

CAPÍTULO 20. Automation Controllers

v.1.1 ENERO 2025



Ricardo Moraleda Gareta

[Director departamento de software de GDO Software]





AUTOMATION
CONTROLLERS

Weidmüller 



Weidmüller

UC20-
WL2000
-AC

M3000/
M4000

GW30

AUTOMATION CONTROLLERS

v.1.1 ENERO 2025



u-OS



u-link

u-OS

Node-
RED

Code
sys

NVL/
UDP





Weidmüller 

PORTFOLIO

Weidmüller 



Portfolio Weidmüller

En las modernas aplicaciones de IoT, la tecnología de automatización debe ser cada vez más potente y adaptable. u-mation ofrece un programa de componentes de hardware perfectamente coordinado, diseñado para la puesta a punto de tus máquinas y sistemas para el futuro.

u-control 2000 es uno de los sistemas de control más compactos y avanzados del mercado y constituye la pieza central de nuestro programa de hardware. u-control 2000 es la base para poder usar cualquiera de nuestras soluciones de ingeniería: u-OS, u-create web, studio e IoT. El diseño modular del controlador se basa en el sistema de I/O remoto u-remote y permite ampliar la instalación según las necesidades para incluir otros componentes de la familia u-mation.

Grupos de controllers

Weidmüller divide los automation controllers en 3 grupos:

- U-control web

- UC20-WL2000-AC
- UC20-WL2000-AC-CAN



- U-control

- UC20-M3000
- UC20-M4000



- IoT Gateways

- IOT-GW30
- IOT-GW30-4G-EU



Todos tienen acceso remoto VPN por **u-link** (PC-cloud-device)



U-CONTROL



U-control

Módulos de I/O y remotas

CPUs → de izquierda a derecha de menor a mayor prestaciones.

https://www.linkedin.com/posts/iotueli_jotueli-weidmaessler-iiot-activity-7117156995014483968-o-Ke/

Los modelos M3000 y 4000 son PLCs (autómatas) aunque se pueden utilizar con carácter Edge, también.

Tanto el WL2000AC, como M3000 y M4000 son modulares y aceptan hasta 64 tarjetas IO en el mismo bastidor o remotas con **u-remote**.

Los GW30 son compactos aunque vienen con un mínimo de I/O.

Logos supported: u-OS, hawkBit, docker, CODESYS, Node-RED, OPC UA, MQTT, Modbus TCP/IP, API, Python, HTML, CSS, JS, portainer.io, EtherCAT, influxdb.

Models shown: GW30, GW30 / LTE 4G, WL2000AC, M3000 (NEW), M4000 (NEW).



| | |
|----------|----|
| Analog | 28 |
| Digital | 29 |
| Function | 17 |

74 tarjetas diferentes



UC20-WL2000-AC

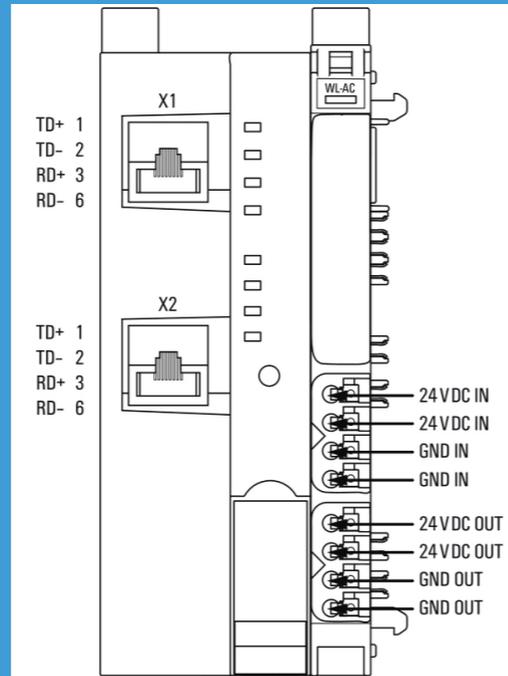


CPU

Prestaciones

Me centraré en esta CPU ya que es con la que he trabajado y puesto en marcha proyectos.

