



FIAP

Engenharia de Software

EDGE COMPUTING & COMPUTER SYSTEMS

02 – Padrões de Comunicação

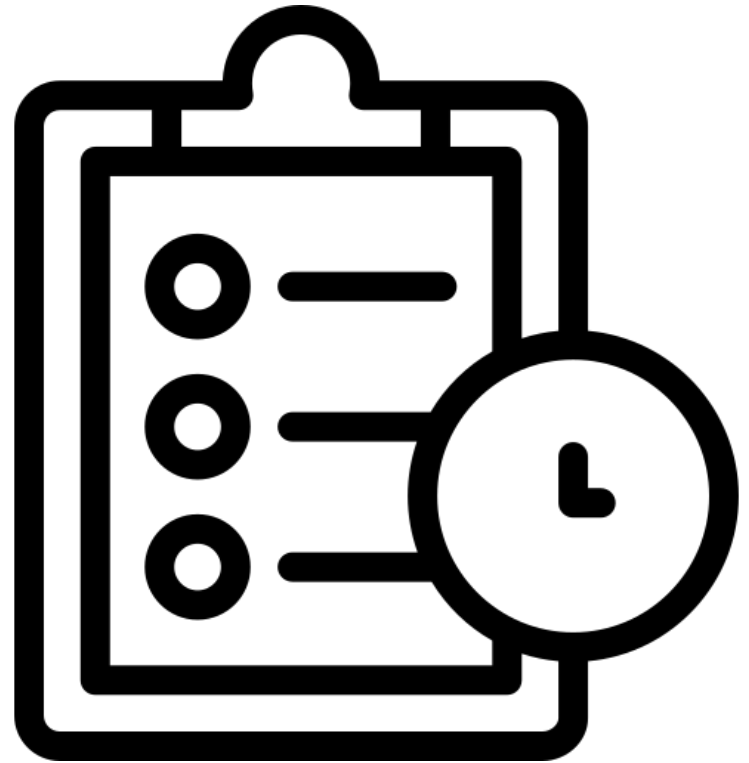


Prof. Airton Y. C. Toyofuku



profairton.toyofuku@fiap.com.br

- O que é comunicação?
- Comunicação entre Computadores;
- Comunicação Paralela;
- Comunicação Serial;
- Características da Comunicação Serial;
- Padrão UART;
- Padrão SPI;
- Padrão I2C;
- Laboratório;
- Exercícios;



O que é comunicação?

FIAP



O que é comunicação?

Comunicação vem do latim (*Communicatio.onis*), que significa “**ação de participar**”.



O que é comunicação?

Comunicação vem do latim (*Communicatio.onis*), que significa “**ação de participar**”.

É um processo que envolve a **troca de informações** entre dois ou mais interlocutores por meio de **signos** e **regras** mutuamente entendíveis.



O que é comunicação?

Comunicação vem do latim (*Communicatio.onis*), que significa “**ação de participar**”.

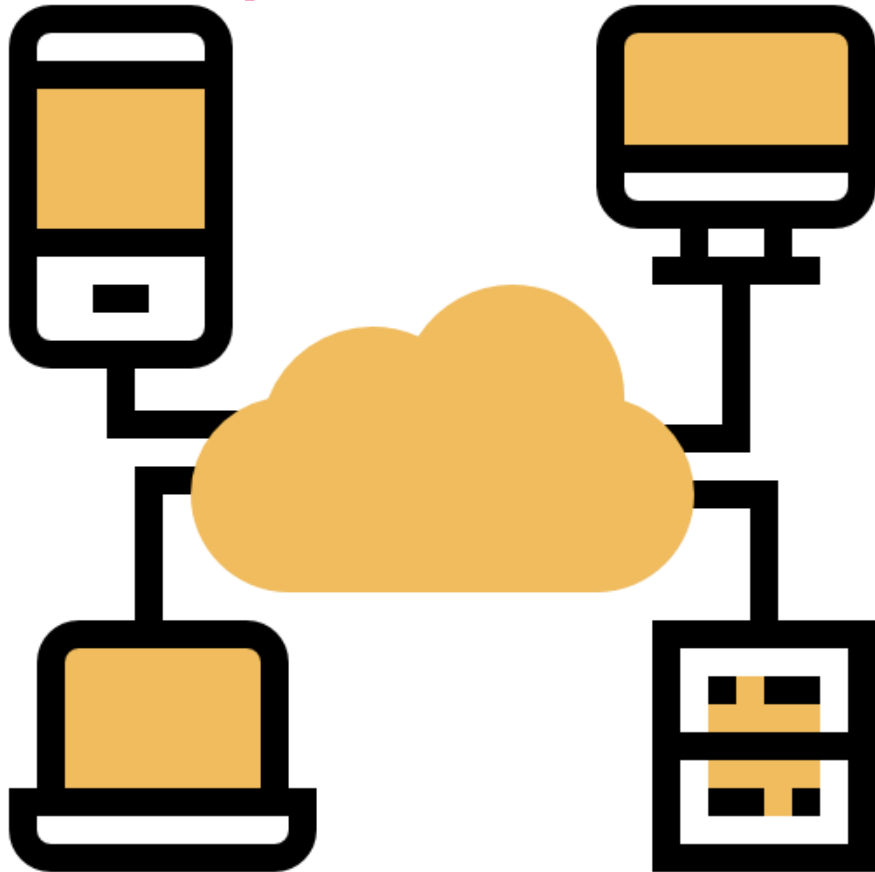
É um processo que envolve a **troca de informações** entre dois ou mais interlocutores por meio de **signos** e **regras** mutuamente entendíveis.

É um processo que permite **criar e interpretar** mensagens que provocam uma resposta.

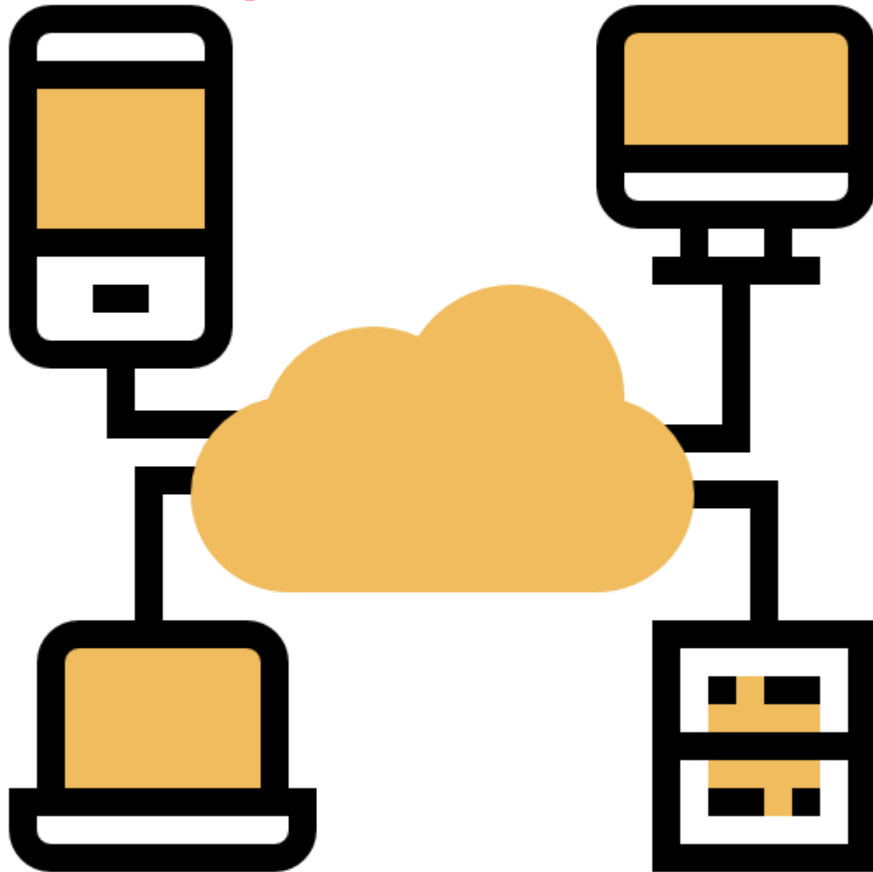


Comunicação entre Computadores

FIA/P

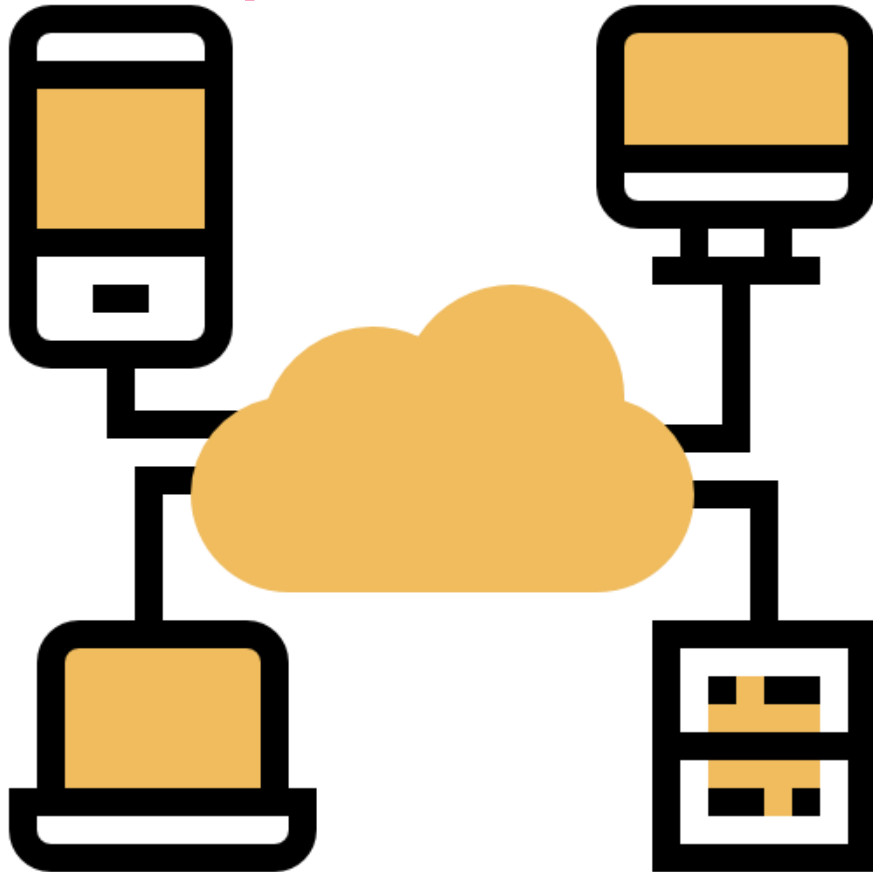


Comunicação entre Computadores



Processo de troca de **informações, dados e recursos entre dois** ou mais **computadores** através de meios de transmissão, como cabos, redes sem fio ou a internet.

