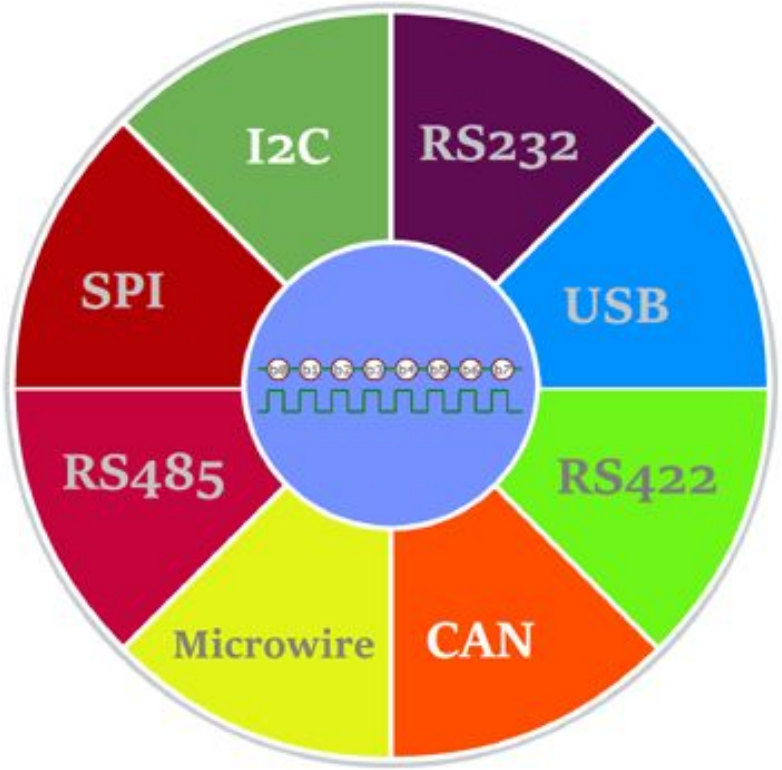
CSE360-Computer Interfacing
BRAC University



Serial Communication Protocols

Communication Protocol

BRAC UNIVERSITY
Inspiring Excellence

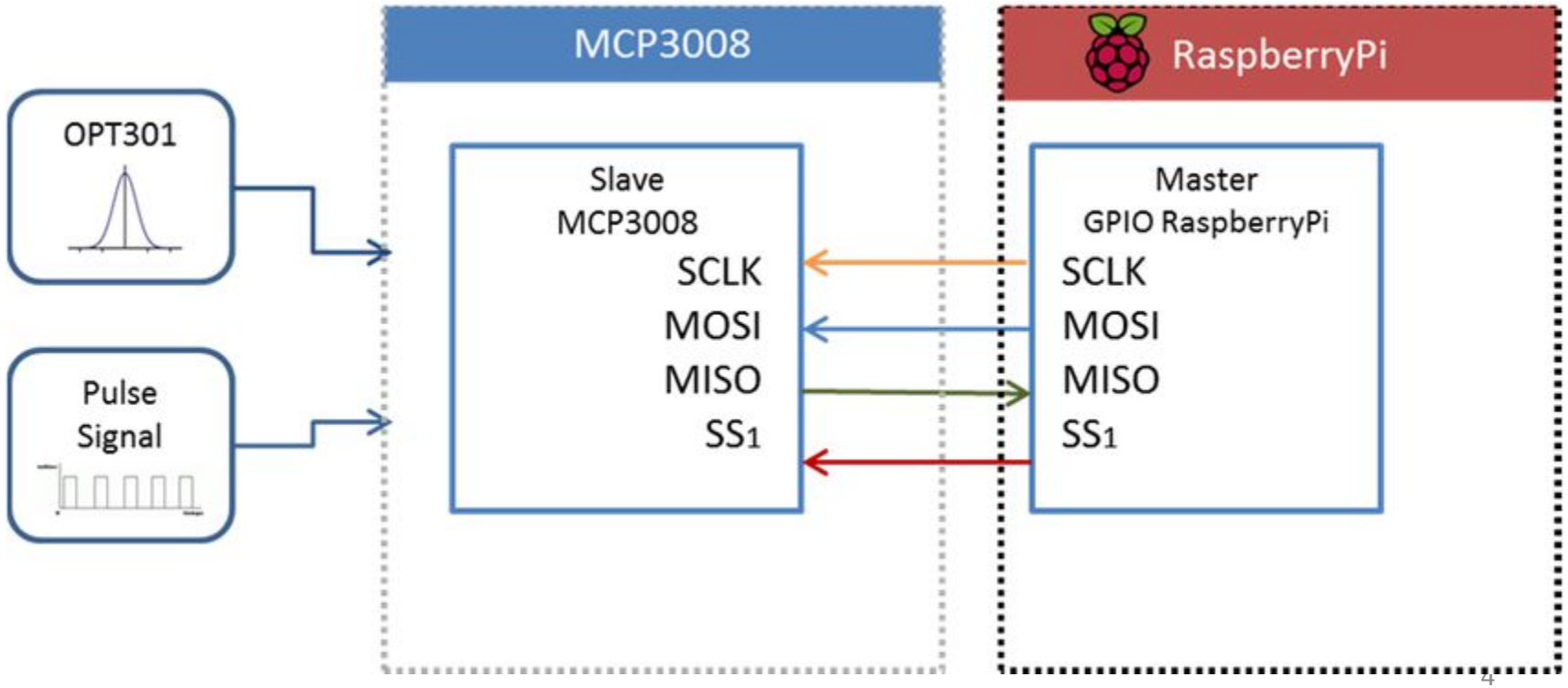
- According to Wikipedia, “A **communication protocol** is a system of rules that allow two or more entities of a Communication system to transmit information via any kind of variation of a physical quantity. The protocol defines the rules, syntax, semantics and synchronization of communication and possible error recovery methods. Protocols may be implemented by hardware, software, or a combination of both.”
- Examples: SPI, I2C, UART, USB etc.

Serial Peripheral Interface (SPI)

- SPI stands for Serial Peripheral Interface
- Used for moving data simply and quickly from one device to another
- Master-Slave(Multiple Slaves, Single Master)
- Synchronous transmission
- Synchronous- Data clocked with Clock Signal
- Data Rate-10mbps
- Full Duplex Protocol
- Low power than I2C (no need of Pull ups)
- Supports Single master and multiple slaves
- No hardware slave acknowledgement

Master – Slave Protocol

- Simple SPI Protocol Specifies 4 Signal Wires
  1. Master Out Slave In (MOSI)
  2. Master In Slave Out (MISO)
  3. Serial Clock (SCLK)
  4. Slave Select (SS)

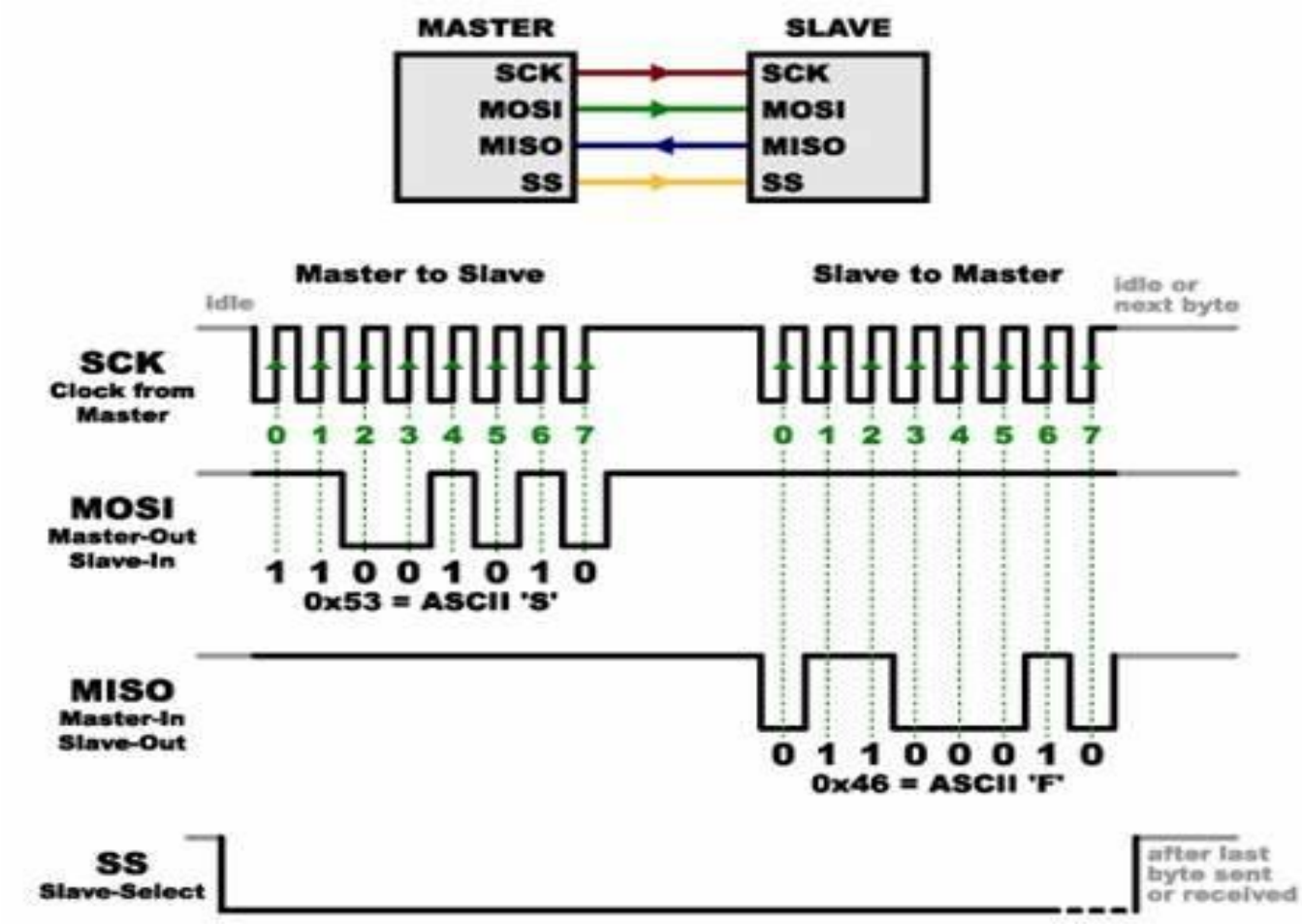


Master – Slave Protocol

- SPI is a Master-Slave protocol
- The Master device controls the clock (SCK)
- No data is transferred unless a clock signal is present
- All slaves are controlled by the master clock
- The slave devices may not manipulate the clock
- Master Set Slave Select low
- Master Generates Clock
-