Esta CPU es la gama media de Weidmüller, para proyectos Edge / IoT.



- 2 conectores RJ45 (Eth) / 2 redes diferentes.
- 4/8 GB memoria Flash
- Dual Core ARM Cortex A9, 624 MHz, 512 MB RAM
- u-create web, **u-OS**

u-OS } y más.

Linux

Node-RED

CODESYS



UC20-WL2000-AC

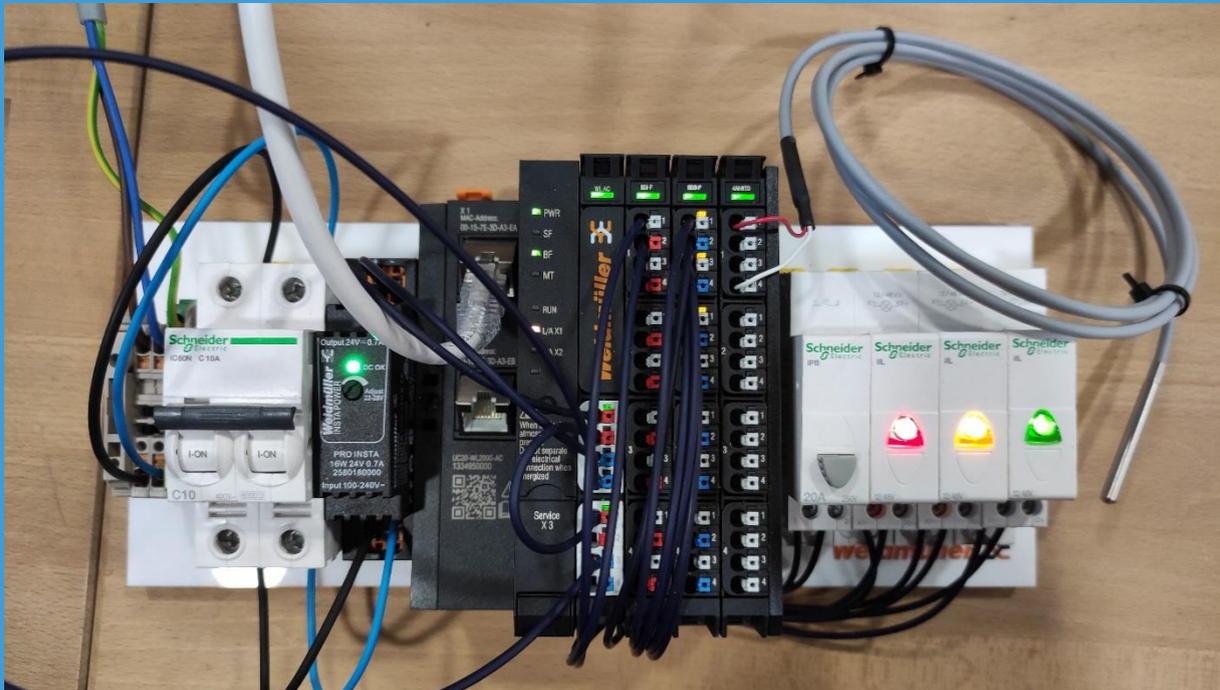


Testing

Partes de la maqueta

Testearemos la siguiente maqueta. [LinkedIn](#)

https://www.linkedin.com/posts/ricardo-moraleda-gareta-9421099_weidmaesller-iot-codesys-activity-7138438193942626304-QDCv/



La maqueta consta de:

- Protecciones
- Fuente de alimentación 24 VDC
- CPU → UC20-WL2000-AC
- Módulo 8-DI-P (digital inputs)
- Módulo 8-DO-P (digital outputs)
- Módulo 4AI-RTD (analog inputs)



Para comprobar (módulos de entrada y salida)

- Pulsador (Entrada digital)
- Red led (Salida digital)
- Yellow led (Salida digital)
- Green led (Salida digital)
- Sonda de temperatura PT1000 2w (Entrada analógica)





UC20-WL2000-AC



Testing

Versiones de software

Testearemos la siguiente maqueta.

