PLANO Y ESQUEMÁTICO DEL CIRCUITO — MÓDULO 05

> ## Montaje en Protoboard (Estilo Tinkercad / Fritzing), Placa de Circuito Impreso Soldada (PCB FR-4), Interconexión RS-485 y Concurrencia del Puente

1. Descripción General del Sistema Físico

El sistema físico del Módulo 05 integra dos dominios de hardware interconectados mediante un bus diferencial **RS-485 Half-Duplex**:

- **Dominio de Campo OT (Arduino UNO R3 en Protoboard o Placa PCB FR-4 Soldada):** Ejecuta el firmware esclavo Modbus RTU (`ID = 1`), adquiriendo en tiempo real la temperatura (`A0`), iluminación (`A1`) y nivel de estanque ultrasónico (`D3/A2`), y gobernando las etapas de potencia mediante relés (`K52B-M-DC5` / `SRD-05VDC-SL-C`) para el motor extractor (`D8`), la bomba de riego (`D4`), el servomotor de escotilla (`D5`) y la botonera/indicadores LED locales (`D6..D12`).
- **Dominio de Borde IT/OT (Conversor USB-RS485 + Raspberry Pi 4B):** Acopla el par trenzado diferencial (`A+` , `B-` , `GND`) hacia el puerto `/dev/ttyUSB0` , donde el servicio `puente_modbus_a_mqtt_edge.py` arbitra el puerto serie con un cerrojo de exclusión mutua (`threading.Lock`) y publica/suscribe hacia el **Broker Eclipse Mosquitto 2.x**.

