SPI vs. I2C vs. UART: Differences Between These Communication Interfaces

In an embedded system, communication plays an important role. Peripheral expansion is a high-power-consuming and highly complex system without knowing the communication protocols. All of the embedded systems make use of serial communication interfaces to communicate with the peripherals. Among them, the most common communication protocols in microcontroller development are SPI, UART, and IC2. These protocols have the significant benefits of low data loss and high speed.

This article has compared all three protocols, including their advantages, disadvantages, and uses. Thus, after reading this article, you can choose the correct protocol per your requirement.

In this article,

What is a Serial Peripheral Interface (SPI)?

ion interfaces of synchronous types. In an embedded
s the full-duplex communication protocol. It allows
PIC, ARM, and AVR controllers are some of the

and microcontroller, while the slaves are the

0

Contact Us

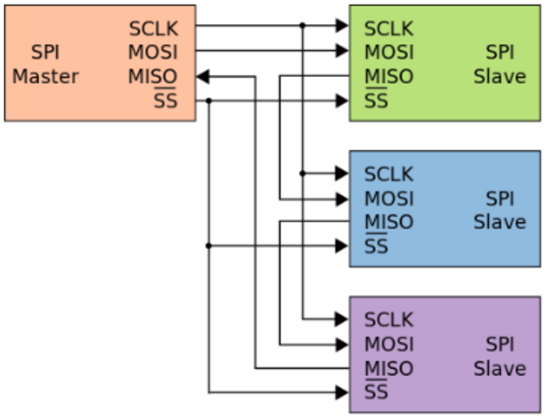
with the master device thus, SPI protocol's major benefit is the speed used where speed is crucial. Furthermore, SPI protocol's applications include SD cards, display modules, etc.

SPI supports two communication interface modes; point-to-point and standard mode. In point-to-point mode, a single controller follows the single slave, while in standards mode, a single master controller can communicate with two slave devices enabling the select chip lines.

SPI Working Principle

Two methods define the SPI working:

- The first method selects each device using the CS, which is the select chip line. Each of these devices needs the unique Chip Select line.
- The second method is daisy chaining. In this method, each device is connected to the other via the data out of one to the data in line or another.



SPI devices can connect to various unlimited devices. But the hardware selects line limits this connection. An SPI interface provides efficient, simple, point-to-point communication without addressing operations, allowing full-duplex communication.

SPI Working Protocol

- SPI has four communication ports:
- MISO – master data input, slave data output.
- MOSI – Master Data Output, Slave Data Input.
- SCLK – clock signal generated by the master device,
- NSS/CS – Slave-enabled signal, controlled by the master device or also called a select chip line.
- An I2C system has only one enable signal, while a multi-slave system needs multiple enable signals.

rs.

itted bit by bit. Front bits are high, and back bits are

sly over the SPI interface. The master device

faster communication medium than I2C and offers a few Mbps of data transfer speed.

Pros

- Supports full duplex communication.
- Simple and fast data transmission rate.
- Offers simple software implications.
- No, start and stop bits allow continuous data transmission with no interruptions.
- MOSI and MISO allow simultaneous data send and receive operations.
- The use of a master clock doesn't require precious oscillators.

Cons

- More use of slave devices complicates the wiring.
- It has a single master device.
- Limits the number of slave's devices to connect with the Master.
- No error check mechanism.
- Don't have an acknowledgment line of data receiving a message.

SPI in Microcontrollers

SPI Driver/Adapter

It is one of the easy tools to control SPI devices. It is compatible with all operating systems. The live logic analyzer displays the SPI traffic on the screen. The operating voltage of the SPI driver is 3.3 V -5 V.

MCP 3008

It is a 10-bit ADC having 8 channels. Moreover, it connects to the Rasberry Pi with the help of an SPI serial connection.

SPI Seeeduino V 4.2

A master Arduino and a slave Arduino can communicate using SPI serial communication interfaces with Arduino. The main aim is to communicate over a short distance at a higher speed.