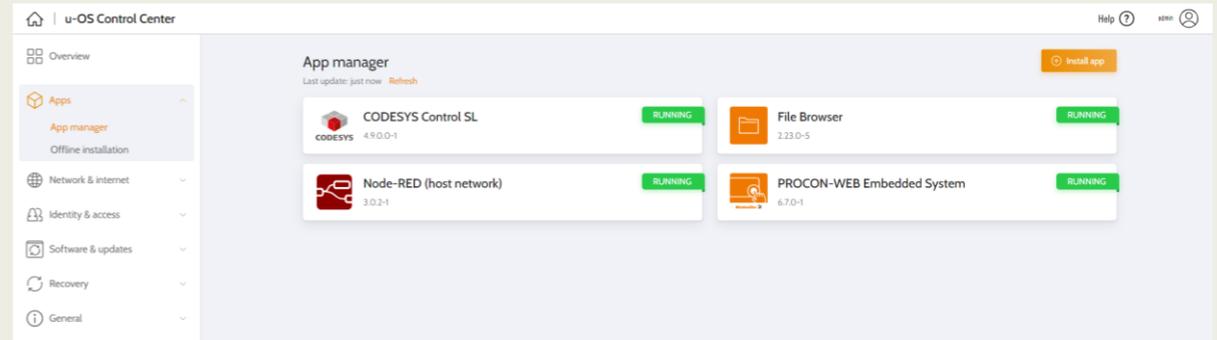
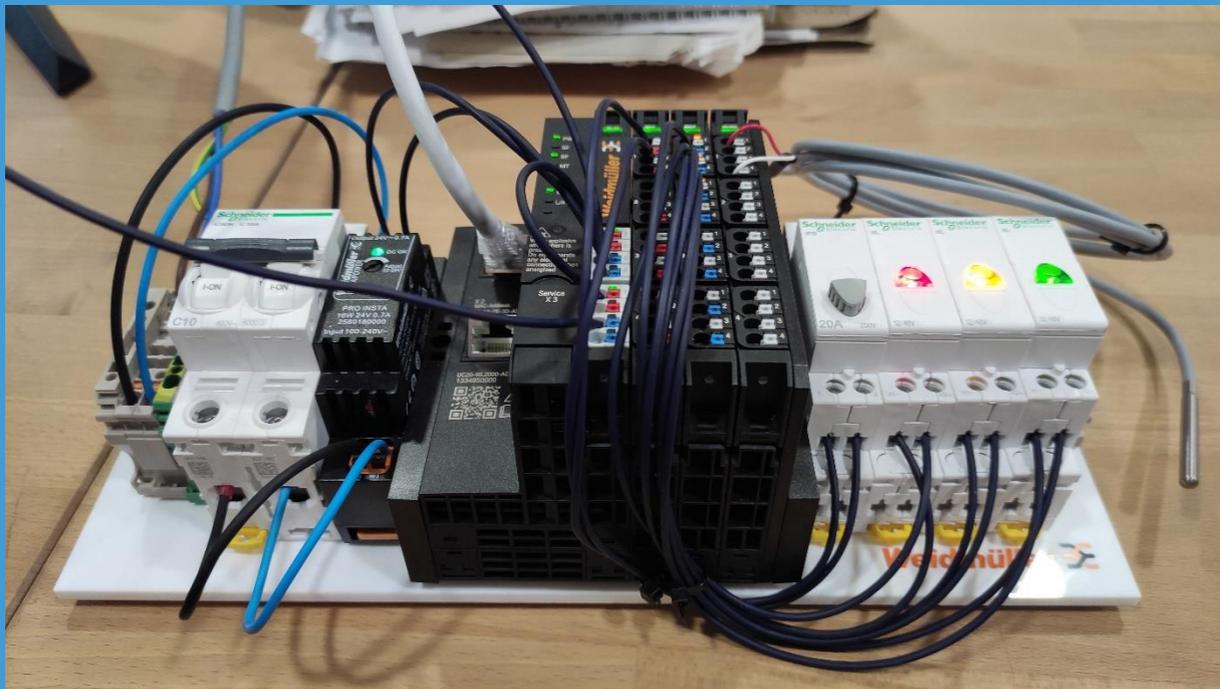
A nivel de software:

