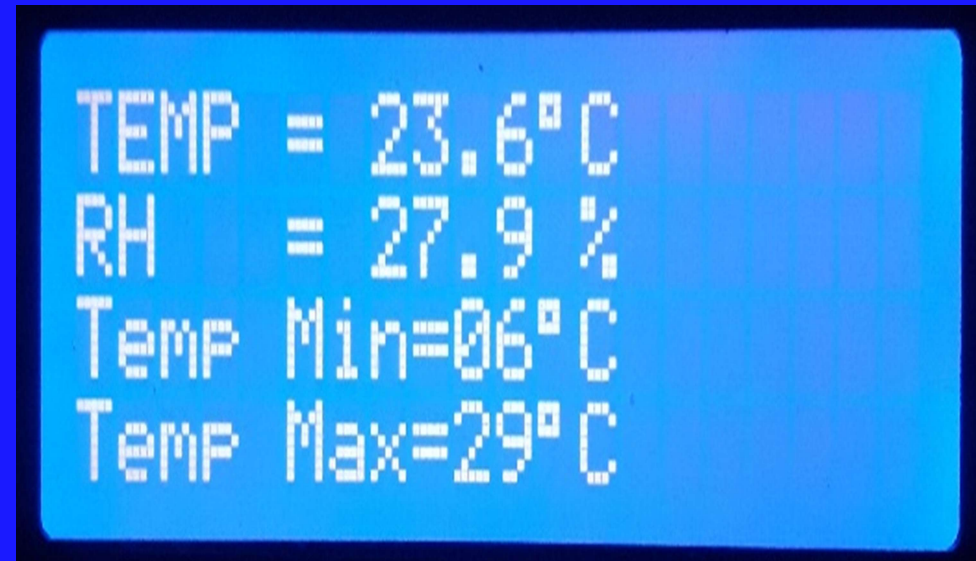
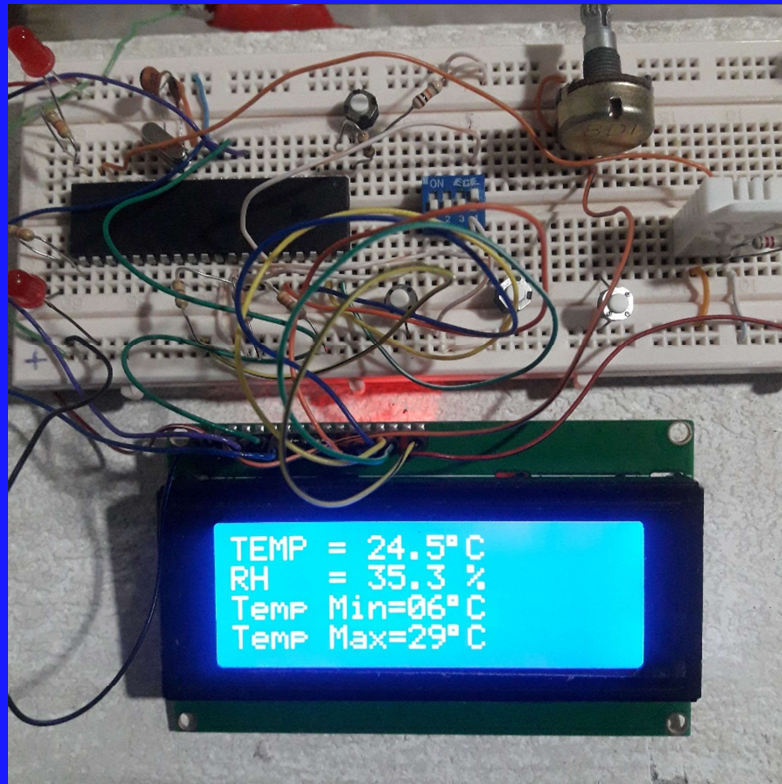
SENSOR LM35

