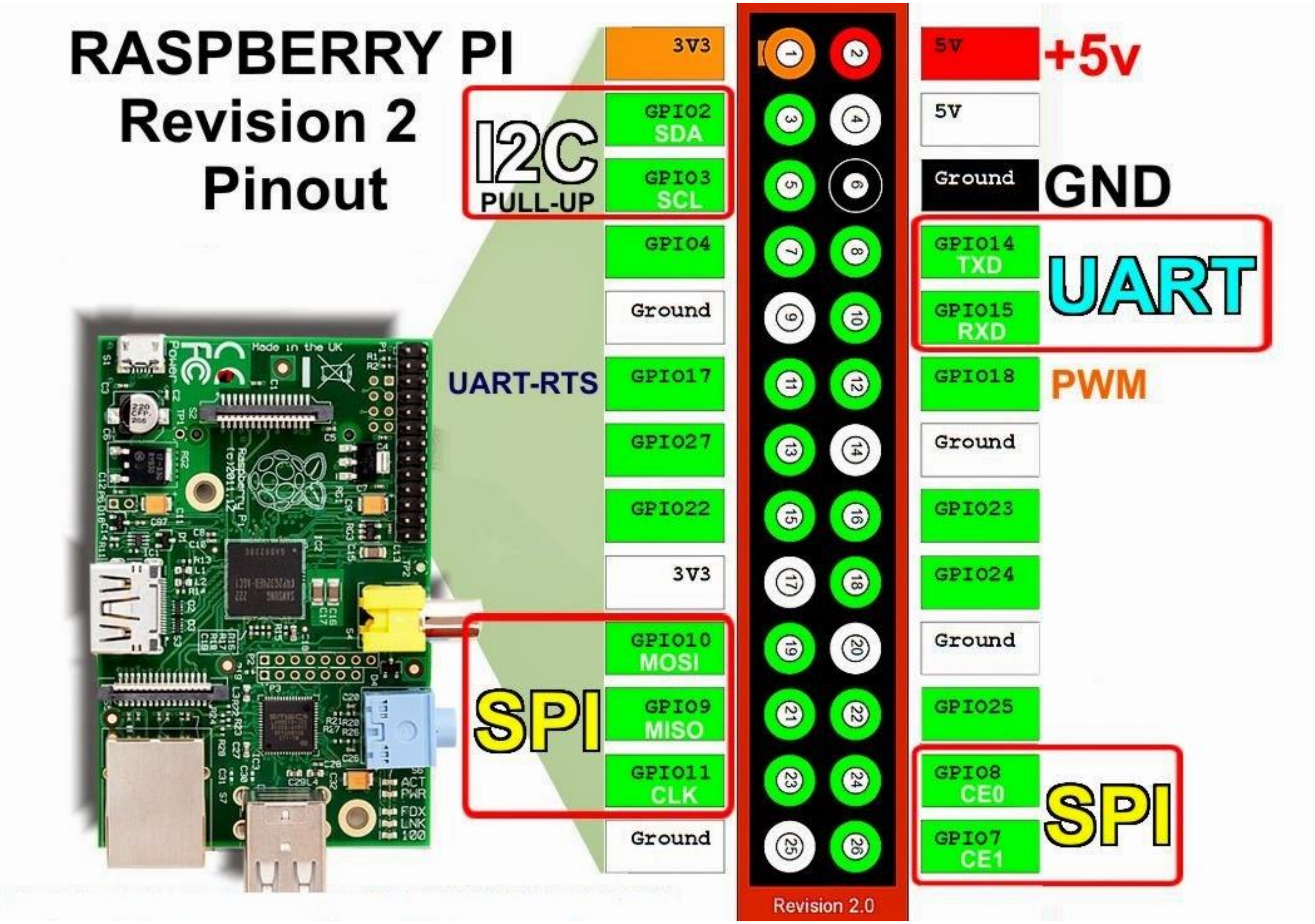
AMRITA
VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY

# I/O interfaces

## 22AIE211 Introduction To Communication & IoT

# External I/O interfaces

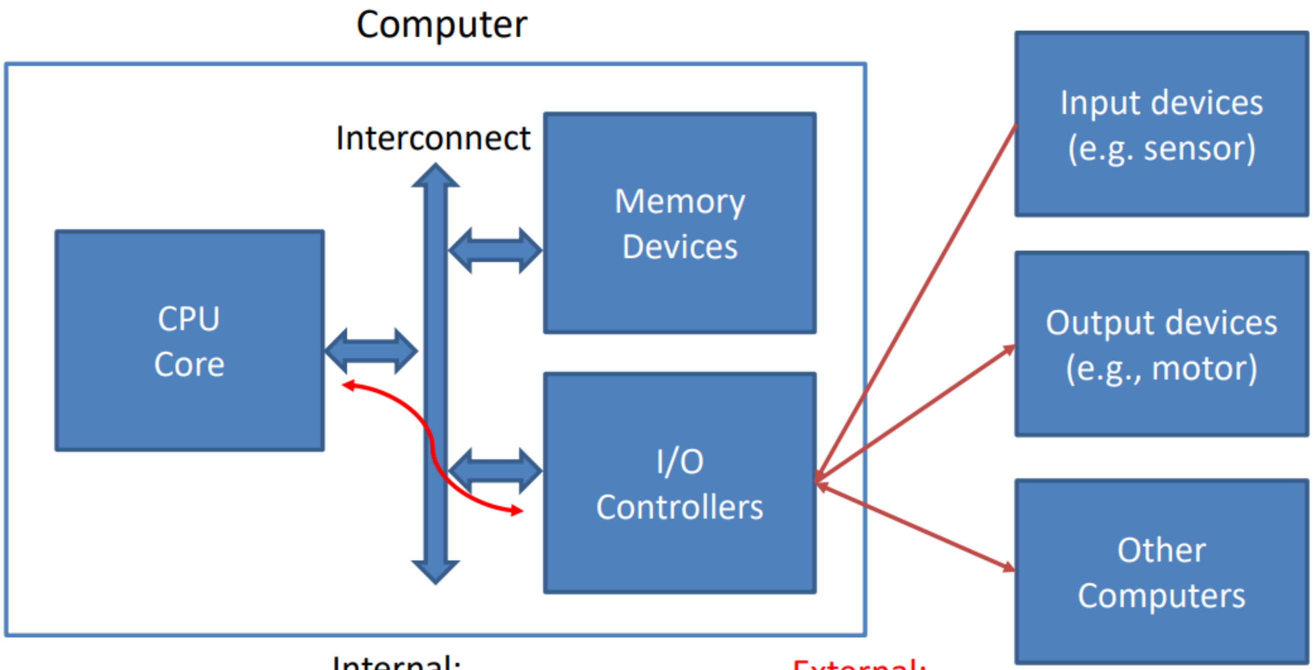


RASPBERRY PI
Revision 2
Pinout

| Group | Function | Pin | Pin | Function | Group |
|---|---|---|---|---|---|
| | 3V3 | 1 | 2 | 5V | +5v |
| I2C PULL-UP | GPIO2 SDA | 3 | 4 | 5V | |
| I2C PULL-UP | GPIO3 SCL | 5 | 6 | Ground | GND |
| | GPIO4 | 7 | 8 | GPIO14 TXD | UART |
| | Ground | 9 | 10 | GPIO15 RXD | UART |
| UART-RTS | GPIO17 | 11 | 12 | GPIO18 | PWM |
| | GPIO27 | 13 | 14 | Ground | |
| | GPIO22 | 15 | 16 | GPIO23 | |
| | 3V3 | 17 | 18 | GPIO24 | |
| SPI | GPIO10 MOSI | 19 | 20 | Ground | |
| SPI | GPIO9 MISO | 21 | 22 | GPIO25 | |
| SPI | GPIO11 CLK | 23 | 24 | GPIO8 CE0 | SPI |
| | Ground | 25 | 26 | GPIO7 CE1 | SPI |

