

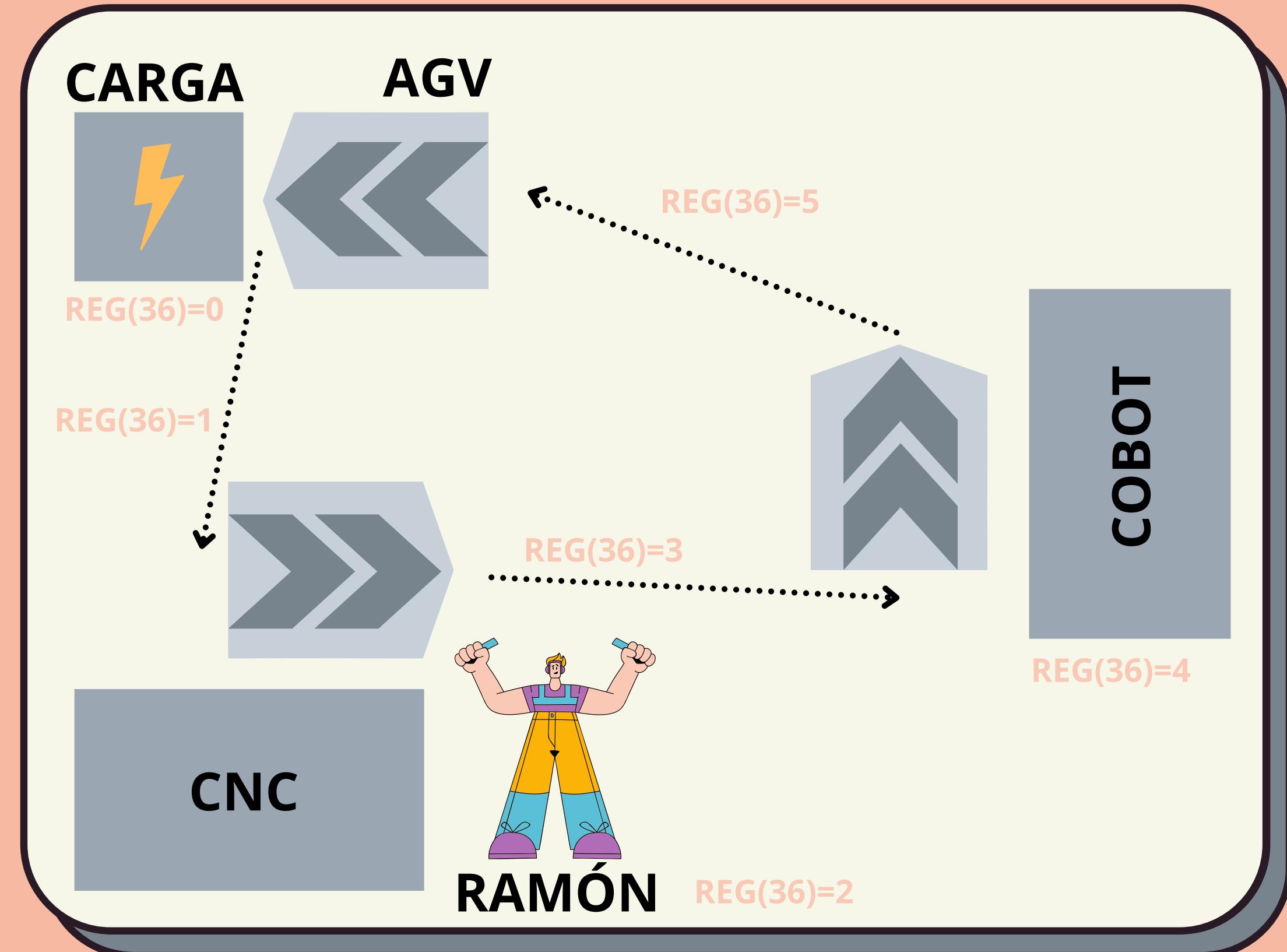
TRES3ES

Manual de Usuario

01

ESQUEMA FÁBRICA

02

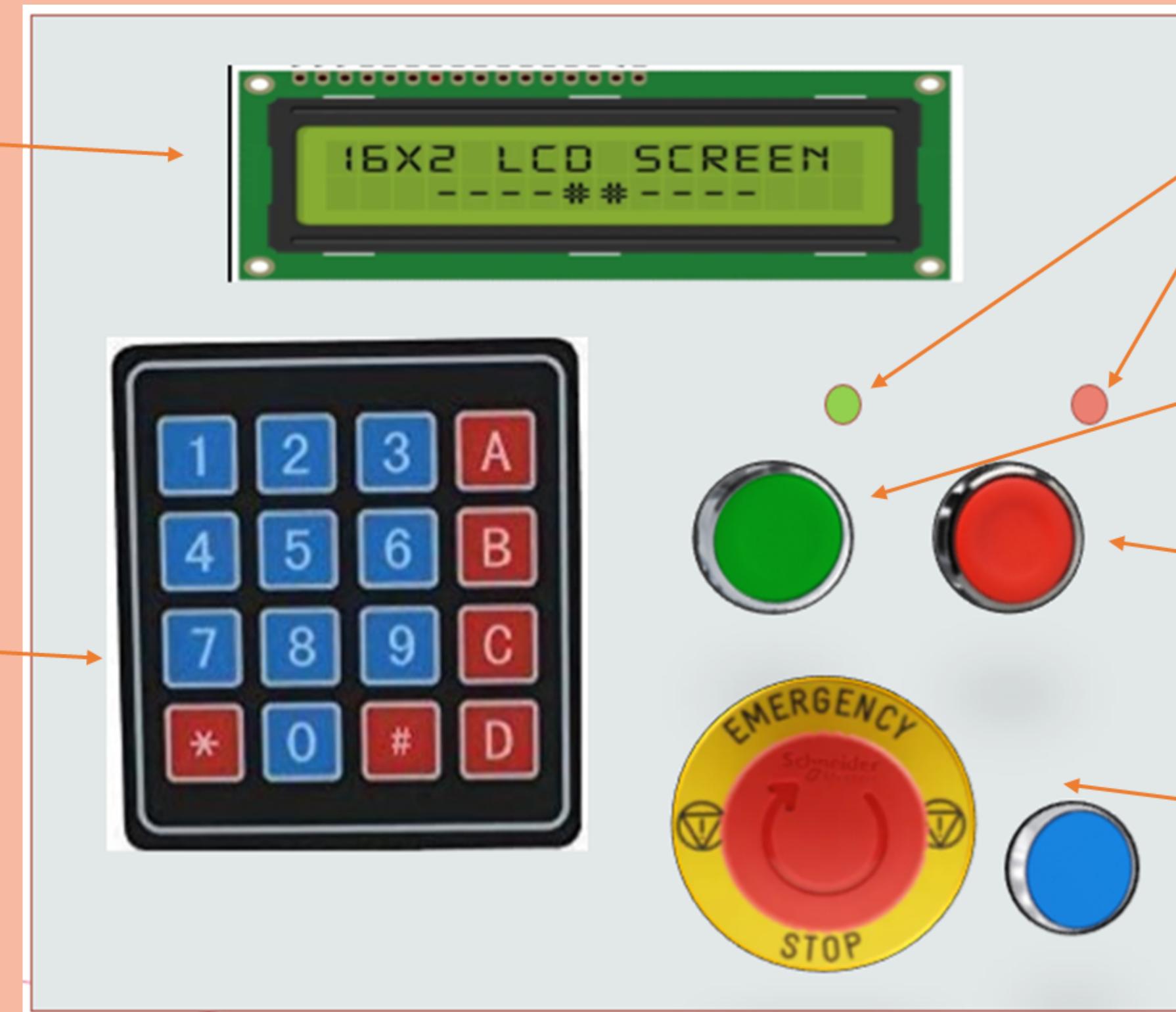


INTERFAZ

03

Display de estado de AGV
y entrada de datos

Teclado para introducir
Errores definidos
previamente



Leds indicadores de si
botón activo

Pulsador llamada al AGV
al CNC

Pulsador envío del AGV
al COBOT

Parada emergencia y
Botón de rearne

FUNCIONALIDAD

04

AGV EN ESPERA



AGV se encuentra en base de carga

**LED indicador pulsador de llamada
ENCENDIDO**

**Únicamente podremos llamar al AGV
cuando el operario quiera insertar
piezas**

INTERNAMENTE: $\text{reg}(36) = 0$



AGV LLEGANDO

