



# SNS COLLEGE OF TECHNOLOGY

Coimbatore-35  
An Autonomous Institution



Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A++' Grade  
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

## DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

### 19ECT312 – EMBEDDED SYSTEM DESIGN

Communication From Serial  
Devices /19ECT312/Embedded  
systems Design /  
Mrs.E.Ramya/AP/ECE/SNSCT

III YEAR/ VI SEMESTER  
1

### UNIT 2 :DEVICES AND EMERGING BUS STANDARDS

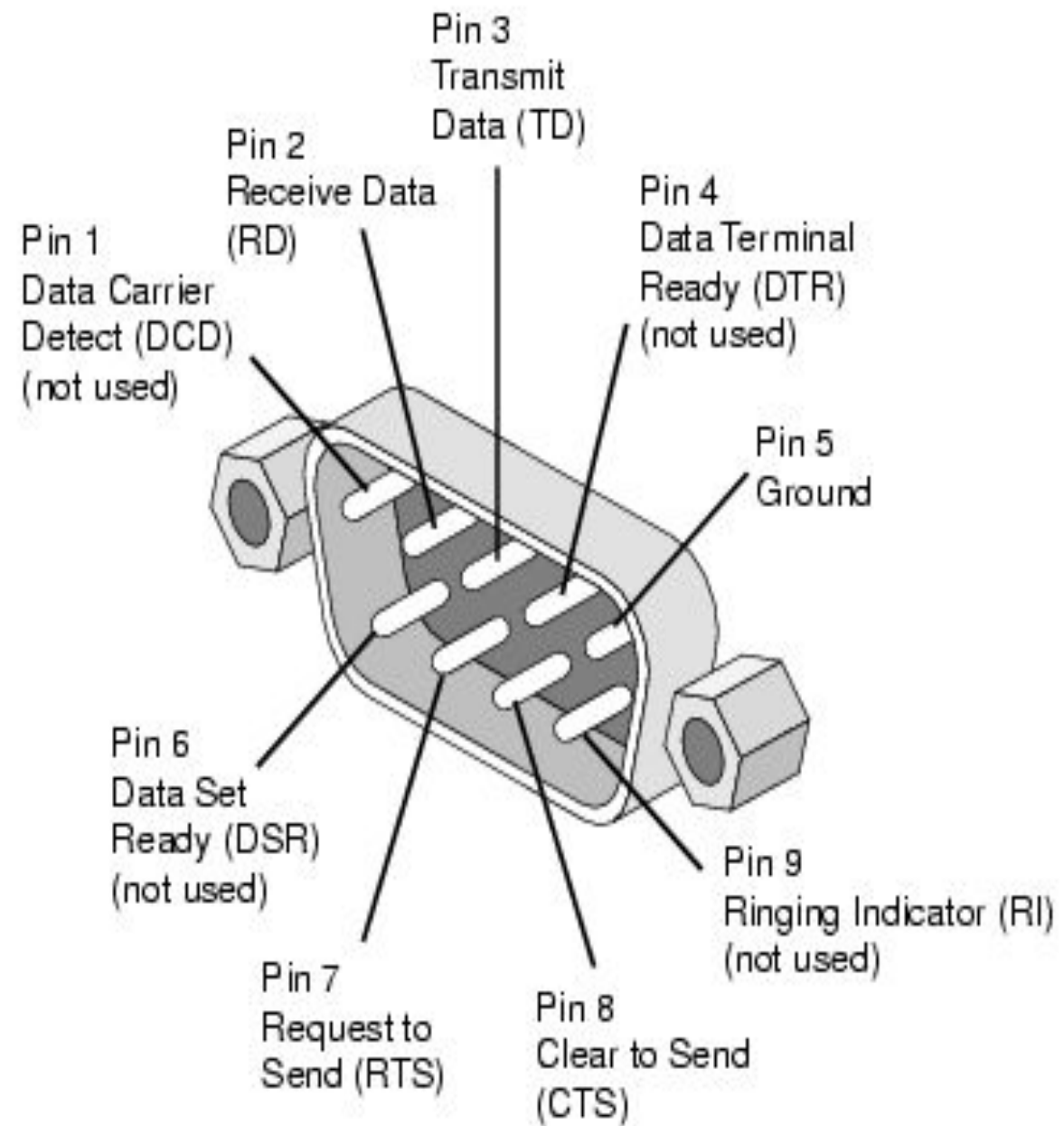
TOPIC : Communication from serial devices-UART,SPI,I2C