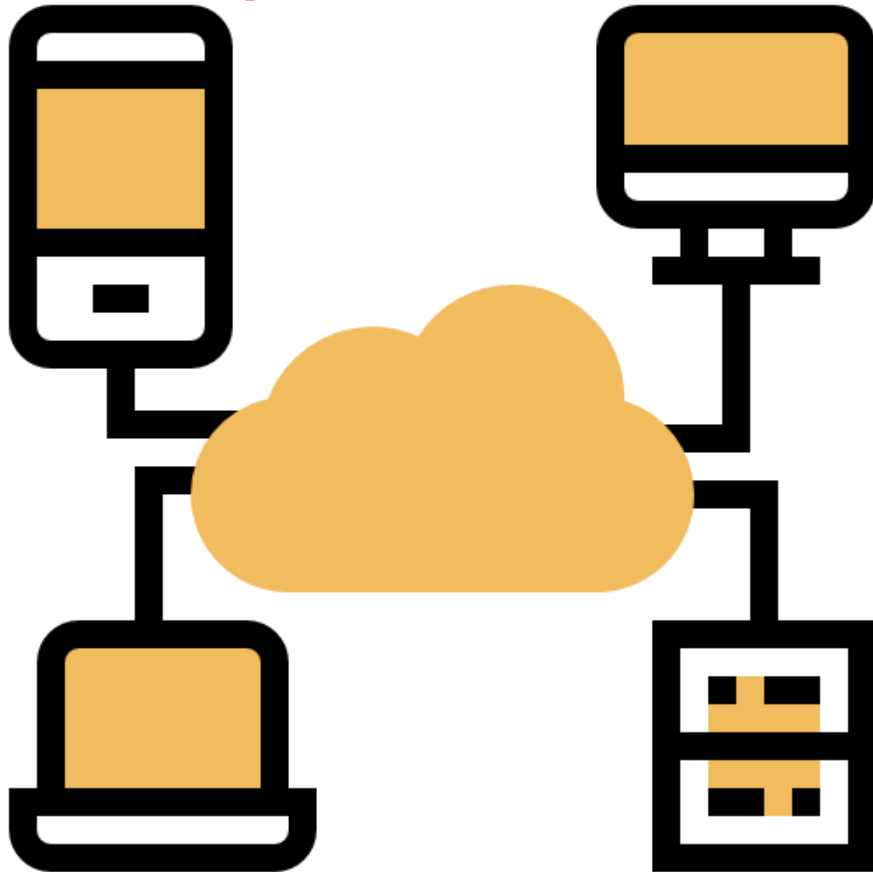
Comunicação entre Computadores



Processo de troca de **informações, dados e recursos entre dois** ou mais **computadores** através de meios de transmissão, como cabos, redes sem fio ou a internet.

Envolve a **transferência** de dados em formato **binário**, que é a representação digital dos dados, utilizando **protocolos** e **padrões** de comunicação bem estabelecidos.

Comunicação entre Computadores



Processo de troca de **informações, dados e recursos entre dois** ou mais **computadores** através de meios de transmissão, como cabos, redes sem fio ou a internet.

Envolve a **transferência** de dados em formato **binário**, que é a representação digital dos dados, utilizando **protocolos** e **padrões** de comunicação bem estabelecidos.

Os **protocolos** definem as **regras** e **formatos padronizados** para que os computadores possa se entender mutuamente.

Os principais elementos envolvidos na comunicação entre computadores são:

1

2

3

4

5

Comunicação entre

Os principais elementos envolvidos na comunicação entre computadores são:

1

2

3

4

5

EMISSOR

O computador ou dispositivo que envia os dados.

Os principais elementos envolvidos na comunicação entre computadores são:

1

2

3

4

5

EMISSOR

O computador ou dispositivo que envia os dados.

RECEPTOR

O computador ou dispositivo que recebe os dados enviados pelo emissor.

Os principais elementos envolvidos na comunicação entre computadores são:

1

2

3

4

5

EMISSOR

O computador ou dispositivo que envia os dados.

RECEPTOR

O computador ou dispositivo que recebe os dados enviados pelo emissor.

MEIO DE TRANSMISSÃO

O canal físico ou lógico que possibilita a troca de dados entre os computadores, como cabos, fibras ópticas, ondas de rádio ou luz.

Os principais elementos envolvidos na comunicação entre computadores são:

1

2

3

4

5

EMISSOR

O computador ou dispositivo que envia os dados.

RECEPTOR

O computador ou dispositivo que recebe os dados enviados pelo emissor.

MEIO DE TRANSMISSÃO

O canal físico ou lógico que possibilita a troca de dados entre os computadores, como cabos, fibras ópticas, ondas de rádio ou luz.

PROTOCOLOS

Conjunto de regras e convenções que definem como os dados serão formatados, enviados, recebidos e interpretados.

Os principais elementos envolvidos na comunicação entre computadores são:

1

2

3

4

5

EMISSOR

O computador ou dispositivo que envia os dados.

RECEPTOR

O computador ou dispositivo que recebe os dados enviados pelo emissor.

MEIO DE TRANSMISSÃO

O canal físico ou lógico que possibilita a troca de dados entre os computadores, como cabos, fibras ópticas, ondas de rádio ou luz.

PROTOCOLOS

Conjunto de regras e convenções que definem como os dados serão formatados, enviados, recebidos e interpretados.

ENDEREÇAMENTO

Cada computador possui um endereço único para identificação e direcionamento dos dados, podendo ser um endereço IP ou Mac Address.

Comunicação Paralela

FIAP

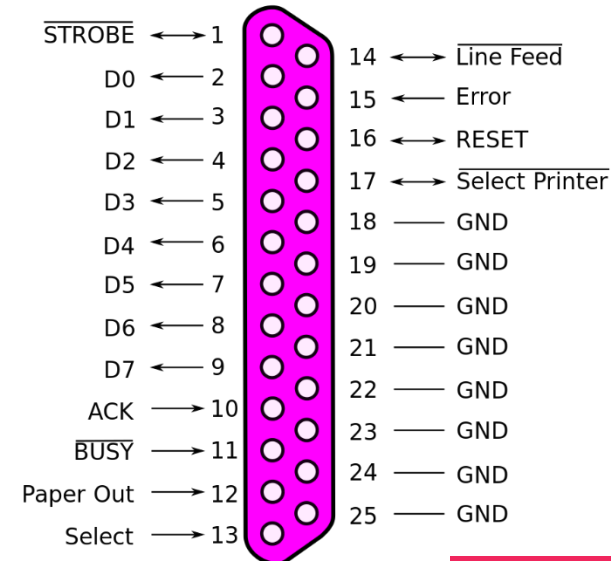
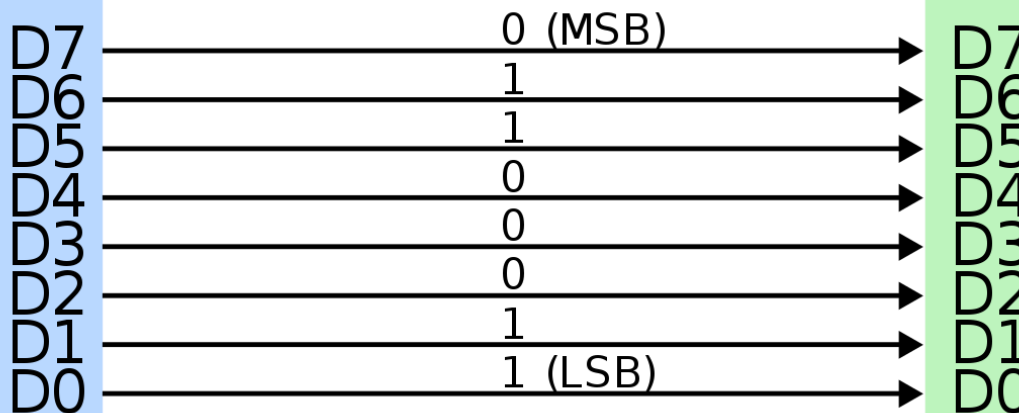
Neste tipo de transmissão, **todos os bits** que o dispositivo transmissor é capaz de manipular **são transmitidos simultaneamente** ao receptor, através de vias paralelas.



Parallel interface example

Transmitter (TX)

Receiver (RX)



Fonte:
https://en.wikipedia.org/wiki/Parallel_port#/media/File:Parallel_port_pinouts.svg

Comunicação Serial

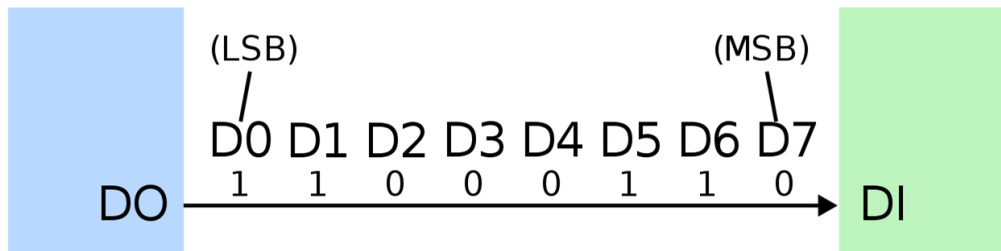
Este tipo de transmissão é caracterizado por **enviar um bit por vez**, sequencialmente, num canal de comunicação ou barramento.



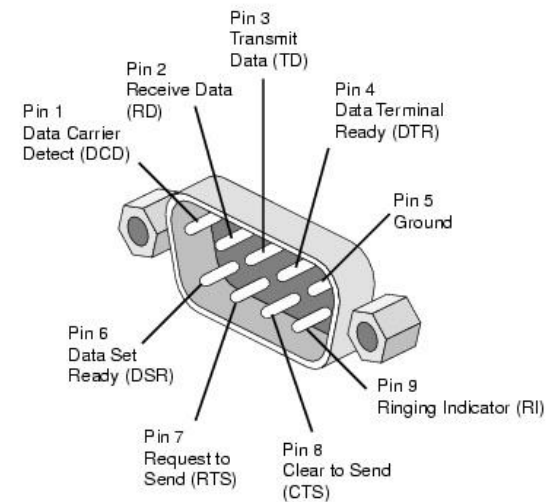
Serial interface example

Transmitter (TX)

Receiver (RX)



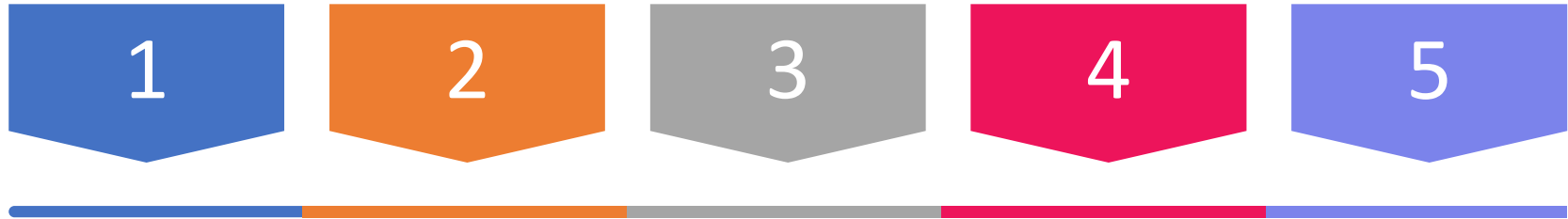
Fonte: https://upload.wikimedia.org/wikipedia/commons/thumb/3/3d/Serial_vs._parallel_transmission.svg/1280px-Serial_vs._parallel_transmission.svg.png



Fonte: <https://www.eltima.com/article/9-pin-serial-port.html>

Características da Comunicação Serial FIAP

Taxa de Comunicação



Características da Comunicação Serial FIAP

Taxa de Comunicação

1

2

3

4

5

Representa a
velocidade da
comunicação

Características da Comunicação Serial FIAP

Taxa de Comunicação

1

2

3

4

5

Representa a
velocidade da
comunicação

Medida em
bps (bits por
segundo)

Características da Comunicação Serial FIAP

Taxa de Comunicação

1

Representa a
velocidade da
comunicação

2

Medida em
bps (bits por
segundo)

3

Em
comunicação
síncronas é
chamado de
clock

4

5

Características da Comunicação Serial FIAP

Taxa de Comunicação

1

Representa a
velocidade da
comunicação

2

Medida em
bps (bits por
segundo)

3

Em
comunicação
síncronas é
chamado de
clock

4

Em
comunicação
s assíncronas
é chamada de
Baud Rate

5

Características da Comunicação Serial FIAP

Taxa de Comunicação

1

Representa a
velocidade da
comunicação

2

Medida em
bps (bits por
segundo)

3

Em
comunicação
síncronas é
chamado de
clock

4

Em
comunicação
assíncronas
é chamada de
Baud Rate

5

Os
dispositivos
devem estar
configurados
com a mesma
taxa de
comunicação

Características da Comunicação Serial FIAP

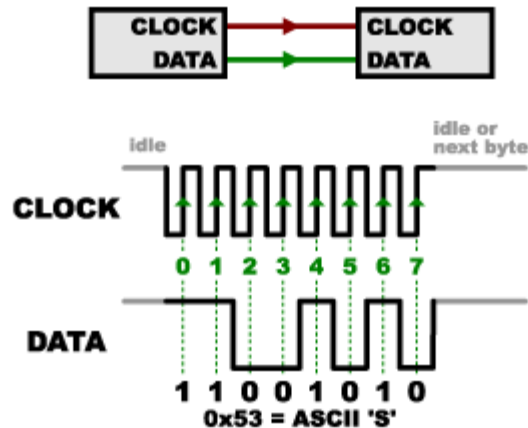


Métodos

Fonte: <https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi>

Características da Comunicação Serial

Métodos



Fonte: <https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi>

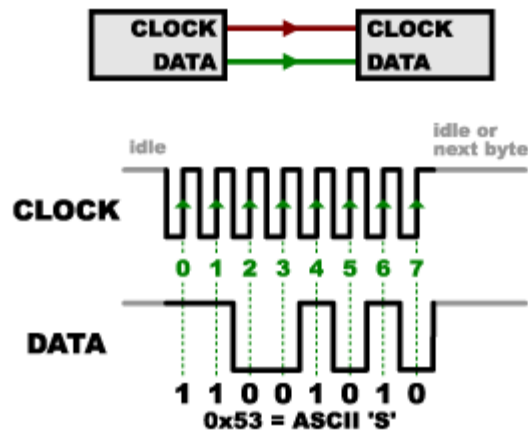
Síncrono

É o método de comunicação que depende de um sinal de clock, ou seja, para enviar cada bit é necessário um pulso de sincronismo para “avisar” da transmissão.