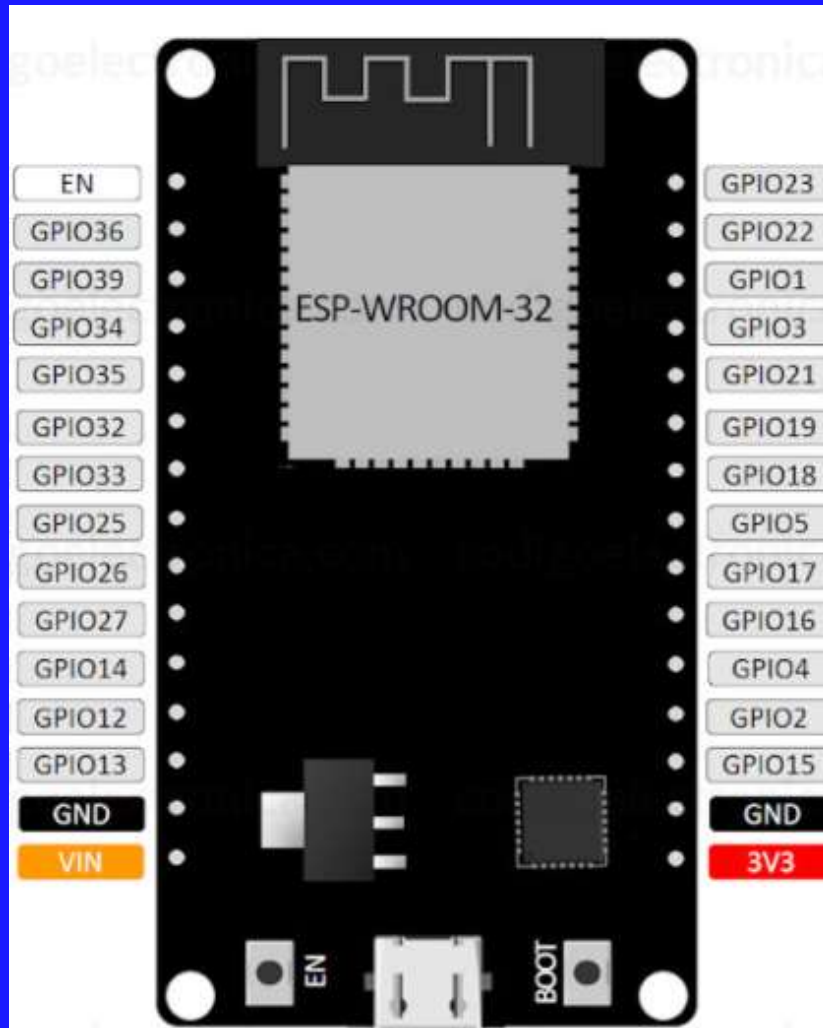


DHT22

CIRCUITO DEL PROTOTIPO MEDIDOR DE TEMPERATURA Y HUMEDAD

DISPLAY LCD 20 X 4 SENSOR DHT 22

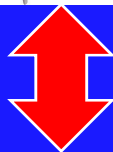
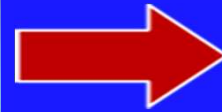
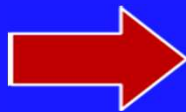
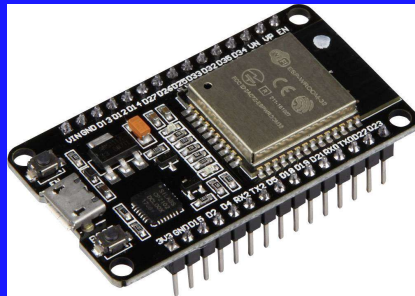




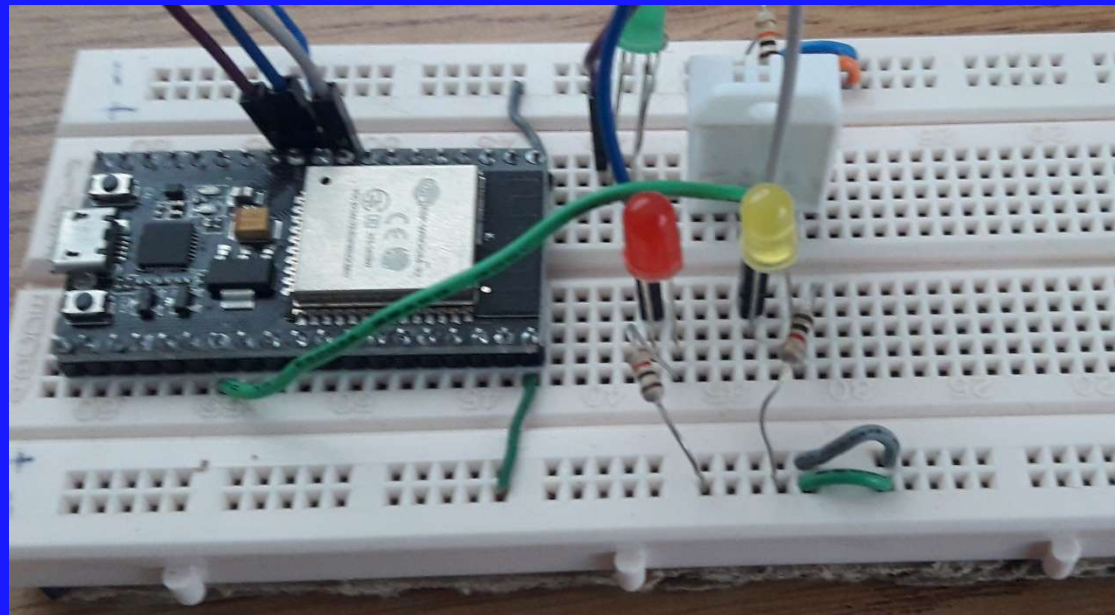
ESP 32

- Microprocesador de 32 – bits
- Interfaz serie-USB y regulador de voltaje
- Comunicación inalámbrica Wifi – Bluetooth
- Interfases periféricas: I2C, SPI, CAN, PWM, etc
- Entradas digitales y analógicas de 12 bit
- Salidas Analógicas PWM
- Controlador SD/MMC
- Controlador remoto infrarrojo
- Algunos modelos tienen integrados: Cámaras, pantallas Oled, Modulos LoRa, etc.