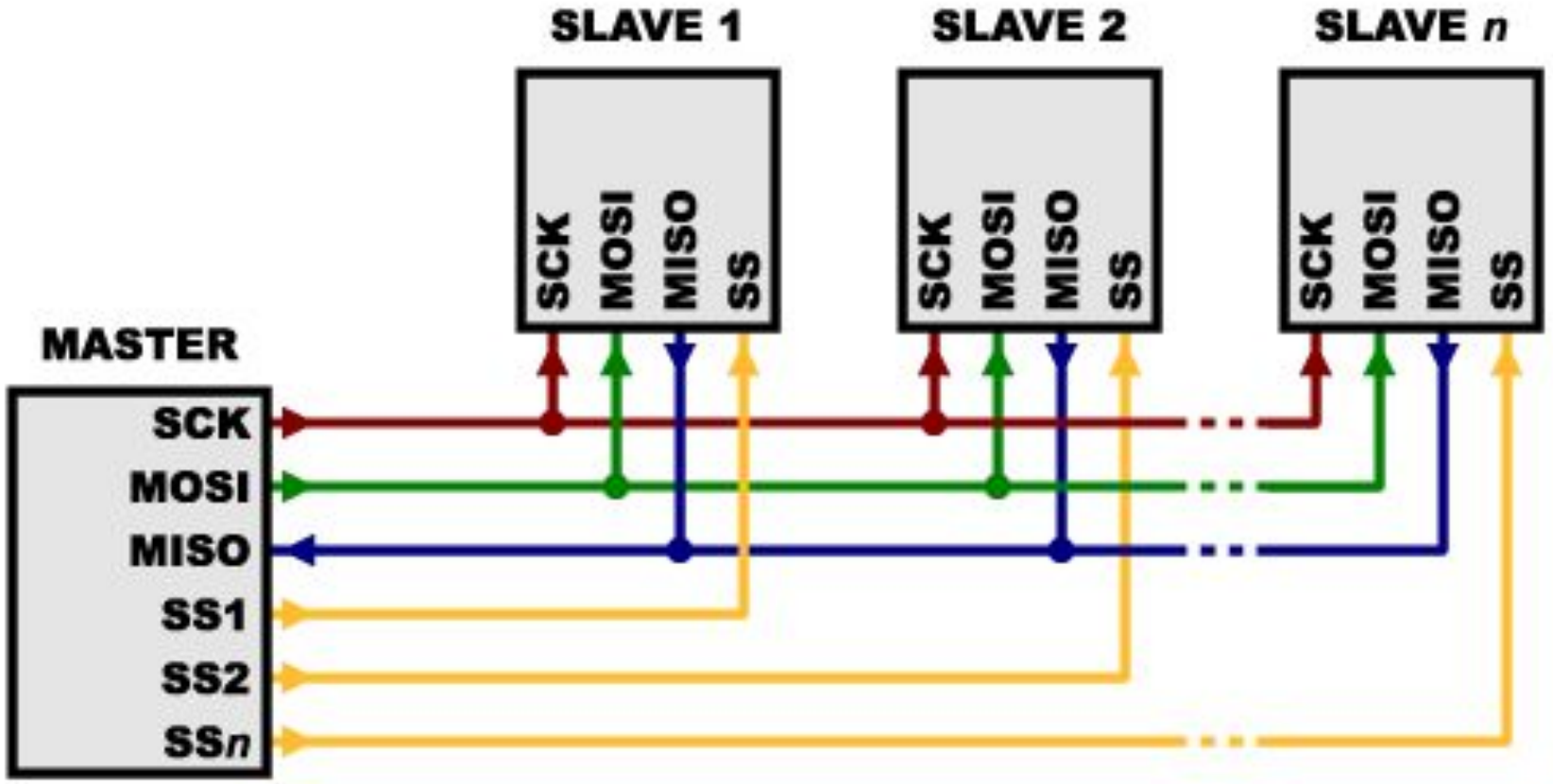
How SPI Works

Single master – single slave



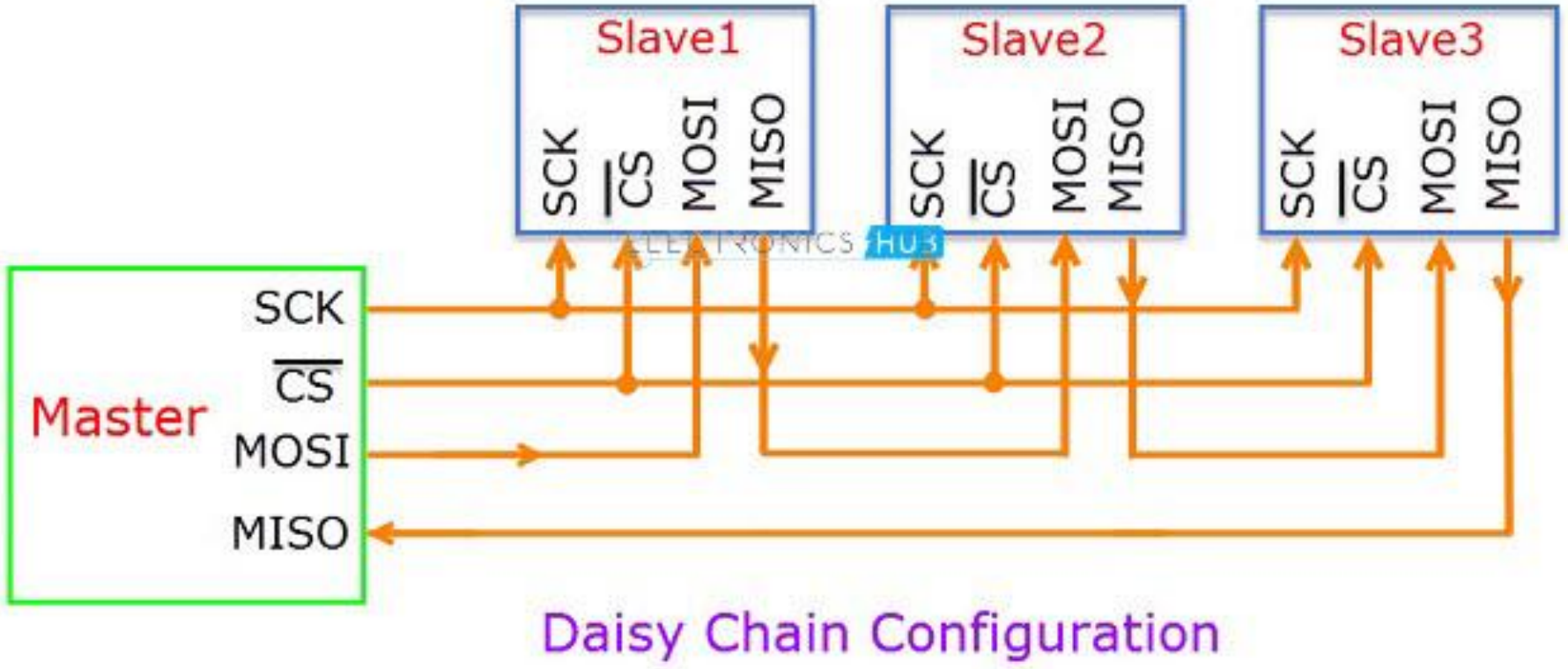
How SPI Works

Single master – Multiple slaves



How SPI Works

Single master – Multiple slaves



Daisy Chain Configuration

Although the system becomes more simple, the time taken for the data to travel to the destined node increases.

How SPI Works

In summary the steps of **SPI:**

1. The master outputs the clock signal

2. The master switches the SS/CS pin to a low voltage state, which activates the slave

3. The master sends the data one bit at a time to the slave along the MOSI line. The slave reads the bits as they are received

4. If a response is needed, the slave returns data one bit at a time to the master along the MISO line. The master reads the bits as they are received

SPI

BRAC UNIVERSITY
Inspiring Excellence

Advantages:

- Full duplex protocol. Separate MISO and MOSI lines, so data can be sent and received at the same time
- No start and stop bits, so the data can be streamed continuously without interruption
- No complicated slave addressing system like I2C
- Higher data transfer rate than I2C (almost twice as fast)
- Not Limited to 8 bit words in case of bit transferring
- Arbitrary choice of message size, content and Purpose
- Low Power

SPI

BRAC UNIVERSITY
Inspiring Excellence

Disadvantages:

- Requires more pins than I2C
- No hardware flow control
- No Slave Acknowledgement
- Multi Master Difficult to Implement
- No form of error checking like the parity bit in UART
- It usually requires separate SS lines to each slave, which can be problematic if numerous slaves are needed.

SPI

BRAC UNIVERSITY
Inspiring Excellence

SPI Peripherals:

- Converters (ADC, DAC)
- Memories (EEPROM, RAM's, Flash)
- Sensors (Temperature, Humidity, Pressure)
- Real Time Clocks
- Misc.- Potentiometers, LCD controllers, UART's, USB controller, CAN controller, amplifiers

.

UART

UART

Introduction:

- Universal Asynchronous Receiver/Transmitter
- It also called Serial Communication Interface(SCI)
- Full-duplex communication.
- Asynchronous communication.
- Compatible with PC .
- The Standard bit rates are: 100, 200, 300, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200 bps.
- Example: Connecting GPS modules, Bluetooth modules, and RFID card reader modules to Raspberry Pi, Arduino, or other microcontrollers.