- Se ha migrado de u-create web a u-OS 2.1.0 **u-OS**
- Se han instalado las siguientes apps:



- CODESYS Control SL 4.9.0.0-1 para ser programado mediante el standard IEC 61131-3 (requiere licencia)

- Node-RED (host network) 3.0.2-1
- otros





UC20-WL2000-AC

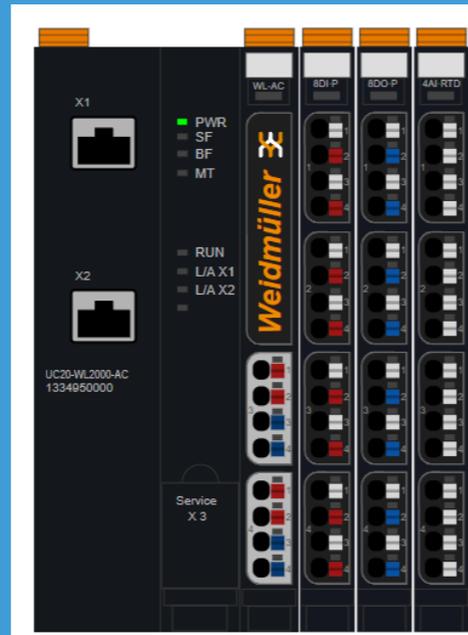
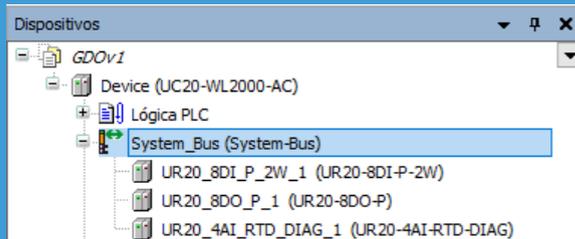


PC



Instalamos la versión 3.5 SP19 y agregaremos la librería **IoDrvUR20Control** de Weidmüller (CODESYS Control SL for Weidmueller u-OS 4.7.1).

Creamos el proyecto y configuramos el dispositivo y su hardware (System-Bus)



Input / Output modules

Módulos de E/S:

- UR20-8DI-P-2W. Usaremos el channel 0 para el pulsador.
- UR20-8DO-P. Usaremos el channel 0, 1 y 2 para los leds rojo, amarillo y verde.
- UR20-4AI-RTD-DIAG. Usaremos el channel 0 para la sonda de temperatura PT1000 2w.





UC20-WL2000-AC



POU



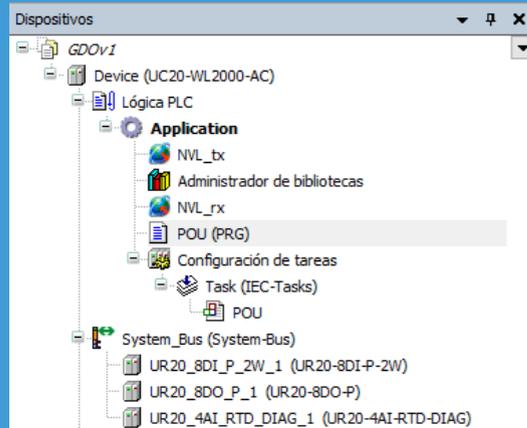
TASK



Se creará la lógica del dispositivo a través del POU (Program Organization Unit) y se ejecutará a través de una Task cíclicamente cada 4 ms.