A vantagem é que a comunicação pode atingir altas velocidades, com a desvantagem de precisar de uma via extra para o sinal de clock.

Características da Comunicação Serial FIAP

Métodos

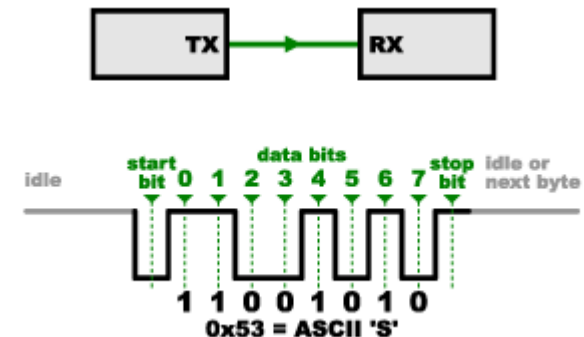


Fonte: <https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi>

Síncrono

É o método de comunicação que depende de um sinal de clock, ou seja, para enviar cada bit é necessário um pulso de sincronismo para “avisar” da transmissão.

A vantagem é que a comunicação pode atingir altas velocidades, com a desvantagem de precisar de uma via extra para o sinal de clock.



Fonte: <https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi>

Assíncrono

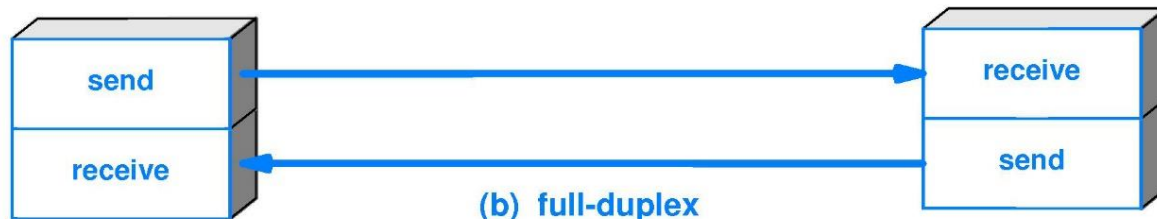
Envia os dados através de uma única via de transmissão e por não ter um sinal de sincronismo, requer um controle mais complicado e é susceptível a erros. Por isso utiliza como parâmetro o Baud Rate, que especifica a velocidade de recepção e envio dos bits.

Características da Comunicação Serial FIAP

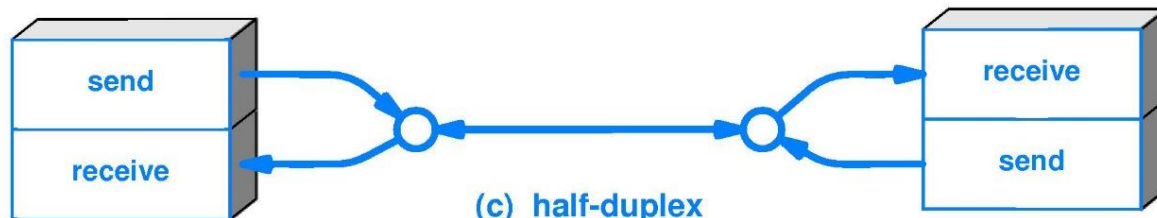
Sentido de transmissão



(a) simplex



(b) full-duplex



(c) half-duplex

Características da Comunicação Serial FIAP



Terminologia

Características da Comunicação Serial FIAP

Terminologia

RX/TX

- RX é o termo usado para o pino receptor;
- TX é o termo usado para o pino transmissor;
- Entre dois dispositivos, os pinos devem ser ligados da seguinte forma:

TX → RX
RX → TX

Características da Comunicação Serial FIAP

Terminologia

RX/TX

- RX é o termo usado para o pino receptor;
- TX é o termo usado para o pino transmissor;
- Entre dois dispositivos, os pinos devem ser ligados da seguinte forma:

TX → RX
RX → TX

Master e Slave

- É o método de comunicação em que há vários dispositivos se comunicando;
- Apenas um dispositivo deve iniciar a comunicação, enviando comandos, controlando velocidades, etc. Sendo este responsável por coordenar a comunicação;
- O transmissor é denominado Master, e os receptores denominados Slaves;

Características da Comunicação Serial FIAP

Terminologia

RX/TX

- RX é o termo usado para o pino receptor;
- TX é o termo usado para o pino transmissor;
- Entre dois dispositivos, os pinos devem ser ligados da seguinte forma:

TX → RX
RX → TX

Master e Slave

- É o método de comunicação em que há vários dispositivos se comunicando;
- Apenas um dispositivo deve iniciar a comunicação, enviando comandos, controlando velocidades, etc. Sendo este responsável por coordenar a comunicação;
- O transmissor é denominado Master, e os receptores denominados Slaves;

Nível Lógico

- São os estados que um bit pode assumir, sendo nível alto (1) e nível baixo (0);
- Os níveis lógicos são interpretados de acordo com os protocolos e baseados nas tensões que recebe;
- Por exemplo, o protocolo TTL considera de 2V a 5V como nível lógico alto e de 0V a 0,8V como nível lógico baixo;

Padrão UART

Universal Asynchronous
Receiver/Transmitter;

Protocolo para troca de informações
de forma serial entre dois dispositivos;

Usa apenas dois “fios”:
TX → RX em cada direção

Pode operar em Simplex, Half-Duplex
ou Full Duplex

A informação é transmitida no formato
de **FRAMES**

Padrão UART

Universal Asynchronous Receiver/Transmitter;

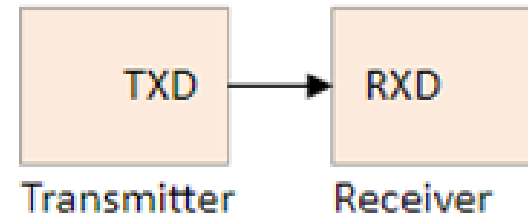
Protocolo para troca de informações de forma serial entre dois dispositivos;

Usa apenas dois “fios”:
TX → RX em cada direção

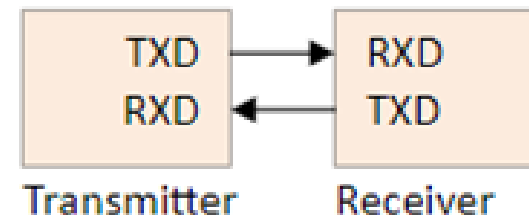
Pode operar em Simplex, Half-Duplex ou Full Duplex

A informação é transmitida no formato de **FRAMES**

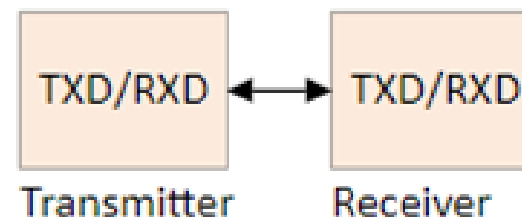
Simplex UART system



Full-duplex UART system



Half-duplex UART system



Padrão UART

Foi um dos primeiros protocolos de comunicação utilizados

Aplicado em Serial (COM) Ports, RS-232, Modems, etc.

Seu uso tem decaído devido a outros protocolos como SPI, I2C, USB e mesmo Ethernet

Ainda é importante para aplicações de baixa velocidade (115200 bps), e baixo throughput

Usamos UART para Mandar mensagens no Console do Arduino



Padrão UART

Foi um dos primeiros protocolos de comunicação utilizados

Aplicado em Serial (COM) Ports, RS-232, Modems, etc.

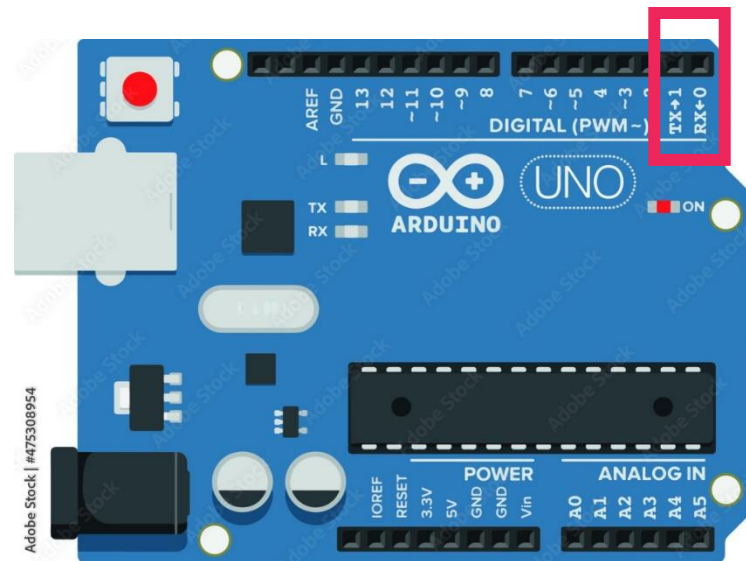
Seu uso tem decaído devido a outros protocolos como SPI, I2C, USB e mesmo Ethernet

Ainda é importante para aplicações de baixa velocidade (115200 bps), e baixo throughput

Usamos UART para Mandar mensagens no Console do Arduino



UART



Padrão UART

UART é Assíncrona, ou seja, não depende de uma fonte de clock

O Receptor e o Transmissor devem operar na mesma velocidade (Baud rate)

Eles também devem usar a mesma estrutura de Frame

Baud Rates Comuns em uma comunicação UART

4800 bps

9600 bps

19200 bps

57600 bps

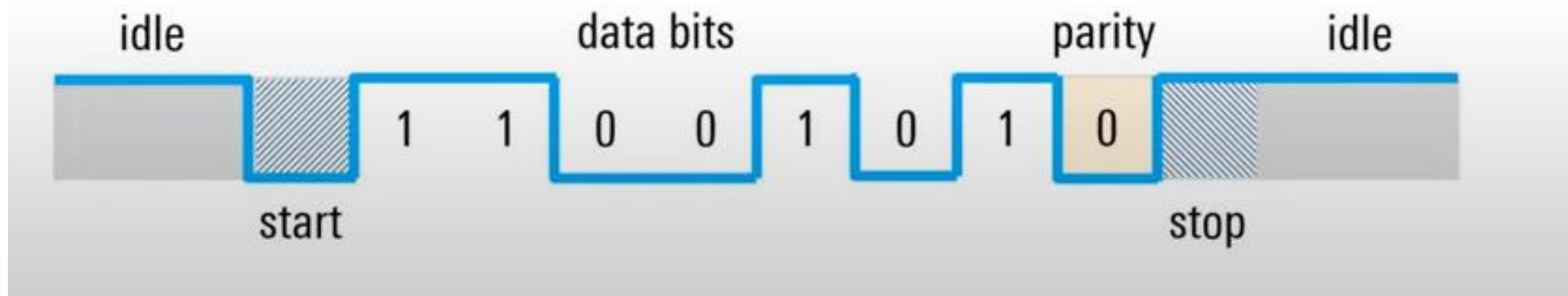
115200 bps

Padrão UART

Bits de START e STOP:

Start bit: Indica informação a caminho;

Stop bit: Indica termino da informação



Padrão UART

FIAP

Bits de informação ? Data Bits

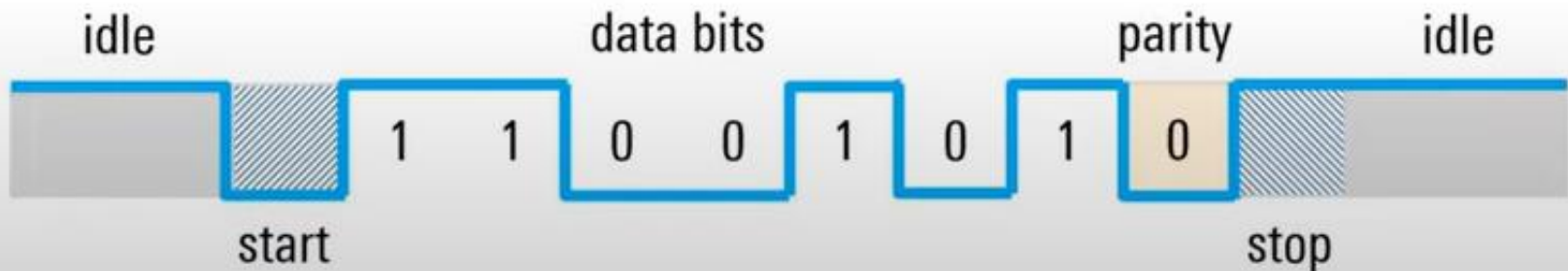
Podem ser de 5 a 9 bits.