UART

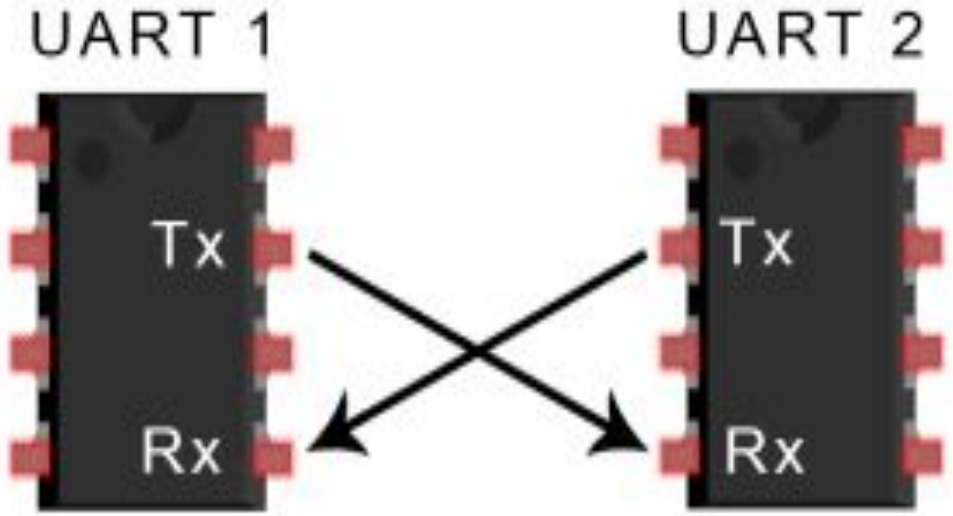
- A UART's main purpose is to transmit and receive serial data.
- One of the best things about UART is that it only uses two wires to transmit data between devices

Why UART?

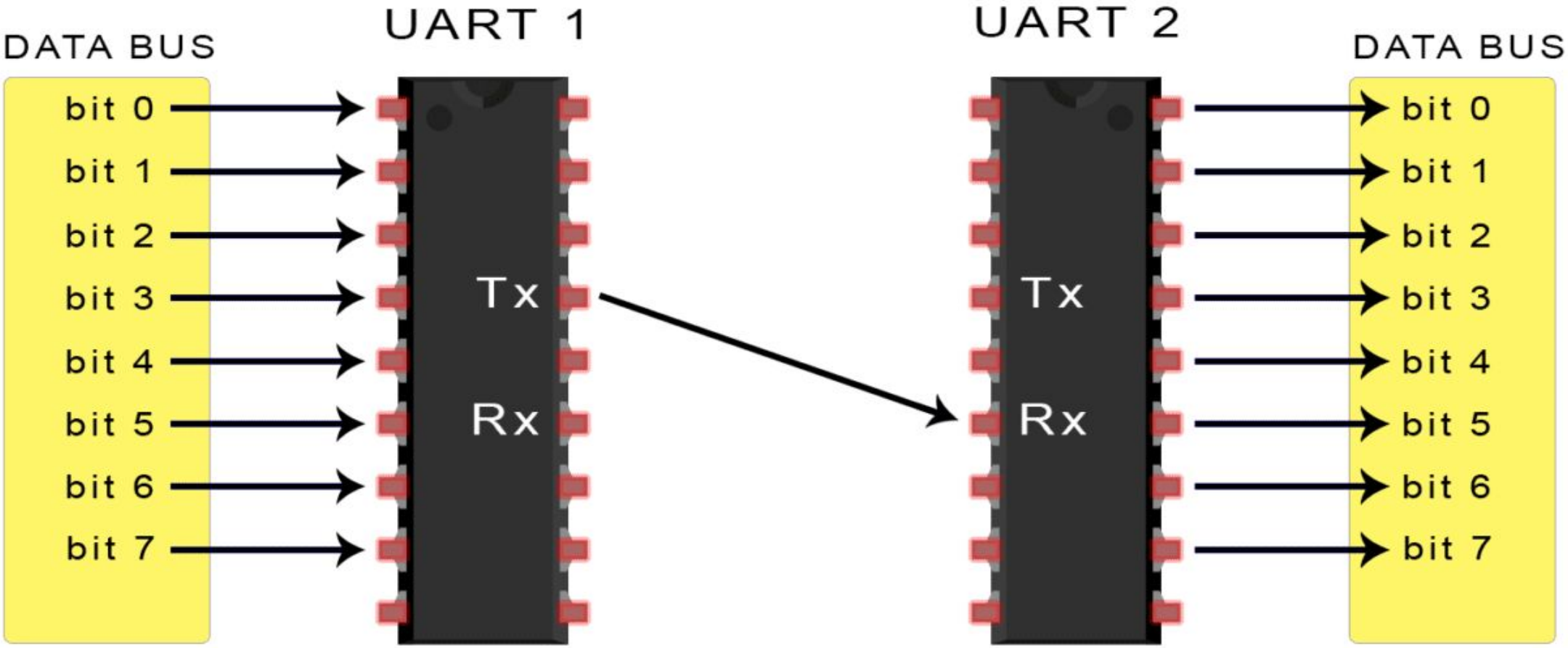
- A UART may be used when:-
  - High speed is not required
  - An inexpensive communication link between two devices is required.
- UART communication is very cheap:-
  - Single wire for each direction(and ground wire).
  - Simple hardware.

UART

- In UART communication, two UARTs communicate directly with each other. Data flows from the Tx pin of the transmitting UART to the Rx pin of the receiving UART:



UART


DATA BUS
bit 0
bit 1
bit 2
bit 3
bit 4
bit 5
bit 6
bit 7
UART 1
Tx
Rx
UART 2
Tx
Rx
DATA BUS
bit 0
bit 1
bit 2
bit 3
bit 4
bit 5
bit 6
bit 7

UART

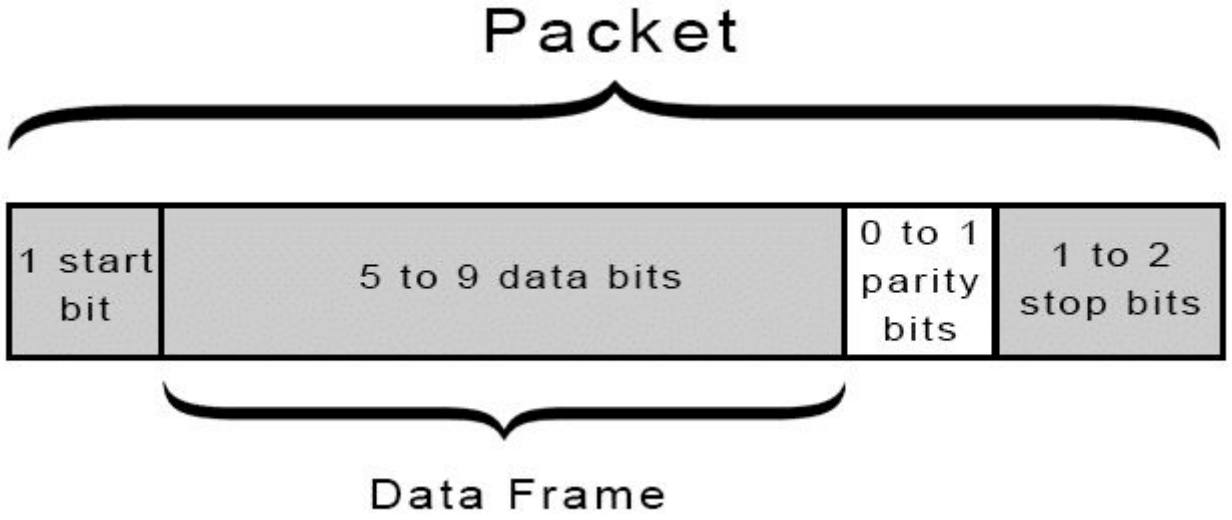
- The transmitting UART converts parallel data from a controlling device like a CPU into serial form, transmits it in serial to the receiving UART, which then converts the serial data back into parallel data for the receiving device.

- 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.

UART

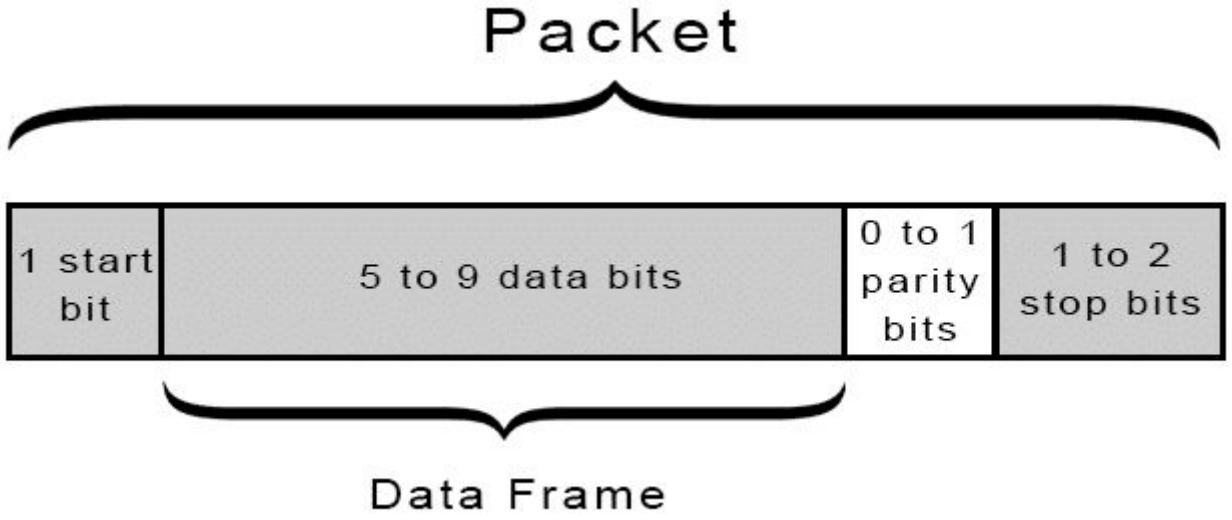
- Next, the data packet is output serially, bit by bit at the Tx pin. The receiving UART reads the data packet bit by bit at its Rx pin. The receiving UART then converts the data back into parallel form and removes the start bit, parity bit, and stop bits. Finally, the receiving UART transfers the data packet in parallel to the data bus on the receiving end.
- When the receiving UART detects a start bit, it starts to read the incoming bits at a specific frequency known as the baud rate. Baud rate is a measure of the speed of data transfer, expressed in bits per second (bps).
- Both UARTs must operate at about the same baud rate. The baud rate between the transmitting and receiving UARTs can only differ by about 10% before the timing of bits gets too far off.

How UART Communicate?