ESP32 MCU - DHT 22



Sensor
DHT22





```

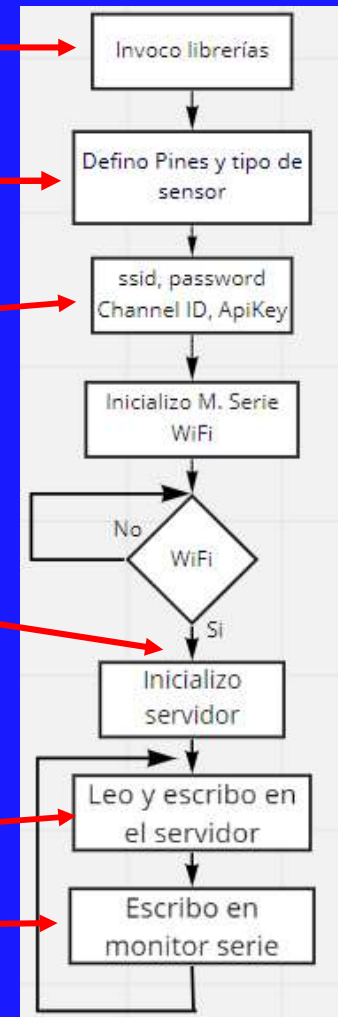
1 #include "ThingSpeak.h" // agregado para comunicación thingspeak
2 #include "WiFi.h" // agregado para comunicación thingspeak
3 #include "DHT.h"
4
5
6 #define DHTPIN 4 // defino el pin donde estará conectado el
7 #define DHTTYPE DHT22 // defino el tipo de sensor
8
9
10 const char* ssid = "xxxxxxxxx"; // agregado para comunicac
11 const char* password = "xxxxxxxxx"; // agregado para comuni
12
13 unsigned long channelID =144965; // agregado para comun
14 const char* WriteAPIKey = "1ZLADKSNDS30OU"; // agregado para con
15

```

```

.....WIFI CONECTADO!
Temperatura DHT22: 23.00 °C.
Humedad DHT22: 36.10 %.
-----
datos enviados a Write APIKey
Temperatura DHT22: 23.10 °C.
Humedad DHT22: 35.90 %.
-----
datos enviados a Write APIKey
Temperatura DHT22: 23.50 °C.
Humedad DHT22: 35.90 %.
-----
datos enviados a Write APIKey

```





SISTEMAS
EMBEBIDOS
2021



Computadora Industrial
Abierta Argentina
Desarrollo colectivo



ThingSpeak™ Canales Aplicaciones Apoyo ▾

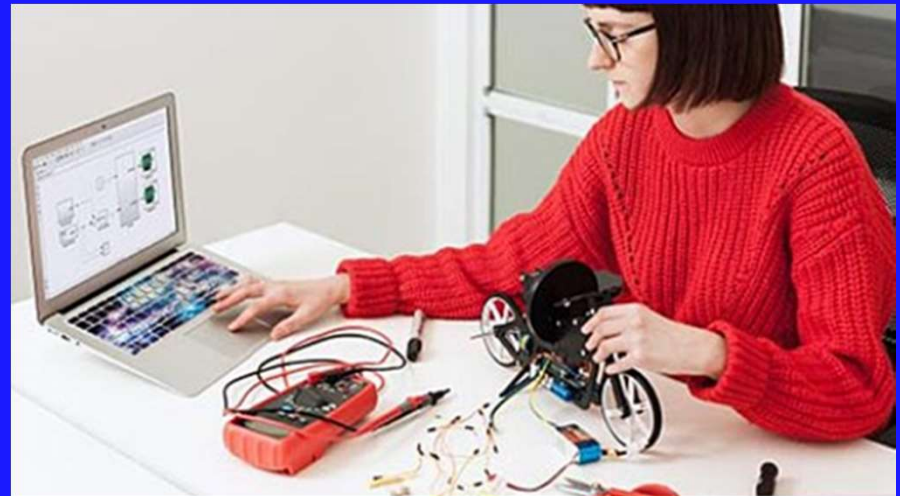
Uso comercial Cómo comprar

ThingSpeak para proyectos de IoT

Recopilación de datos en la nube con análisis de datos avanzado usando MATLAB

Empiece gratis

Aprende más



ThingSpeak para estudiantes y educadores

Implemente proyectos de investigación de IoT rápidamente con herramientas integradas de análisis de datos de MATLAB y recopilación de datos de sensores en tiempo real

Acerca de ThingSpeak

ThingSpeak es un servicio de plataforma de análisis de IoT que le permite agregar, visualizar y analizar flujos de datos en vivo en la nube. Puede enviar datos a ThingSpeak desde sus dispositivos, crear una visualización instantánea de datos en vivo y enviar alertas.



Recoger

Envíe los datos del sensor de forma privada a la nube.



Analizar

Analice y visualice sus datos con MATLAB.



actuar

Desencadena una reacción.



SISTEMAS EMBEBIDOS 2021



Computadora Industrial Abierta Argentina Desarrollo colectivo



To use ThingSpeak, you must sign in with your existing MathWorks account or create a new one.

