Electronic Hardware

Onboarding: Communication Protocols

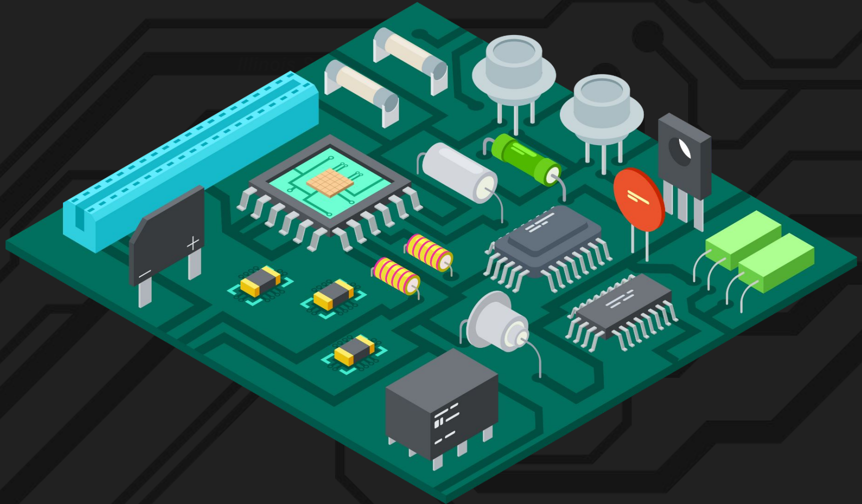
#004

2023

Introduction

Agenda:

1. Logic Levels
2. Pull Resistors
3. Push-Pull & Open-Drain
4. Parallel vs Serial
5. UART
6. SPI
7. I2C
8. SD/MMC
9. PWM



Goal:

Understand the various ways embedded digital components communicate with one another

Slide Designer: Peter Giannetos

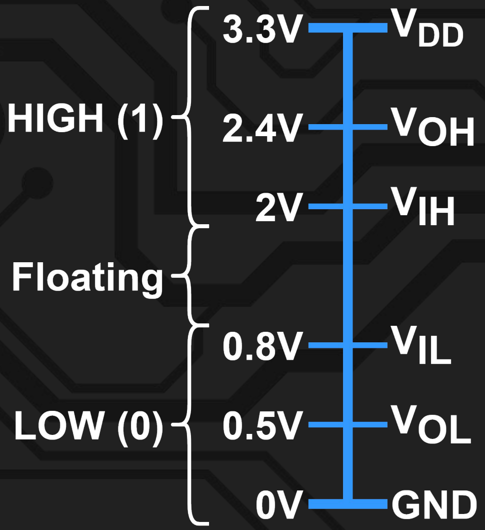
Logic Levels

Intro

- The binary ON (1) OFF (0) states of a digital circuits via voltage
- Different logic "levels" exists 5V, 3.3V, 1.8V referring to V_{DD} value
  (Need logic level shifting chips to convert between levels)

Thresholds

- V_{DD} - Power supply voltage, max voltage
- V_{OH} - Output High, minimum output voltage for high state
- V_{IH} - Input High, minimum input voltage for high state
- V_{OL} - Output Low, maximum output voltage for low state
- V_{IL} - Input Low, maximum input voltage for low state
- GND - Common voltage reference, lowest voltage
- *Different per logic level! 5V, 3.3V, etc…*



3.3V Logic Families
(Based on 74LVT04 Hex Inverter)

Pull Resistors

Pull-Up

- A resistor used to pull a line up to voltage supply

Pull-Down

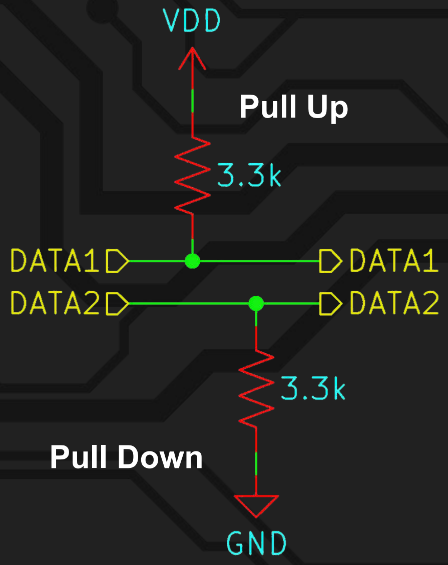
- A resistor used to pull a line down to ground