# COMMUNICATION FROM SERIAL DEVICES



## Serial buses

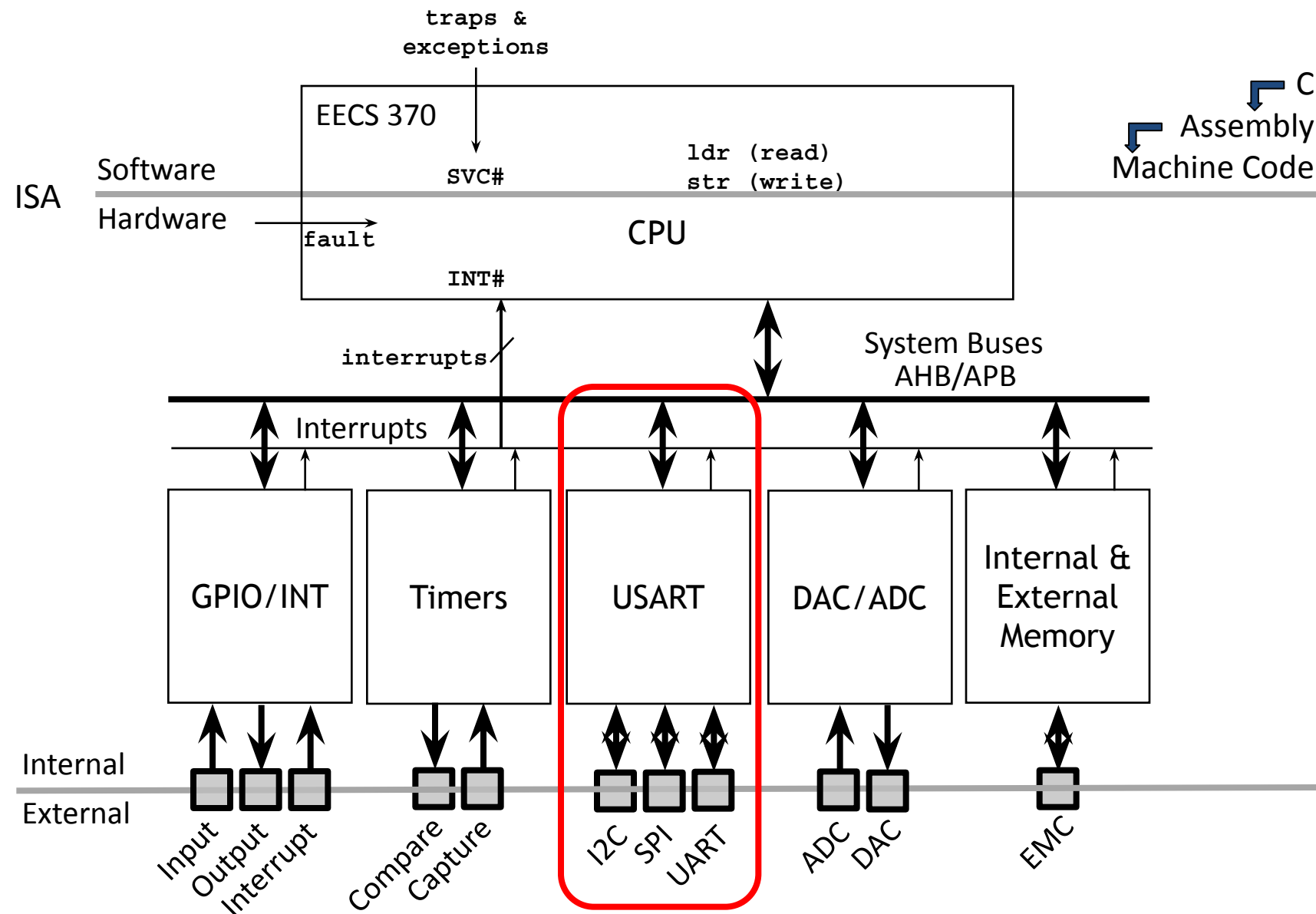




# COMMUNICATION FROM SERIAL DEVICES



## Serial interfaces





# COMMUNICATION FROM SERIAL DEVICES



## Outline

- **Introduction to Serial Buses**
- **UART**
- **SPI**
- **I2C**



# SERIAL BUS INTERFACE MOTIVATIONS



- Motivation
  - Connect different systems together
    - Two embedded systems
    - A desktop and an embedded system
  - Connect different chips together in the same embedded system
    - MCU to peripheral
    - MCU to MCU
  - Without using a lot of I/O lines
    - I/O lines require I/O pads which cost \$\$\$ and size
    - I/O lines require PCB area which costs \$\$\$ and size
  - Often at relatively low data rates
  - But sometimes at higher data rates
- So, what are our options?
  - Universal Synchronous/Asynchronous Receiver Transmitter
  - Also known as USART (pronounced: “you-zart”)



# COMMUNICATION FROM SERIAL DEVICES



## Serial bus design space

- Number of wires required?
- Asynchronous or synchronous?
- How fast can it transfer data?
- Can it support more than two endpoints?
- Can it support more than one master (i.e. txn initiator)?
- How do we support flow control?
- How does it handle errors/noise?



