WORKSHEET

A12.1 Communication methods

Networks and the web · A level · OCR H446 1.3.3, AQA 7517 4.9.1.1, Eduqas A500QS 2.2 · about 25 min

BugBotLab

Name _____ Class _____ Date _____

What this lesson is about

Serial and parallel, synchronous and asynchronous, baud rate, bit rate, bandwidth, latency and protocols.

- Serial and parallel
- Synchronous and asynchronous
- Baud rate and bit rate
- Bandwidth and latency
- Protocols

Questions

5 marks in all

1. Why is serial transmission preferred to parallel over long distances? [1 mark]

- ☐ **A** Parallel wires suffer skew and crosstalk, which corrupt data at high speeds
- ☐ **B** Serial sends several bits at the same moment
- ☐ **C** Parallel needs start and stop bits
- ☐ **D** Serial does not need a protocol

2. In asynchronous serial transmission, what is the purpose of the start bit? [1 mark]

- ☐ **A** It changes the line from its idle state so the receiver can synchronise its clock for the character
- ☐ **B** It tells the receiver how many data bits follow
- ☐ **C** It is a parity bit for error checking
- ☐ **D** It returns the line to its idle state

3. A link uses 16 different signal levels and a baud rate of 2400. What is its bit rate in bits per second? [1 mark]

4. What does this program print? [1 mark]

```
baud = 4800
bits_per_frame = 1 + 8 + 1
print(baud // bits_per_frame, 'characters per second')
```

5. Which statement about bandwidth and bit rate is correct? [1 mark]

- ☐ **A** Bit rate is directly proportional to bandwidth
- ☐ **B** Bit rate is always equal to baud rate
- ☐ **C** Increasing bandwidth increases latency
- ☐ **D** Bandwidth is the delay before data arrives

The task: frame a byte

The LED can be a one-wire serial line: white for 1, off for 0. Write `frame(ch)`. Its input `ch` is a single character with a code from 0 to 255. It returns a string of exactly 10 characters, each `0` or `1`: a start bit `0`, the 8 data bits of the character's code least significant bit first, then a stop bit `1`. For each character of `Hi!`, in order: 1. print the character, a space and its frame, for example `H 0000100101`; 2. put each of the 10 bits on the LED in turn, `led("white")` for 1 and `led("off")` for 0, with `wait(0.05)` after each. Then print `bits sent: <n>`, where n is the number of bits you put on the LED, counted by your loop. Last, print how long 1200 characters take at 9600 baud with a 2-level signal, using the length of one frame: `time for 1200 characters: <seconds> s`, where seconds is the result of the division printed as Python prints it.

```
# the two lines every program starts with: the commands, then the robot
from bugbot import *
connect()

BAUD = 9600

def frame(ch):
    data = format(ord(ch), "08b")
    return data

for ch in "Hi!":
    print(ch, frame(ch))
```

Plan your program here, then type it in and press Run.

Do it on the robot

www.bugbotlab.com/learn/a12-1-communication-methods/

The simulator checks it and tells you when it passes. Nothing to install, no account.

Challenges

1. Add an even parity bit between the data and the stop bit. How many characters per second does 9600 baud carry now?
2. Write `unframe(bits)` that checks the start and stop bits and returns the character, or `None` if the framing is wrong.
3. A line uses 8 voltage levels at 4800 baud. Work out its bit rate, then check it with `bit_rate`.