Listado de Materiales (BOM) — Módulo 04: Comunicación Directa UART, I²C, SPI, RS-485 y Modbus RTU

Curso: De Sistemas Embebidos a IoT — Integración Práctica con Arduino y Raspberry Pi

Módulo 04: Comunicación Directa Arduino ↔ Raspberry Pi

Referencia en Video 04: Diapositivas 3, 5, 6, 8, 9, 10 y 14

---

1. Tabla Principal de Componentes de Hardware y Comunicaciones

Ítem	Cant.	Designador	Componente / Descripción Técnica	Especificación Eléctrica / Mecánica	Función en el Módulo 04
01	1	`SBC1`	**Raspberry Pi 4 Model B** (o RPi 5)	SoC ARM64 (`3.3 V` CMOS), puertos USB y cabezal GPIO de 40 pines	Nodo Maestro `Modbus RTU` / Maestro `I²C` y `SPI`
02	1	`U1`	**Arduino UNO R3** (`ATmega328P`)	Lógica TTL de `5.0 V`, reloj de $16\ \text{MHz}$, `UART` hardware (`D0` / `D1`), `I²C` (`A4` / `A5`)	Nodo Esclavo `Modbus RTU` (`ID = 1`) / Coprocesador `I²C` (`0x08`)
03	1	`U2`	**Adaptador Industrial USB a RS-485**	Chip `FT232RL` + `MAX485` (o `CH340E`), protección `TVS` contra transitorios y bornes `A(+)` , `B(-)` , `GND`	Interfaz física `RS-485` en la Raspberry Pi (`/dev/ttyUSB0`)
04	2	`U3, U4`	**Módulo Transceptor RS-485 (`MAX485CSA`)**	Alimentación `5.0 V` , semidúplex hasta $2.5\ \text{Mbps}$, hasta 32 cargas unitarias (`1 UL = 12 kΩ`)	Conversión entre `UART` TTL (`RO` , `DI` , `DE` / `/RE`) y par diferencial `RS-485`
05	1	`U5`	**Conversor Bidireccional de Nivel Lógico `BSS138` (4 Canales)**	4 transistores MOSFET Canal N `BSS138` con resistencias *pull-up* de $10\ \text{k}\Omega$ en `LV` (`3.3 V`) y `HV` (`5.0 V`)	Adaptación bidireccional de colector abierto para buses `I²C` (`SDA` / `SCL`) y `SPI`
06	2	`R_TERM1, R_TERM2`	**Resistencias Terminadoras de Línea RS-485**	$120\ \Omega\ \pm\ 1\%$, película metálica, potencia de $1/4\ \text{W}$ ($0.25\ \text{W}$)	Adaptación de impedancia característica ($Z_0 = 120\ \Omega$) en los 2 extremos físicos del bus
07	1	`R_PD`	**Resistencia Pull-Down de Habilitación TX**	$10\ \text{k}\Omega\ \pm\ 1\%$, $1/4\ \text{W}$	Mantiene los pines `DE` / `/RE` (`Pin D4`) en modo escucha (`0 V`) durante el arranque del AVR
08	2	`R_PU1, R_PU2`	**Resistencias Pull-Up Externas para Bus I²C**	$4.7\ \text{k}\Omega\ \pm\ 1\%$, $1/4\ \text{W}$	Ajuste del tiempo de subida ($\tau = R_p \cdot C_{bus}$) hacia el riel de `3.3 V` si hay múltiples esclavos
09	1	`R1, R2`	**Par de Resistencias para Divisor UART Directo**	$R_1 = 1.8\ \text{k}\Omega$ y $R_2 = 3.3\ \text{k}\Omega$ ($\pm\ 1\%$, $1/4\ \text{W}$)	Atenuación de la línea `TX` de `5.0 V` del Arduino a $3.24\ \text{V}$ hacia `GPIO 15 (RXD)`
10	2	`C_DEC1, C_DEC2`	**Condensadores de Desacoplo Cerámicos**	$100\ \text{nF}$ ($0.1\ \mu\text{F}$), dieléctrico `X7R` , $50\ \text{V}$	Filtrado de picos de corriente en el pin `VCC` de cada transceptor `MAX485`
11	5 m	`CAB1`	**Cable de Par Trenzado Blindado (`STP` / `RS-485`)**	Impedancia característica $Z_0 = 120\ \Omega$, calibre `24 AWG` con malla de drenaje	Línea troncal diferencial para interconexión en cadena (*Daisy-Chain*)

2. Cálculos de Verificación Eléctrica

2.1 Impedancia Equivalente y Corriente en el Bus RS-485

Con únicamente las dos resistencias terminadoras de $120\ \Omega$ instaladas en los extremos inicial y final de la línea troncal:

$$R_{eq} = \frac{R_{term,1} \cdot R_{term,2}}{R_{term,1} + R_{term,2}} = (120\ \Omega \cdot 120\ \Omega) / (120\ \Omega + 120\ \Omega) = 60\ \Omega$$

Para una tensión diferencial típica de salida $V_{OD} = 2.0\ V$ entregada por el transmisor `MAX485`:

- **Corriente en el bucle diferencial**: $I_{bus} = (2.0\ V) / (60\ \Omega) = 33.3\ mA$ (dentro del límite nominal de $60\ mA$ del `MAX485`).
- **Potencia disipada en cada terminador**: $P_{term} = ((2.0\ V)^2) / (120\ \Omega) = 33.3\ mW \ll 250\ mW$ ($1/4\ W$).

2.2 Tiempo de Carácter y Silencio de Trama Modbus RTU a 19200 bps (`8E1`)

Cada carácter en formato `8E1` consta de $11\ bits$ ($1\ Start + 8\ Datos + 1\ Paridad\ Par + 1\ Stop$):

$$T_{char} = (11\ bits) / (19200\ bits/s) = 572.92\ \mu s \quad\Longrightarrow\quad T_{3.5\ char} = 3.5 \times 572.92\ \mu s \approx 2005\ \mu s \approx 2.0\ ms$$