Pin States

- Active High: A high signal actives the system or specific function
- Active Low: A low signal actives the system or specific function
  (Pin may be show with line over name Ex: $\overline{CS}$)
- High Impedance: A circuit that is not driven high or low and is floating, requires external circuit to change its state
  (Impedance = Resistance + Reactance)
- Floating: Undefined, achieved when pin isn't connected to V_{DD} or GND
  (Unintentional floating circuits may behave unexpectedly)

Note

- Large resistors lead to weak pull-ups and pull-downs.



Push-Pull & Open-Drain

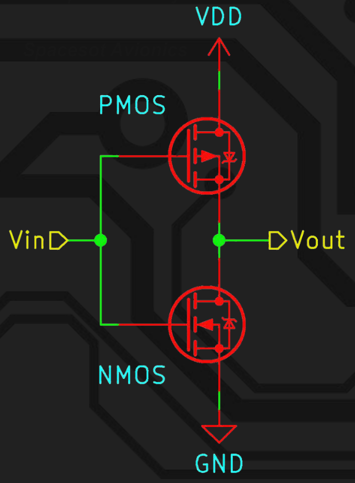
How do output pins change from high to low?

Push-Pull

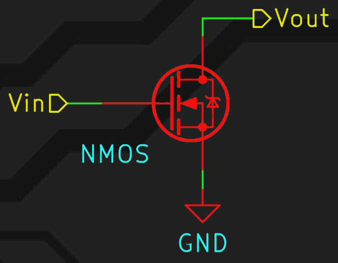
- Two transistors switch between load voltage and GND
- Commonly used for unidirectional lines (SPI, UART)

Open-Drain

- One transistors ties the output to GND
- An external resistor pulls the line high by default
- Commonly used for bidirectional lines (I2C)



Push-Pull



Open-Drain

Push-Pull

Push

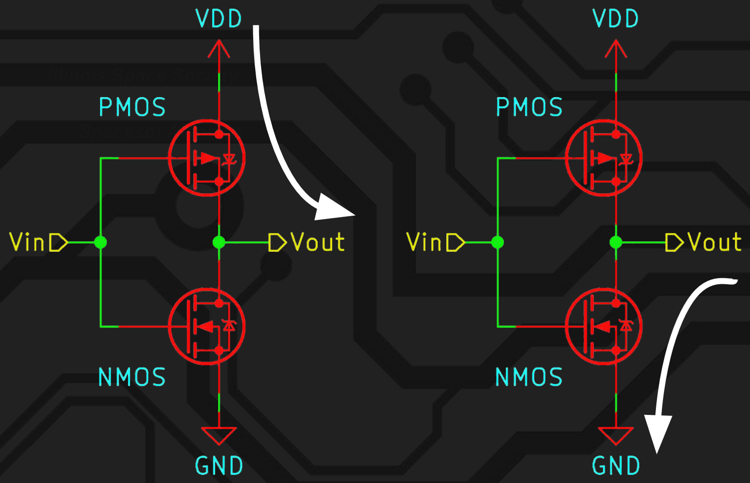
- When the output pin is driven to 1 transistors
- "Sourcing current to the load"

Pull

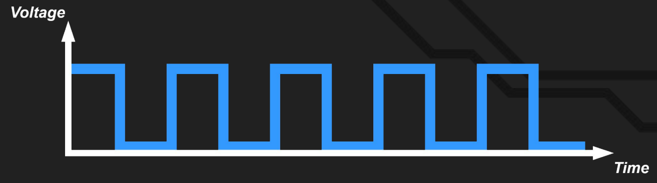
- When the output pin is driven to 0 by the transistors
- "Sinking current from the load"