Non-commercial users may use ThingSpeak for free. Free accounts offer limits on certain functionality. Commercial users are eligible for a time-limited free evaluation. To get full access to the MATLAB analysis features on ThingSpeak, log in to ThingSpeak using the email address associated with your university or organization.

To send data faster to ThingSpeak or to send more data from more devices, consider the [paid license options](#) for commercial, academic, home and student usage.

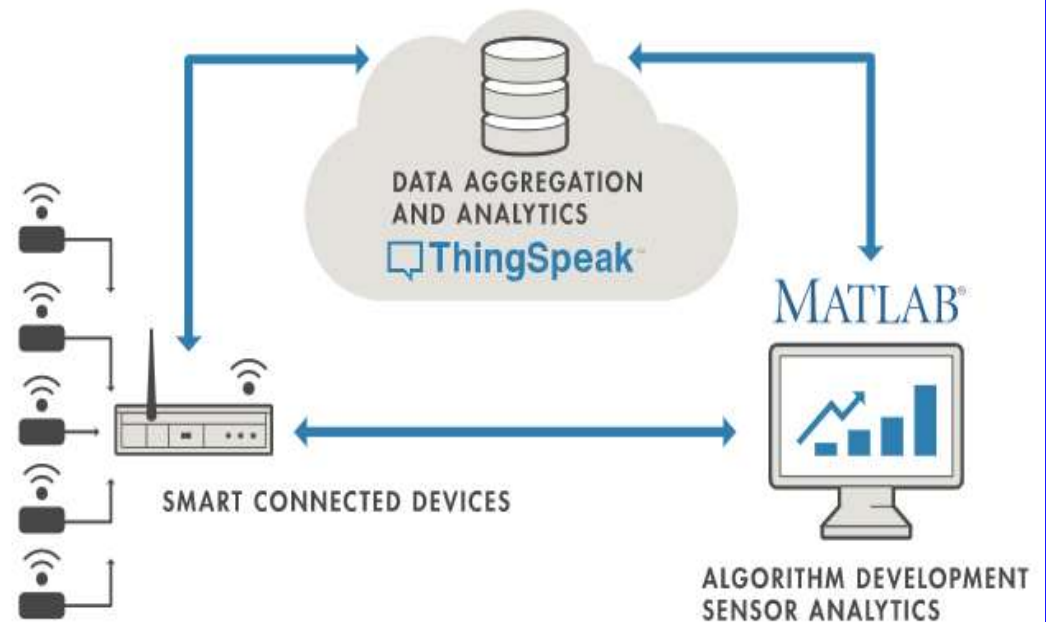


Email

No account? [Create one!](#)

By signing in you agree to our [privacy policy](#).

Next





SISTEMAS
EMBEBIDOS
2021



Computadora Industrial
Abierta Argentina
Desarrollo colectivo



Author: [mwa0000023124537](#)

Access: Private

Private View

Public View

Channel Settings

Sharing

API Keys

Data Import / Export

+ Add Visualizations

+ Add Widgets

Export recent data

MATLAB Analysis

MATLAB Visualization

Channel Stats

Created: [3 months ago](#)

Last entry: [a day ago](#)

