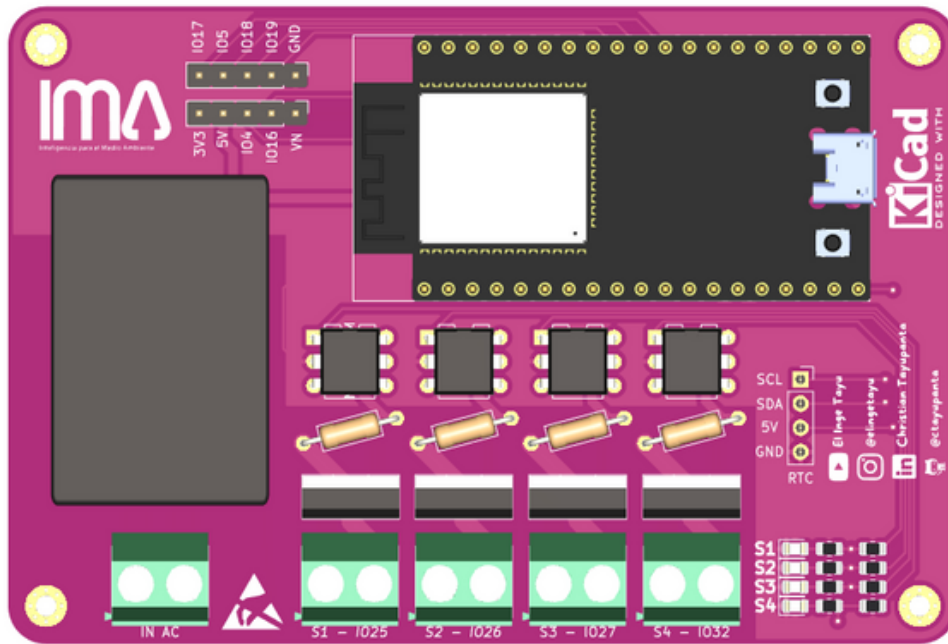


Smart Controller Home

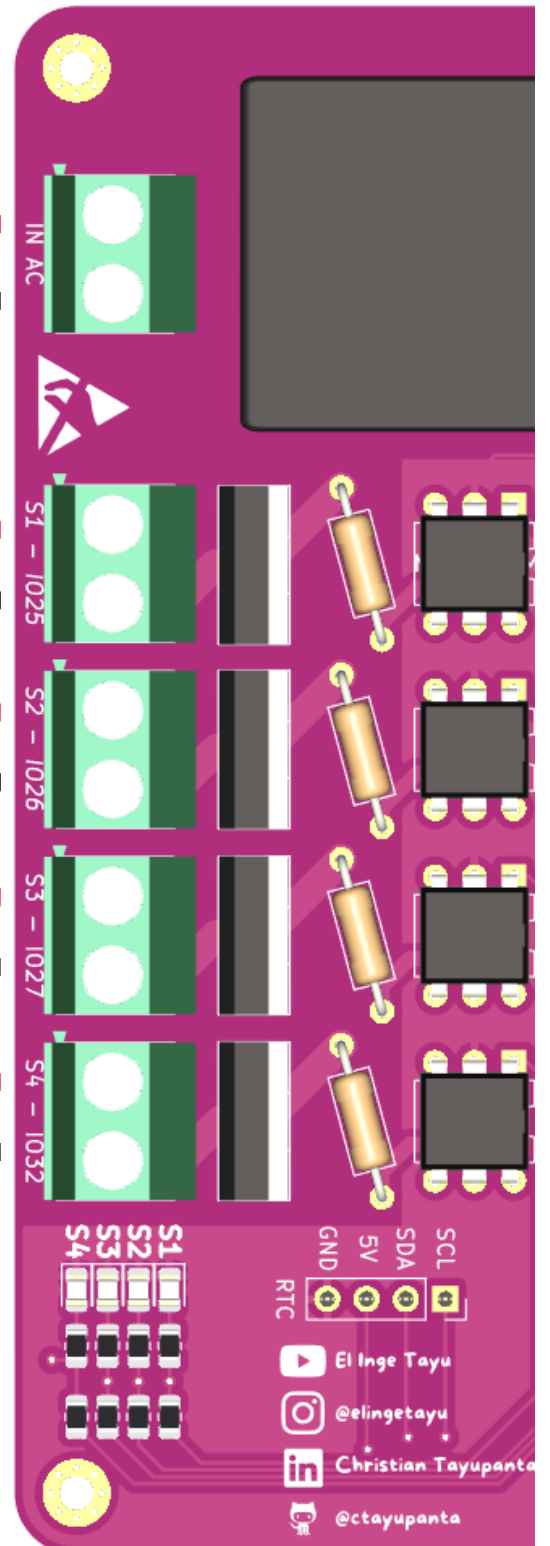
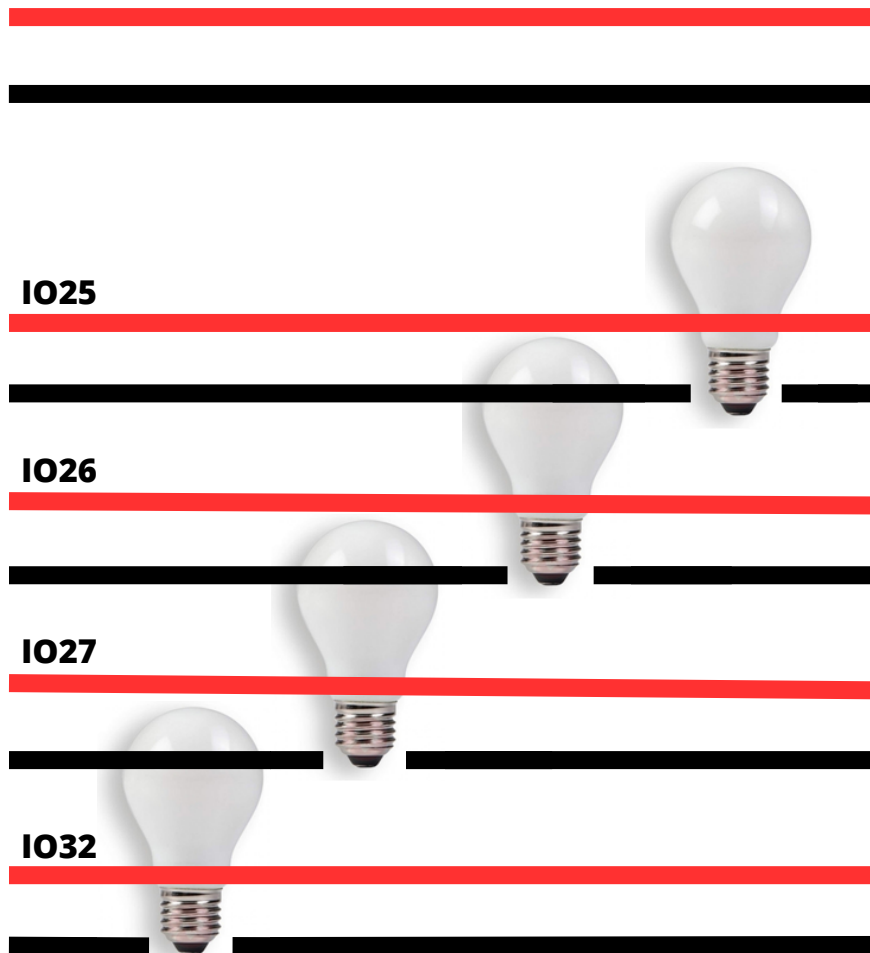