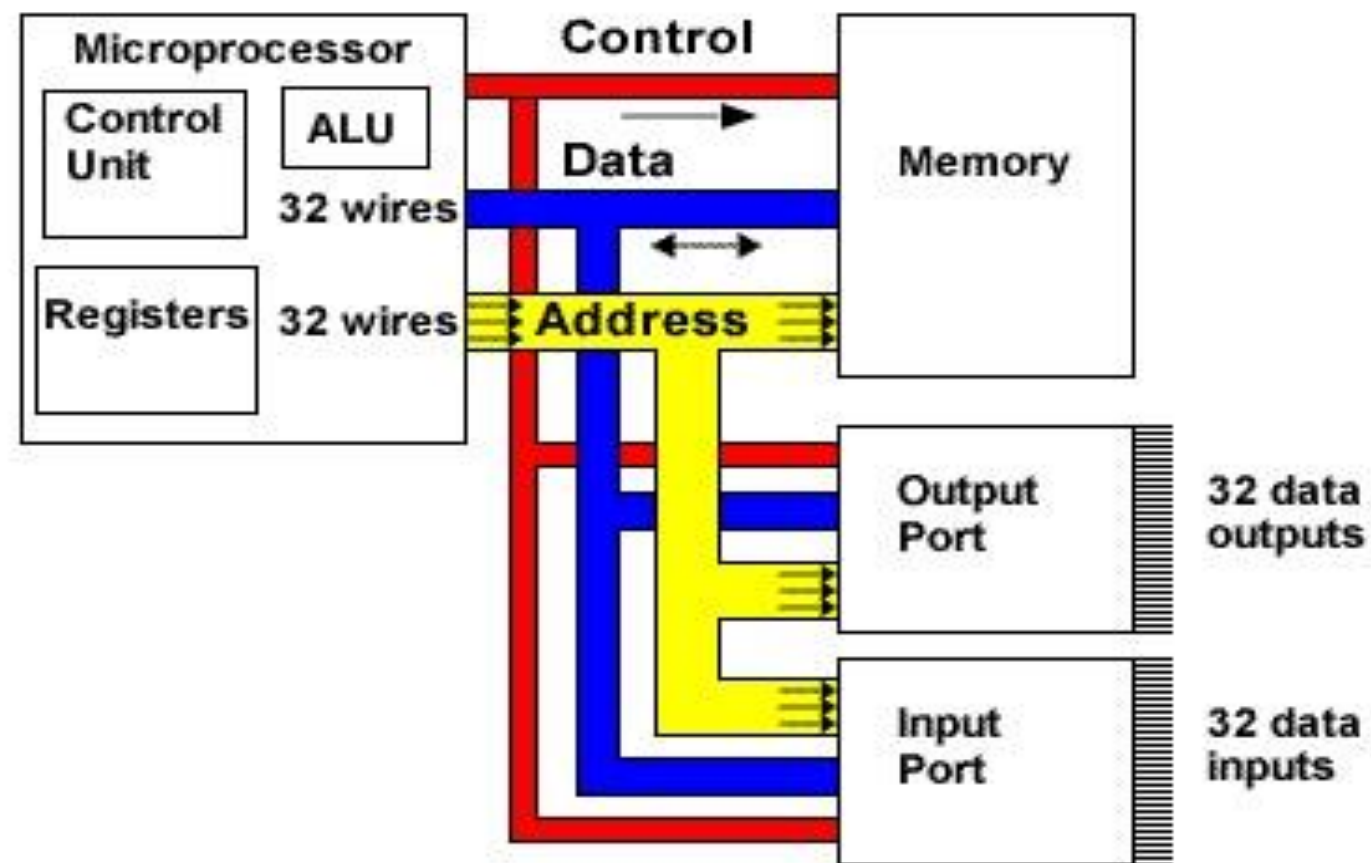


# COMMUNICATION FROM SERIAL DEVICES



## Fun with buses

- A multidrop bus (MDB) is a computer bus in which all components are connected to the same set of electrical wires. (from Wikipedia)
  - In the general case, a bus may have more than one device capable of driving it.
    - That is, it may be a “multi-master” bus as discussed earlier.