ThingSpeak™

Channels ▾

Apps ▾

Devices ▾

Support ▾

Commercial Use

How to Buy



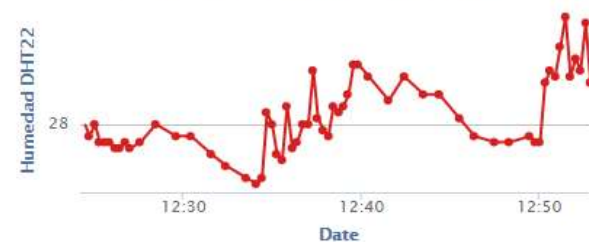
Estación de Medición - Temperatura



Temperatura DHT22



Estación de Medición - Humedad



Humedad





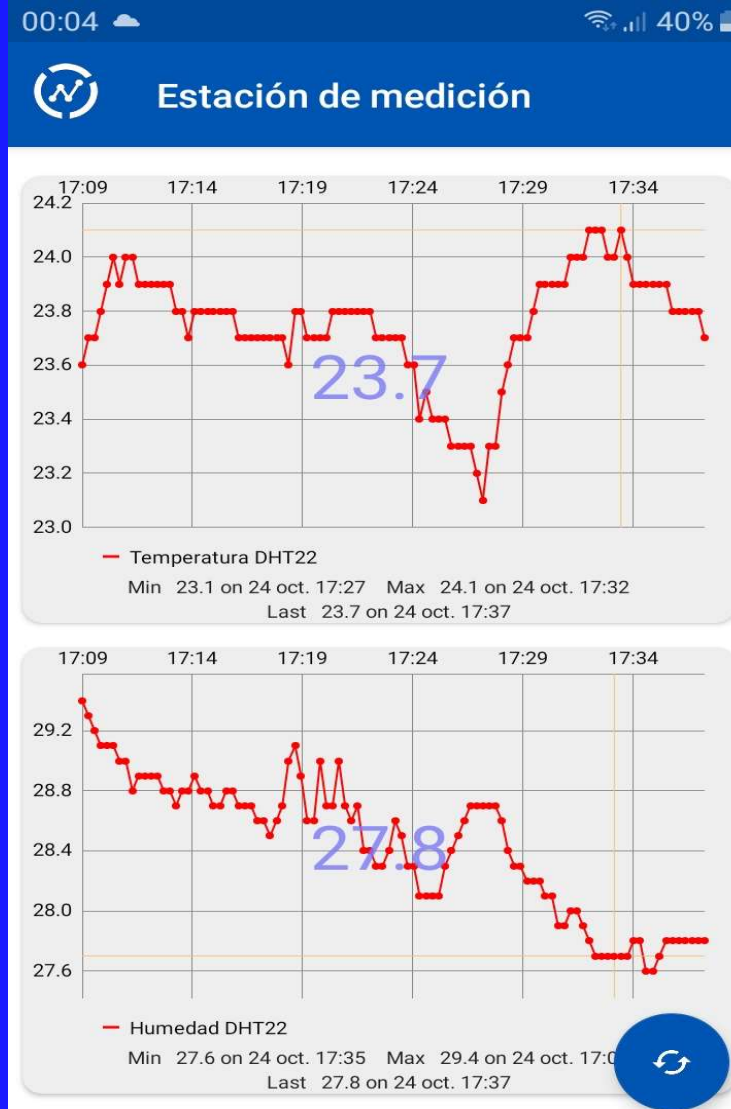
SISTEMAS
EMBEBIDOS
2021



Computadora Industrial
Abierta Argentina
Desarrollo colectivo



Aplicación desde
el celular





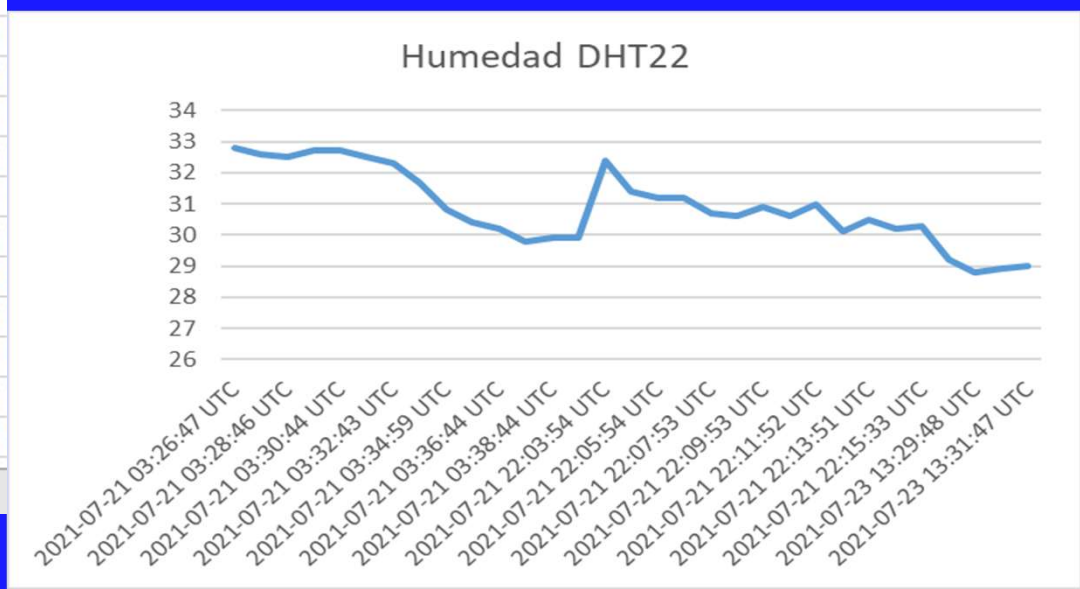
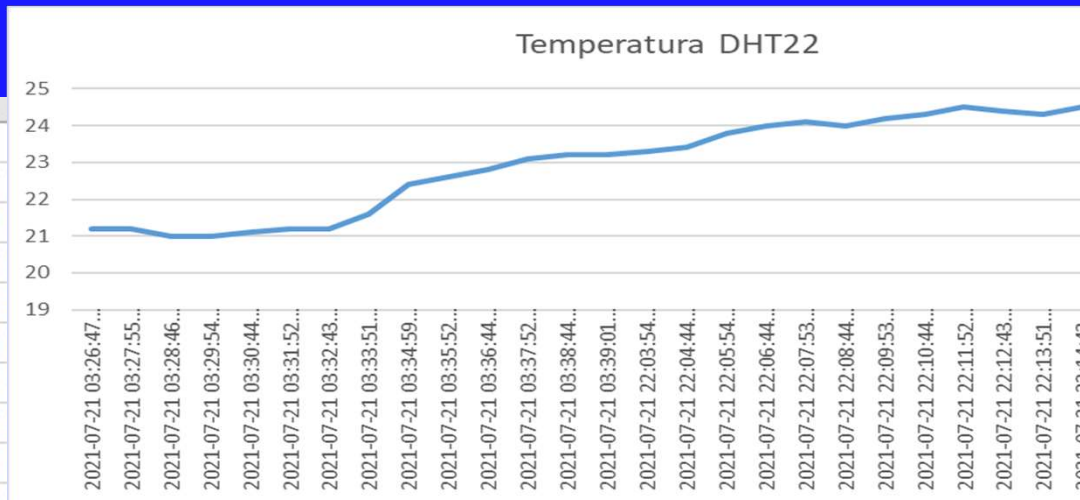
SISTEMAS
EMBEBIDOS
2021