What is I2C?

Inter-integrates circuit (I2C) is the serial communication protocol. This protocol is effective for the sensors and modules but not for the PCB device communications. The bus transmits information bidirectionally between connected devices using only two wires. With them, you can connect up to 128 devices to the mainboard while maintaining a clear communication pathway between them. They are ideal for projects that require many different parts to work together (e.g., sensors, pins, expansions, and drivers). Also, I2C speed depends on the speed of data,

also uses two-wire interfaces for connecting low-

aces, etc.

ne acceptance port, respectively. The clock line CL is

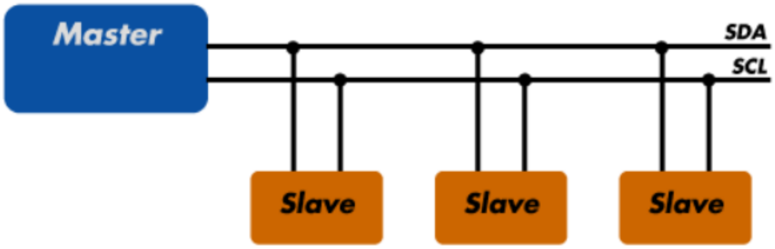
received through SDA lines.

ock signal. This clock signal opens the transferred

device, and any address devices are considered slave devices.

want to send data to the slave. This will stop the data transfer. At the same time, the Master should address the slave if it wants to receive the data from the slave devices. Finally, the receiver terminates the receiving process by receiving the data sent by the slave.

Additionally, the host generates the timing clock and completes data transfers. Power supply connections must also be made through pull-up resistors. Both lines operate at high power levels when the bus is idle.



I2C Working Protocol

Data Transmission Method

The master devices send the signal to each of the slaves connected. This is carried out by switching SDA and SCL lines from a high to a low voltage level.

The master device is responsible for sending 7 or 10 bits of the address, including the read/write bit, to the slave for communicating.

The slave compares the address, and if it matches, the ACK acknowledgment bit returns. This would switch the SDA line low for one bit, else there is no change in the SDA line, leaving it high.

The master also transceives the data frame, and receiving device acknowledges the successful transmission by sending an ACK bit. The master device sends a stop signal to stop the data transmission, where the SCL switch is high before the SDA switch to high.

Clock Synchronisation

Data transmission requires clock signals from every master on the SCL line. Data in I2C transmissions remain valid only during the high period of the clock.

Mode of Transmission

Quick mode and high-speed mode are the two data transmission modes.

Quick Mode

Devices and transceive data rate at 400 kbit/s. They must sync with this transmission to slow it down and extend the SCL signal's low period.

ckward compatibility with the standard mode. This modes.

- It offers multi-slave and multi-master communication.
- This protocol is flexible and adaptable too.

Cons

- I2C is a bit slower protocol because of the need for pull-up resistors.
- It takes up more space.
- The architecture is more complex with the increasing number of devices.
- This protocol is half-duplex, which is quite problematic and requires different devices for complete communication.

I2C in Microcontroller

Raspberry Pi's 4 channel 16 bit ADC

It is the Seeed product that is compatible with the Rasberry Pi. This 16-bit ADC is used when a more precise ADC is required in the circuit.

I2C Driver/Adapter

It is an open-source tool that is easy to use. Usually, it is used for controlling I2C devices. It is compatible with all OS. It offers a built-in color screen that provides the live dashboard of I2C activity. Thus, when an I2C drive connects to the I2C bus, it displays the traffic on the screen. Besides, it can help to debug the I2C issues and troubleshoot them.

I2C Arduino

I2C Communication interfaces between two Arduino boards are also possible. It is used for short-distance communication interfaces and uses the synchronized clock pulse. This I2C Arduino is used while communicating with the other sensors and devices that need to send the information to the Master.