Normalmente usamos de 7 a 8 bits;

Bit menos significativo primeiro (LSB);

Por Exemplo:

- A letra 'S' = 0x52 = 1010011
- Na Ordem LSB = 1100101



Padrão UART

FIAP

Bits de Paridade 2 Opcional

Usado para detectar erros

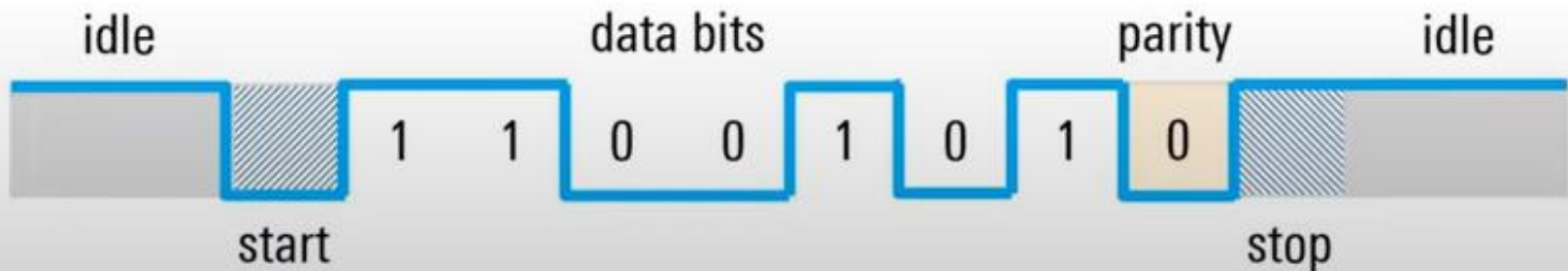
Paridade PAR: a quantidade de 1 deve ser par → Bit de Paridade = 1

Paridade IMPAR: a quantidade de 1 deve ser ímpar → Bit de Paridade = 0

Só consegue detectar erro de um único bit

Por Exemplo:

- A letra 'S' = 0x52 = 1010011 → possui Quatro 1's
- Paridade Par!



Padrão SPI

Interface de comunicação com 4 “fios”, desenvolvido pela Motorola em 1980s

Velocidade superior aos padrões UART e I2C

Usado para transferir informações entre um controlador (smart) e um periférico (less smart)

Um Controlador pode se comunicar com um ou mais periféricos!

Ao contrário da UART, as informações são enviadas em Conjuntos de Bytes!

Padrão SPI

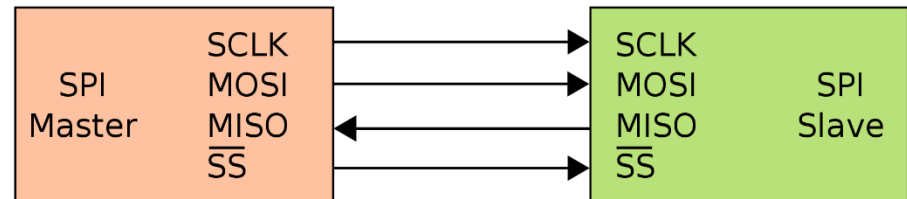
Interface de comunicação com 4 “fios”, desenvolvido pela Motorola em 1980s

Velocidade superior aos padrões UART e I2C

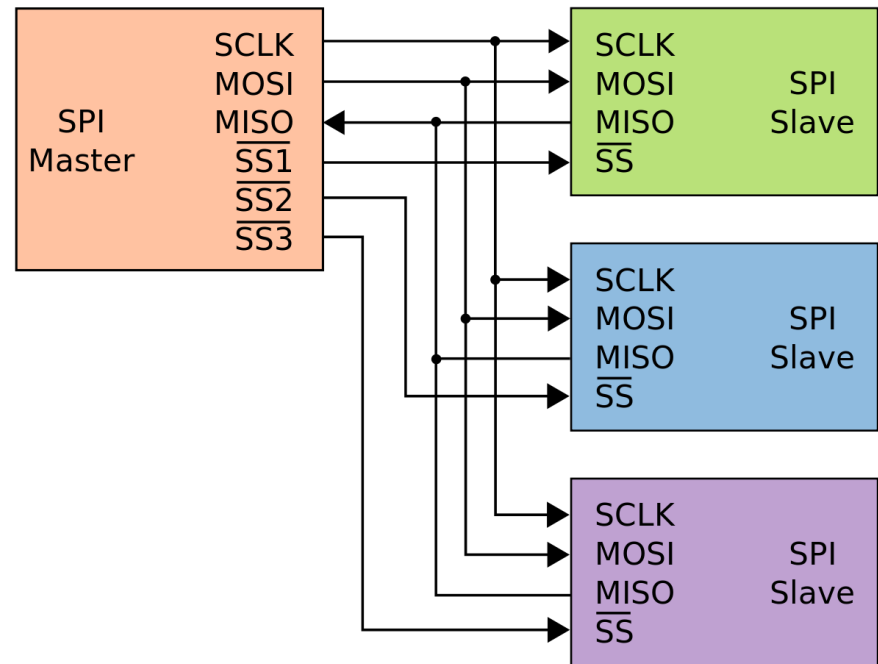
Usado para transferir informações entre um controlador (smart) e um periférico (less smart)

Um Controlador pode se comunicar com um ou mais periféricos!

Ao contrário da UART, as informações são enviadas em Conjuntos de Bytes!



Fonte: https://en.wikipedia.org/wiki/Serial_Peripheral_Interface#/media/File:SPI_single_slave.svg



Fonte: https://en.wikipedia.org/wiki/File:SPI_three_slaves.svg

Padrão SPI

SS ou **CS** (Chip Select) ☐ Escolhe o periférico de destino da comunicação

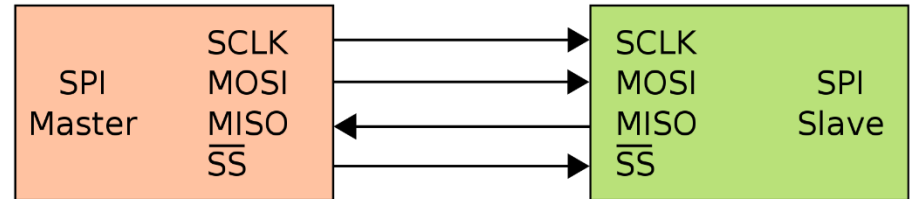
SCLK (Synchronous Clock) ☐
Fornece os pulsos de sincronismo da comunicação

MOSI (Master Out – Slave In) ☐
Informação transmitida pelo Controlador

MISO (Master In – Slave Out) ☐
Informação enviada pelo Periférico

No Arduino, os pinos são mapeados como:

13 - SCK
12 - MISO
11 - MOSI
10 - SS/CS



Fonte: https://en.wikipedia.org/wiki/Serial_Peripheral_Interface#/media/File:SPI_single_slave.svg

Padrão SPI

SS ou **CS** (Chip Select) ☐ Escolhe o periférico de destino da comunicação

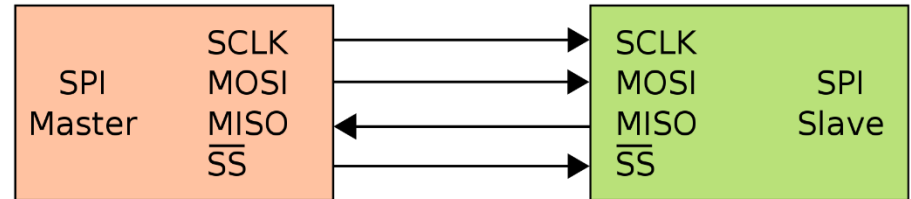
SCLK (Synchronous Clock) ☐
Fornece os pulsos de sincronismo da comunicação

MOSI (Master Out – Slave In) ☐
Informação transmitida pelo Controlador

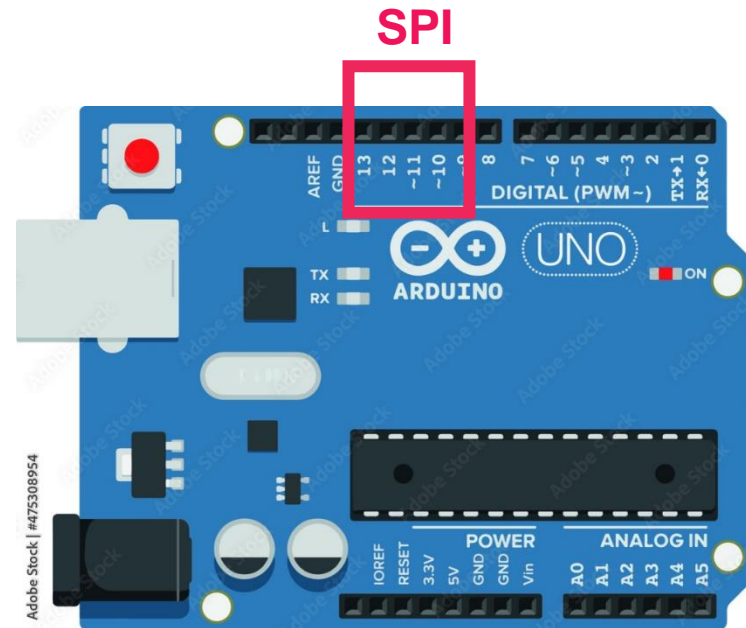
MISO (Master In – Slave Out) ☐
Informação enviada pelo Periférico

No Arduino, os pinos são mapeados como:

13 - **SCK**
12 - **MISO**
11 - **MOSI**
10 - **SS/CS**

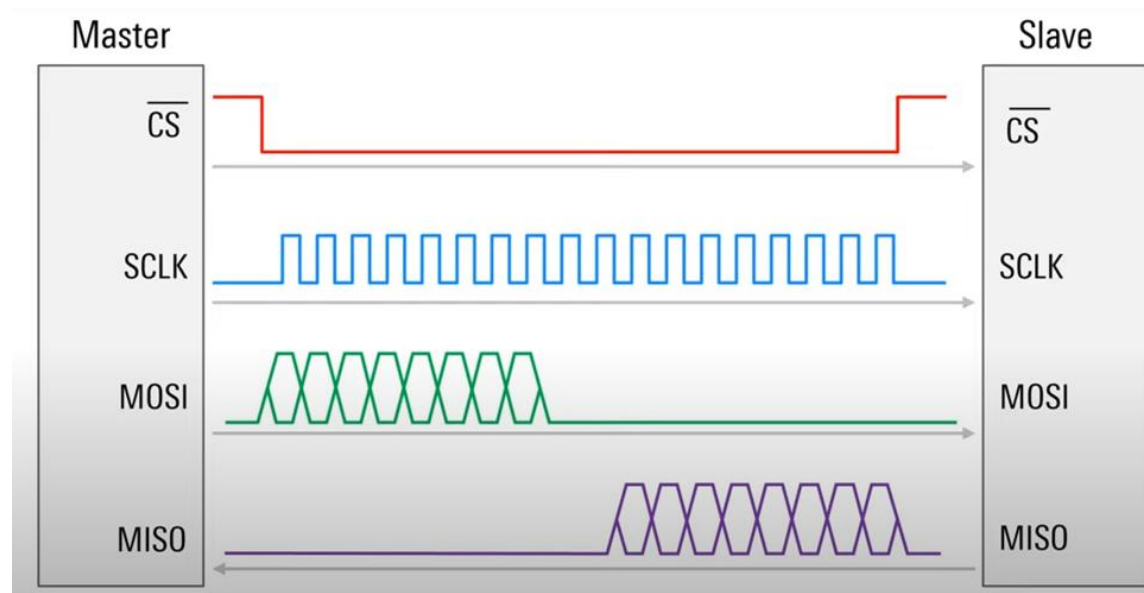
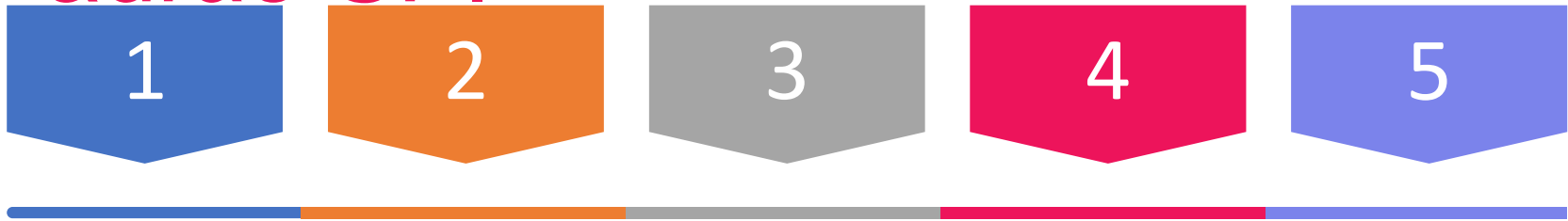


Fonte: https://en.wikipedia.org/wiki/Serial_Peripheral_Interface#/media/File:SPI_single_slave.svg