Revision 2.0

# Input and Output (I/O) Interfaces

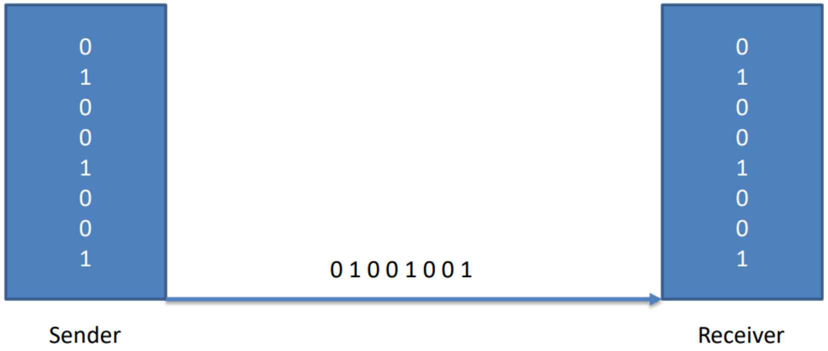


Internal:
AXI, TileLink, …

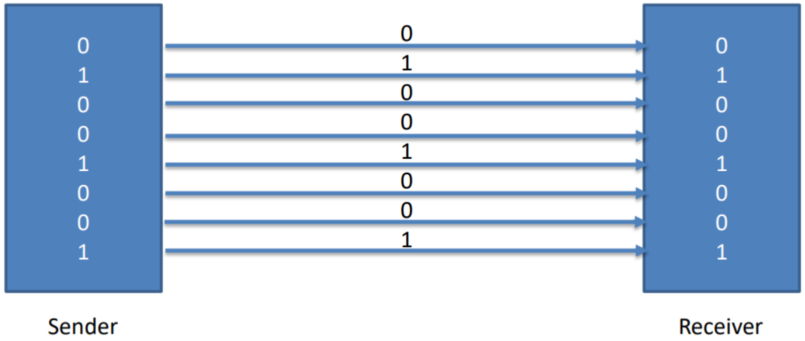
External:
GPIO, UART, USB, I2C,
SPI, CAN, ATA, PCI,
WiFi, Ethernet, …

# Serial vs. Parallel

- Serial communication
  - use single line



- Parallel communication
  - use multiple lines



# Serial vs. Parallel Interfaces

- Serial interfaces
  - RS-232: serial communication standard
  - USB: universal serial bus
  - I$^2$C: inter-integrated circuit
  - SPI: serial peripheral interface bus
  - SATA: serial ATA
  - …

I$^2$C

SATA

USB

RS-232

# Serial vs. Parallel Interfaces

- Parallel interfaces
  - Parallel ATA: advanced technology attachment
  - SCSI: small computer system interface
  - PCI: peripheral component interface
  - …

