

Serial Ports

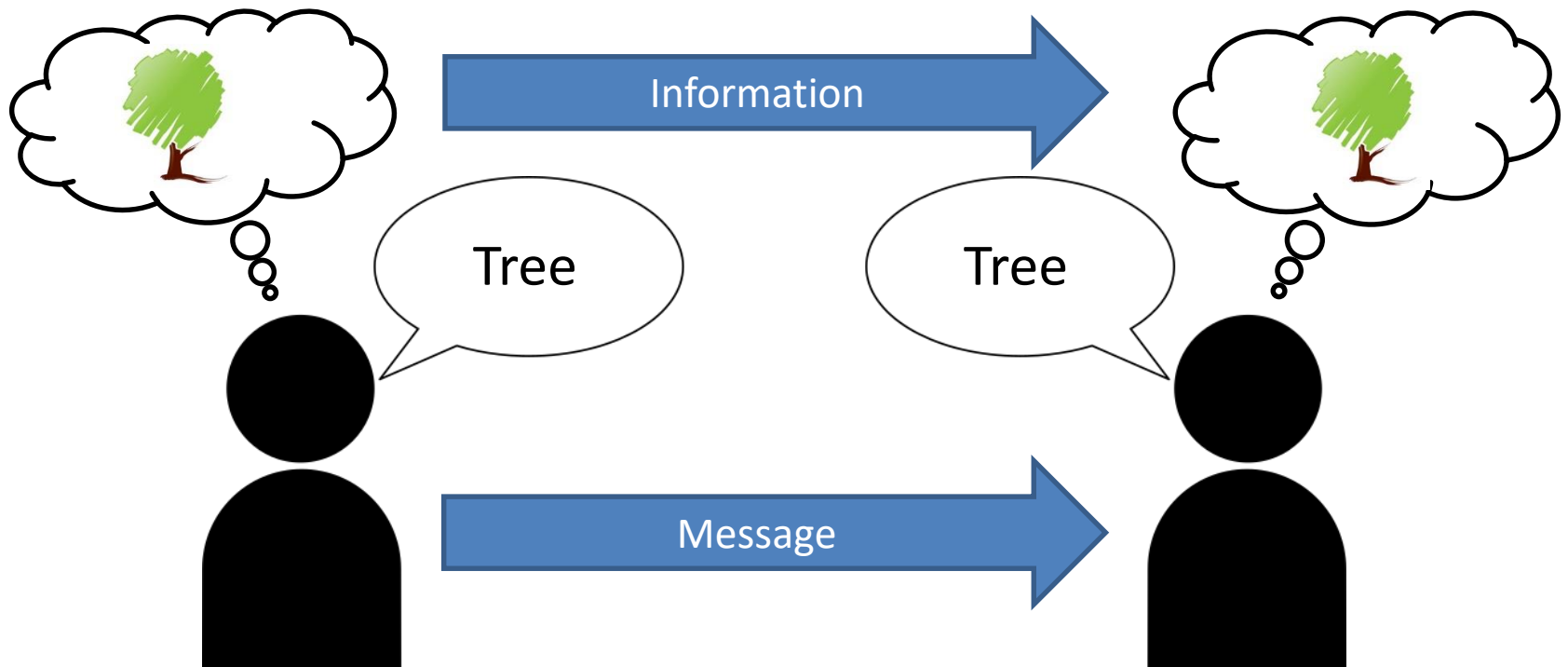
Networks and Embedded Systems

Second Grade Level

Wolfgang Neff