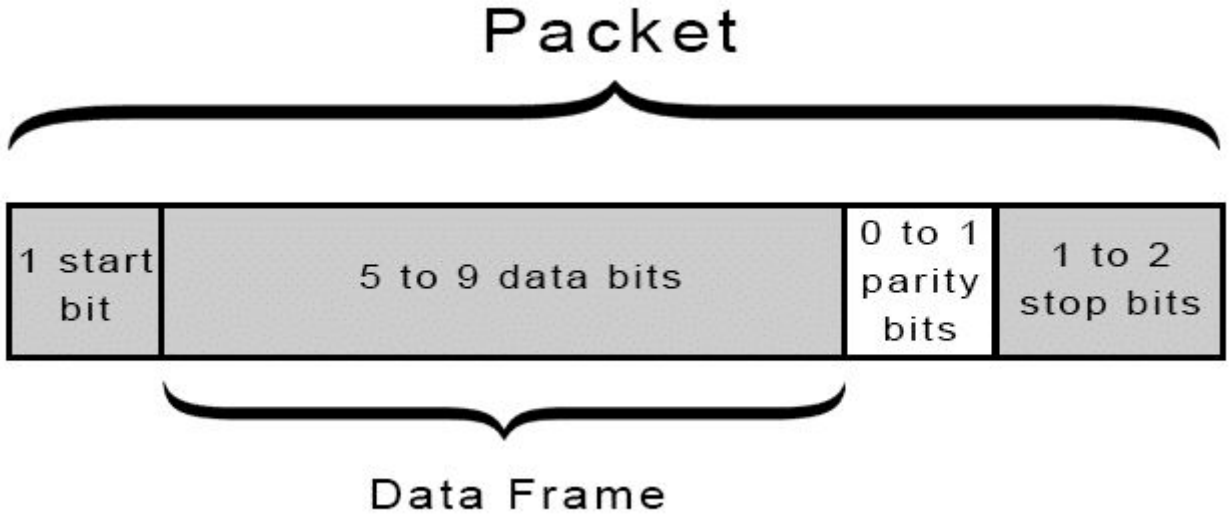
- **START BIT:** The UART data transmission line is normally held at a high voltage level when it's not transmitting data.
  - To start the transfer of data, the transmitting UART pulls the transmission line from high to low for one clock cycle.
  - When the receiving UART detects the high to low voltage transition, it begins reading the bits in the data frame at the frequency of the baud rate.

How UART Communicate?



DATA FRAME: The data frame contains the actual data being transferred. It can be 5 bits up to 8 bits long if a parity bit is used.
If no parity bit is used, the data frame can be 9 bits long. In most cases, the data is sent with the least significant bit first.

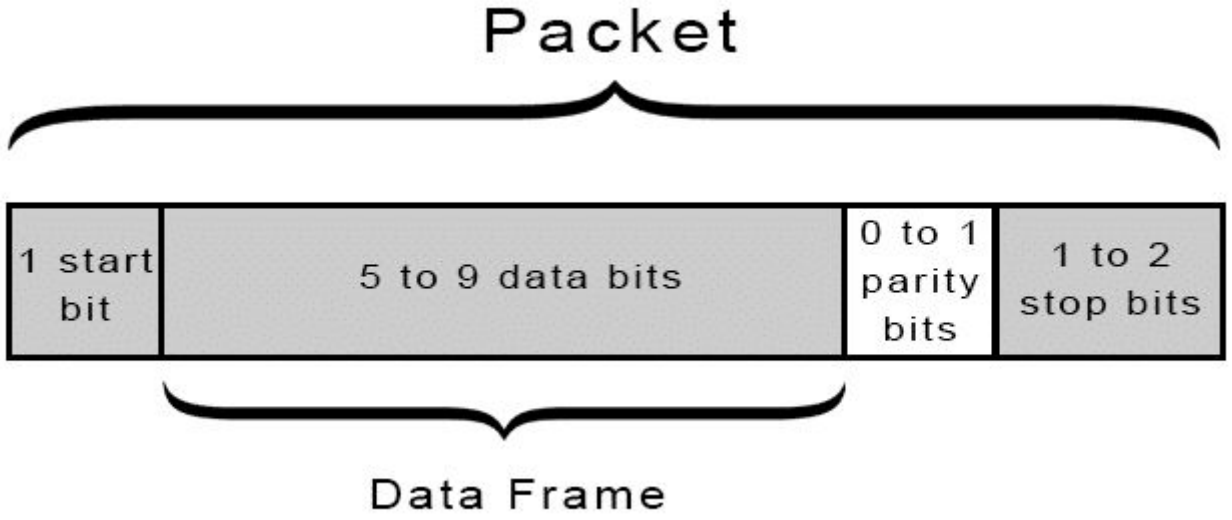
How UART Communicate?



PARITY: Parity describes the evenness or oddness of a number.

- The parity bit is a way for the receiving UART to tell if any data has changed during transmission(Bits can be changed by electromagnetic radiation, mismatched baud rates, or long distance data transfers).
- After the receiving UART reads the data frame, it counts the number of bits with a value of 1 and checks if the total is an even or odd number.
- If the parity bit is a 0 (even parity), the 1 bits in the data frame should total to an even number. If the parity bit is a 1 (odd parity).

How UART Communicate?

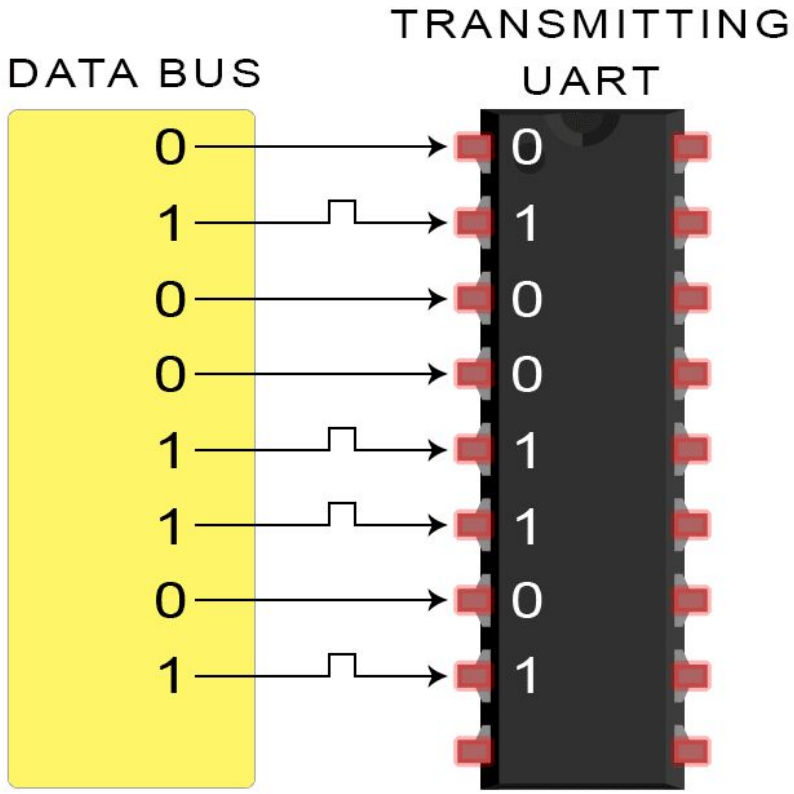


- **STOP BITS:** To signal the end of the data packet, the sending UART drives the data transmission line from a low voltage to a high voltage for at least two bit durations

STEPS OF UART TRANSMISSION

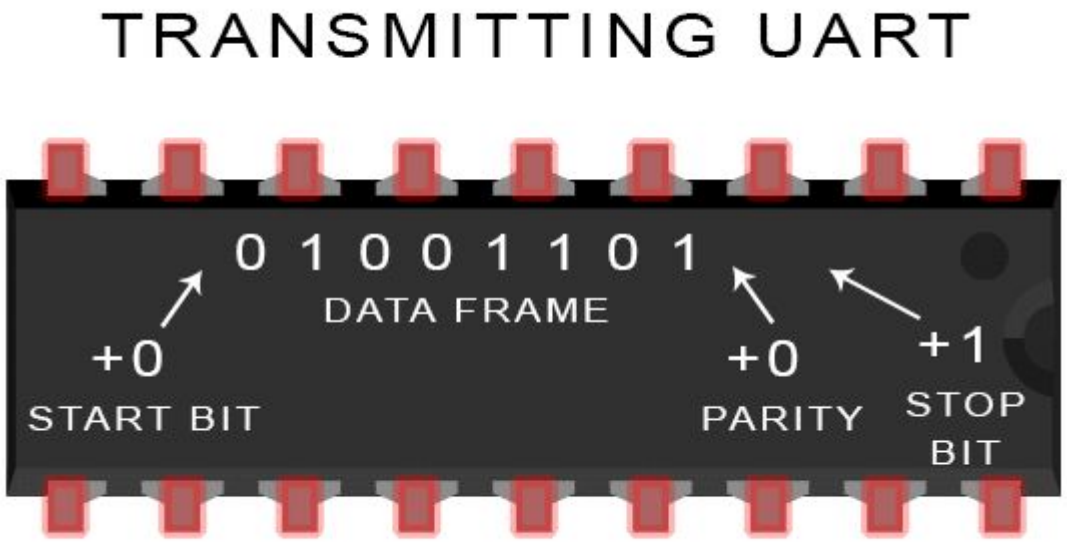
BRAC UNIVERSITY
Inspiring Excellence

- The transmitting UART receives data in parallel from the data bus.



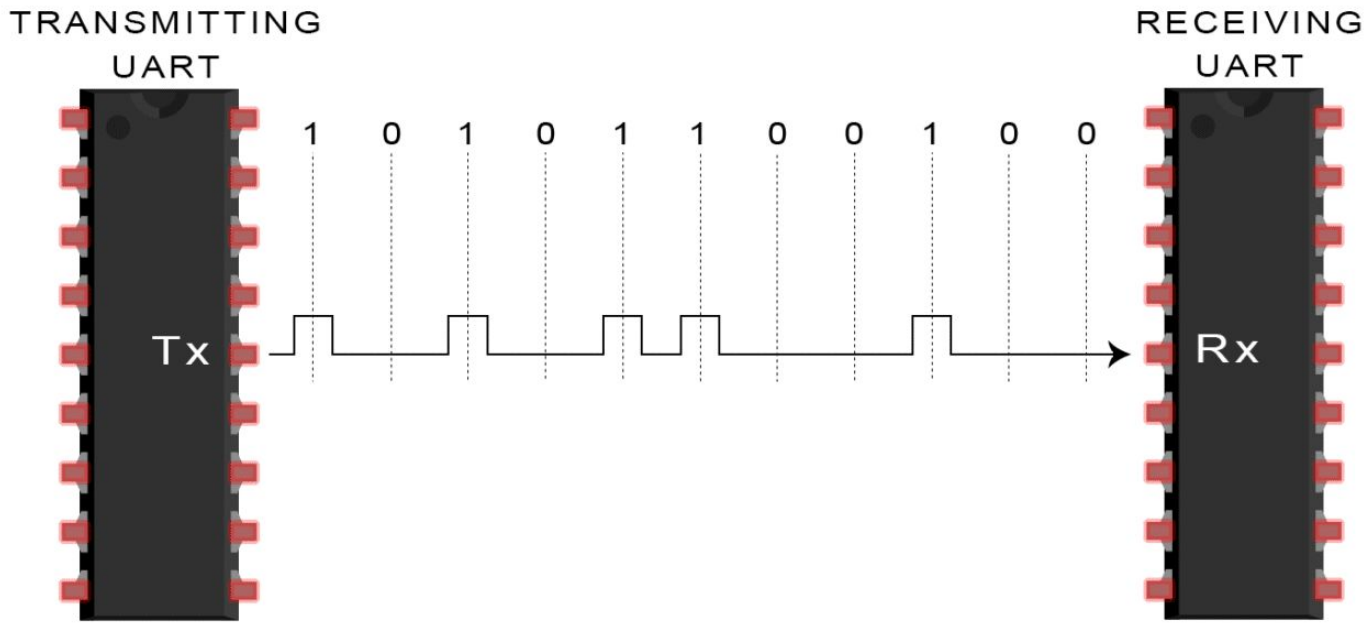
STEPS OF UART TRANSMISSION

- The transmitting UART adds the start bit, parity bit, and the stop bit(s) to the data frame.