PCI

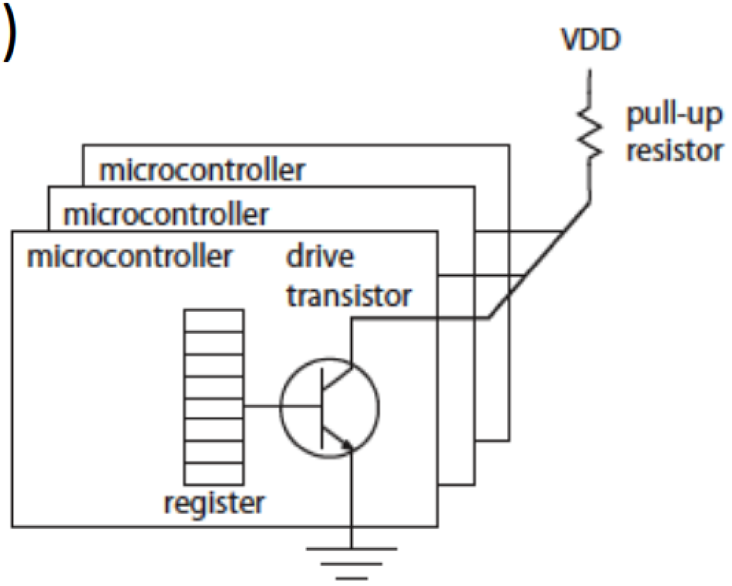
SCSI

PATA

Parallel port

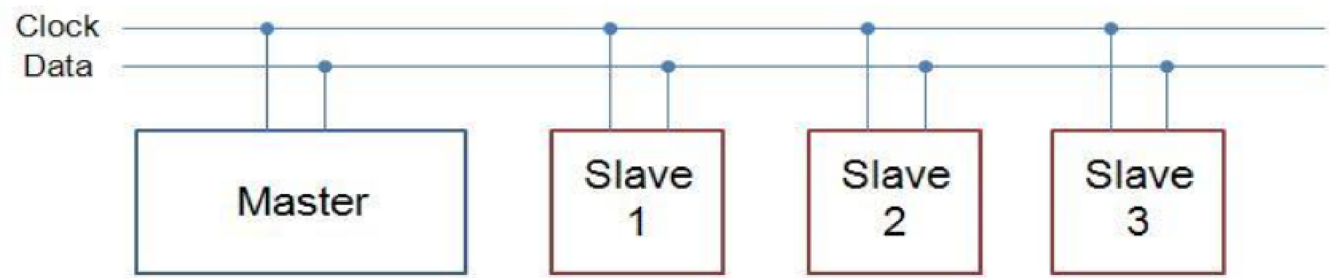
# General Purpose I/O (GPIO)

- Programmable digital input/output pins
- Use voltage levels to represent digital signals
  - 5V = logic 1  (for 5V CPUs)
  - 0V = logic 0
- Can be configured as
  - Input or output
- Useful to interact with external devices



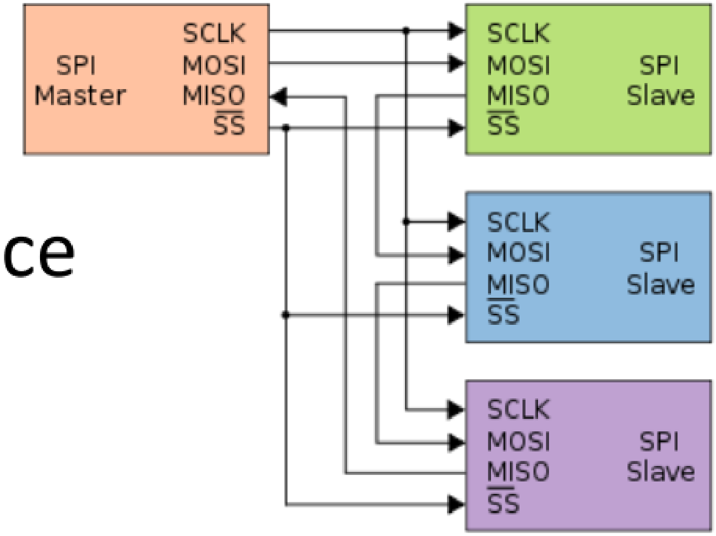
# I$^2$C

- Inter-integrated circuit protocol (I$^2$C), by NXP
- Synchronous serial communication protocol
- Use two wires (SCL - clock, SDA - data)
- Multi-master, multi-slave support
- Use 7bit ID. Connect up to 127 devices.
- Half duplex, relatively slow