Padrão SPI

FIAP



Padrão SPI

FIAP

1

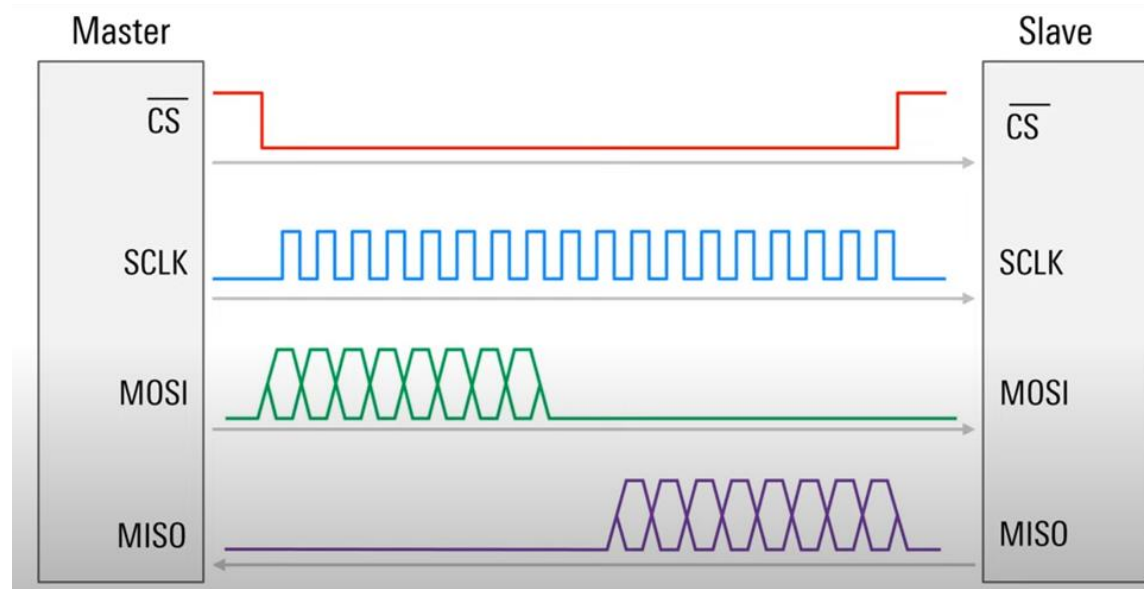
2

3

4

5

O Controlador
coloca o sinal
do CS em
baixa



Padrão SPI

FIAP

1

2

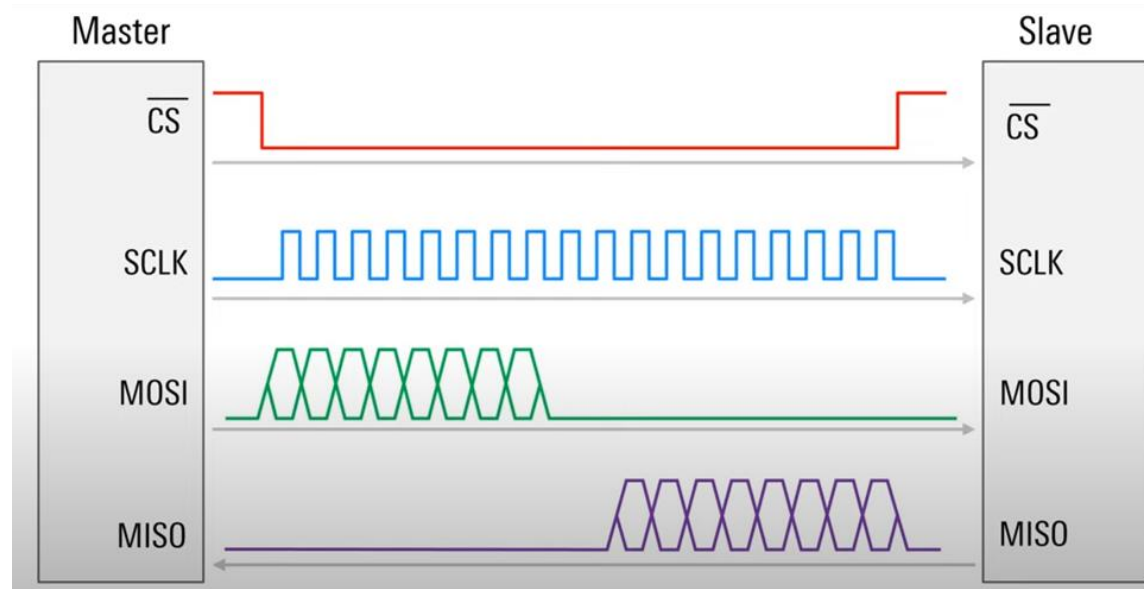
3

4

5

O Controlador
coloca o sinal
do CS em
baixa

Em seguida,
começa a
fornecer o
sinal de
sincronismo
pelo SCLK



Padrão SPI

FIAP

1

2

3

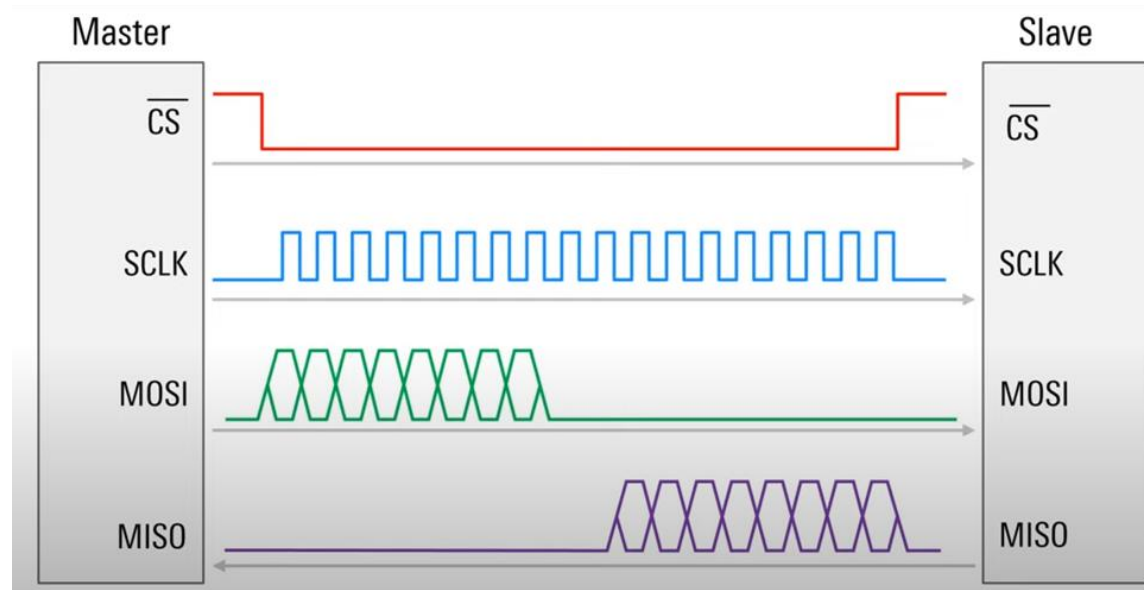
4

5

O Controlador coloca o sinal do CS em baixa

Em seguida, começa a fornecer o sinal de sincronismo pelo SCLK

E transmite a informação pelo MOSI



Padrão SPI

FIAP

1

2

3

4

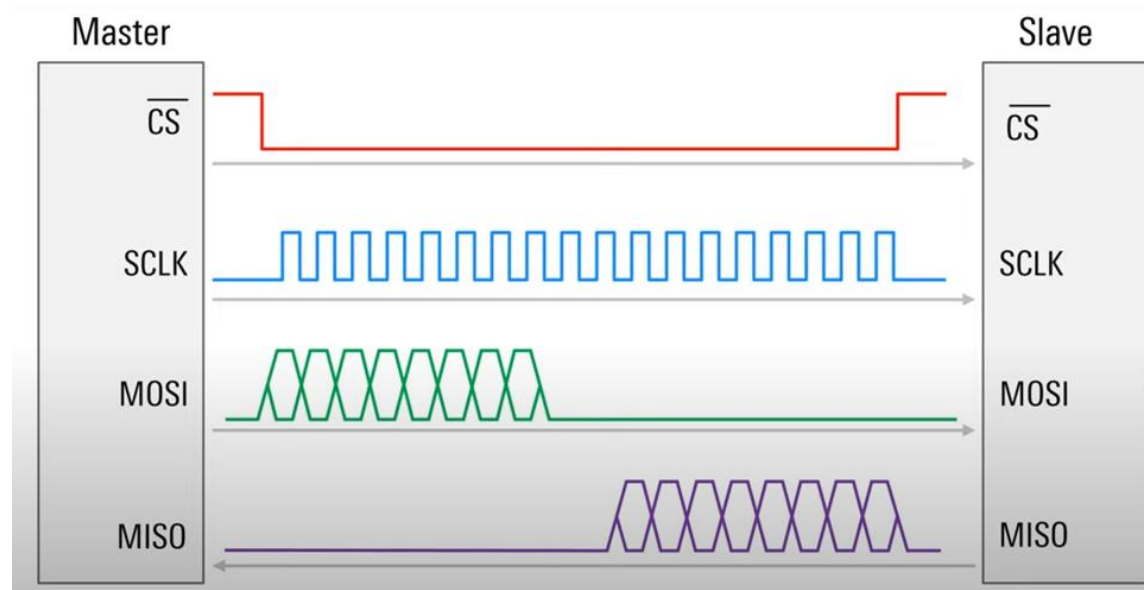
5

O Controlador coloca o sinal do CS em baixa

Em seguida, começa a fornecer o sinal de sincronismo pelo SCLK

E transmite a informação pelo MOSI

Se o periférico precisar enviar alguma informação, é enviado pelo MISO



Padrão SPI

FIAP

1

2

3

4

5

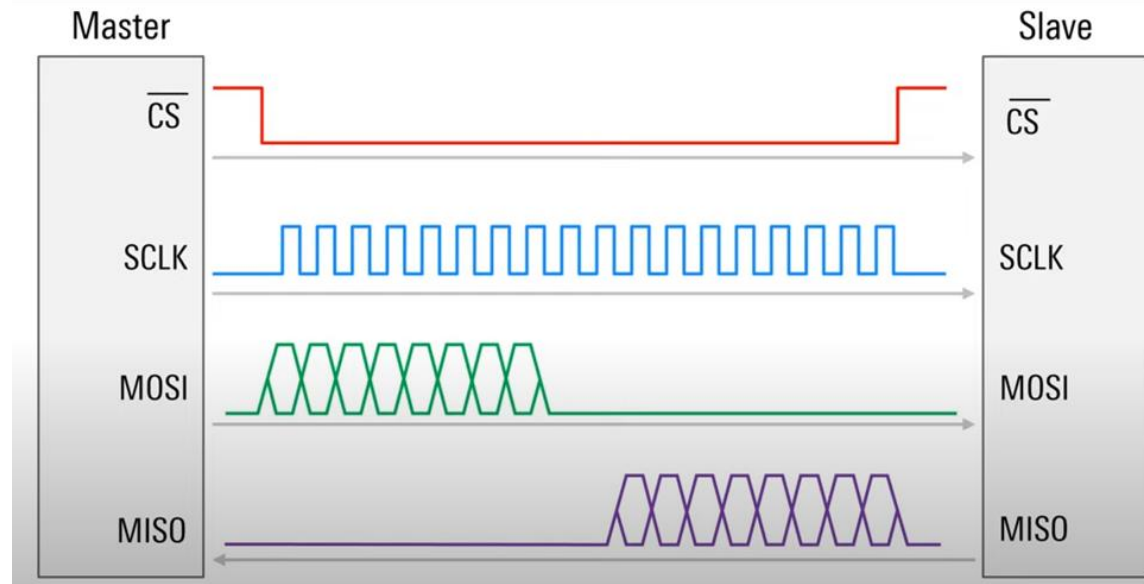
O Controlador coloca o sinal do CS em baixa

Em seguida, começa a fornecer o sinal de sincronismo pelo SCLK

E transmite a informação pelo MOSI

Se o periférico precisar enviar alguma informação, é enviado pelo MISO

Com o término da comunicação, o Controlador coloca o sinal de CS em alta e para o sincronismo



Padrão I2C

Inter-Integrated Circuit (I2C), desenvolvido pela Philips em 1982

Usado para comunicação de curta distância (alguns centímetros)

Comunicação bidirecional, Half Duplex, com diferentes velocidades entre os periféricos

A comunicação é feita por dois “fios”;
SCL → Serial Clock
SDA → Serial Data

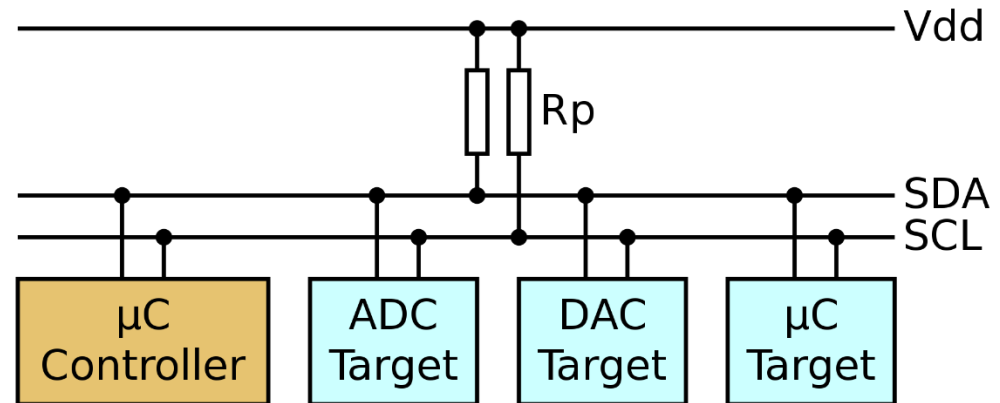
No Arduino, os pinos são mapeados como:

A4 - SDA

A5 - SCL

16 - SDA

17 - SCL



Fonte: https://en.wikipedia.org/wiki/I%C2%B2C#/media/File:I2C_controller-target.svg

Padrão I2C

Inter-Integrated Circuit (I2C), desenvolvido pela Philips em 1982

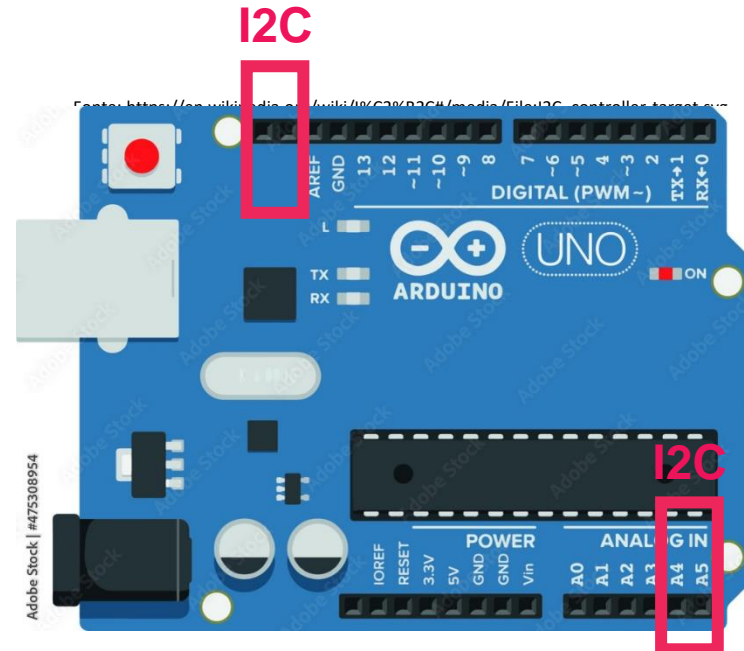
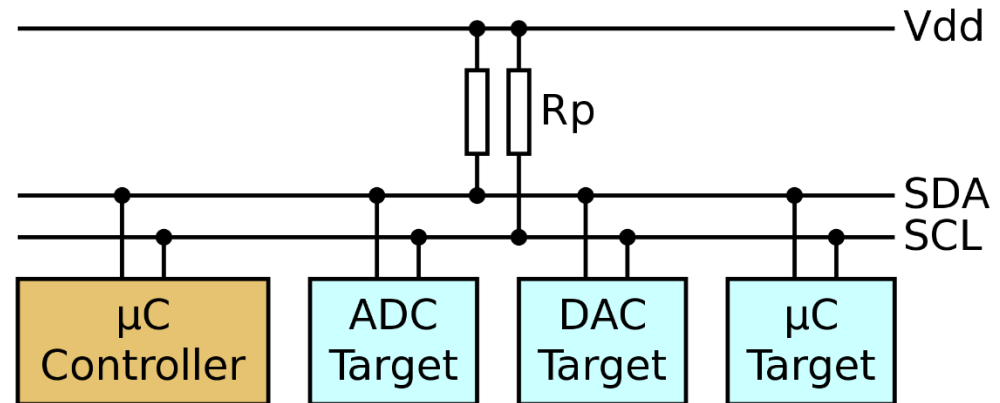
Usado para comunicação de curta distância (alguns centímetros)

Comunicação bidirecional, Half Duplex, com diferentes velocidades entre os periféricos

A comunicação é feita por dois “fios”;
SCL → Serial Clock
SDA → Serial Data

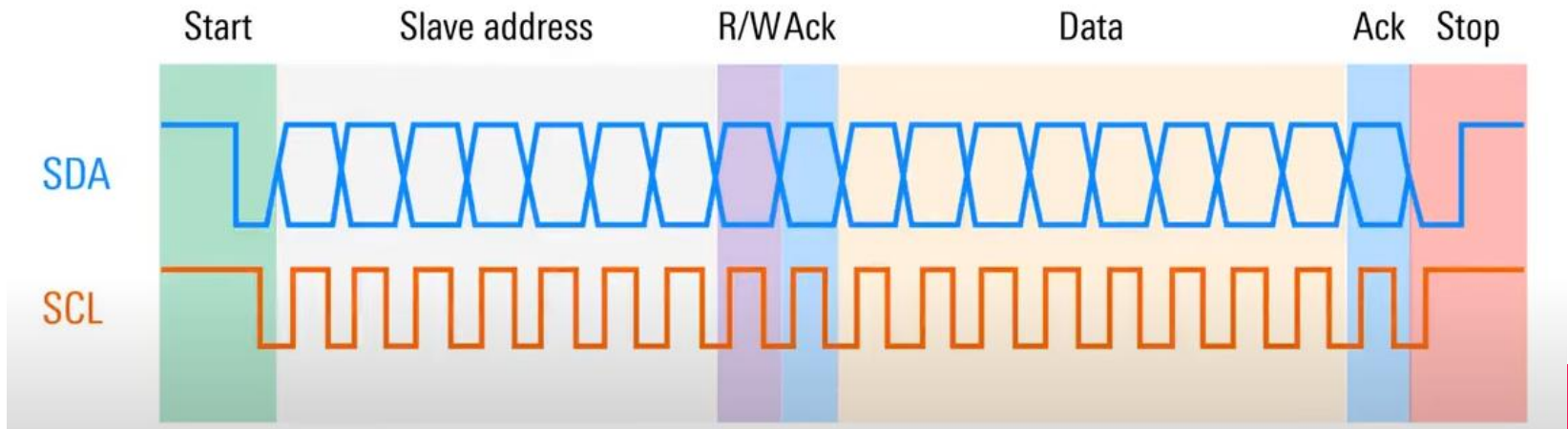
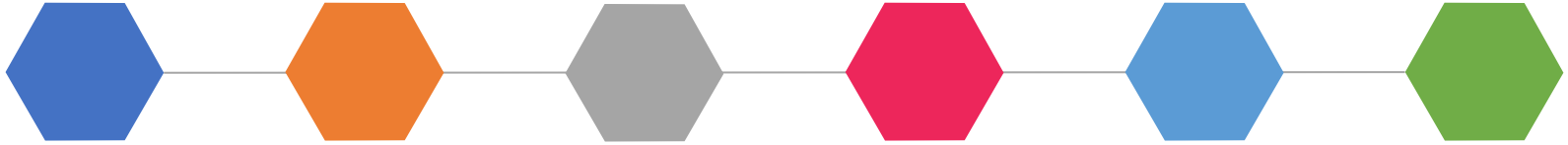
No Arduino, os pinos são mapeados como:

A4 - SDA
A5 - SCL
16 - SDA
17 - SCL



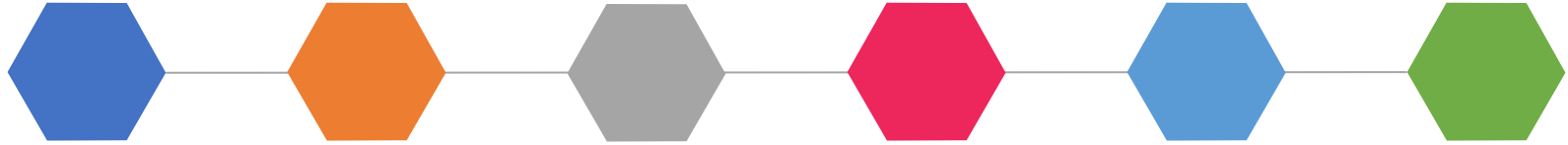
Padrão I2C

FIAP



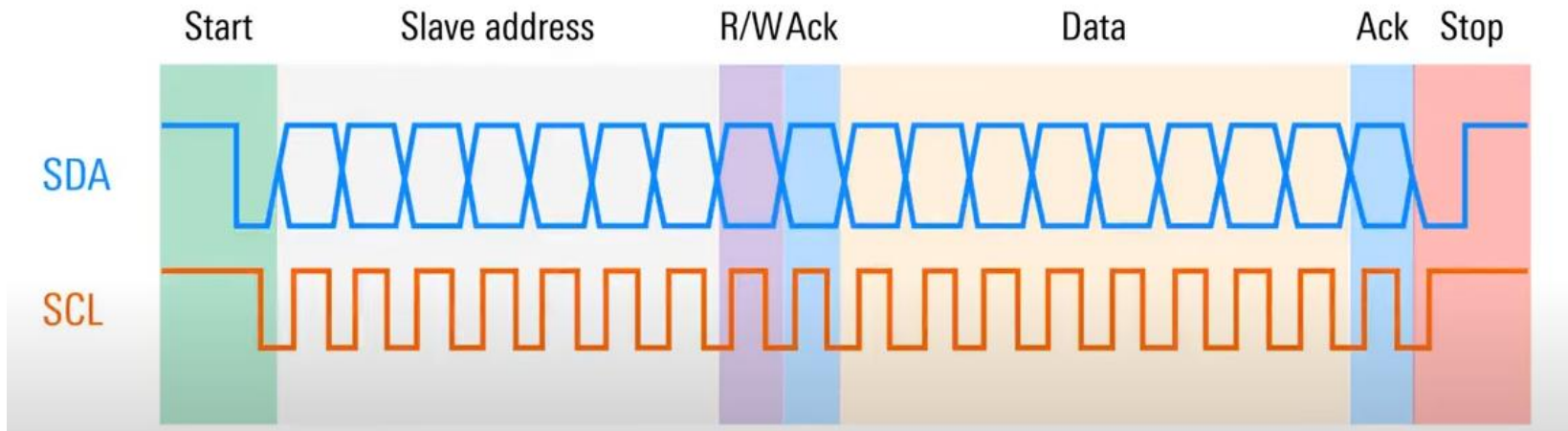
Padrão I2C

FIAP



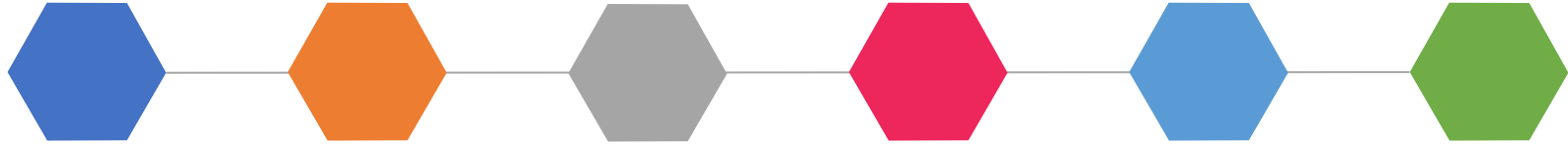
START

- Star ocorre quando um node coloca o Sinal SDA em baixa, seguido pelo sinal SCL;
- Isso indica que aquele node está requisitando o barramento;
- Em seguida, ele fornece o sinal de sincronismo no pino SCL;



Padrão I2C

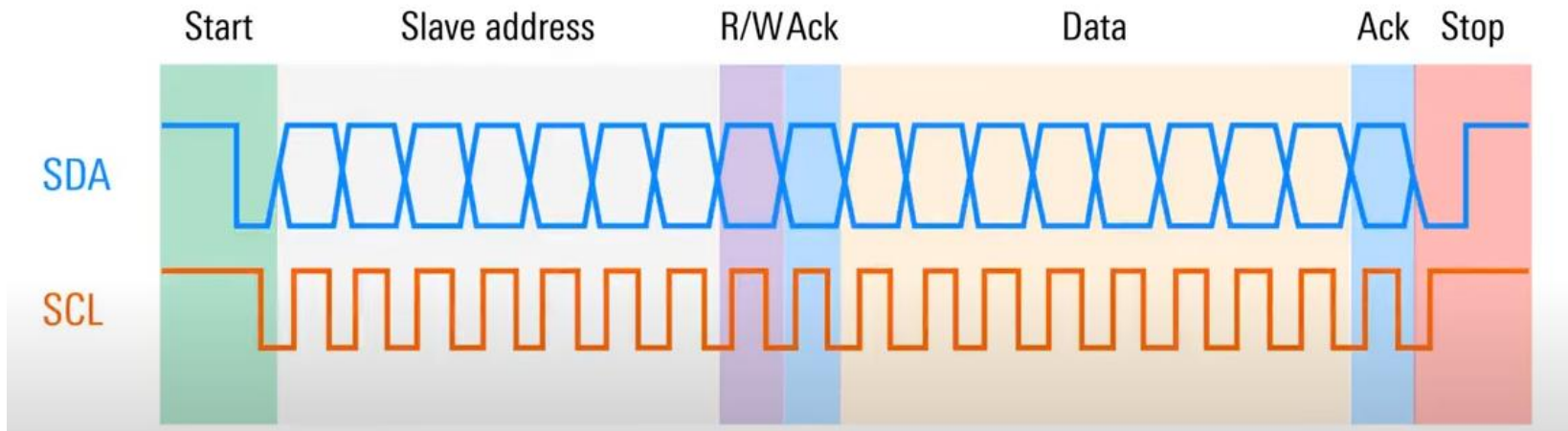
FIAP



START

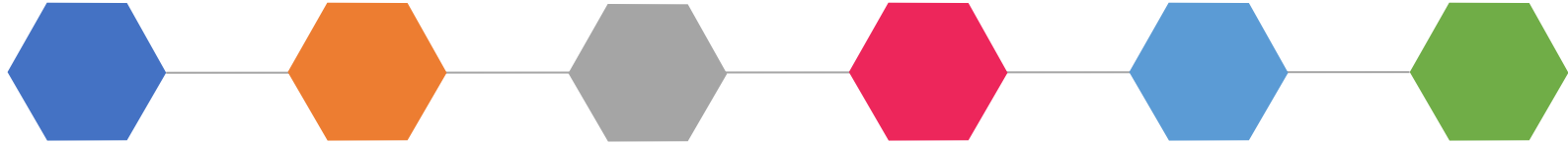
Slave Address

- Star ocorre quando um node coloca o Sinal SDA em baixa, seguido pelo sinal SCL;
 - Isso indica que aquele node está requisitando o barramento;
 - Em seguida, ele fornece o sinal de sincronismo no pino SCL;
- Em Seguida, ele informa o endereço do periférico com quem quer falar;
 - Normalmente, o endereço possui 7 bits, MSB;
 - Endereços podem ser "hard coded" ou parcialmente configuráveis por jumpers.



Padrão I2C

FIAP

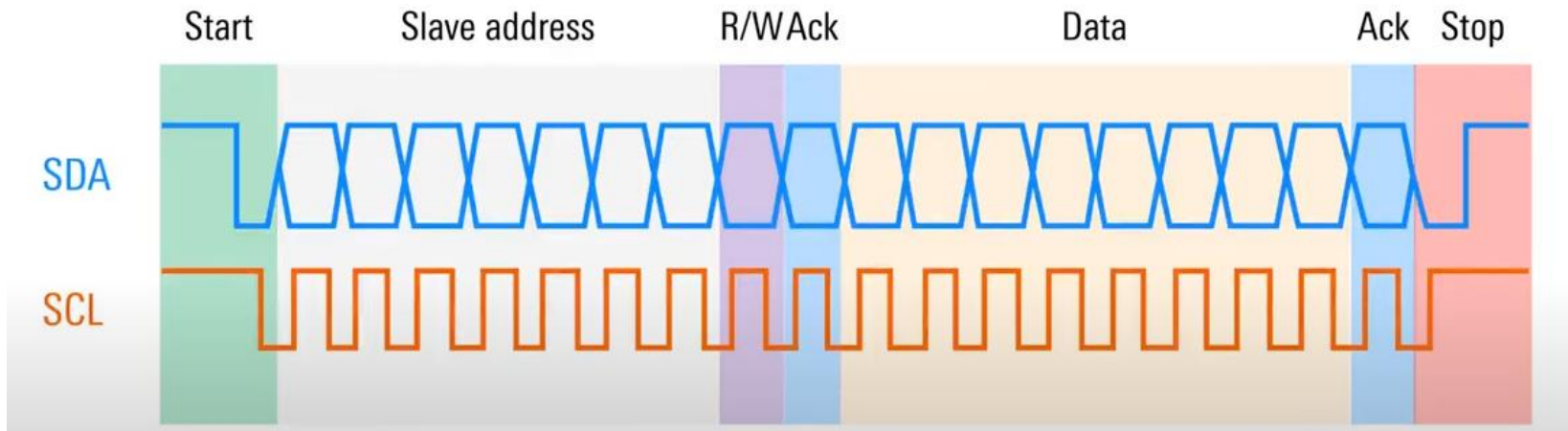


START

Slave Address

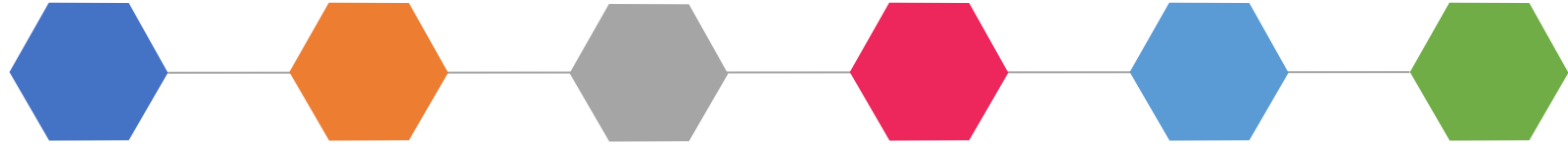
R/W Ack

- Star ocorre quando um node coloca o Sinal SDA em baixa, seguido pelo sinal SCL;
 - Isso indica que aquele node está requisitando o barramento;
 - Em seguida, ele fornece o sinal de sincronismo no pino SCL;
- Em Seguida, ele informa o endereço do periférico com quem quer falar;
 - Normalmente, o endereço possui 7 bits, MSB;
 - Endereços podem ser "hard coded" ou parcialmente configuráveis por jumpers.
- R/W indica Read or Write;
 - 0 → Controlador quer escreve no periférico
 - 1 → Controlador quer ler do periférico
 - ACK é enviado pelo receptor da mensagem:
 - 0 → Entendido;
 - 1 → Não Entendido