AGV se encuentra de camino al CNC

**LED indicador pulsador de llamada
APAGADO**

ESPERAR

INTERNAMENTE: reg(36) = 1

AGV EN CNC



AGV se encuentra con Ramón

**LED indicador pulsador de envío al
cobot ENCENDIDA**

OPCIONES:

- PULSAR 1: AÑADIR ERRORES**
- PULSAR 2: BORRAR EL REGISTRO ANTERIOR**
- PULSAR ROJO: ENVIA AGV AL COBOT**
- PULSAR VERDE?: ENVÍO DE VUELTA A CARGA**

INTERNAEMENTE: $\text{reg}(36) = 2$

1. INTRO ERROR



**LED indicador pulsador de envío al
cobot APAGADO**

TIPO DE ERROR:

- A1, A2, B1, B2, C1, C2 -> POR DEFINIR
- OTRO -> ERROR NO EXISTENTE

CANTIDAD INTRODUCIDA

- 1-9 -> INTRODUCIENDO...

**Vuelve a AGV EN CNC y habilita de
nuevo botón de envío a cobot**



2. BORRAR ANTE

**LED indicador pulsador de envío al
cobot APAGADO**

BORRANDO...

**vuelve a AGV EN CNC y habilita de
nuevo botón de envío a cobot**

Internamente

Los errores están asociados a cada palo, por ello al introducirse, se guardaran el los registros 37 - 41, coincidentes con los palos 1 - 5 respectivamente, la cantidad introducida en cada uno.

Posteriormente el COBOT leerá los registros y así tendrá la información.

36	0	Equipo 3	$* 41071 + 41072$		
37	0	Palo1	$* 41073 + 41074$		
38	0	Palo2	$* 41075 + 41076$		
39	0	Palo3	$* 41077 + 41078$		
40	0	Palo4	$* 41079 + 41080$		
41	0	Palo5	$* 41081 + 41082$		