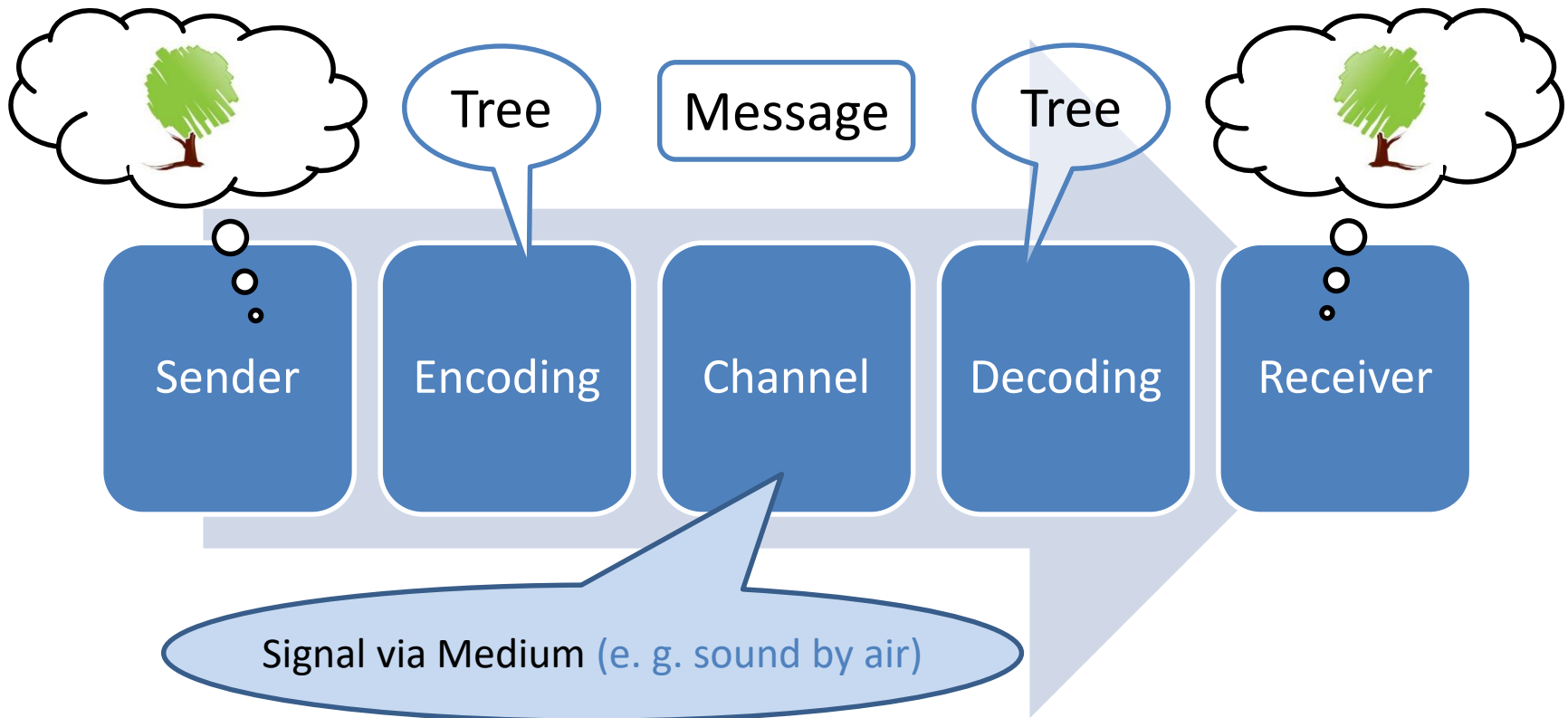
Data Transmission (1)

- Communication: simple view



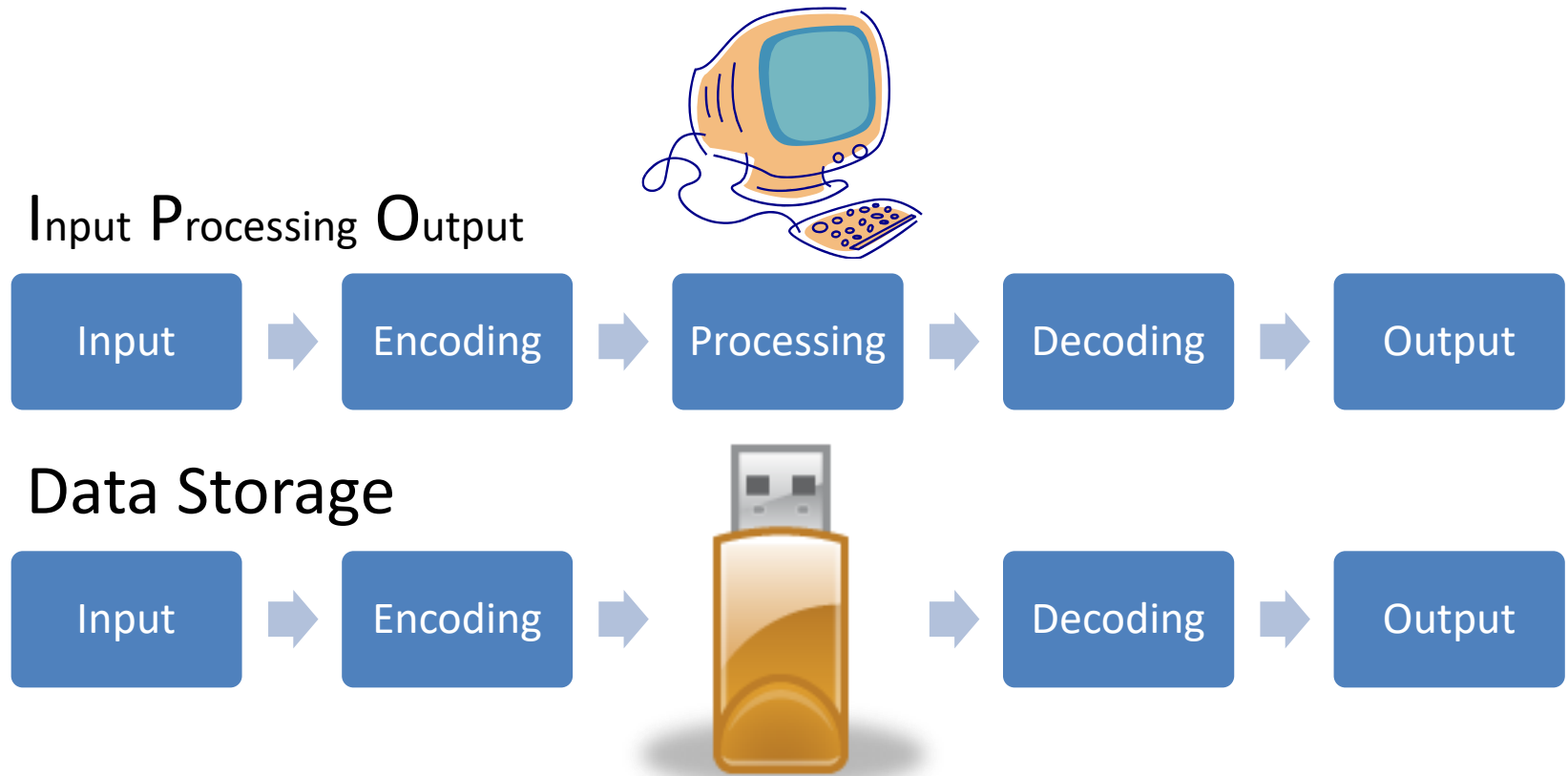
Data Transmission (2)