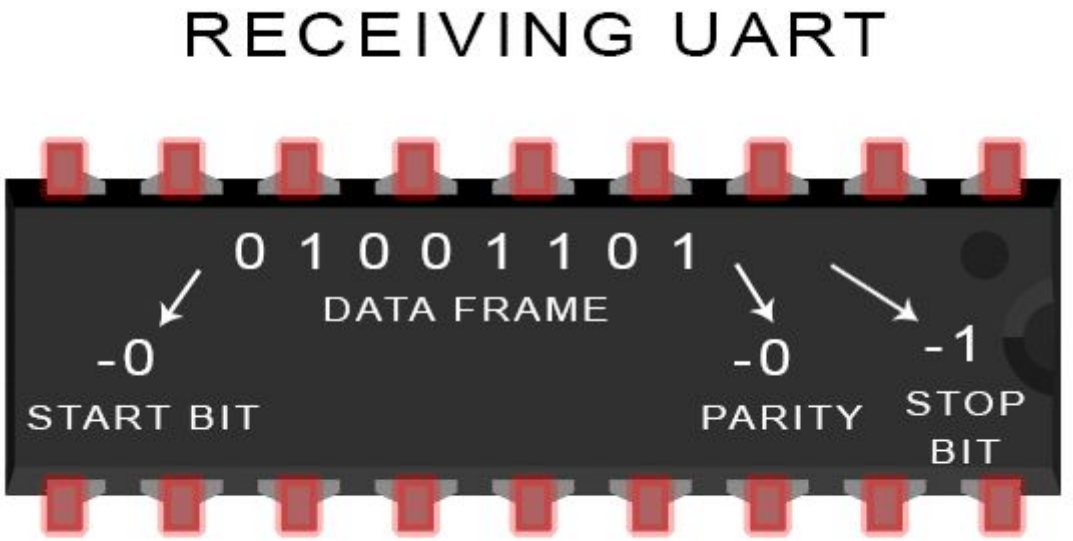
STEPS OF UART TRANSMISSION

- The entire packet is sent serially from the transmitting UART to the receiving UART. The receiving UART samples the data line at the pre-configured baud rate.



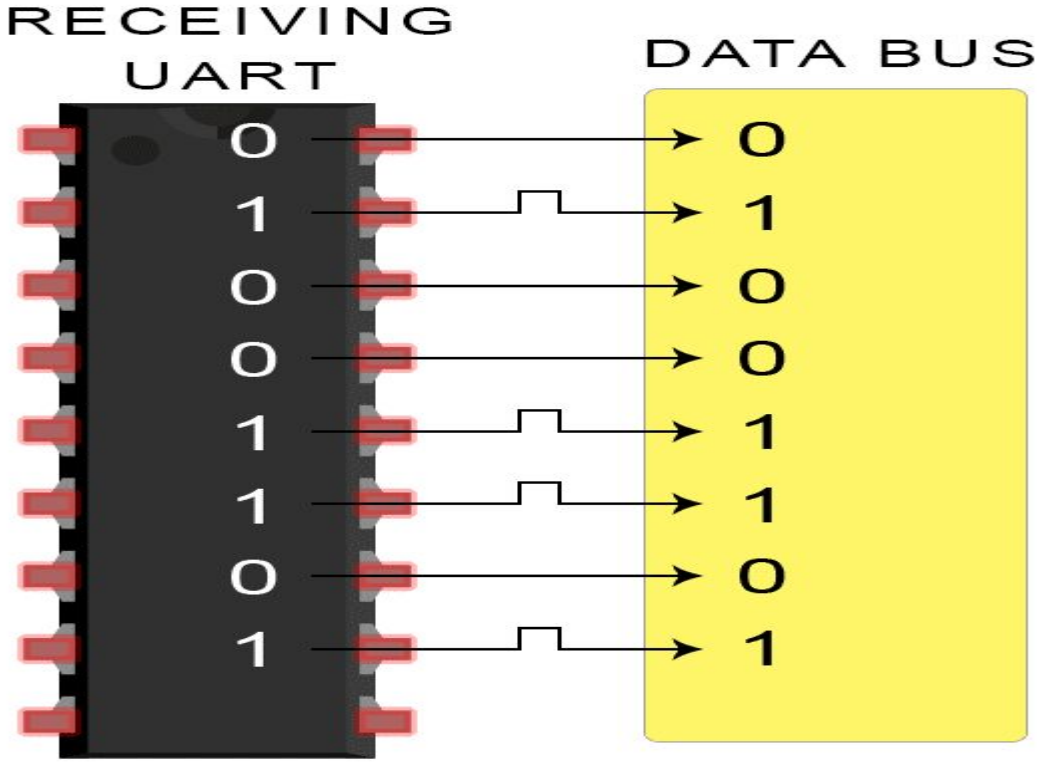
STEPS OF UART TRANSMISSION

- The receiving UART discards the start bit, parity bit, and stop bit from the data frame.



STEPS OF UART TRANSMISSION

- The receiving UART converts the serial data back into parallel and transfers it to the data bus on the receiving end.



UART

BRAC UNIVERSITY
Inspiring Excellence

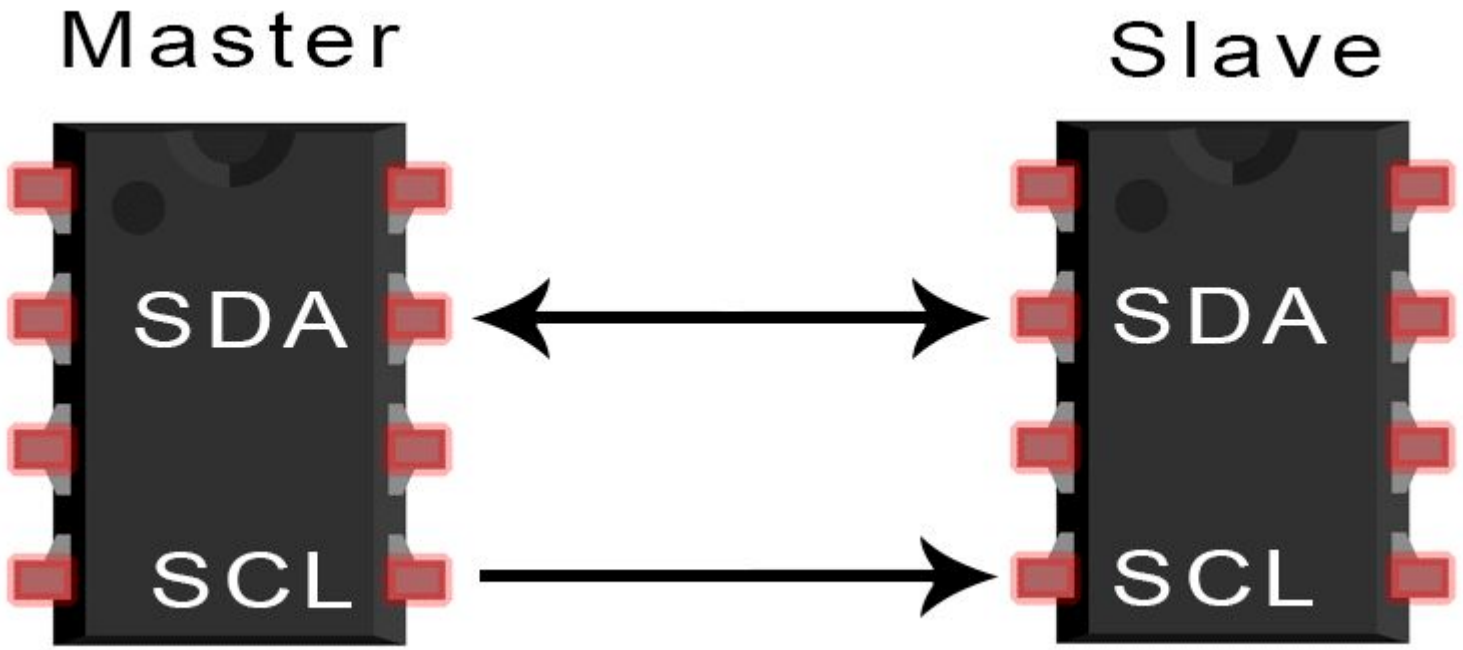
ADVANTAGES

- Only uses two wires
- No clock signal is necessary
- Has a parity bit to allow for error checking
- The structure of the data packet can be changed as long as both sides are set up for it
- Well documented and widely used method

DISADVANTAGES

- The size of the data frame is limited to a maximum of 9 bits
- Doesn’t support multiple slave or multiple master systems