TASK → Ejecuta el POU cada 4 ms



POU → Structured Text ST

```

1 PROGRAM POU
2 VAR
3     Temp: INT;
4     Puls: BOOL;
5     Rojo: BOOL;
6     Amarillo: BOOL;
7     Verde: BOOL;
8 END_VAR
9
10

```

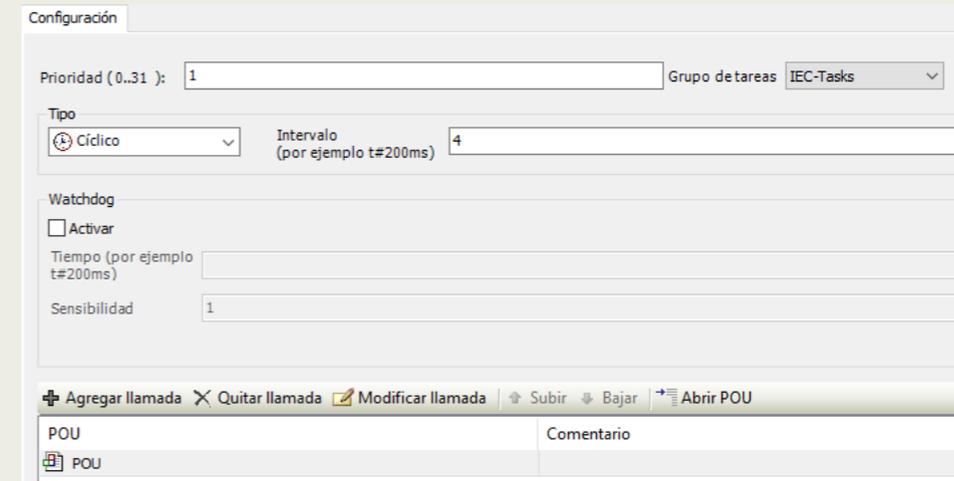
Declaración variables

```

1 NVL_tx.Temperatura := Temp;
2 NVL_tx.Pulsador := Puls;
3 Rojo := NVL_rx.Red;
4 Amarillo := NVL_rx.Yellow;
5 Verde := NVL_rx.Green;

```

Ejecución programa





UC20-WL2000-AC



Mapeo señales con variables



NVL (UDP) ↔



UR20-8DI-P-2W

| Variable | Asignación | Canal | Dirección | Tipo | Unidad | Descripción |
|----------------------|------------|---------|-----------|------|--------|-------------------|
| Application.POU.Puls | | Input | %IB0 | BYTE | | |
| | | Value 0 | %IX0.0 | BOOL | | Pulsador (ON/OFF) |
| | | Value 1 | %IX0.1 | BOOL | | |
| | | Value 2 | %IX0.2 | BOOL | | |
| | | Value 3 | %IX0.3 | BOOL | | |
| | | Value 4 | %IX0.4 | BOOL | | |
| | | Value 5 | %IX0.5 | BOOL | | |
| | | Value 6 | %IX0.6 | BOOL | | |
| Value 7 | %IX0.7 | BOOL | | | | |

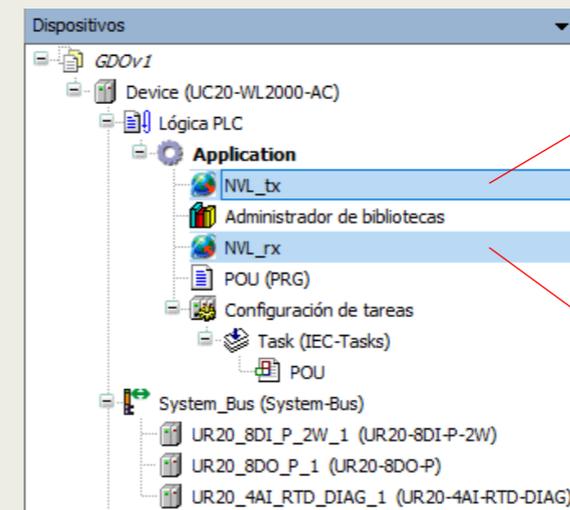
UR20-8DO-P

| Variable | Asignación | Canal | Dirección | Tipo | Unidad | Descripción |
|----------------------|------------|---------|-----------|------|--------|-------------|
| Application.POU.Rojo | | Output | %QB0 | BYTE | | |
| | | Value 0 | %QX0.0 | BOOL | | Red Led |
| | | Value 1 | %QX0.1 | BOOL | | Yellow Led |
| | | Value 2 | %QX0.2 | BOOL | | Green Led |
| | | Value 3 | %QX0.3 | BOOL | | |
| | | Value 4 | %QX0.4 | BOOL | | |
| | | Value 5 | %QX0.5 | BOOL | | |
| | | Value 6 | %QX0.6 | BOOL | | |
| Value 7 | %QX0.7 | BOOL | | | | |