Computadora Industrial
Abierta Argentina
Desarrollo colectivo

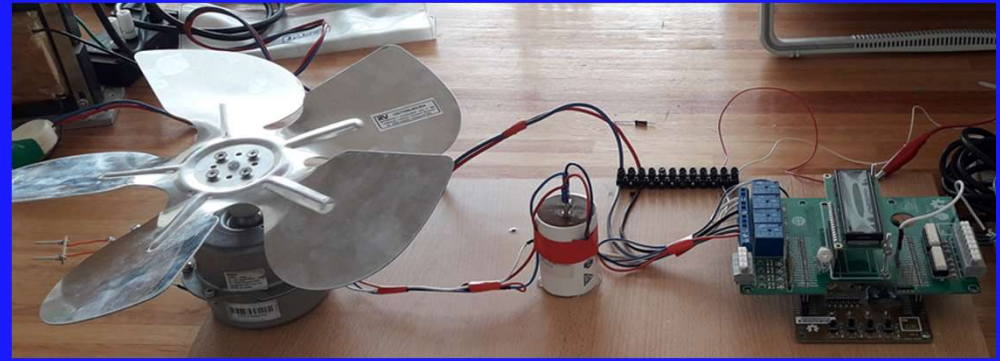
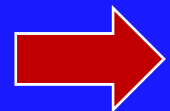
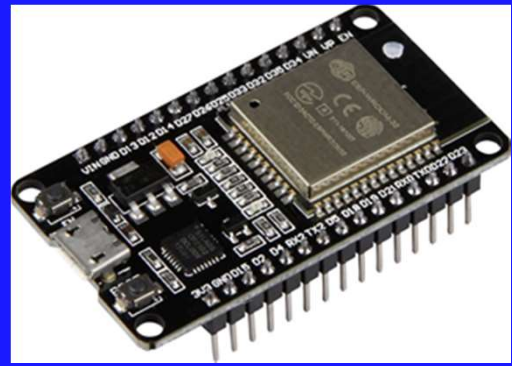
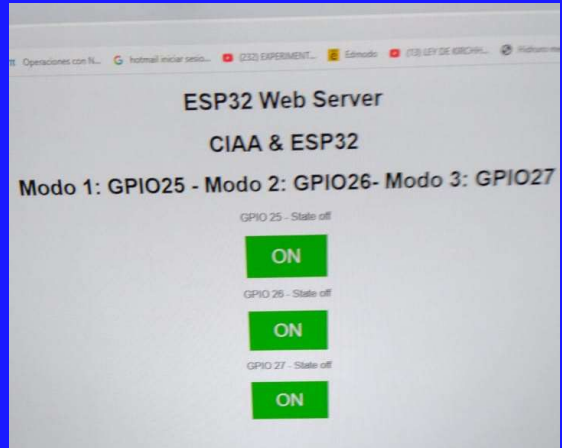
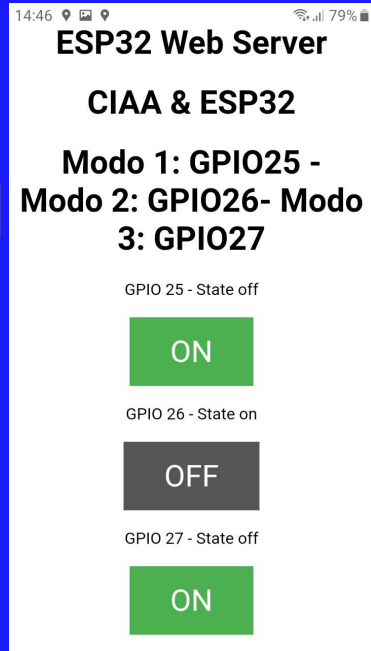
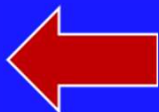


	A	B	C
1	created_at	field1	field2
2			
3	2021-07-21 03:26:47 UTC	21.2	32.8
4	2021-07-21 03:27:55 UTC	21.2	32.6
5	2021-07-21 03:28:46 UTC	21	32.5
6	2021-07-21 03:29:54 UTC	21	32.7
7	2021-07-21 03:30:44 UTC	21.1	32.7
8	2021-07-21 03:31:52 UTC	21.2	32.5
9	2021-07-21 03:32:43 UTC	21.2	32.3
10	2021-07-21 03:33:51 UTC	21.6	31.7
11	2021-07-21 03:34:59 UTC	22.4	30.8
12	2021-07-21 03:35:52 UTC	22.6	30.4
13	2021-07-21 03:36:44 UTC	22.8	30.2
14	2021-07-21 03:37:52 UTC	23.1	29.8
15	2021-07-21 03:38:44 UTC	23.2	29.9
16	2021-07-21 03:39:01 UTC	23.2	29.9
17	2021-07-21 22:03:54 UTC	23.3	32.4
18	2021-07-21 22:04:44 UTC	23.4	31.4
19	2021-07-21 22:05:54 UTC	23.8	31.2
20	2021-07-21 22:06:44 UTC	24	31.2
21	2021-07-21 22:07:53 UTC	24.1	30.7