PCF 8574

It provides the general purpose remote I/O expansion through two bidirectional I2C buses.

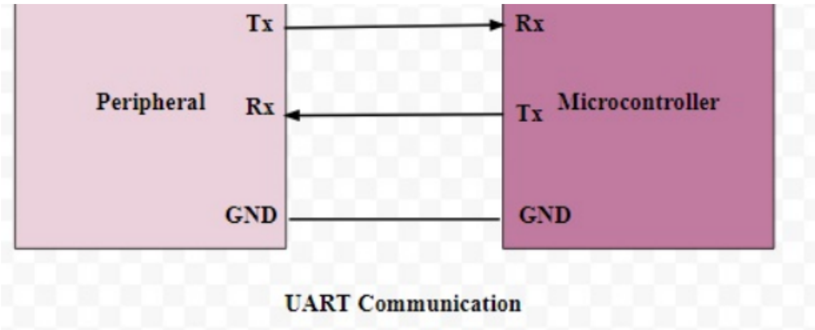
What is UART?

Universal Asynchronous Reception and Transmission (UART) is the serial communication interface that provides the communication interface of the host with that of the auxiliary device. This protocol enables full-duplex serial communication. UARTs are chips designed for asynchronous communication.

Using start and stop bits, it transmits data bits in order from the smallest to the largest. Driver circuits control the electric signaling levels external to the UART.

Thus, this UART protocol supports bi-directional, serial, and asynchronous data transmission. Through data lines 0
of two data lines, transmit and receive.

lem between serial devices and computers. The UART
doesn't synchronize its output with the receiving
g start and stop bits. So that the receiving UART
inning and end of the data packet.



UART works in three various ways;

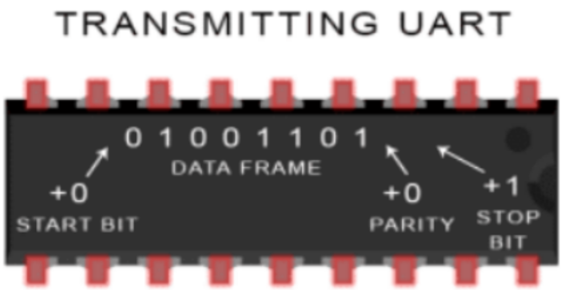
- Simplex: In a simplex system, data is transmitted in only one direction.
- Half-duplex: In a half-duplex system, data is transmitted in both directions but not simultaneously.
- Full-duplex: In a full-duplex system, data is simultaneously transmitted in both directions.

After the successful connection, data is transmitted from the TX pin of the transmitting UART to the RX pin of the receiving pin of UART. UART transmits serial data from the master device to the receiving UART by converting it from parallel. UART has no clock signal thus, UART transmits start and stops bits to represent the beginning and end of a message.

The functioning of the UART is possible only if the baud rate or the speed of data transmission of two UARTs syncs with each other. Further, the baud rate difference of more than 10 % is unacceptable as it makes the data useless. UART transmits data in the form of packets. Each of these data packets has 1 start bit, depending on the UART second, the 5 to 9 data bits, the parity bit for informing the receiver if the data is changed or not, and finally, the stop bits (1 or 2).

In summary, the UART data transmission includes

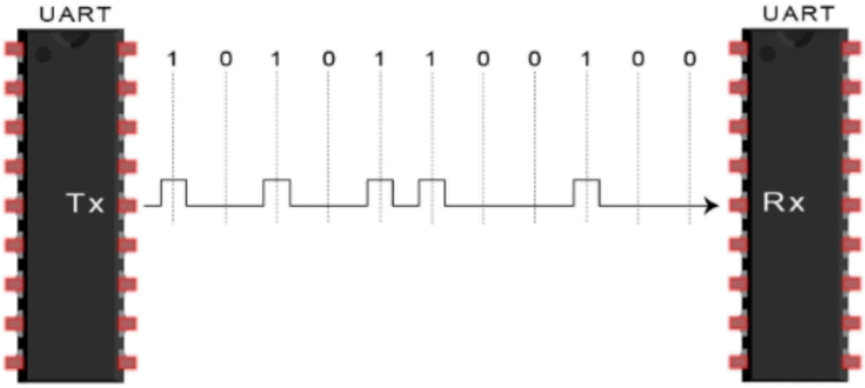
- Initially, transmitting UART gets the data from the data bus in parallel form.
- After receiving data by transmitting UART, it configures the bit by adding star, parity, and stop bits in the data frame.