2. Plano 1: Montaje Completo en Protoboard del Nodo Esclavo (Estilo Tinkercad / Fritzing)

📐 Plano 1: Montaje en Protoboard Estilo Tinkercad / Fritzing

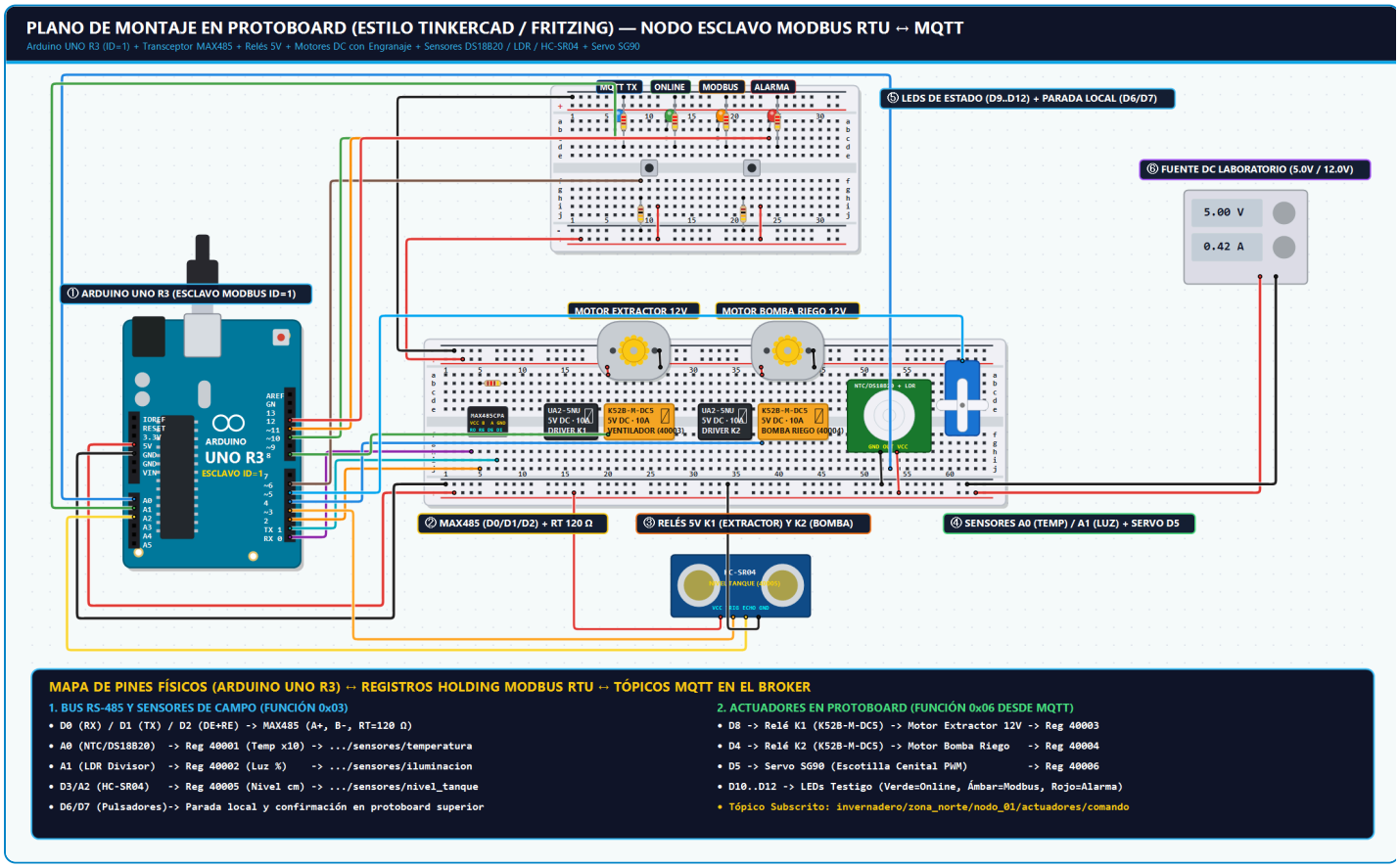


Tabla de Asignación de Pines del Arduino UNO R3 y Registros Modbus

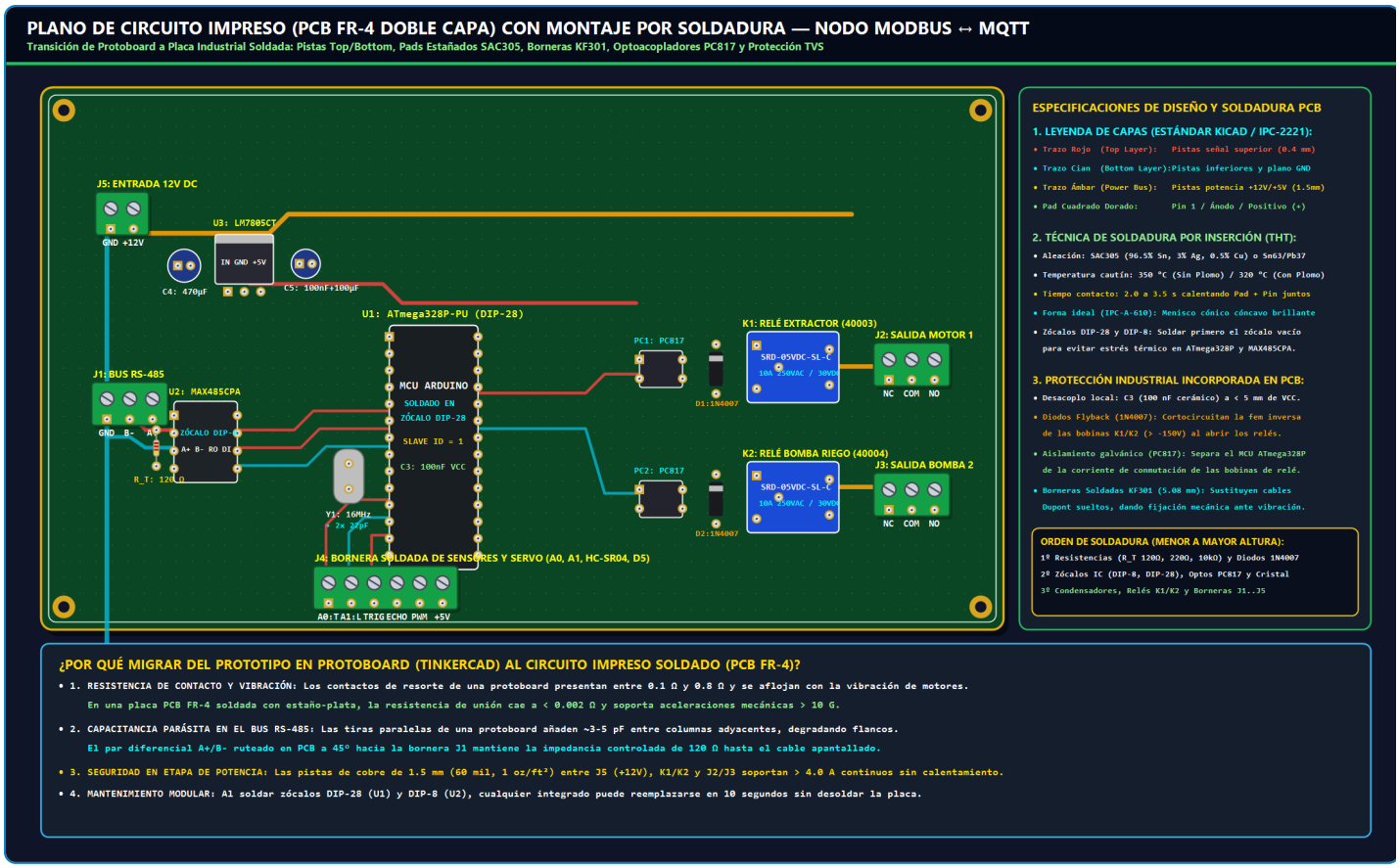
Pin Arduino UNO	Dispositivo en Protoboard / PCB	Registro Holding Modbus	Dirección de Datos y Tópico MQTT Asociado
`D0 (RX)` / `D1 (TX)`	Transceptor **MAX485** (`RO` / `DI`)	—	Tramas serie Modbus RTU a $9600\ bps$ (`8N1`).
`D2`	Transceptor **MAX485** (`DE + /RE` puenteados)	—	Control de dirección Half-Duplex (`HIGH` =TX, `LOW` =RX).
`A0`	Sensor Térmico **NTC / DS18B20**	`40001` (Índice 0)	Lectura `0x03` ($\times\ 10$) → `.../sensores/temperatura`
`A1`	Divisor Fotorresistivo **LDR +** $10\ k\Omega$	`40002` (Índice 1)	Lectura `0x03` ($0-100\ \%$) → `.../sensores/iluminacion`
`D8`	**Relé K1 (`K52B-M-DC5`)** → Motor Extractor 12V	`40003` (Índice 2)	Lectura `0x03` / Escritura `0x06` $\leq ftarrow$ `.../actuadores/comando`
`D4`	**Relé K2 (`K52B-M-DC5`)** → Motor Bomba Riego	`40004` (Índice 3)	Lectura `0x03` / Escritura `0x06` $\leq ftarrow$ `.../actuadores/comando`
`D3 (TRIG)` / `A2 (ECHO)`	Sensor Ultrasónico **HC-SR04** / Consigna	`40005` (Índice 4)	Nivel de estanque / Consigna térmica ($\times\ 10$) vía `0x06` .
`D5 (PWM)`	Micro Servomotor **SG90** (Escotilla)	`40006` (Índice 5)	Apertura proporcional de ventilación cenital ($0^\circ-90^\circ$).
`D6` , `D7`	Pulsadores Táctiles con Pull-Down $10\ k\Omega$	—	Parada local de emergencia y confirmación manual en planta.
`D9..D12`	Diodos LED 5 mm con resistencias de $220\ \Omega$	—	Indicadores luminosos: `MQTT TX` , `ONLINE` , `MODBUS` , `ALARMA` .