Celular

Notebook

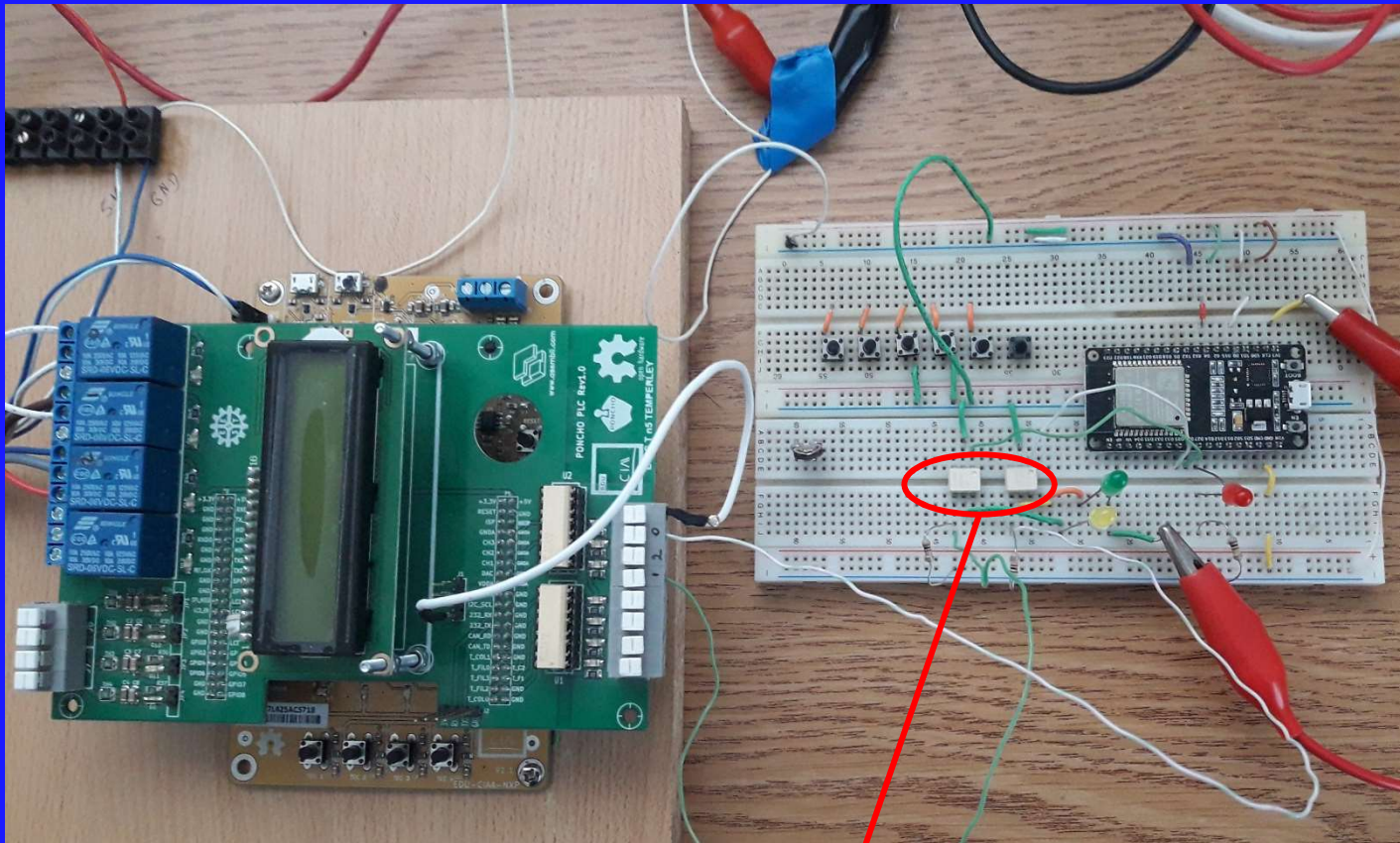




SISTEMAS
EMBEBIDOS
2021



Computadora Industrial
Abierta Argentina
Desarrollo colectivo

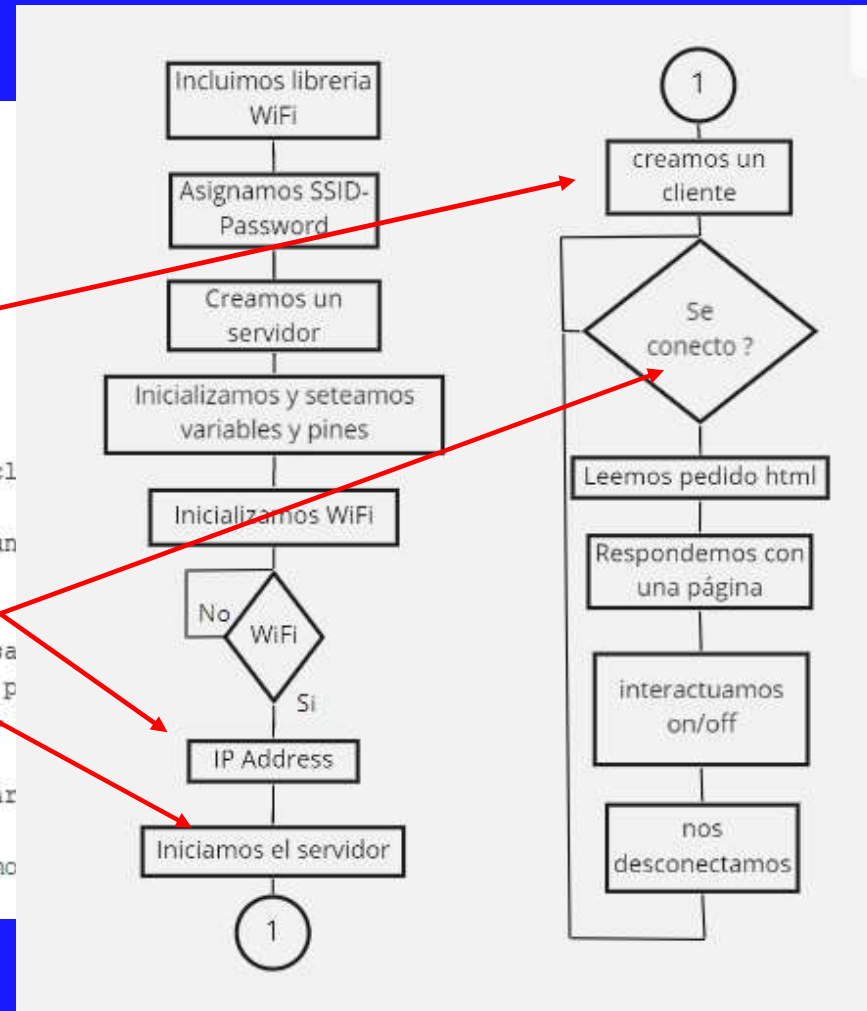


Optoacopladores

```

50 }
51 // Imprima la dirección IP y active el servidor |
52 Serial.println("");
53 Serial.println("WiFi connected.");
54 Serial.println("IP address: ");
55 Serial.println(WiFi.localIP());
56 server.begin();
57 ]
58
59 void loop(){
60   WiFiClient client = server.available(); // Escucha a los cl
61
62   if (client) { // si se conecta un
63     currentTime = millis();
64     previousTime = currentTime;
65     Serial.println("New Client."); // escribe un mensa
66     String currentLine = ""; //Hace una cadena p
67     while (client.connected() && currentTime - previousTime <=
68       currentTime = millis();
69       if (client.available()) { // Si hay datos par
70         char c = client.read(); // lea un byte
71         Serial.write(c); // imprima en el mo
72         header += c;

```





SISTEMAS
EMBEBIDOS
2021



Computadora Industrial
Abierta Argentina
Desarrollo colectivo



14:46 79%

ESP32 Web Server

CIAA & ESP32

**Modo 1: GPIO25 -
Modo 2: GPIO26- Modo
3: GPIO27**

GPIO 25 - State off

GPIO 26 - State on

GPIO 27 - State off

```

/ Mostrar la página web html
client.println("<!DOCTYPE html><html>");
client.println("<head><meta name=\"viewport\" content=\"width=device-width, initial-scale=1\">");