ART and UARTs sample data at the predetermined

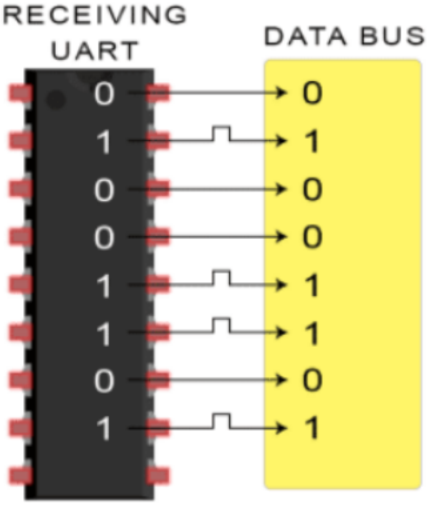
0

Contact Us



- UARTs receiving data frames discard the start, parity, and stop bits.

- Finally, UARTs on the receiving end convert serial data into parallel and transmit it to a data bus which completes the data transmission in UART.



UART Working Protocol

Data Transmission and Receiving

- Initially, UARTs need data buses, such as CPUs, to receive data for transmission.
- UART then adds three bits called start, parity, and stop bits.
- Next, it adds the three bits, namely, the start bit, parity bit, and stop bit. The TX pin will transmit the three data packets to the receiving UART.
- Finally, when the transmitting UART exhausts its data, data transmission stops.

ents like

- Receiving Timeout

FIFO Operation

UART module contains two of the 16-byte FIFOs for data transmission and receiving. The UART receives 4 data when the FIFO triggers an interrupt at 1/4.

In transmitting FIFO, the data transmission process starts immediately after entering data. While this process executes, all the data continues to enter this FIFO until its full. After then, this transmitting FIFO sends data until it's empty. An extra slot adds after data is clear in the transmitting FIFO.

In receiving FIFO, hardware stores the data, which is an automatic process. Thus there should be enough space so that there is no problem with storage in receiving FIFO.

There is no chance of data loss in this FIFO as any issue is cross-checked in sending and receiving the files.

Loopback

A UART has an internal loopback function that sends data from TX to RX, which can be used for diagnostics or debugging.

Serial Infrared Protocol

- There is an encoder and decoder module for IrDA Serial Infrared (SIR) on UART. SIR modules translate asynchronous UART data to semi-duplex serial interfaces for IrDA.
- This device provides the digitally encoded output and decoded input to a UART. IrDA SIR physically connects the UART to an infrared transceiver.

Pros and Cons of UART

Pros

- It only uses two lines or wires helping to clean circuitry
- It doesn't require a clock signal.
- It has a parity bit for error checking.
- It is an easy and widely used method of communication.

Cons

- It has a limited data size of 9 bits only.
- It doesn't support multiple Master and slave devices.
- The baud rate or the data transmission speed should be below 10 % or each other.
- The data transmission rate is lower compared to the remaining protocols.

Example of UART in Microcontroller

to-serial UART interfaces provide a USB-to-serial
number of external components while using the
bus converter can convert USB files into serial files.

0

Contact Us

this USB CP2102 Serial Converter.

UART Seeduino V4.2

The Arduino boars contain at least one serial port that usually communicates the digital pins Tx and Rx through USB. It is compatible to work with Arduino board. This board can be programmed through a MicroUSB cable.