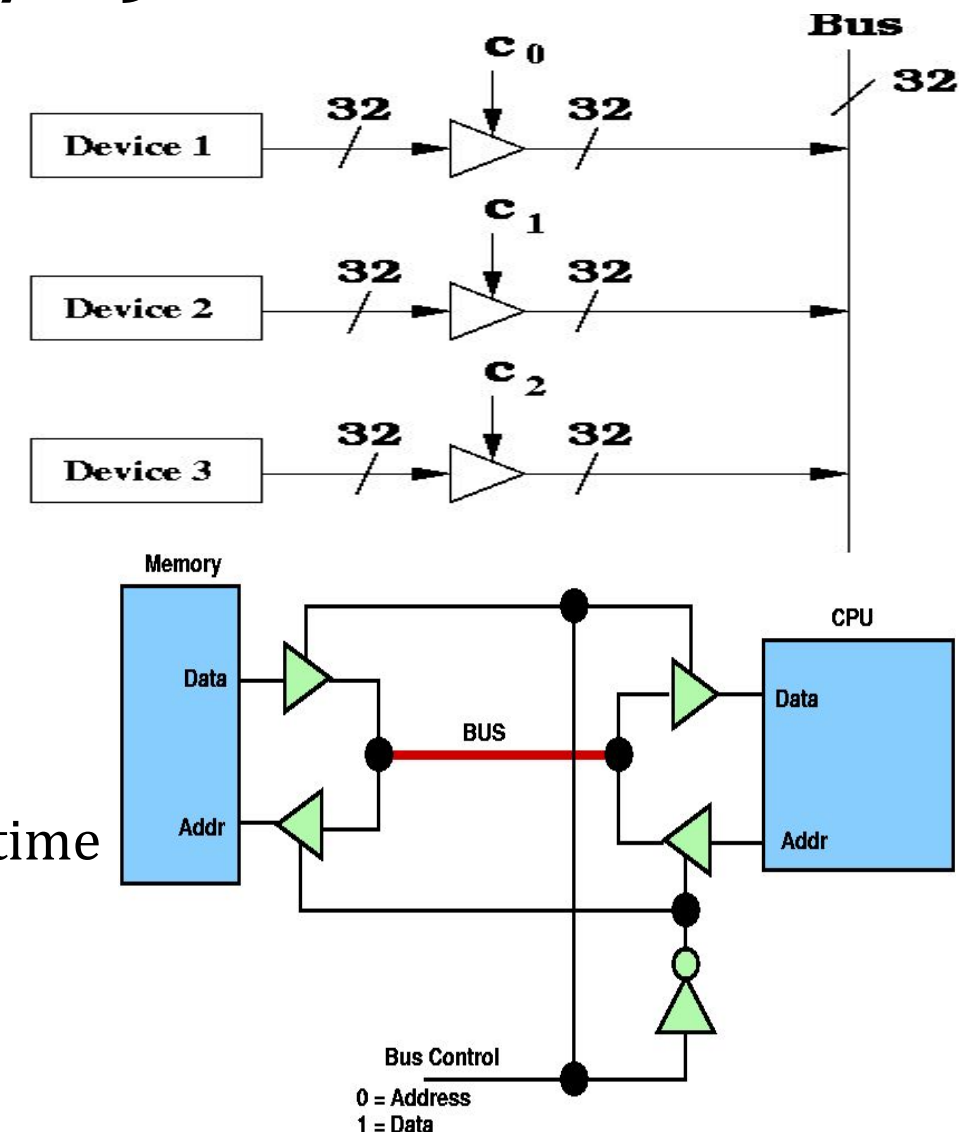


# COMMUNICATION FROM SERIAL DEVICES



## How can we handle multiple (potential) busdrivers? (1/3)

- Tri-state devices, just have one device drive at a time. Everyone can read though
  - Pros:
    - Very common, fairly fast, pin-efficient.
  - Cons:
    - Tri-state devices can be slow.
      - Especially drive-to-tristate?
    - Need to be sure two folks not driving at the same time
      - Let out the magic smoke.
  - Most common solution (at least historically)
    - Ethernet, PCI, etc.







## How can we handle multiple (potential) bus drivers? (2/3)

- MUX
  - Just have each device generate its data, and have a MUX select.
    - That's a LOT of pins.
      - Consider a 32-bit bus with 6 potential drivers.
        - » Draw the figure.
        - » How many pins needed for the MUX?
    - Not generally realistic for an “on-PCB” design as we’ll need an extra device (or a lot of pins on one device)
      - But reasonable on-chip
        - In fact AHB, APB do this.