client.println("<link rel=\"icon\" href=\"data:,\">
```

```

client.println("<style>html { font-family: Helvetica; display: inline-block; margin: 0px auto; text-align: ce
client.println(".button { background-color: #4CAF50; border: none; color: white; padding: 16px 40px;");
client.println("text-decoration: none; font-size: 30px; margin: 2px; cursor: pointer;});");
client.println(".button2 {background-color: #555555;}</style></head>");

// Cabecera de la página web
client.println("<body><h1>ESP32 Web Server</h1>");
client.println("<body><h1>CIAA & ESP32 </h1>"); //agregado el 22/7
client.println("<body><h1>Modo 1: GPIO25 - Modo 2: GPIO26- Modo 3: GPIO27 </h1>");//agregado 23/7

//Muestra el estado actual y los botones on/off del gpio 25
client.println("<p>GPIO 25 - State " + output25State + "</p>");
// Si el estado de 25 es off muestra el boton on
if (output25State=="off") {
  client.println("<p><a href=\"/25/on\"><button class=\"button\">ON</button></a></p>");
} else {
  client.println("<p><a href=\"/25/off\"><button class=\"button button2\">OFF</button></a></p>");
}

//Muestra el estado actual y los botones on/off del gpio 26
client.println("<p>GPIO 26 - State " + output26State + "</p>");
// muestra el estado actual y los botones on/off del gpio 26
if (output26State=="off") {

```

<https://www.youtube.com/watch?v=RAaM173FSd4>



SISTEMAS
EMBEBIDOS
2021



**Computadora Industrial
Abierta Argentina**
Desarrollo colectivo



MUCHAS GRACIAS POR SU ATENCIÓN