(Inter-Integrated Circuit) I2C or I^2C Bus



I2C Bus

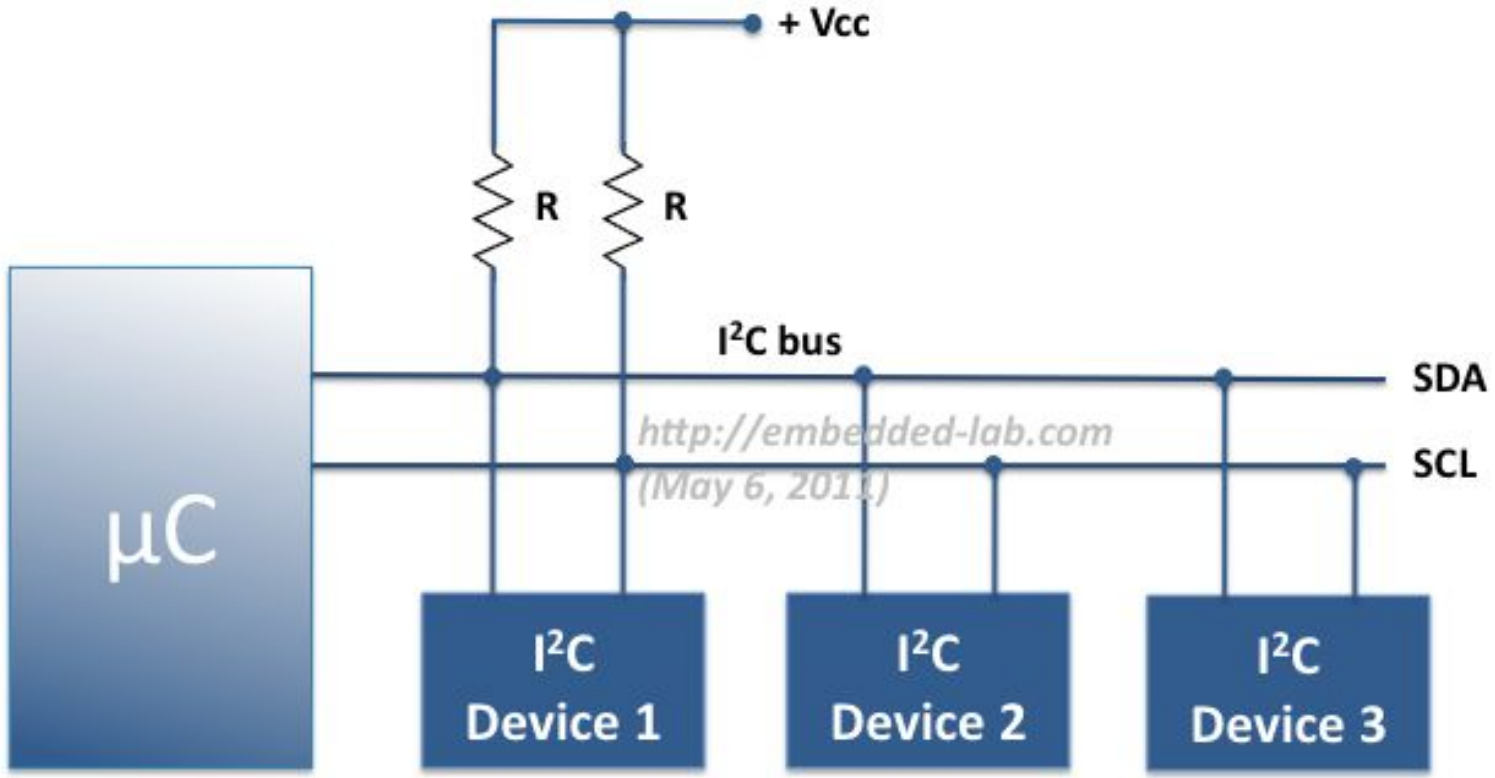
BRAC UNIVERSITY
Inspiring Excellence

- I2C is not only used on single boards but also to connect components which are linked via cable. Simplicity and flexibility are key characteristics that make this bus attractive to many applications.

Most significant features include:

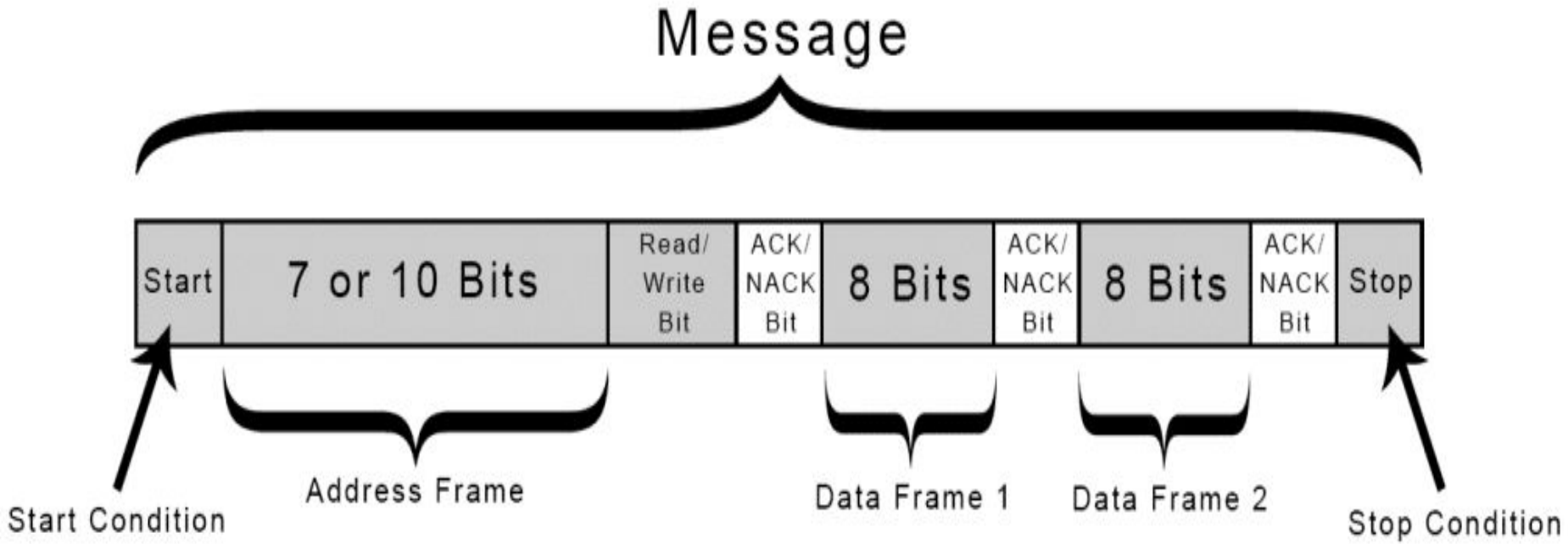
- Only two bus lines are required: SDA (Serial Data), SCL (Serial Clock)
- No strict baud rate requirements like for instance with RS232, the master generates a bus clock
- Simple master/slave relationships exist between all components Each device connected to the bus is software-addressable by a unique address
- I2C is a true multi-master bus providing arbitration and collision detection