AGV LLEGANDO

AGV se encuentra de camino al COBOT

**LED indicador pulsador de llamada
ENCENDIDO**

**Podremos llamarlo en caso de que nos
hayamos olvidado una pieza**

INTERNAMENTE: $\text{reg}(36) = 3$



AGV CON COBOT

AGV se encuentra con el COBOT

**LED indicador pulsador de llamada
APAGADO**

**EL AGV AVISARÁ AL COBOT Y PODRÁN
A JUGAR JUNTOS**

INTERNAMENTE: $\text{reg}(36) = 4$

AGV VUELTA BASE DE CARGA



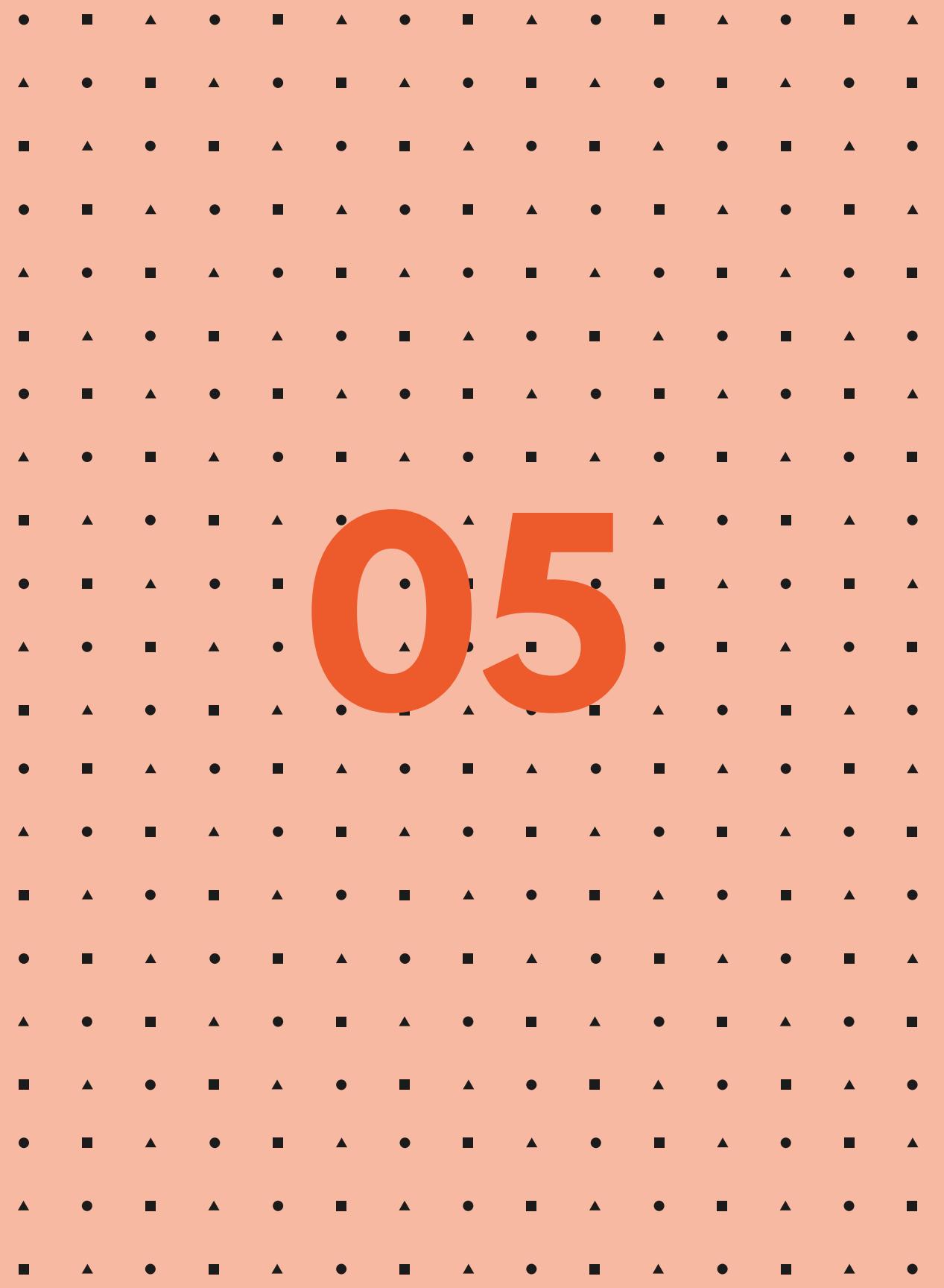
Cuando el COBOT finalice enviará al AGV a casa, poniendo a 0 los registros asociados a los palos.