UR20-4AI-RTD-DIAG

| Variable | Asignación | Canal | Dirección | Tipo | Unidad | Descripción |
|----------------------|------------|---------|-----------|------|--------|-----------------------------|
| Application.POU.Temp | | Value 0 | %IW1 | INT | | Temperatura sonda (°C X 10) |
| | | Value 1 | %IW2 | INT | | |
| | | Value 2 | %IW3 | INT | | |
| | | Value 3 | %IW4 | INT | | |

Además, para comunicar con otras aplicaciones (en este caso Node-RED para visualización GUI) se utilizan unas variables llamadas NVL (**Network Variable List**), dependiendo si es de entrada o salida, serán sender o receiver y se comunicarán por protocolo **UDP**.



```

1 {attribute 'qualified_only'}
2 VAR_GLOBAL
3     Temperatura: INT;
4     Pulsador: BOOL;
5 END_VAR

```

```

1 //Esta lista de variables globales se recibirá a través de la
2 //Remitente: Importado del archivo 'C:\codesys\NVL.gvl'
3 //Protocolo: UDP
4
5 {attribute 'qualified_only'}
6 VAR_GLOBAL
7     Red: BOOL;
8     Yellow: BOOL;
9     Green: BOOL;
10 END_VAR

```



UC20-WL2000-AC



Flujo Node-RED

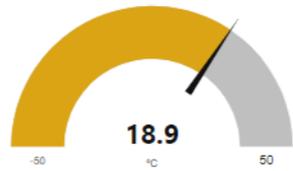


NVL (UDP) ↔



Maqueta UC20-WL2000-AC

Temperatura



Analog Input

Pulsador

Digital Input

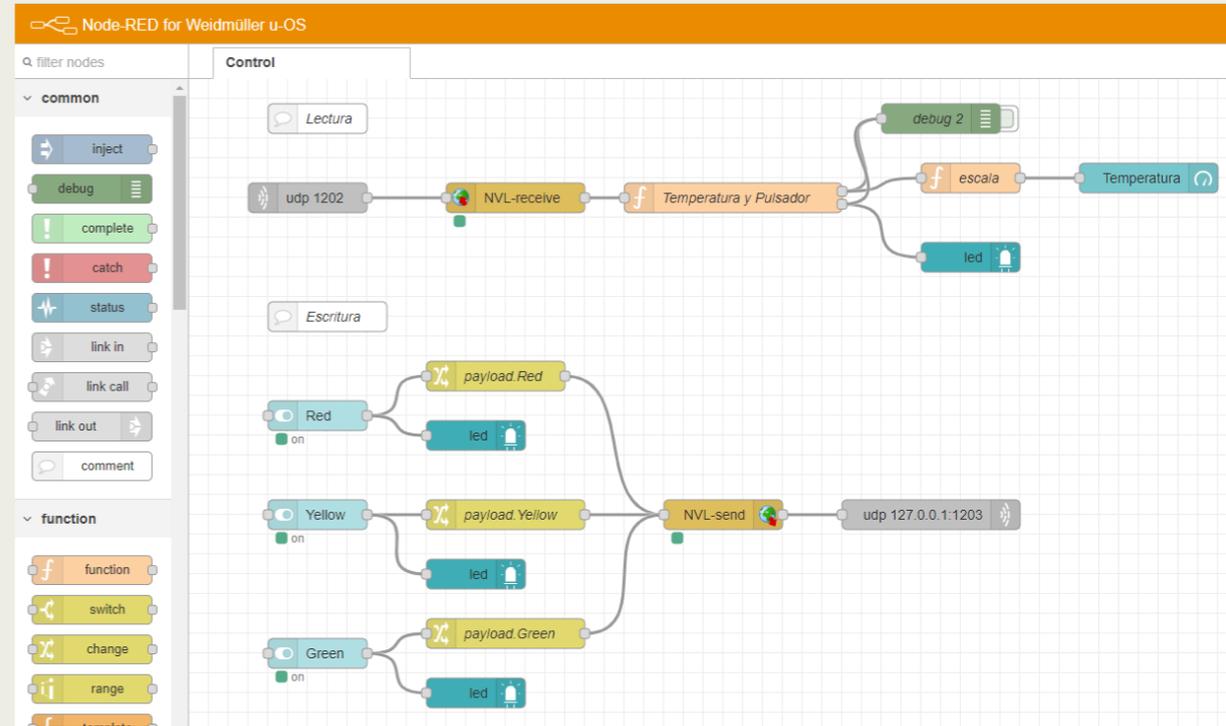
Red

Commands

Yellow

Green

Digital Outputs





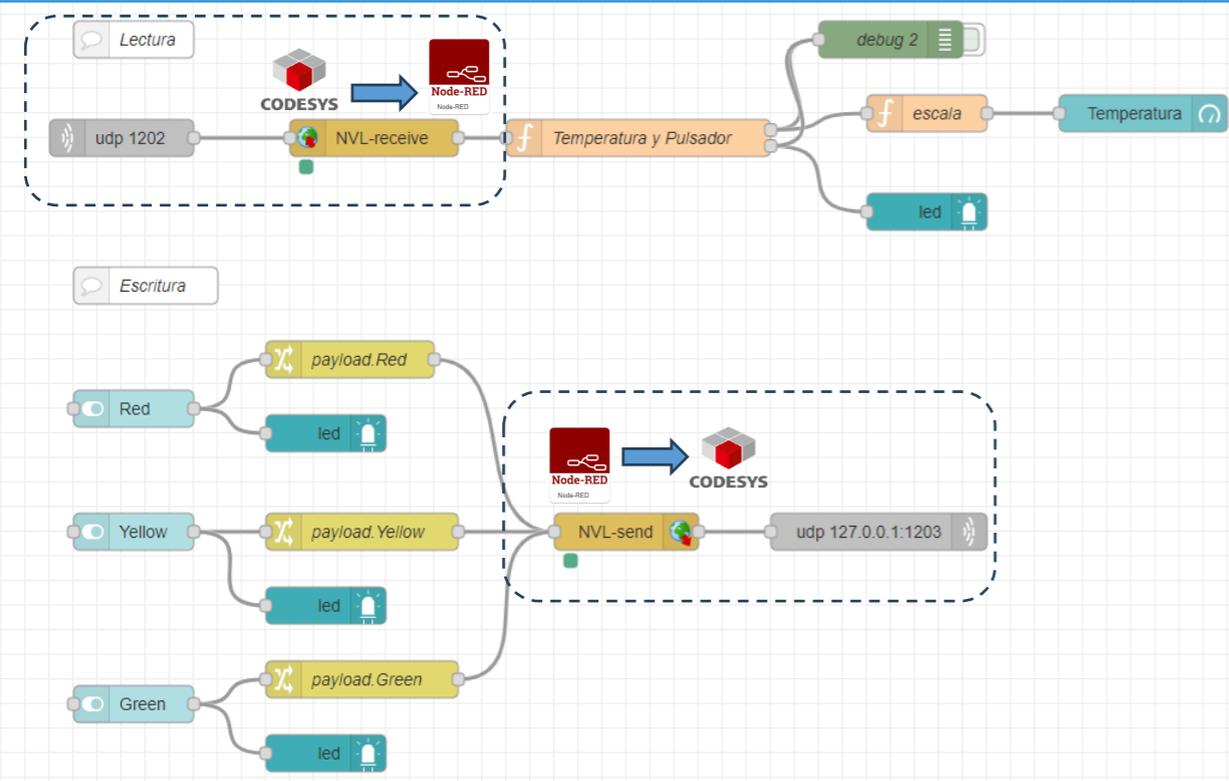
UC20-WL2000-AC



Flujo Node-RED



NVL (UDP) ↔



Por el puerto 1202 UDP, Node-RED recibe los datos de Codesys y los representa gráficamente. En este caso, el pulsador y la temperatura de la sonda.