# SPI

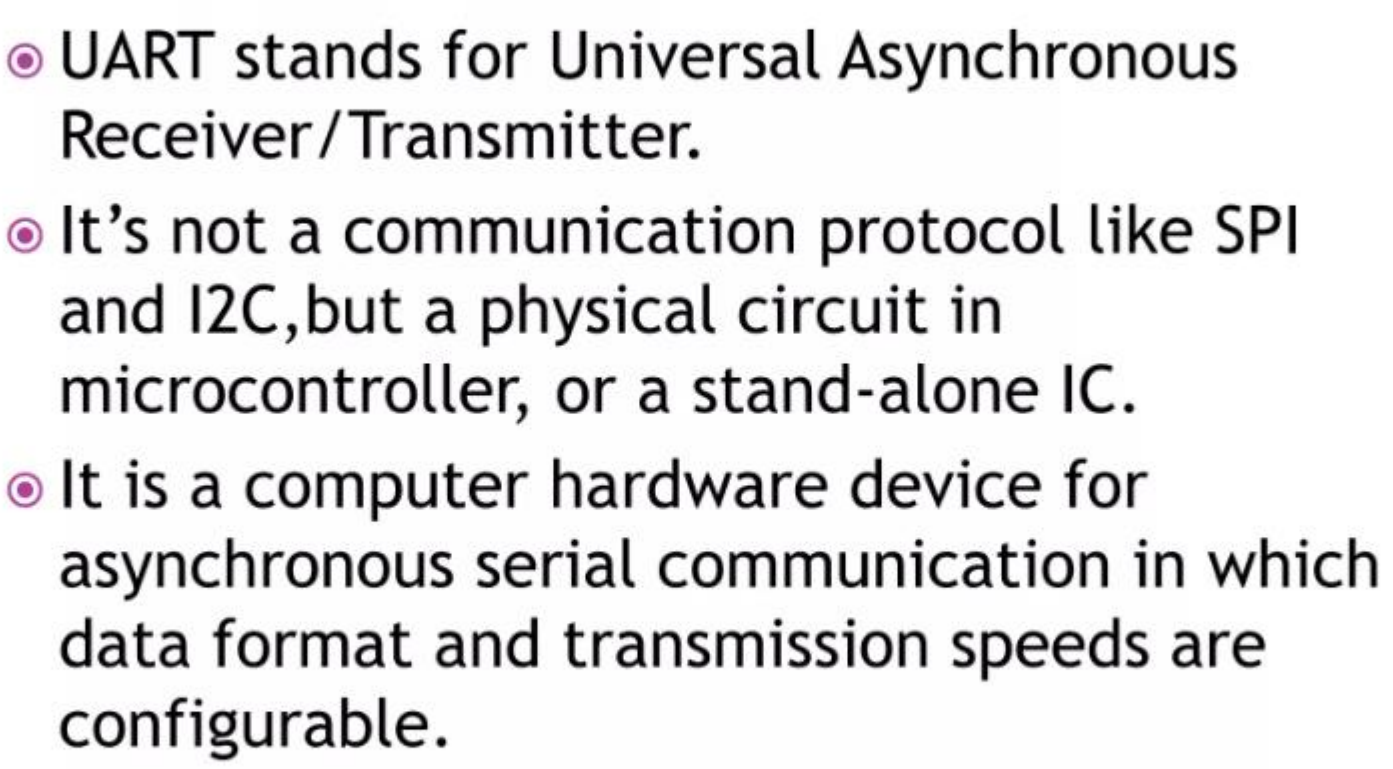
- Serial peripheral interface (SPI), by Motorola
- Synchronous serial communication protocol
- Use 4 lines, full-duplex, (relatively) fast
- Single master, multi-slave
- No start/stop bits
- Good for fast, short distance communication



# USB

- Universal serial bus
- One host, multiple devices, can form a tree
- Use two wires for single differential signal
  - Reduce noise caused by electromagnetic interference
- Device needs to process the protocol messages

# UART



- UART stands for Universal Asynchronous Receiver/Transmitter.
- It’s not a communication protocol like SPI and I2C,but a physical circuit in microcontroller, or a stand-alone IC.
- It is a computer hardware device for asynchronous serial communication in which data format and transmission speeds are configurable.

- A UART’s main purpose is to transmit and receive serial data.
- One of the best things about UART is that it only uses two wires to transmit data between devices
- In UART communication, two UARTs communicate directly with each other.
- The transmitting UART converts parallel data from a controlling device like a CPU into serial form, transmits it in serial to the receiving UART, which then converts the serial data back into parallel data for the receiving device.

- The UART that is going to transmit data receives the data from a data bus. The data bus is used to send data to the UART by another device like a CPU, memory, or microcontroller. Data is transferred from the data bus to the transmitting UART in parallel form. After the transmitting UART gets the parallel data from the data bus, it adds a start bit, a parity bit, and a stop bit, creating the data packet.

- Next, the data packet is output serially, bit by bit at the Tx pin. The receiving UART reads the data packet bit by bit at its Rx pin. The receiving UART then converts the data back into parallel form and removes the start bit, parity bit, and stop bits. Finally, the receiving UART transfers the data packet in parallel to the data bus on the receiving end.

# Summary

- USB
- GPIO
- I2C
- SPI
- UART