Notes on Serial Communication Protocols for Hardware Hacking Module

# Serial Communication Protocols

Serial Communication Protocols are a type of communication protocol used for data transmission via a serial communication link. Serial communication means that data is transmitted in series or one bit at a time over a single channel.

This contrasts parallel communications that use multiple channels to transmit multiple pieces of data at once. While parallel communications allow for faster data transmission rates, some of the benefits of serial communications follow:

* Due to only one communication channel the wiring tends to be simpler than that of parallel communications.
* Serial links also allow for longer cable links also due to their simplicity.
* Serial communication hardware is cheaper.
* Many devices and microcontrollers have built-in compatibility for serial communication protocols. This adds ease in integrating into existing systems.
* Serial communication is less affected by noise and signal interference, which reduces the number of errors and improves reliability.

While both serial and parallel communications are susceptible to hacking, serial communications may be easier to compromise. Due to their simplicity, a hacker needs only to gain access to a single channel to intercept, manipulate, or interrupt data transmission.

## Inter-Integrated Circuit (I2C) Protocol

* Voltage at the Common Collector (VCC)
* Ground (GND)

## Serial Peripheral Interface (SPI) Protocol

* Stuff goes here.

## Universal Asynchronous Receiver/Transmitter (UART) Protocol

* Stuff goes here.

## Controller Area Network (CAN) Protocol

* Stuff goes here.

## Universal Serial Bus (USB) Protocol

* Stuff goes here.

# References

## Wikipedia & Miscellaneous References

* <https://en.wikipedia.org/wiki/I%C2%B2C>
* <https://en.wikipedia.org/wiki/Serial_Peripheral_Interface>
* <https://en.wikipedia.org/wiki/Universal_asynchronous_receiver-transmitter>
* <https://en.wikipedia.org/wiki/CAN_bus>
* <https://en.wikipedia.org/wiki/USB>
* <https://youtu.be/i_HMmKIh1Nk>
* <https://www.youtube.com/watch?v=IyGwvGzrqp8&t=143s>
* <https://www.youtube.com/watch?v=B5rQSoOmR5w>

## Inter-Integrated Circuit (I2C)

* <https://www.nxp.com/docs/en/application-note/AN10216.pdf>
* <https://www.ti.com/lit/ml/slva704/slva704.pdf?ts=1679934516663&ref_url=https%253A%252F%252Fwww.google.com%252F>
* <https://cujo.com/hardware-hacking-101-e01-i2c-sniffing/>
* <https://www.instructables.com/Arduino-I2C-Sniffer/>
* <https://www.digikey.com/en/maker/projects/raspberry-pi-pico-rp2040-i2c-example-with-micropython-and-cc/47d0c922b79342779cdbd4b37b7eb7e2>
* <https://www.youtube.com/watch?v=j9yx8LOslng>
* <https://learn.sparkfun.com/tutorials/i2c/all>
  + Image
* <https://www.ti.com/lit/ml/slva704/slva704.pdf?ts=1680254815867&ref_url=https%253A%252F%252Fduckduckgo.com%252F>

## Serial Peripheral Interface (SPI)

* Reference

## Universal Asynchronous Receiver/Transmitter (UART)

* Reference

## Controller Area Network (CAN)

* <https://www.ti.com/lit/ml/sloa101b/sloa101b.pdf?ts=1679922241944>

## Universal Serial Bus (USB)

* Reference