---

3. Plano 2: Circuito Impreso Industrial (PCB FR-4 Doble Capa) con Montaje por Soldadura

Para despliegues definitivos en invernaderos o naves industriales sometidas a vibraciones mecánicas, humedad relativa elevada y ciclos térmicos, el prototipo en protoboard se transfiere a una **placa de circuito impreso (PCB FR-4 de doble capa) con componentes soldados por inserción (`Through-Hole Technology - THT`) y bornes de tornillo (`KF301` de $5.08\ mm$)**:

📐 Plano 2: Circuito Impreso PCB FR-4 con Montaje por Soldadura



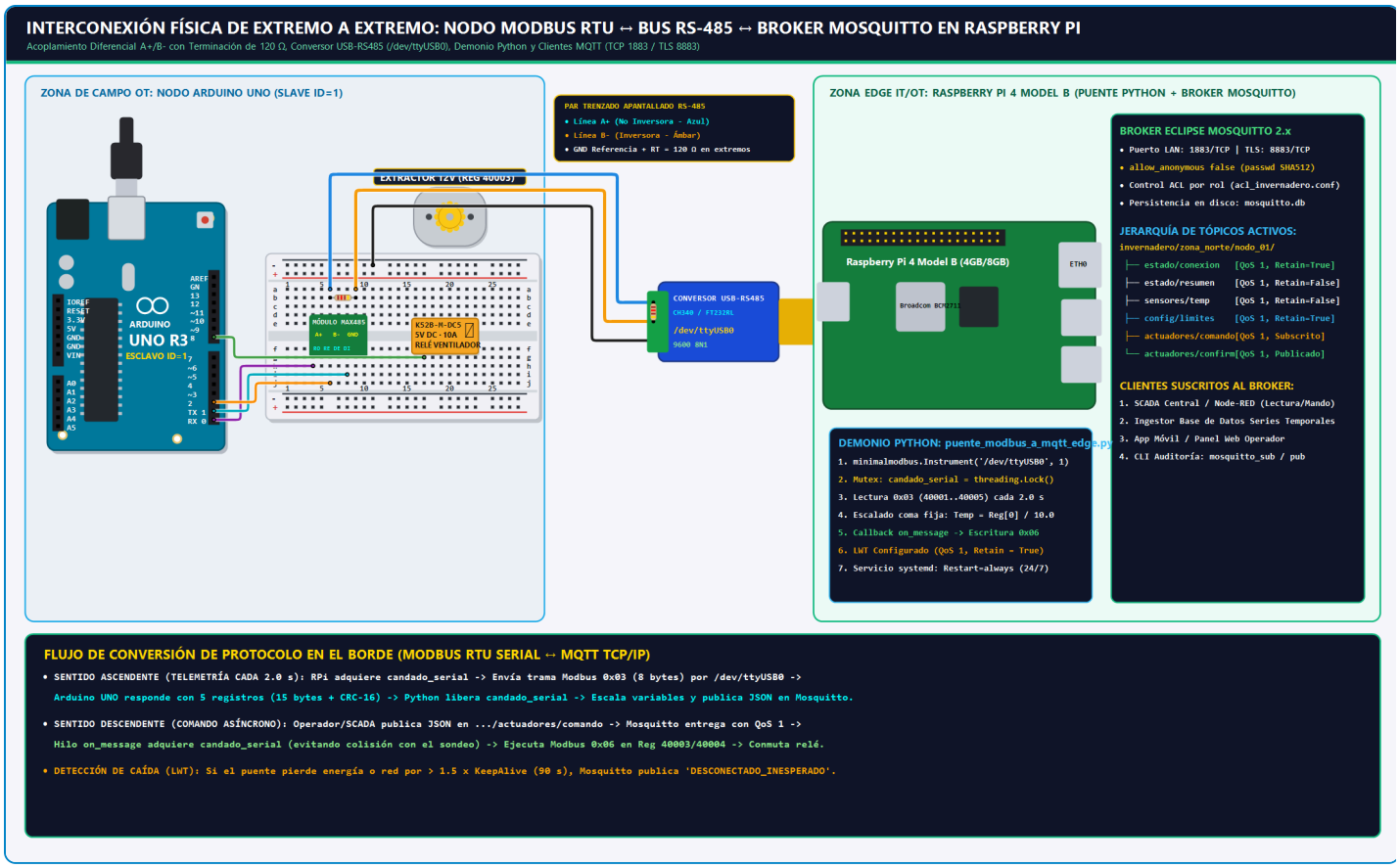
Especificaciones de Fabricación PCB y Proceso de Soldadura (`IPC-2221` / `IPC-A-610`)