- Communication: detailed view



Data Transmission (3)

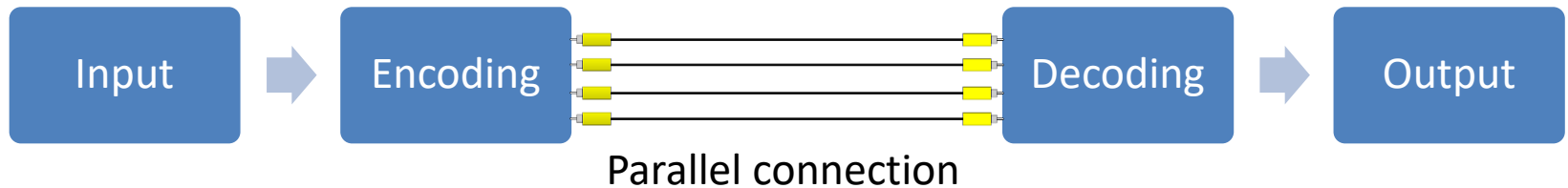
- Data processing and data storage



Data Transmission (4)

- Data communication

Parallel communication



Serial communication



Data Transmission (5)

- Encoding example: ASCII code table

| ASCII | | Lower Hex Digit | | | | | | | | | | | | | | | |
|------------------|---|-----------------|-----|-----|-----|-----|-----|-----|-----|-----|----|-----|-----|----|----|----|-----|
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| Higher Hex Digit | 0 | NUL | SOH | STX | ETX | EOF | ENQ | ACK | BEL | BS | HT | LF | VT | FF | CR | SO | SI |
| | 1 | DLE | DC1 | DC2 | DC3 | DC4 | NAK | SYN | ETB | CAN | EM | SUB | ESC | FS | GS | RS | US |
| | 2 | | ! | " | # | \$ | % | & | ' | (|) | * | + | , | - | . | / |
| | 3 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | : | ; | < | = | > | ? |
| | 4 | @ | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O |
| | 5 | P | Q | R | S | T | U | V | W | X | Y | Z | [| \ |] | ^ | _ |
| | 6 | ` | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o |
| | 7 | p | q | r | s | t | u | v | w | x | y | z | { | | } | ~ | DEL |