AGV se encuentra de camino a la base de carga

LED indicador pulsador de llamada ENCENDIDO

Podremos llamar al AGV cuando el operario quiera insertar piezas

INTERNAMENTE: reg(36) = 5



05

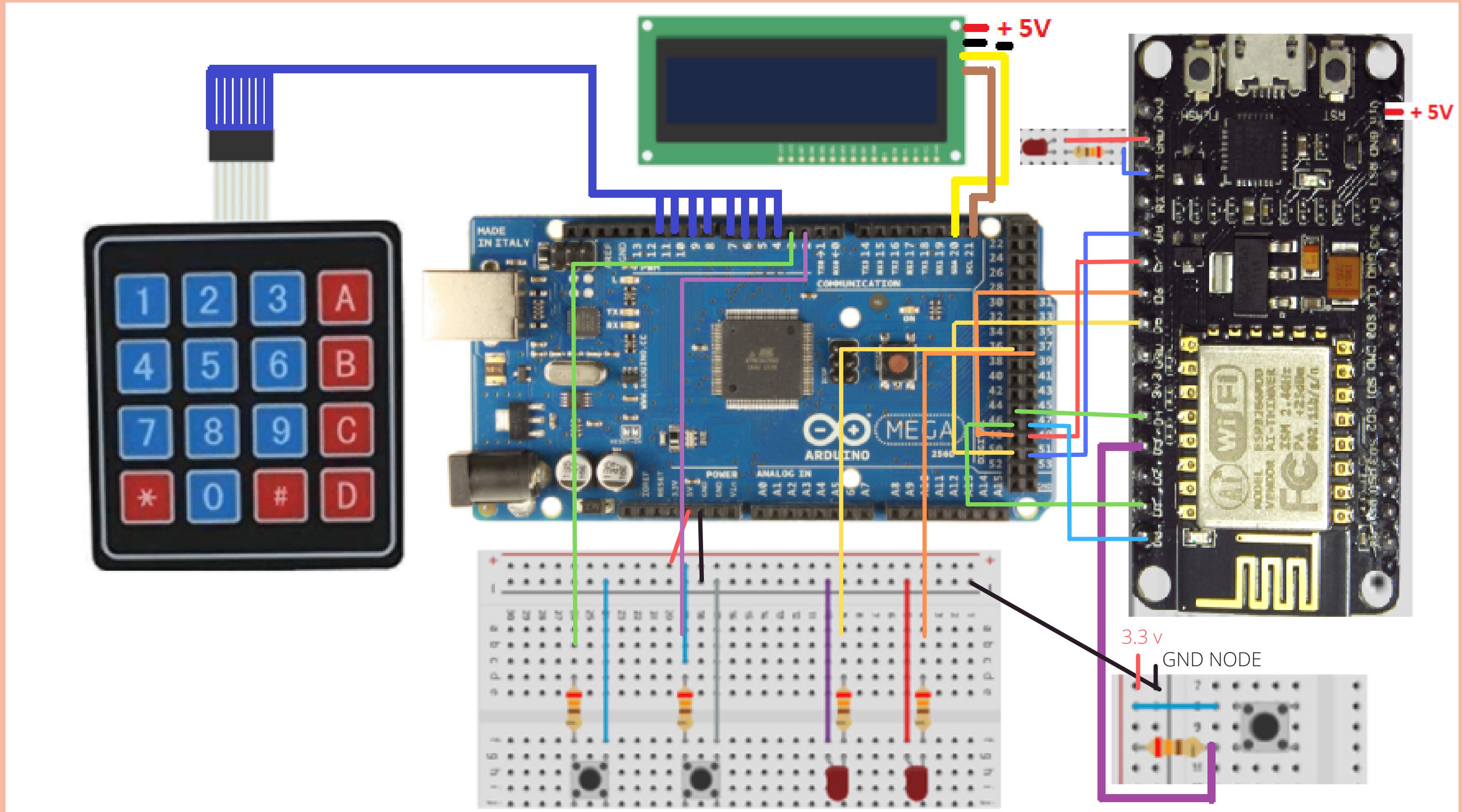
ESQUEMA ELÉCTRICO

**EL NODE SE COMUNICA POR
PROTOCOLO SPI CON ARDUINO**

**NODE HACE LECTURA PERÍODICA DEL ESTADO
(reg(36)) Y SE LA ENVÍA AL ARDUINO**

**ARDUINO TIENE LAS DIFERENTES FUNCIONES
Y CONTROLA LA INTERFAZ**

**HABILITA SEÑALES DIGITALES CUANDO SE
QUIEREN ACCIONAR MISIONES EN EL NODE**



BANCO DE PRUEBAS

06

- ✓ • KEYPAD
- ✓ • LCD
- ✓ • LCD-KEYPAD
- ✓ • BOTONES
- ✓ • LCD-KEYPAD-BOTONES
- ✓ • LIBRERÍA GBUTTON
- ✓ • BOTONES POR INTERRUPCIONES
- ✗ • PRUEBA CONEXIÓN 12C (CONEXIÓN DESCARTADA)
- ✓ • CONEXIÓN SPI
- ✓ • CONEXIÓN SPI NODE
- ✓ • CONEXIÓN SPI BIDIRECCIONAL
- ✓ • LIBRERÍA MODBUS ESP8266