Subsistema en PCB	Referencia Serigrafía	Especificación de Ruteo y Soldadura	Función de Ingeniería
Microcontrolador	`U1` (`ATmega328P-PU`) + `Y1` (16 MHz)	Soldar primero zócalo `DIP-28` sin el chip; insertar `U1` en frío. Condensador `C3` (100 nF) a < 5 mm de `VCC`.	Evita daño térmico al silicio durante la soldadura a $350\ ^\circ C$ y permite reemplazo rápido en campo.
Transceptor RS-485	`U2` (`MAX485CPA`) + `R_T` ($120\ \Omega$) + `J1`	Pistas diferenciales `A+` / `B-` de 0.6 mm ruteadas con codos a 45° hacia bornera soldada `J1`.	Elimina reflexiones de impedancia y capacitancias parásitas propias de las protoboards (3–5 pF).
Etapa de Potencia Aislada	`PC1/PC2` (`PC817`) + `D1/D2` (`1N4007`) + `K1/K2`	Pistas de potencia en cobre de 1.5 mm (60 mil, $1\ oz/ft^2$) entre relés `SRD-05VDC-SL-C` y borneras `J2` / `J3`.	Aísla galvánicamente el microcontrolador y suprime picos inductivos inversos (> -150 V) al desconectar motores.
Regulación de Energía	`U3` (`LM7805CT`) + `C4` ($470\ \mu F$) + `J5`	Pad cuadrado dorado identifica el terminal positivo (`+12V` / Ánodo). Aleación recomendada: `SAC305` o `Sn63/Pb37`.	Convierte la línea industrial de 12 V DC en 5.0 V estabilizados con resistencia de contacto $< 0.002\ \Omega$.

---

4. Plano 3: Acoplamiento Físico RS-485 hacia Raspberry Pi y Broker Mosquitto

📐 Plano 3: Acoplamiento RS-485 a Broker Mosquitto en Raspberry Pi



Reglas de Cableado e Integridad de Señal en el Bus RS-485

- **Polaridad Homóloga Directa (Sin Cruce):** A diferencia de UART punto a punto (donde `TX` se cruza con `RX`), en el bus diferencial RS-485 el borne `A+` **del módulo MAX485 se conecta directamente al borne** `A+` **del conversor USB-RS485**, y el borne `B-` **se conecta directamente al borne** `B-`.
- **Doble Terminación Resistiva ($R_T = 120\ \Omega$):** Instale una resistencia de $120\ \Omega$ entre `A+` y `B-` en el nodo Arduino UNO (final de línea) y otra de $120\ \Omega$ en la bornera del conversor USB-RS485 (cabecera de línea en la Raspberry Pi).
- **Referencia Común de Masa (`GND`):** Conecte siempre un tercer conductor de referencia `GND` entre la masa lógica del nodo de campo y el terminal `GND` del conversor USB-RS485 para mantener el voltaje de modo común dentro del rango operativo del transceptor ($-7\ V$ a $+12\ V$).

---

5. Plano 4: Jerarquía de Tópicos ISA-95, Niveles QoS y Máquina de Estados LWT

📐 Plano 4: Mapa de Jerarquía de Tópicos y Flujo QoS MQTT

ARQUITECTURA DE TÓPICOS JERÁRQUICOS ISA-95, COMODINES (+ / #), NIVELES QoS Y CICLO LWT

Diseño de Espacio de Nombres MQTT, Handshakes de Entrega (QoS 0, QoS 1, QoS 2), Mensajes Retenidos y Testamento de Última Voluntad