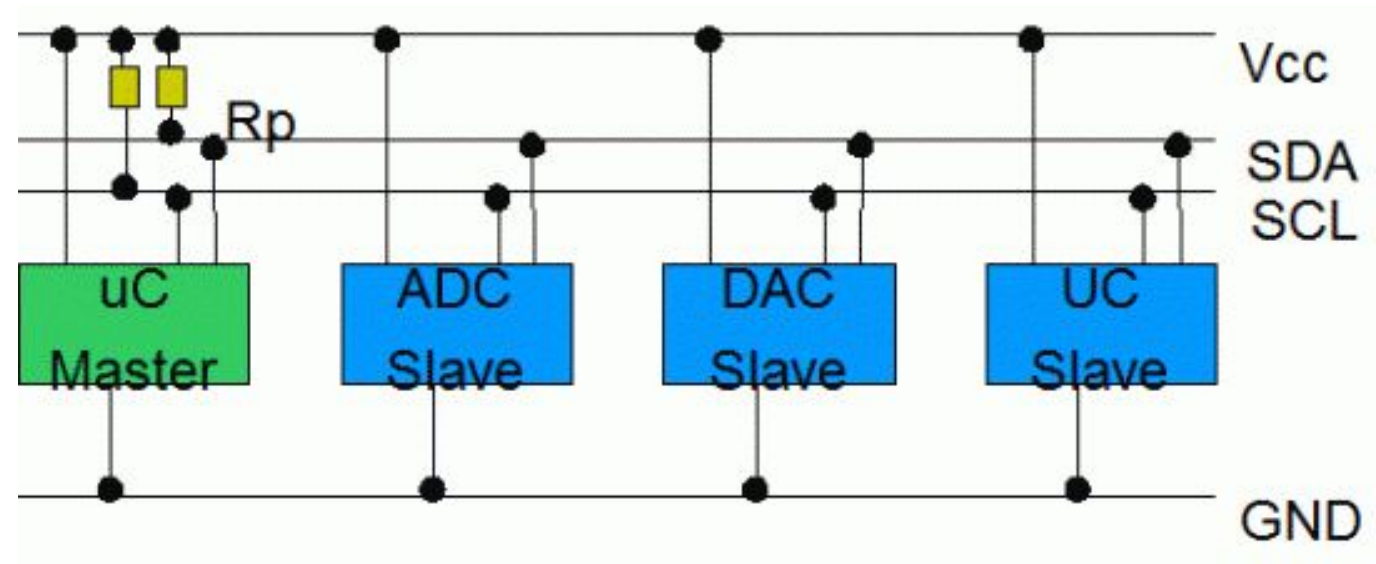
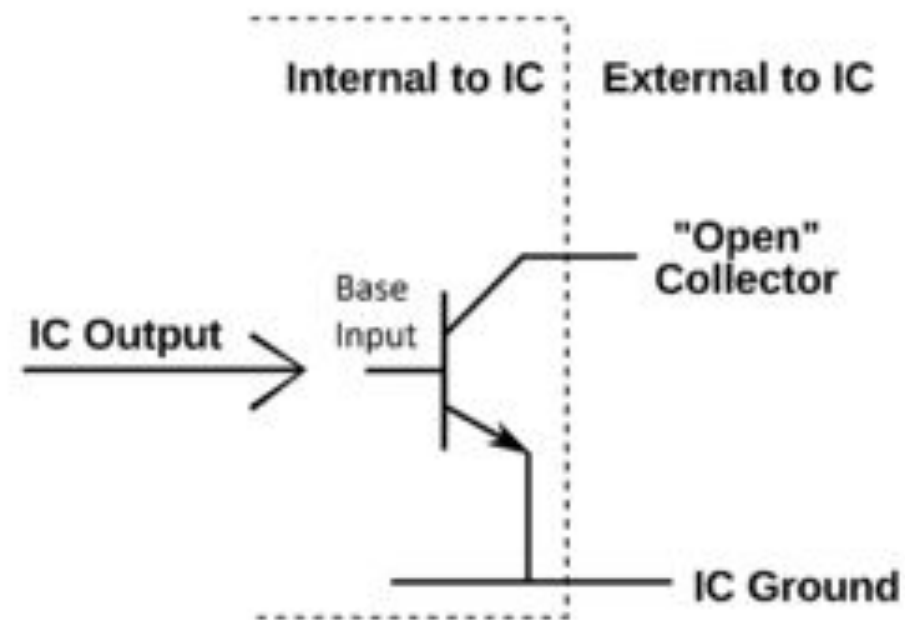


# COMMUNICATION FROM SERIAL DEVICES



## How can we handle multiple (potential) bus drivers? (3/3)

- “pull-up” aka “open collector” aka “wired OR”
  - Wire is pulled high by a resistor
  - If any device pulls the wire low, it goes low.
- Pros:
  - If two devices both drive the bus, it still works!
- Cons:
  - Rise-time is very slow.
  - Constant power drain.
- Used in I2C, CAN





## SUMMARY & THANK YOU