- ✓ • NODEMCU COMO ESCLAVO
- ✓ • NODEMCU COMO MAESTRO
- ✓ • ESCRITURA/LECTURA
- ✓ • REGISTROS/COIL
- ✓ • CREAR MISIONES EN AGV
- ✓ • LEDS
- ✓ • TRIGGERS PARA MISIONES EN AGV
- ✓ • LANZAMIENTO DE MISIONES
- ✓ • EMPLEO DE COILS PREDEFINIDOS (PARADA, BORRAR COLA, ...)
- ✓ • INTERFAZ REGISTRO DE DATOS
- ✓ • FUNCIONES POR INTERRUPCIONES NODEMCU
- ✓ • LECTURA DEL ESTADO EN NODE Y COMUNICACIÓN A ARDUINO
- ✓ • INTERRUPCIÓN DIGITAL ARDUINO A NODE
- ✓ • INTERFAZ ARDUINO
- ✓ • INTERFAZ NODE
- ✓ • PRUEBAS DE FUNCIONAMIENTO
- ✓ • CONEXIÓN ERRORES CON BASE DE DATOS
- ✓ • COMUNICACIÓN IOT COBOT CON AGV/NODEMCU
- ✓ • PRUEBA DE FUNCIONAMIENTO COMPLETA

BASE DE DATOS

PASOS PREVIOS:

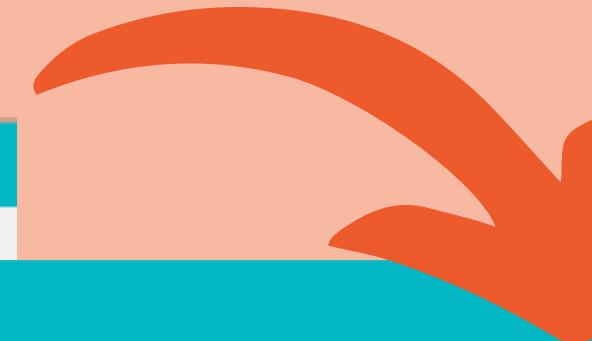
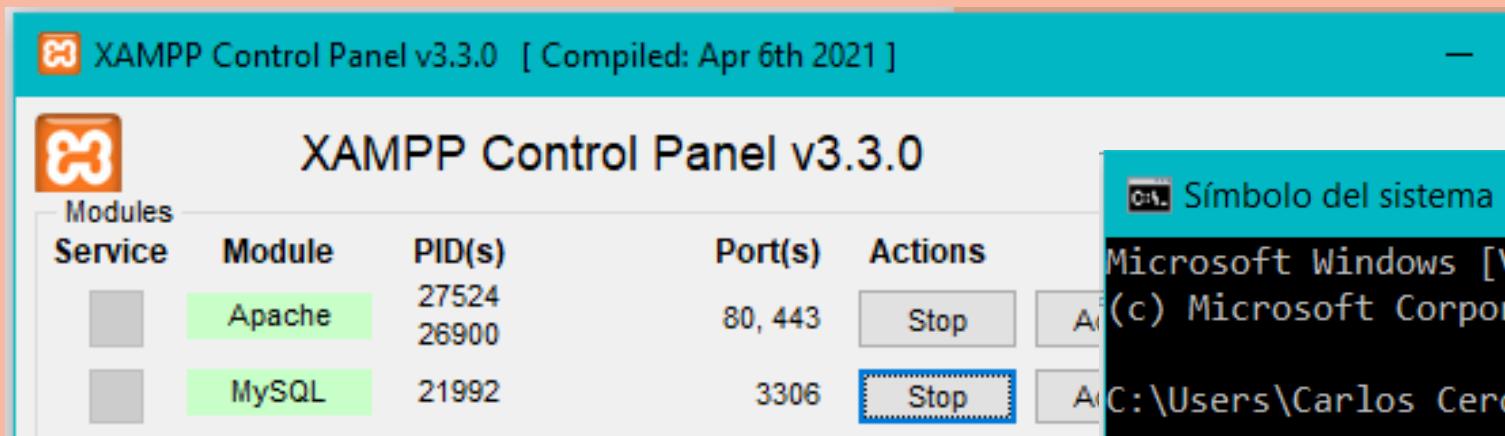
Para cargar la base de datos habrá que cargar un programa de conexión y activar XAMPP en el ordenador.