1. ÁRBOL JERÁRQUICO DE TÓPICOS (PLANTA → ZONA → NODO → RECURSO)

```
invernadero/                          [Nivel 1: Planta / Sitio]
└── zona_norte/                       [Nivel 2: Área / Célula]
    └── nodo_01/                      [Nivel 3: Unidad Edge / ID=1]
        ├── estado/
        │   ├── conexion              -> QoS 1 | Retain=True  (Online / LWT)
        │   └── resumen               -> QoS 1 | Retain=False (JSON Completo)
        ├── sensores/
        │   ├── temperatura           -> QoS 1 | Retain=False (24.5 °C)
        │   ├── iluminacion           -> QoS 1 | Retain=False (68 %)
        │   └── nivel_tanque          -> QoS 0 | Retain=False (142 cm)
        ├── configuracion/
        │   └── limites               -> QoS 1 | Retain=True  (Consigna 28.0 °C)
        └── actuadores/
            ├── comando               -> QoS 1 | Retain=False (Hacia Puente)
            └── confirmacion          -> QoS 1 | Retain=False (Eco Modbus 0x06)
```

REGLAS DE SUSCRIPCIÓN CON COMODINES (SOLO EN SUBSCRIBE):

- **Comodín Un Nivel (+):** `invernadero/+/+/sensores/temperatura`
- **Comodín Multinivel (#):** `invernadero/zona_norte/nodo_01/#`
- Antipatrones Prohibidos: Nunca iniciar con '/' ni usar espacios o '#'

2. NIVELES DE CALIDAD DE SERVICIO (QoS 0, QoS 1 Y QoS 2)

QoS 0 — A LO SUMO UNA VEZ (AT MOST ONCE / FIRE AND FORGET)

```
Emisor  ──────[ PUBLISH (QoS=0) ]──────►  Broker Mosquitto   (1 paquete MQTT, sin acuse)
```

Uso recomendado: Variables de alta frecuencia no críticas donde el siguiente muestreo reemplaza al anterior.

QoS 1 — AL MENOS UNA VEZ (AT LEAST ONCE — ESTÁNDAR INDUSTRIAL IIoT)

```
Emisor  ──────[ 1. PUBLISH (PacketId=K, DUP=0) ]──────►  Broker Mosquitto
Emisor  ◄──────[ 2. PUBACK  (PacketId=K)        ]──────  Broker Mosquitto   (2 paquetes MQTT)
```

Uso recomendado: Telemetría agregada, alarmas, LWT y comandos de actuación con idempotencia.

QoS 2 — EXACTAMENTE UNA VEZ (EXACTLY ONCE — HANDSHAKE DE 4 FASES)

```
Emisor ──[1. PUBLISH]──► Broker ──[2. PUBREC]──► Emisor ──[3. PUBREL]──► Broker ──[4. PUBCOMP]
```

Sobrecarga: 4 paquetes por mensaje. Reservado para facturación fiscal o dosificación no repetible.

Fórmula Paquetes Totales (1 Emisor + M Suscriptores en QoS q): $N_{paq} = (2^q) * (1 + M)$

3. MENSAJES RETENIDOS (RETAIN=TRUE) Y MÁQUINA DE ESTADOS DEL TESTAMENTO (LWT)

- FASE 1 (CONEXIÓN INICIAL): Puente envía paquete CONNECT registrando el LWT en RAM del Broker:
  Tópico LWT: .../estado/conexion | Payload: 'DESCONECTADO_INESPERADO' | QoS=1 | Retain=True
- FASE 2 (MENSAJE DE NACIMIENTO / BIRTH): Inmediatamente tras recibir CONNACK, el puente publica:
  Tópico: .../estado/conexion | Payload: 'CONECTADO' | QoS=1 | Retain=True
- FASE 3 (SUPERVISIÓN KEEP-ALIVE): El cliente envía PINGREQ si no hay tráfico en T_keepalive = 60 s.
- FASE 4 (DISPARO AUTOMÁTICO LWT): Si transcurre T_LWT = 1.5 * T_keepalive = 90 s sin PINGREQ ni DISCONNECT,
  el Broker Mosquitto cierra el socket y publica automáticamente el LWT retenido a todos los SCADA suscritos.

