

8.5 Serial protocols

Name: _____ Class: _____ Date: _____

Total: 23 marks
KEY WORDS UART • baud rate • chip select • open-drain

Objective

Configure UART/SPI/I2C from their timing rules, compute throughput and framing budgets, and choose a protocol from requirements – every baud and clock computation checked from its definition.

1 Key knowledge and worked examples

■ The three protocols

- **UART** : no clock wire; both ends agree a **baud rate** ; each byte is framed as start + 8 data + optional parity + stop (10 bits for 8N1). Both ends must agree the frame too.
- **SPI**: master drives SCLK; **MOSI/MISO** carry both directions simultaneously; one **chip select** per device. Fast, wired per device.
- **I2C**: two shared wires (SDA/SCL), **open-drain** with pull-ups; 7-bit addressing (up to ~112 devices + reserved), multi-master capable, clocked at 100/400 kHz (standard/fast).

■ Worked example: the UART arithmetic

115 200 baud, 8N1 $\rightarrow$ 10 bits per byte $\rightarrow$ 11 520 bytes/s; one byte takes $10/115\,200 = 86.8\text{ }\mu\text{s}$. A 100-byte message needs 8.68 ms – it does NOT fit a 1 ms control period (the sizing decides). **SPI** at 10 MHz: a byte in $0.8\text{ }\mu\text{s}$ ($108\times$ faster than this UART). **I2C** at 400 kHz: a byte needs 9 clock pulses (8 data + 1 ack) = $22.5\text{ }\mu\text{s}$ – slower than both, but two wires for the whole board. Speed vs wiring vs addressing: the three protocols are three points on that trade.

■ I2C mechanics

Start condition (SDA falls while SCL high), 7-bit address + R/W, ack bit per byte, stop condition. Any device can hold SCL low (**clock stretching**) – the open-drain wiring makes the bus AND-wired: slow devices are first-class, which is why I2C fits sensors.

2 Practice

2.1 A 9600-baud 8N1 link: bytes per second, and the time for a 20-byte message. [3]

2.2 SPI at 4 MHz with 8-bit frames: time to clock one byte (full duplex – what is transferred in BOTH directions in that time)? [3]

2.3 How many devices can share one I2C bus (7-bit addressing), and what physically limits the practical number? [2]

3 Exam-style questions

3.1 A sensor board reports a 12-byte packet every 10 ms over UART 9600 8N1.

Compute the airtime of one packet and the link utilisation (%). The requirement doubles: packets every 5 ms. Compute the minimum baud rate from 8N1 framing (packet time ≤ 5 ms, and state the standard baud you would choose: 19200, 38400 or 57600). Then compute the same requirement over I2C fast mode (400 kHz: 9 clocks per byte including ack, plus the 2-byte address+register overhead per packet), and state which link meets the deadline with more margin. [8]

3.2 An SPI ADC: 16-bit results, SCLK at 2 MHz, chip select active during each conversion read; the ADC needs its CS high ≥ 50 ns between reads.

Compute the time to clock one 16-bit result, the maximum complete conversions per second if CS overhead is 100 ns per read, and the data rate in bytes/s. Then a firmware bug writes the SPI data register while the transmit buffer is full – state what happens to the sample and what the correct flow is. Finally, compare with the same ADC on an I2C bus at 400 kHz (2 address bytes + 2 data bytes + acks: 36 clocks per conversion) and state the conversion rate there. [7]

Solutions

2.1

- $9600/10 = 960$ bytes/s.
- 20 bytes: $20 \times 10/9600 = 20.8$ ms.

2.2

- 8 bits at 4 MHz: $8/(4 \times 10^6) = 2$ μ s.
- In that time one byte travels MOSI (out) AND one MISO (in) – full duplex: 2 bytes of information per 2 μ s.

2.3

- 7-bit addressing $\rightarrow$ 128 addresses, $\sim$ 112 usable (16 reserved).
- Practical limit: bus capacitance (spec $\sim$ 400 pF) – every device adds capacitance, and the open-drain rise time degrades.

3.1

- Packet airtime at 9600: $12 \times 10/9600 = 12.5$ ms.
- Utilisation: $12.5/10 = 125\%$ – ALREADY over the budget; the 10 ms requirement is not met at 9600.
- Doubled requirement (5 ms): need $12 \times 10/\text{baud} \leq 5$ ms $\rightarrow \text{baud} \geq 24000 \rightarrow$ choose 38 400 (19200 gives 6.25 ms – misses).
- I2C fast mode: clocks per packet = $2 + 12 \times 9 = 110$ clocks (2 overhead + 9-bit bytes) $\rightarrow 110/(400 \times 10^3) = 275$ μ s – meets 5 ms with $18\times$ margin.
- I2C meets the deadline with far more margin (275 μ s vs 3.1 ms at 38 400 UART) – for a 12-byte periodic read, the I2C answer is decisive.

3.2

- One 16-bit result: $16/(2 \times 10^6) = 8$ μ s.
- Per read: 8 μ s clock + 100 ns CS overhead ≈ 8.1 μ s $\rightarrow \max 1/(8.1 \times 10^{-6}) = 123\,457$ conversions/s.
- Data rate: 2 bytes each $\rightarrow 246\,907$ bytes/s.
- Writing with a full transmit buffer: the write is typically ignored (or overruns, losing a byte) – the sample is corrupted/lost. Correct flow: wait for TX-empty before writing, and RX-not-empty before reading (status flags or interrupts), one flag per transaction step.
- I2C: 36 clocks per conversion $\rightarrow 400$ kHz/36 = 11 111 conversions/s – $11\times$ slower than SPI here: the protocol choice follows the throughput.