Padrão I2C

FIAP



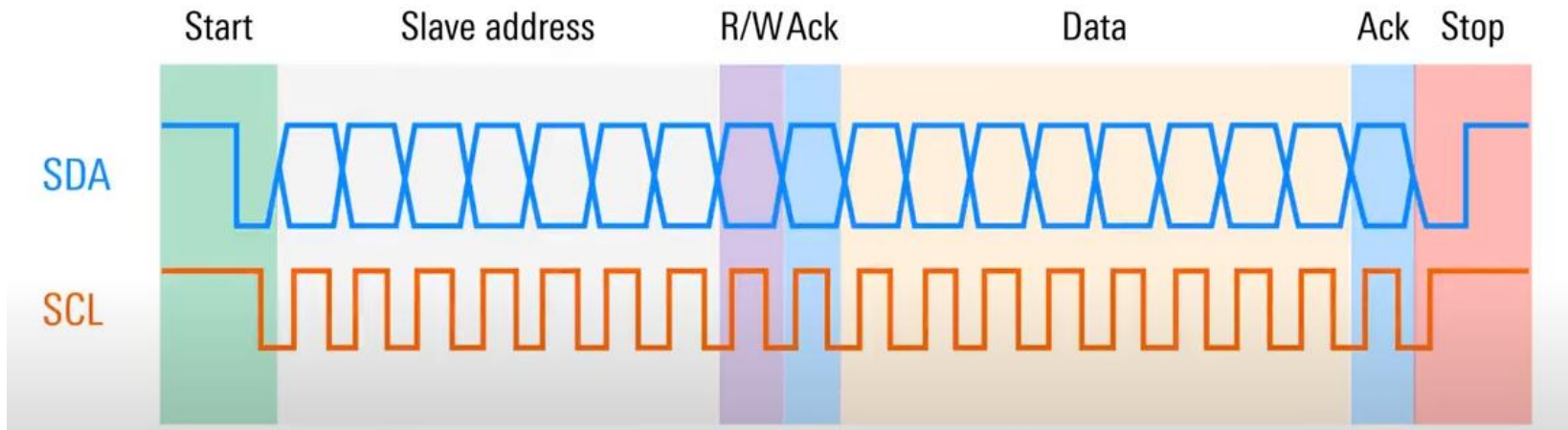
START

Slave Address

R/W Ack

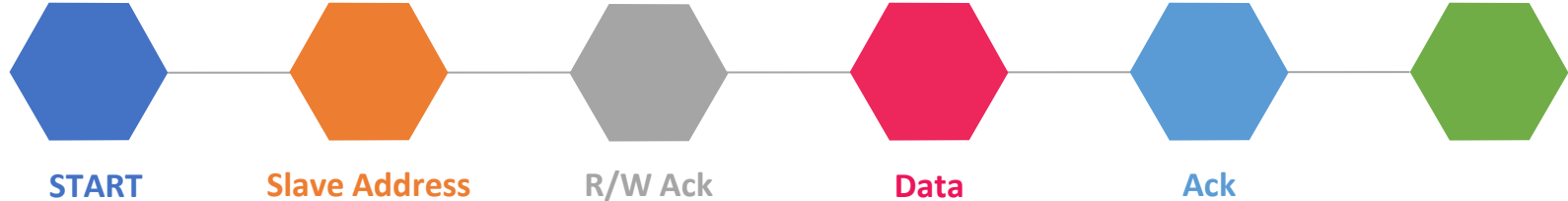
Data

- Star ocorre quando um node coloca o Sinal SDA em baixa, seguido pelo sinal SCL;
 - Isso indica que aquele node está requisitando o barramento;
 - Em seguida, ele fornece o sinal de sincronismo no pino SCL;
- Em Seguida, ele informa o endereço do periférico com quem quer falar;
 - Normalmente, o endereço possui 7 bits, MSB;
 - Endereços podem ser "hard coded" ou parcialmente configuráveis por jumpers.
- R/W indica Read or Write;
 - 0 → Controlador quer escreve no periférico
 - 1 → Controlador quer ler do periférico
 - ACK é enviado pelo receptor da mensagem:
 - 0 → Entendido;
 - 1 → Não Entendido
- Byte contendo a informação transferida entre o Controlador e o Periférico
 - Mensagens de 8 bits, MSB

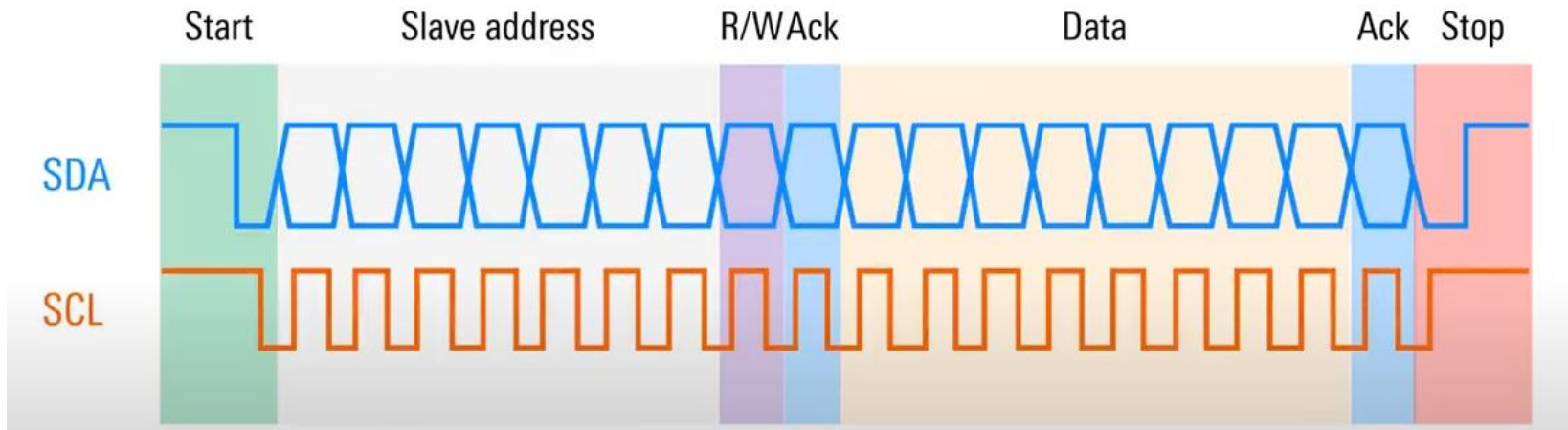


Padrão I2C

FIAP

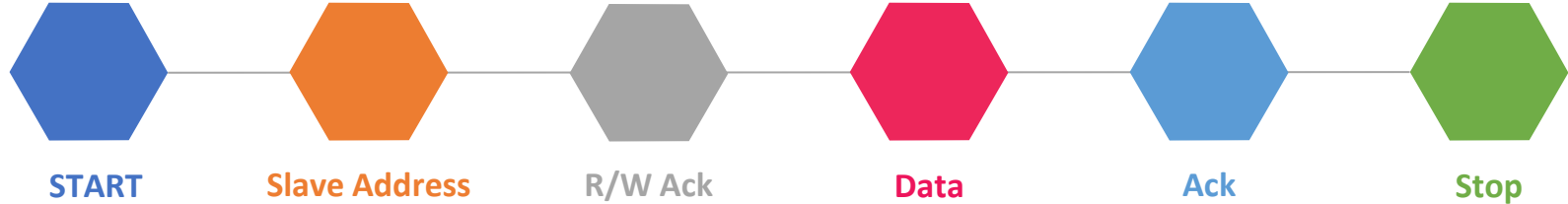


- Star ocorre quando um node coloca o Sinal SDA em baixa, seguido pelo sinal SCL; Isso indica que aquele node está requisitando o barramento;
 - Em seguida, ele fornece o sinal de sincronismo no pino SCL;
- Em Seguida, ele informa o endereço do periférico com quem quer falar;
 - Normalmente, o endereço possui 7 bits, MSB;
 - Endereços podem ser "hard coded" ou parcialmente configuráveis por jumpers.
- R/W indica Read or Write;
 - 0 → Controlador quer escreve no periférico
 - 1 → Controlador quer ler do periférico
 - ACK é enviado pelo receptor da mensagem:
 - 0 → Entendido;
 - 1 → Não Entendido
- Byte contendo a informação transferida entre o Controlador e o Periférico
 - Mensagens de 8 bits, MSB
- ACK é enviado pelo receptor da mensagem:
 - 0 → Entendido;
 - 1 → Não Entendido

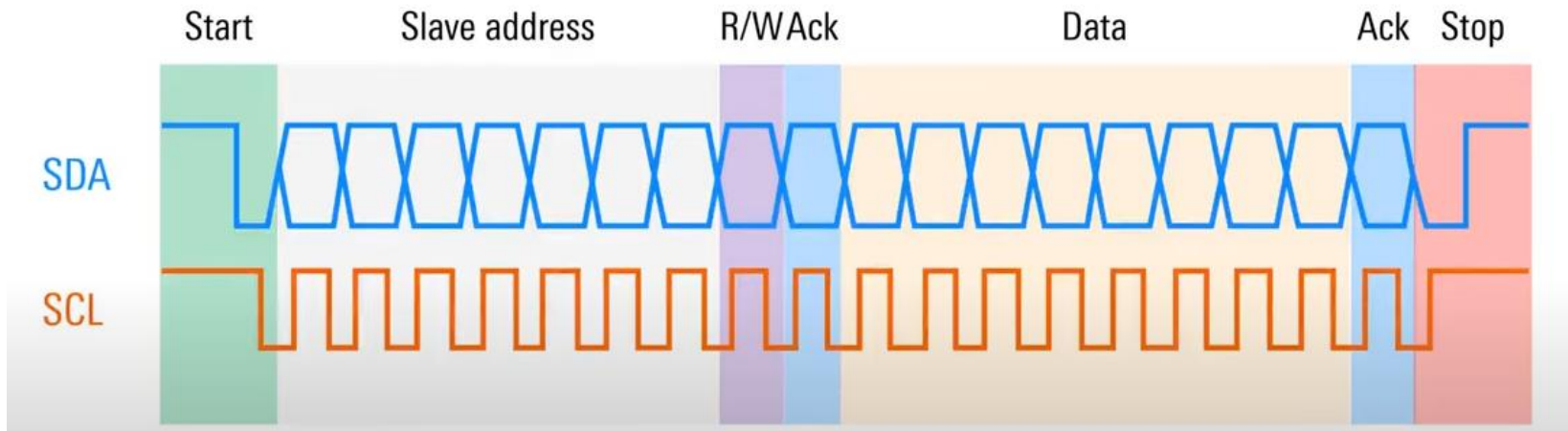


Padrão I2C

FIAP



- | | | | | | |
|---|---|--|---|---|--|
| <ul style="list-style-type: none"> • Star ocorre quando um node coloca o Sinal SDA em baixa, seguido pelo sinal SCL; • Isso indica que aquele node está requisitando o barramento; • Em seguida, ele fornece o sinal de sincronismo no pino SCL; | <ul style="list-style-type: none"> • Em Seguida, ele informa o endereço do periférico com quem quer falar; • Normalmente, o endereço possui 7 bits, MSB; • Endereços podem ser "hard coded" ou parcialmente configuráveis por jumpers. | <ul style="list-style-type: none"> • R/W indica Read or Write; • 0 → Controlador quer escreve no periférico • 1 → Controlador quer ler do periférico • ACK é enviado pelo receptor da mensagem: • 0 → Entendido; • 1 → Não Entendido | <ul style="list-style-type: none"> • Byte contendo a informação transferida entre o Controlador e o Periférico • Mensagens de 8 bits, MSB | <ul style="list-style-type: none"> • ACK é enviado pelo receptor da mensagem: • 0 → Entendido; • 1 → Não Entendido | <ul style="list-style-type: none"> • Stop ocorre quando um node coloca o Sinal SCL em alta, seguido pelo sinal SDA; • Isso indica que o barramento está livre; |
|---|---|--|---|---|--|

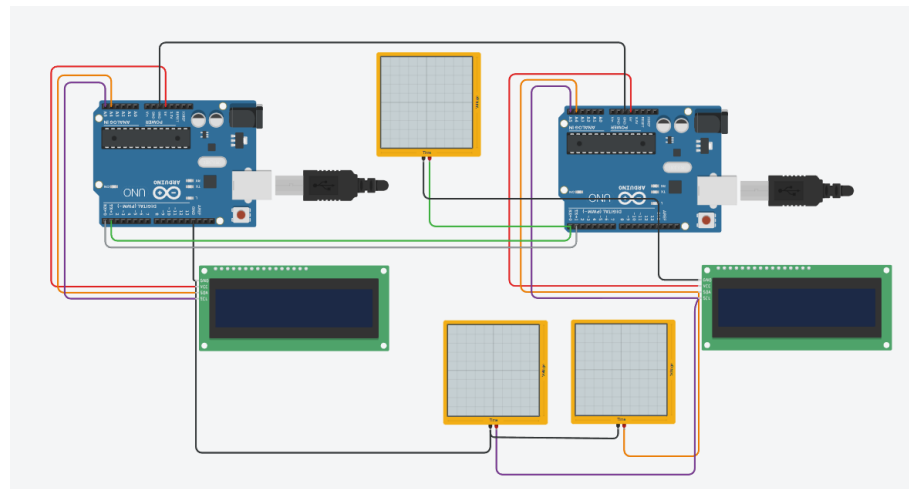


Laboratório – Padrões de Comunicação

Neste laboratório, vamos explorar dois padrões de comunicação no Arduino: UART e I2C.

Para isso vamos fazer dois Arduinos se comunicarem via UART e mostrar as informações num display I2C.

Para nos auxiliar na visualização dos protocolos, vamos usar três Osciloscópios, uma ferramenta que nos permite ver os sinais digitais



Material necessário:

- 2 Arduinos;
- 2 Displays LCD I2C;
- 3 Osciloscópios;
- Jumpers cables.



Link: [Projeto 14 – Padrões de Comunicação](#)

Exercícios

Altere o projeto do laboratório para que os Arduinos se comuniquem via SPI.

Pesquise a biblioteca SPI e veja os exemplos disponíveis no site do Arduino.

Para auxiliar na visualização dos protocolos, use os Osciloscópios.



Copyright © 2023 Prof. **Airton Y. C. Toyofuku**

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).