En un mundo en constante evolución tecnológica, presentamos una solución que redefine el control de cargas resistivas. Los SSR, superando los relés convencionales, garantizan una mayor confiabilidad y durabilidad, asegurando un rendimiento óptimo en todas las aplicaciones.

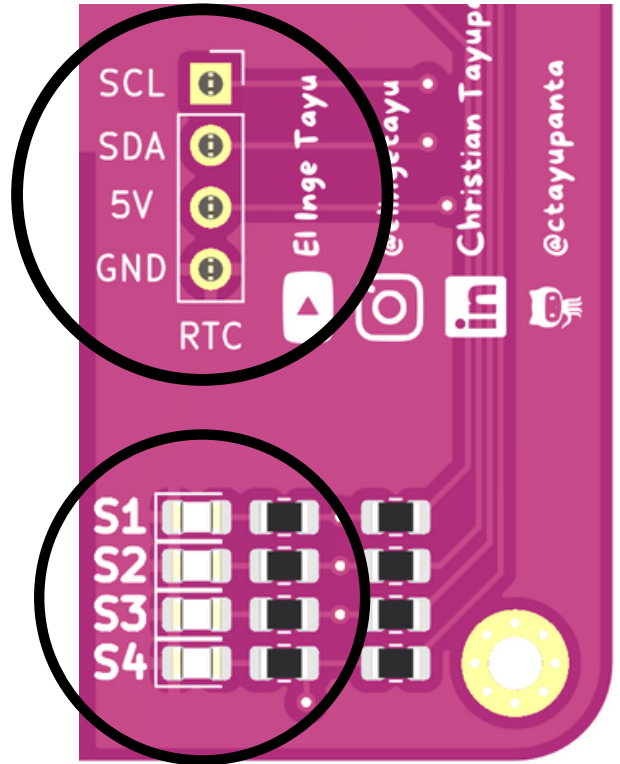
Con 4 salidas estratégicamente diseñadas, esta tarjeta ofrece una flexibilidad excepcional en la gestión de dispositivos. Desde iluminación hasta electrodomésticos, nuestra tarjeta se adapta a una amplia gama de necesidades. La integración de un potente ESP32 agrega un nivel de versatilidad impresionante.