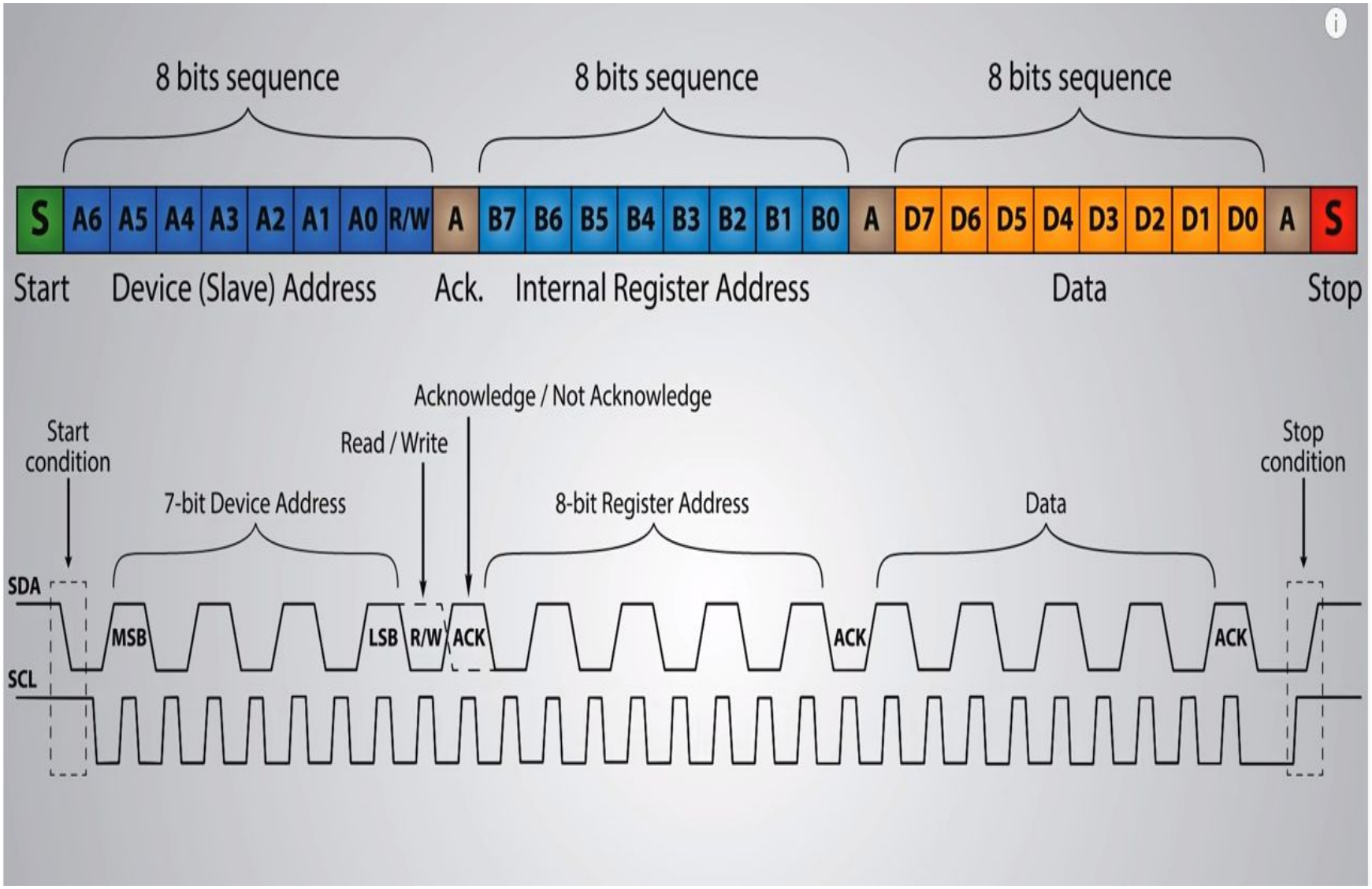
Structure of I2C or I²C Bus



Multiple devices on common I²C bus

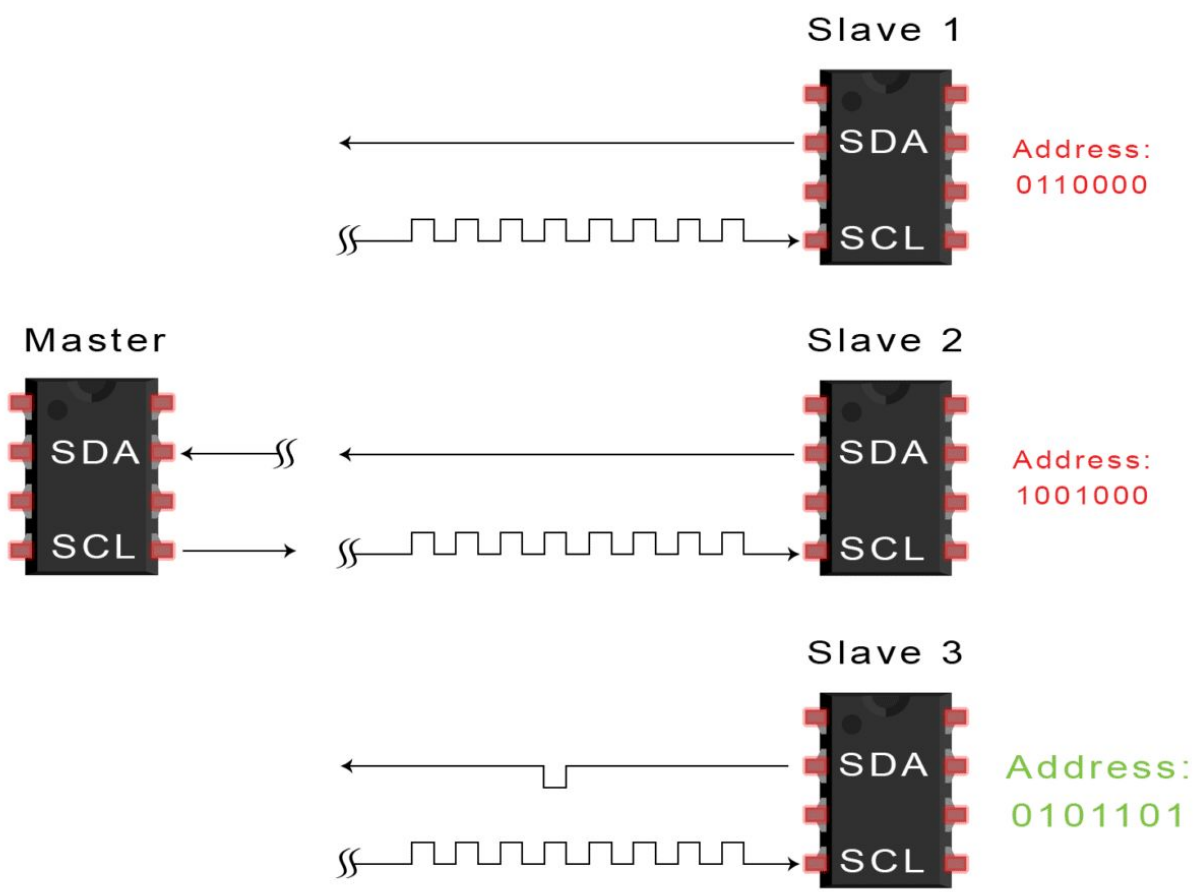
Structure of I2C or I²C Bus

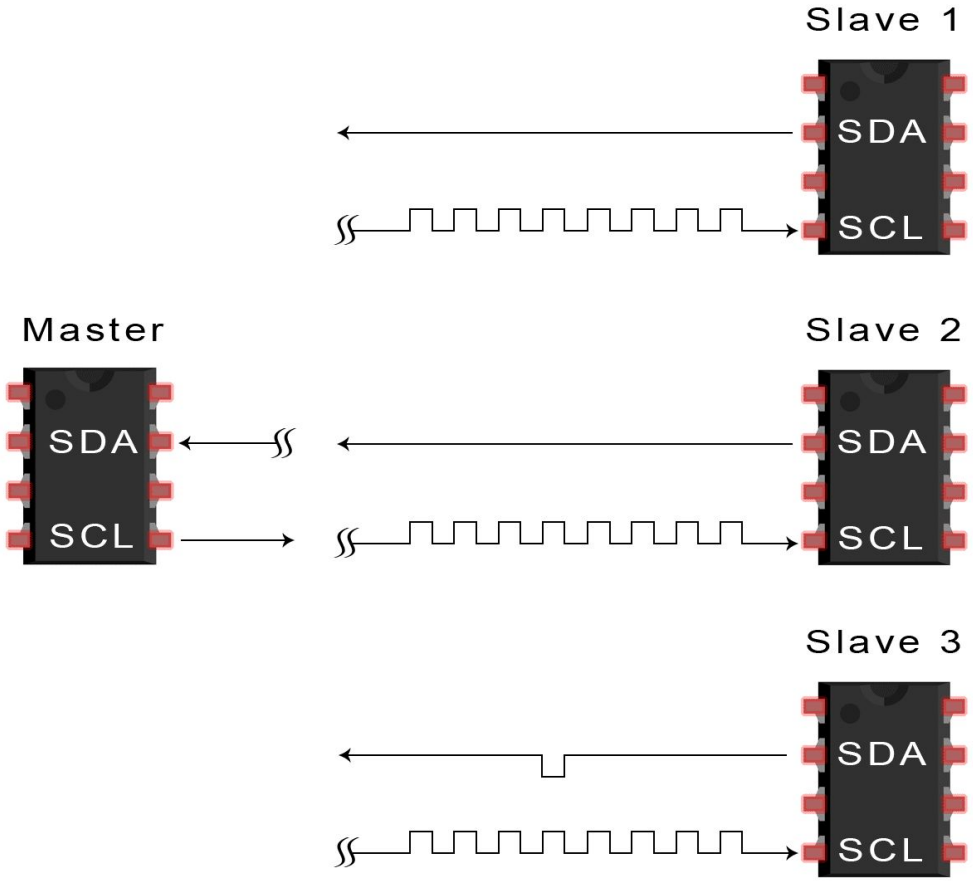



8 bits sequence
8 bits sequence
8 bits sequence
S A6 A5 A4 A3 A2 A1 A0 R/W A B7 B6 B5 B4 B3 B2 B1 B0 A D7 D6 D5 D4 D3 D2 D1 D0 A S
Start
Device (Slave) Address
Ack.
Internal Register Address
Data
Stop
Acknowledge / Not Acknowledge
Start condition
Read / Write
7-bit Device Address
8-bit Register Address
Data
Stop condition
SDA
SCL
MSB
LSB
R/W
ACK
ACK
ACK

Structure of I2C or I^2C Bus

BRAC UNIVERSITY
Inspiring Excellence





I2C Bus

BRAC UNIVERSITY
Inspiring Excellence

I2C Protocol (Basics) Messages are broken up into two types of frame:

- Address frame: The master indicates the slave to which the message is being sent
- Data frame: These are 8-bit data messages passed from master to slave or vice versa.

Data is placed on the SDA line after SCL goes low, and is sampled after the SCL line goes high.

I2C Bus

BRAC UNIVERSITY
Inspiring Excellence

Start Condition

- To initiate the address frame, the master device leaves SCL high and pulls SDA low.
- This puts all slave devices on notice that a transmission is about to start.
- If two master devices wish to take ownership of the bus at one time, whichever device pulls SDA low first wins the race and gains control of the bus.
- It is possible to issue repeated starts, initiating a new communication sequence without relinquishing control of the bus to other masters

I2C Bus

BRAC UNIVERSITY
Inspiring Excellence

Address Frame

- The address frame is always first in any new communication sequence.
- For a 7-bit address, the address is clocked out most significant bit (MSB) first, followed by a R/W bit indicating whether this is a read (1) or write (0) operation.

I2C Bus

BRAC UNIVERSITY
Inspiring Excellence

NACK/ACK bit

- It is the 9th bit of the frame
- This is the case for all frames (data or address).
- Once the first 8 bits of the frame are sent, the receiving device is given control over SDA.
- If the receiving device does not pull the SDA line low before the 9th clock pulse, it can be inferred that the receiving device either did not receive the data or did not know how to parse the message.
- In that case, the exchange halts, and it’s up to the master of the system to decide how to proceed.

I2C Bus

BRAC UNIVERSITY
Inspiring Excellence

Data Frames

- After the address frame has been sent, data can begin being transmitted.
- The master will simply continue generating clock pulses at a regular interval, and the data will be placed on SDA by either the master or the slave, depending on whether the R/W bit indicated a read or write operation.

Stop condition

- Once all the data frames have been sent, the master will generate a stop condition.
- Stop conditions are defined by a 0->1 (low to high) transition on SDA after a 0->1 transition on SCL, with SCL remaining high.
- During normal data writing operation, the value on SDA should not change when SCL is high, to avoid false stop conditions.

.

STEPS OF I2C DATA TRANSMISSION

BRAC UNIVERSITY
Inspiring Excellence

1. The master sends the start condition to every connected slave by switching the SDA line from a high voltage level to a low voltage level *before* switching the SCL line from high to low.

2. The master sends each slave the 7 or 10 bit address of the slave it wants to communicate with, along with the read/write bit.

3. Each slave compares the address sent from the master to its own address. If the address matches, the slave returns an ACK bit by pulling the SDA line low for one bit. If the address from the master does not match the slave's own address, the slave leaves the SDA line high.