Notes

- Consistently driven ≈ better performance
  (Sharper rising edge slope)



Push **Pull**



Open-Drain

Open

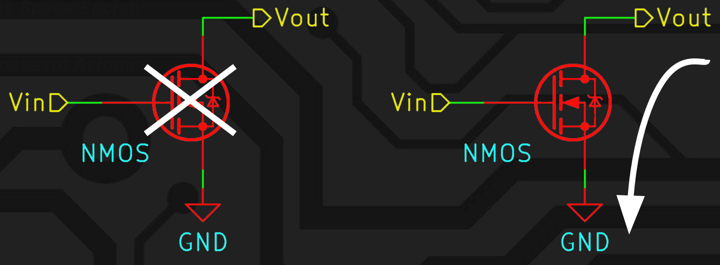
- Line is left floating and pulled up by external resistors
- High impedance (Hi-Z) state

Drain

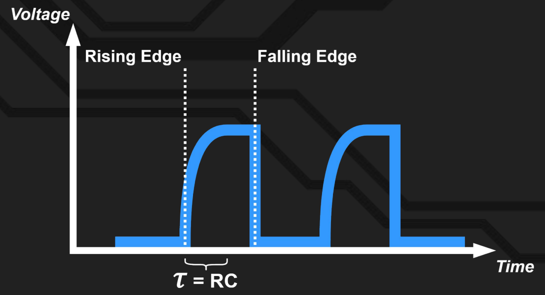
- When the output pin is driven to 0 by the transistors
- "Sinking current from the load"

Rising Edge Slope

- Wires naturally have capacitance so when combined with pull-up resistors it creates a low pass filter leading to curved rising edges
  (Transistors in Push-Pull have less resistance so faster rising edges)
- More resistance = Slower rising edge speeds & more noise
- Less resistance = More power consumption



Open **Drain**



Parallel vs Serial

Parallel

- Each bit in a piece of data is transferred simultaneously

Serial

- Each bit in a piece of data is transferred sequentially
- Slower than parallel but requires less hardware connections

Synchronous

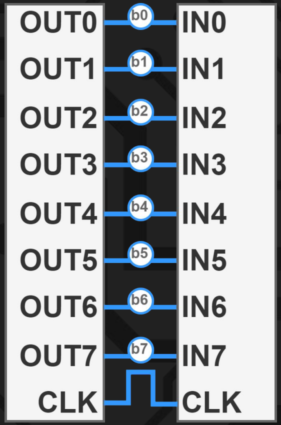
- A common clock is shared amongst all devices on the data bus
- Requires at least one extra wire

Asynchronous

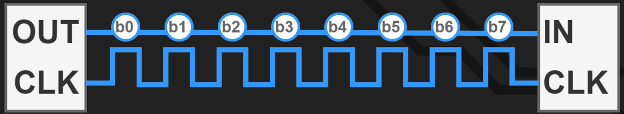
- Data is transferred without a shared common clock

Baud Rate

- The speed data is transferred over serial
- No set limit as long as both devices operate at the same rate
- Standard Rates:
  - 9600, 1200, 2400, 4800, 19200, 38400, 57600, 115200



Synchronous Parallel



Synchronous Serial

Serial Frames

Data Chunk

- The section where actual data is sent
- Variable length, depends on protocol

Synchronization

- Start: A flag used to indicate a new packet
- End: A flag used to indicate the end of a packet

Address *(Sometimes)*

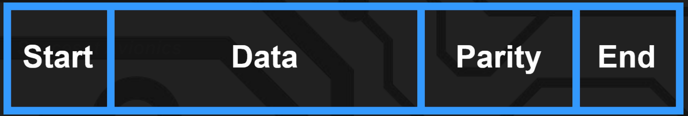
- Data to indicate destination on a multi-device shared bus

Parity

- Low level error checking

Endianness

- The order bits are sent
- Big-Endian
  - Most Significant Bit (MSB) sent first
  - MSB stored at smallest memory address
- Little-Endian
  - Least Significant Bit (LSB) sent first
  - LSB stored at smallest memory address