La conectividad es clave en la era digital, y aquí es donde nuestra tarjeta brilla. A través de conexiones WiFi y Bluetooth, tienes el control total en tus manos, sin importar dónde te encuentres. Pero no nos detenemos ahí: gracias a la compatibilidad con Alexa, la automatización del hogar adquiere una dimensión completamente nueva.

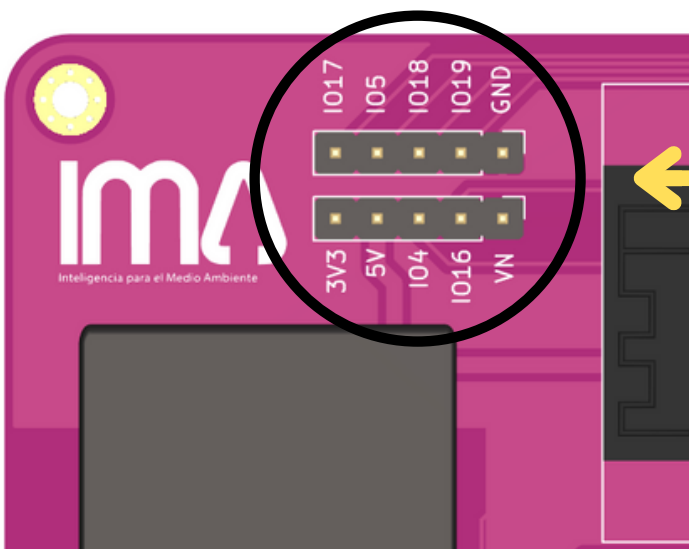
CONEXIONES / PINOUT



→ **RTC** →



→ **LEDS INDICADORES** →



← **PINES EXTRA
DE CONEXIÓN** →

CÓDIGO PARA ARDUINO

```
#include <Arduino.h>
#ifdef ESP32
#include <WiFi.h>
#else
#include <ESP8266WiFi.h>
#endif
#include "fauxmoESP.h"

#define WIFI_SSID "ESCRIBE_TU_RED"
#define WIFI_PASS "ESCRIBE_LA_CONTRA"
#define SERIAL_BAUDRATE 115200

// DEVICES
#define LED_1 25
#define ID_1 "Lampara 1"
#define LED_2 26
#define ID_2 "Lampara 2"
#define LED_3 27
#define ID_3 "Lampara 3"
#define LED_4 32
#define ID_4 "Lampara 4"

fauxmoESP fauxmo;
// -----
// Wifi
// -----

void wifiSetup() {

    // Set WIFI module to STA mode
    WiFi.mode(WIFI_STA);

    // Connect
    Serial.printf("[WIFI] Connecting to %s ", WIFI_SSID);
    WiFi.begin(WIFI_SSID, WIFI_PASS);
```

```
// Wait  
while (WiFi.status() != WL_CONNECTED) {  
  Serial.print(".");  
  delay(100);  
}  
Serial.println();  
  
// Connected!  
Serial.printf("[WIFI] STATION Mode, SSID: %s, IP address: %s\n", WiFi.SSID().c_str(),  
WiFi.localIP().toString().c_str());  
  
}  
  
void setup() {  
  
  Serial.begin(SERIAL_BAUDRATE);  
  
  // LEDs  
  pinMode(LED_1, OUTPUT);  
  digitalWrite(LED_1, LOW);  
  pinMode(LED_2, OUTPUT);  
  digitalWrite(LED_2, LOW);  
  pinMode(LED_3, OUTPUT);  
  digitalWrite(LED_3, LOW);  
  pinMode(LED_4, OUTPUT);  
  digitalWrite(LED_4, LOW);  
  
  // Wifi  
  wifiSetup();
```

```
fauxmo.createServer(true);
fauxmo.setPort(80);
fauxmo.enable(true);
// Add virtual devices
fauxmo.addDevice(ID_1);
fauxmo.addDevice(ID_2);
fauxmo.addDevice(ID_3);
fauxmo.addDevice(ID_4);
fauxmo.onSetState([(unsigned char device_id, const char * device_name,
bool state, unsigned char value) {

    Serial.printf("[MAIN] Device #%d (%s) state: %s value: %d\n", device_id,
device_name, state ? "ON" : "OFF", value);

    if (strcmp(device_name, ID_1)==0) {
        digitalWrite(LED_1, state ? HIGH : LOW);
    }else if (strcmp(device_name, ID_2)==0) {
        digitalWrite(LED_2, state ? HIGH : LOW);
    }else if (strcmp(device_name, ID_3)==0) {
        digitalWrite(LED_3, state ? HIGH : LOW);
    }else if (strcmp(device_name, ID_4)==0) {
        digitalWrite(LED_4, state ? HIGH : LOW);
    }

}]);

}

void loop() {
    fauxmo.handle();
}
```