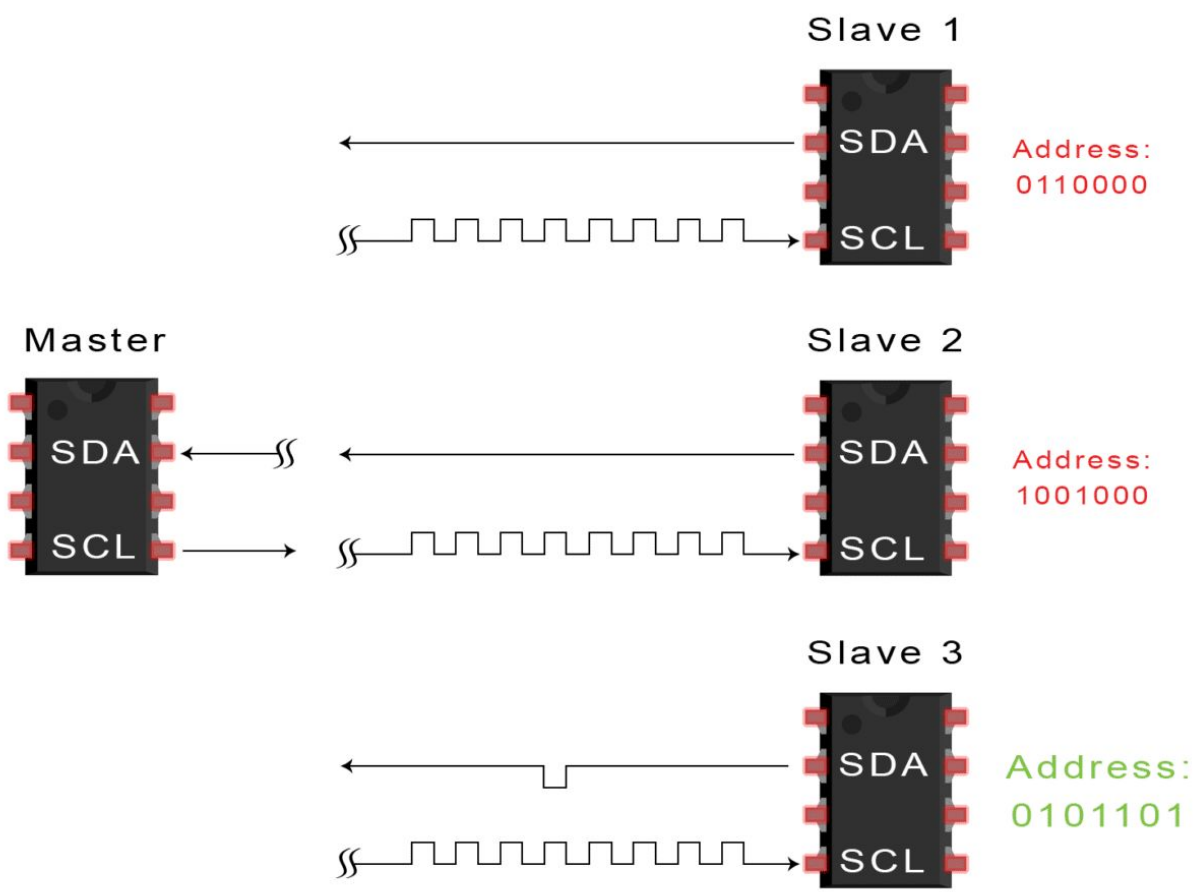
4. The master sends or receives the data frame.

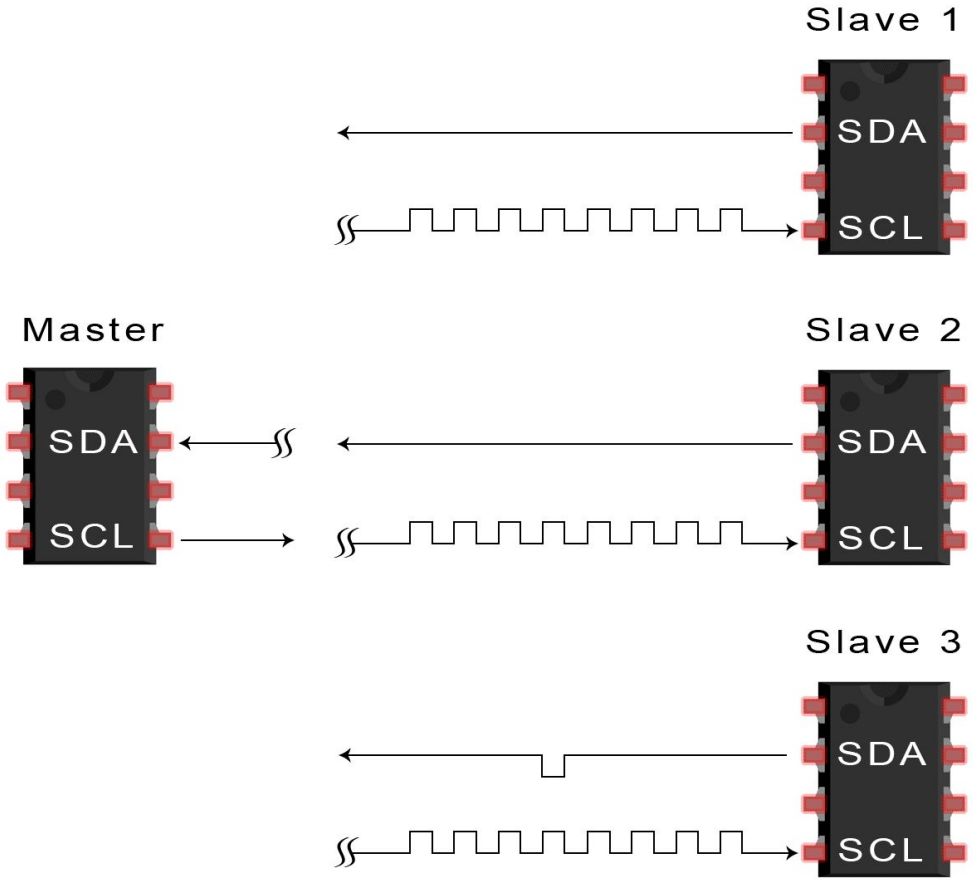
5. After each data frame has been transferred, the receiving device returns another ACK bit to the sender to acknowledge successful receipt of the frame.

6. To stop the data transmission, the master sends a stop condition to the slave by switching SCL high before switching SDA high.

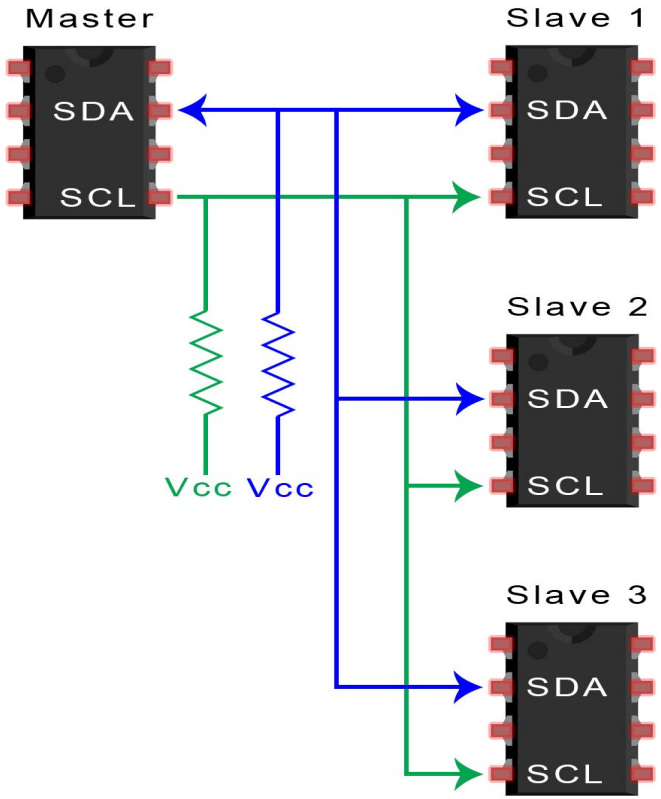
Structure of I2C or I^2C Bus

BRAC UNIVERSITY
Inspiring Excellence

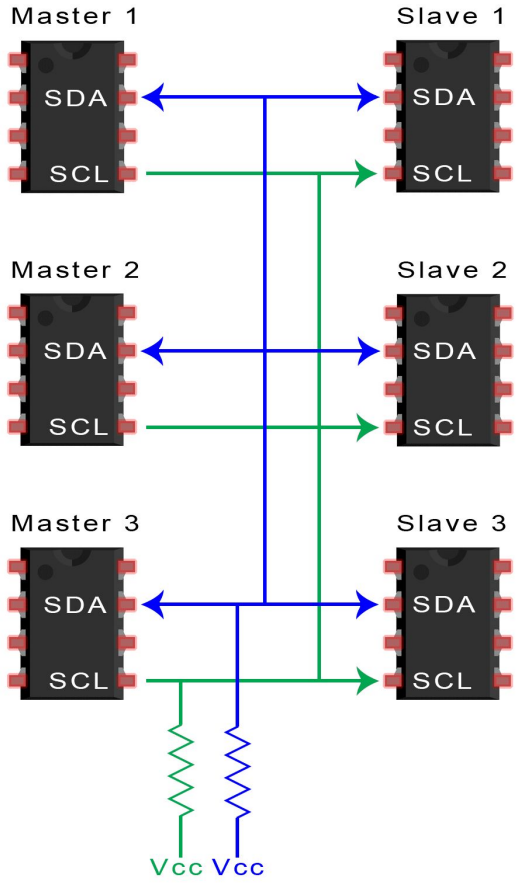




Structure of I2C or I²C Bus



Single Master – Multiple Slave



Multiple Master – Multiple Slave

Advantages

BRAC UNIVERSITY
Inspiring Excellence

- I2C communication or protocol has a significant edge over its peers such as serial port communication and SPI. Let us have a look into the various advantages that renders the I2C protocol so effective for short distance intra-board communication.
- **1. Flexibility** – The I2C protocol supports multi-master, multi-slave communication, which implies you can add a lot of functionality to your design. More than one master IC controlling and communicating with the slave ICs can speed things up and add functionalities to the embedded system.
- **2. Addressing feature** – Yet another advantage of the I2C protocol lies in its inherent ability to use chip addressing. It means that you can easily add components to the bus without any complexity. It eliminates the necessity of SS (Slave select) lines.
- **3. Simplicity** – I2C protocol doesn't complicate the design. It requires only two bidirectional signal lines to establish communication among multiple devices. Further, the pin count is low as well.
- **4. Better error handling mechanism** – To improve the error detection and correction mechanism, the I2C protocol relies on ACK/NACK feature, which is a robust error correction feature. ACK stands for Acknowledgement whereas NACK means No Acknowledgement.
- **5. Adaptable** – The I2C protocol is adaptable in the sense that it can work well with both slow ICs and fast ICs.

Disadvantages

BRAC UNIVERSITY
Inspiring Excellence

1. Conflicts – Due to chip addressing, there's always a possibility of an address conflict.

2. Slower speeds – Slower data transfer rate than SPI

3. Requires more space – More complicated hardware needed to implement than SPI

4. Size of Data Frame - Limited to 8 bits.

Thank You

For Your Attention