Base Shield V2

There is no doubt that Arduino Uno is the most popular Arduino board. However, it can be a bit frustrating once you need to connect many sensors and LEDs to the board, and your jumper wires are tangled. This product eliminates the need for breadboards and jumpers. Adding all the grove modules to the Arduino Uno is convenient since the baseboard has grove connectors.

Difference between SPI, I2C, and UART

Speed

SPI protocol is faster among all three communication interfaces, whereas UART is the slowest.

Number of Devices

SPI can take an unlimited number of devices as long as hardware complexity is not a problem. At the same time, I2C can take 127 devices, but UART support just 2 devices.

Number of Wires

For connecting Master and slave devices, SPI has 4 wires, I2C has 2 wires, and the UART has only one.

Complexity

UART beats both SPI and I2C in terms of complexity. It is simple compared to the remaining two protocols, whose complexity increases with the number of devices.

Communication Types

UART is one of the asynchronous communication interface types with no clock signal, whereas I2C and SPI are synchronous protocols.

Features	UART	SPI	I2C
Full Form	Universal Asynchronous Receiver/Transmitter	Serial Peripheral Interface	Inter-Integrated Circuit
Pin Designations	TxD: Transmit Data RxD: Receive Data	SCLK: Serial Clock MOSI: Master Output, Slave Input MISO: Master Input, Slave Output elect	SDA: Serial Data SCL: Serial Clock
		ports about 10 Mbps s	I2C supports 100 kbps, 400 kbps, and 3.4 Mbps. Some variants also support 10 Kbps and 1 Mbps.
		us	Synchronous

interfaces			
Distance	Lower about 50 feet	highest	Higher
Master Device	One	One	One or more than One
Clock	No use of Common Clock signal. Both devices will use their independent clocks.	There is one common serial clock signal between master and slave devices.	There is a common clock signal between multiple masters and multiple slaves.
Hardware complexity	lesser	less	more
Protocol	For 8 bits of data, one start bit and one stop bit are used.	Each manufacturer has got their own specific protocols to communicate with peripherals. Hence, a datasheet is a must-to-know read/write the protocol for SPI communication to be established.	It uses start and stops bits. It uses an ACK bit for every 8 bits of data which indicates whether data has been received or not.
Software addressing	There is no need for addressing since this is a one-to-one connection between two devices.	Any slave connected with the master is addressed by a slave select line. A master device will have 'n' slave select lines for 'n' slaves.	Any slave connected with the master is addressed by a slave select line. A master device will have 'n' slave select lines for 'n' slaves.

Best Communication Interfaces among SPI, I2C, and UART

All three communication peripherals have their specific upsides and downsides. The selection of the best protocol for your project depends on your requirements. If you are concerned about the speed, you may go for the SPI, while if you want the simple circuitry with one drive, you may go for the UART communication protocol.

While if you want to connect numerous devices with no complex array, you can go for the I2C.

Conclusion

Thus, this article presents communication protocols. Each protocol has its advantages and disadvantages with unique features. Thus, you need to remember that the devices, sensors, or modules should support the communication protocols.

NextPCB can help you with all PCB-related services. You don't need to worry about your devices' communication protocols. We ensure that you receive PCB with proper communication interfaces per your requirements. Contact us for any questions or request a quote to get started.

0

Contact Us

Please enter your content

Comments

Products & Service

PCB Quote
PCB Assembly Quote
PCB Stencil Quote
PCB Capabilities
PCB Assembly Capabilities
HQDFM

Exclusive customer

FAQ
PCB Vocabulary
Products & Capabilities

About NextPCB

About Us
Why Us
Payment Methods
Shipping Methods
Certificate
Blog

Support

Terms of Service
Privacy Policy
Return Policy

Payment Methods

Shipping Methods

Certificate

Social Media

HQonline and NextPCB are both subsidiaries of
Shenzhen Huaqiu Electronics Co., Ltd.

0

Contact Us