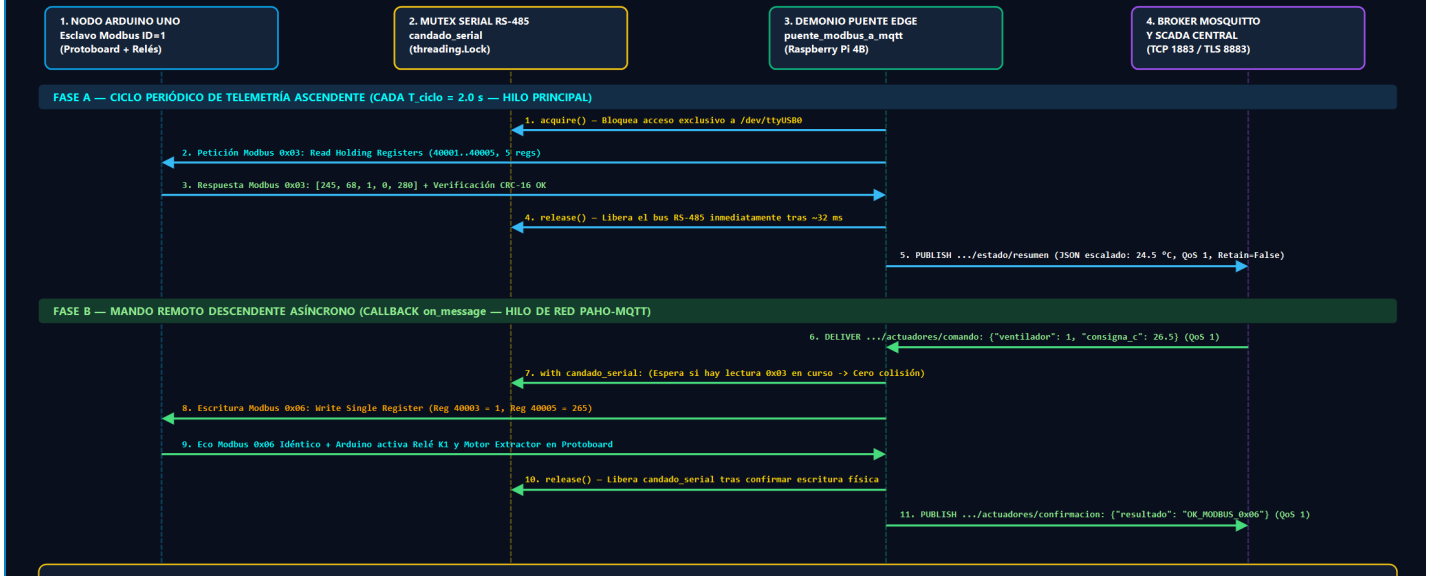
---

6. Plano 5: Secuencia Bidireccional y Exclusión Mutua (`threading.Lock`)

📐 Plano 5: Diagrama de Secuencia del Puente Bidireccional Modbus a MQTT

DIAGRAMA DE SECUENCIA BIDIRECCIONAL Y EXCLUSIÓN MUTUA (MODBUS RTU ↔ PUENTE PYTHON ↔ MOSQUITTO)

Arbitraje del Bus RS-485 con threading.Lock() entre el Bucle de Sondeo (0x03) y el Hilo Asíncrono de Comandos MQTT (0x06)



¿POR QUÉ ES OBLIGATORIO EL MUTEX (threading.Lock) EN EL PUENTE MODBUS ↔ MQTT?

- En paho-mqtt, client.loop_start() ejecuta las recepciones de red (on_message) en un hilo secundario asíncrono, mientras que el bucle principal while True ejecuta el sondeo periódico de los sensores cada 2.0 segundos.
- El bus RS-485 es estrictamente Half-Duplex: si el hilo on_message intenta enviar una trama 0x06 por /dev/ttyUSB0 en el mismo milisegundo en que el bucle principal espera los 15 bytes de respuesta de una lectura 0x03, los bytes de ambas tramas se entrelazan en el UART, corrompiendo el CRC-16 y provocando una excepción de Timeout.
- Con 'with candado_serial:', toda transacción Petición-Respuesta + Silencio de 3.5 caracteres es 100% ATÓMICA.
- Latencia máxima de encolamiento del comando: T_espera_max <= T_transaccion_0x03 (~32 ms a 9600 bps).