Por el puerto 1203 UDP, Node-RED envía las órdenes ON/OFF de cada led (rojo, amarillo y verde) a Codesys para que actúe físicamente.

Properties

- Name: Name
- List Id: 1
- Pack variables: true
- Global datatypes: Add new nvl-datatypes...

Limitations: When using the option "Transmit on change" use the "pack" option and limit the nvl size to 256 bytes.

Network variable list

```

1 {attribute 'qualified_only'}
2 VAR_GLOBAL
3   Temperatura: INT;
4   Pulsador: BOOL;
5 END_VAR

```

Properties

- Name: Name
- Udp port: 1203
- List Id: 1
- Pack variables: true
- Global datatypes: Add new nvl-datatypes...

Network variable list

```

1 VAR_GLOBAL
2   Red: BOOL;
3   Yellow: BOOL;
4   Green: BOOL;
5 END_VAR

```

node-red-contrib-nvl (1.2.1)

Application note:
AN0106v01-UC-Codesys
NVL in Node-RED [3]





Enlaces recomendados



Enlaces recomendados

Arquitecturas / recursos

<https://iotueli.ch/> → (UELI IFF)

Catálogo controles de Weidmüller

https://www.weidmuller.es/es/productos/automatizacion_y_software/controles/index.jsp

[3] <https://support.weidmueller.com/support-center/learning/knowledge-base>

Vídeo maqueta

https://www.linkedin.com/posts/ricardo-moraleda-gareta-9421099_weidmaesller-iot-codesys-activity-7138438193942626304-QDCv/

U-OS

https://www.weidmuller.es/es/productos/automatizacion_y_software/controls/u_os/index.jsp

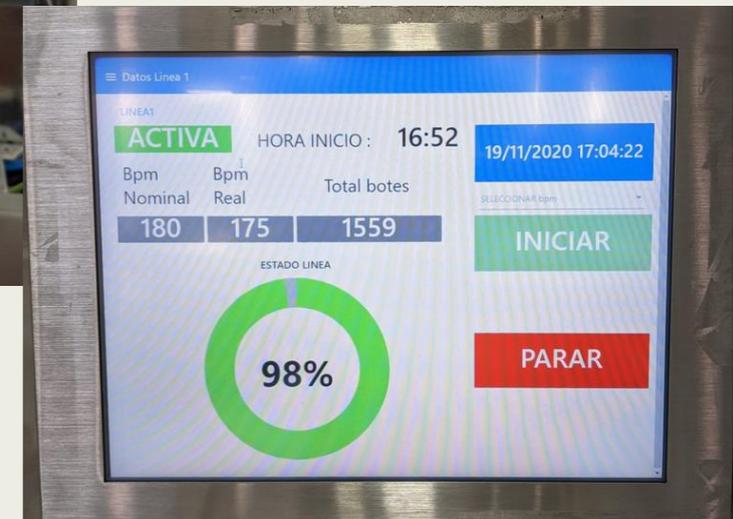
Codesys

<https://www.codesys.com/>

Fotos reales de proyectos en producción:



- Sector del agua (I+D). Adquisición de diferentes tipos de sensores y sondas (multiprotocolo)
- Sector alimentación. Tiempo de ciclo envasadoras. Contaje fotocélulas.
- Sector ferroviario. Adquisición datos, tratamiento y envío en fichero CSV a S3 de AWS



AUTOMATION CONTROLLERS

v.1.1 ENERO 2025



<https://www.linkedin.com/in/ricardo-moraleda-gareta-9421099>

<https://www.linkedin.com/company/gdo-electric1996/>

RICARDO MORALEDA GARETA