Cargando los códigos en el CMD del portátil, situándose en la carpeta de la librería, se extraerán los datos del Serial a nuestra base de datos.

Códigos:

```
CD C:\Users\serial2sql-master  
pip install mysql-connector-python  
pip install pySerial  
python serial2sql.py
```

07

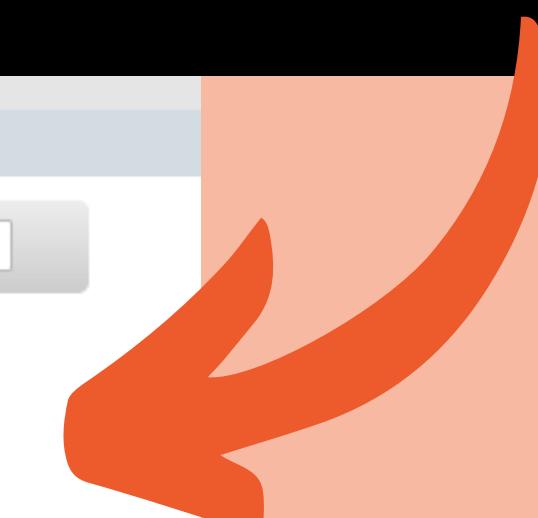


The phpMyAdmin interface for the 'data' database. It shows a table named 'datafromserial'. The table has columns: Fallo, Numero, Lote, borrar, Fecha, Hora, and id. The data in the table is:

	Fallo	Numero	Lote	borrar	Fecha	Hora	id
1	No	-			2022-05-09	11:31:50	65
2	No	-			2022-05-09	11:32:22	66
1	No	-			2022-05-09	11:32:51	67

Below the table, there is a terminal window showing the command line output of a Python script:

```
C:\Users\Carlos Cerdán\Documents\ETSII\Master\Ingenia\serial2sql-master>python serial2sql.py
CREATE TABLE datafromserial (id INT(9) NOT NULL AUTO_INCREMENT,Fallo VARCHAR(2),Numero INT(1),Lote VARCHAR(10),borrar VARCHAR(15),PRIMARY KEY (id))
1050 (42S01): Table 'datafromserial' already exists
Opening port and starting to save data
SELECT * FROM `datafromserial`Press Ctrl+C to stop
```



08

Advertencias

Cargar ambos programas en las placas correspondientes (aconsejable primero NODE y luego Arduino), CON LA SETA PULSADA.

Al finalizar se deberá soltar la seta y presionar el botón de rearme.

La botonera funciona únicamente con el cable del Arduino conectado al ordenador de la base de datos, que le haría de fuente de alimentación además de conexión.

Si se quiere obviar el paso de la base de datos la botonera también funcionaría con una fuente de alimentación conectada.

Muchas gracias