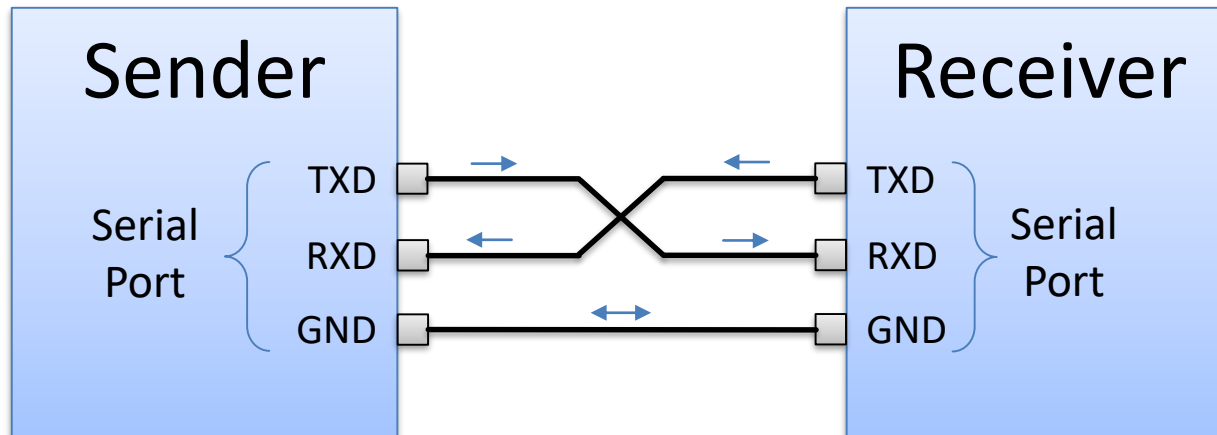
Data Transmission (6)

- Serial communication is already quite old
 - Used to interconnect
 - Teletypewriters
 - Mainframe computers
 - Terminals
 - Printers
 - Etc.



Serial Ports (1)

- Serial Ports are used for serial communication
 - TXD: transmitted data
 - RXD: received data
 - GND: ground, electrical bonding



Serial Ports (2)

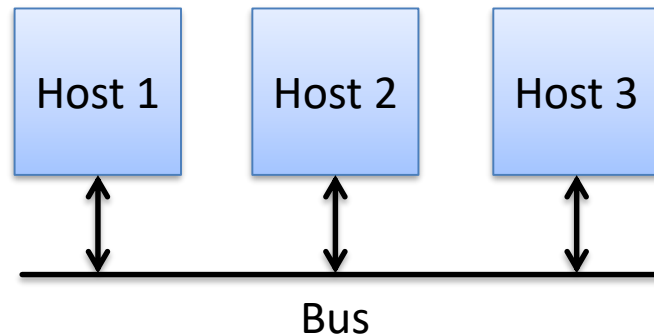
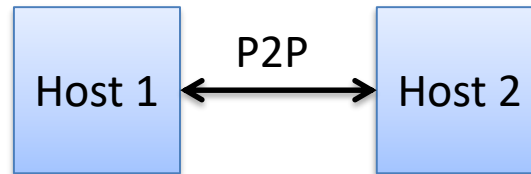
- How hosts communicate with each other

- Point-to-point connection

- UART

- Serial buses

- SPI
- I²C
- CAN
- USB



Serial Ports (3)

- Clock drift may corrupt data

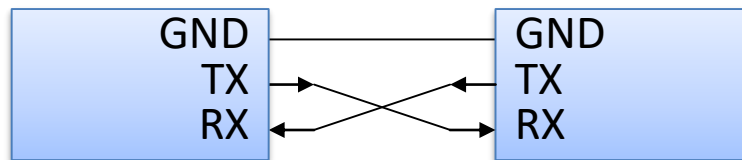
| | | | | | | | | |
|---------|---|---|---|---|---|---|---|---|
| Data | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| Clock A | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Signal | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| | | | | | | | | |
| Clock B | 0 | 1 | 2 | 3 | 4 | 5 | 6 | |
| Data | 0 | 0 | 1 | 0 | 1 | 1 | 0 | |

Data seen by clock A: 8 Bits, 0010 0110, 26_{hex}

Data seen by clock B: 7 Bits, 001 0110, 16_{hex}

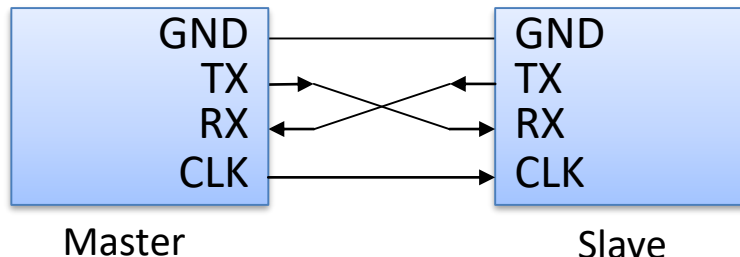
Serial Ports (4)

- Clocks must to be synchronized
 - Synchronization bits (cf. UART)



So-called **Asynchronous**
Data Transmission

- Synchronization line (cf. I2C or SPI)



So-called **Synchronous**
Data Transmission