Big Endian

Data	0	1	0	1	0	0	1	0
Position	7	6	5	4	3	2	1	0
	MSB							LSB

Little Endian

Data	0	1	0	1	0	0	1	0
Position	0	1	2	3	4	5	6	7
	MSB							LSB

UART: Universal Asynchronous Receiver/Transmitter

Protocol Signals

- TX: Transmit
- RX: Receive

Features

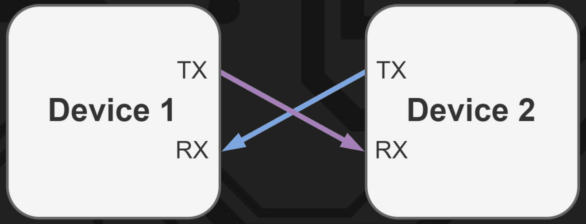
- Asynchronous
- Direct connections
- Full-duplex *(TX & RX simultaneously)*

Common Errors

- TX → RX not TX → TX
- Baud rate mismatch

Implementation

- Converts between serial and parallel *(Internally)*
- May use first-in-first-out (FIFO) buffer to store data



SPI: Serial Peripheral Interface

Protocol Signals

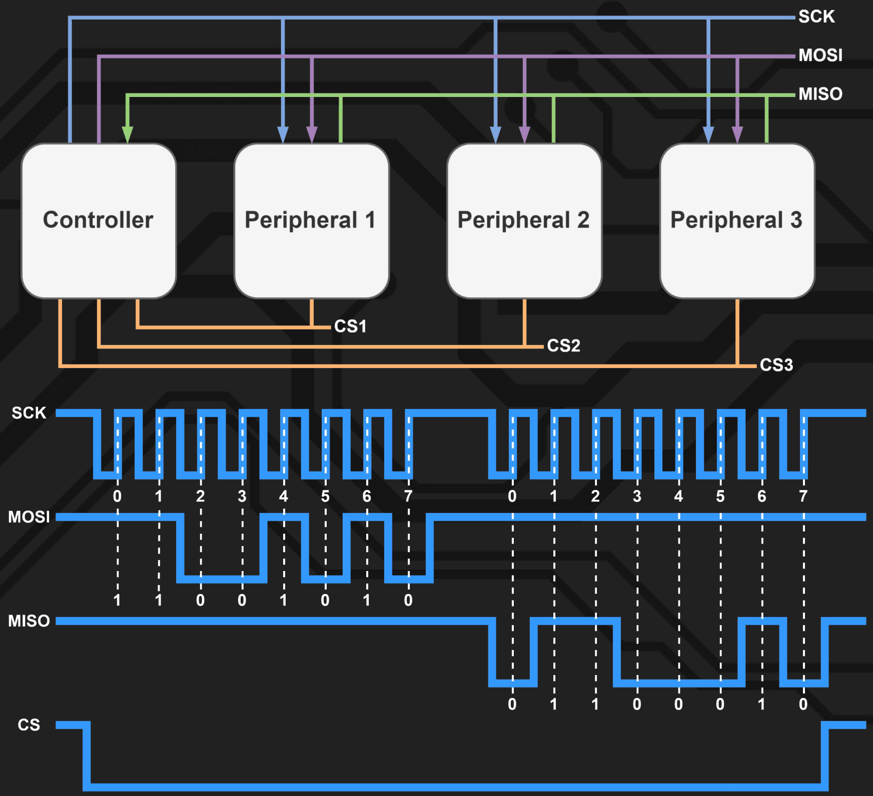
- SCK: Serial Clock
- MOSI: Master Out Slave In
- MISO: Master In Slave Out
- CS: Chip Select

Features

- One controller many peripherals
- Many different speeds depending on hardware
- Extra data line versions (Octo-SPI & Quad-SPI)
- Synchronous
- Full-duplex

Chip Select

- Turns on peripheral
- Only one active peripheral
- Active low



SPI: Serial Peripheral Interface

Implementation

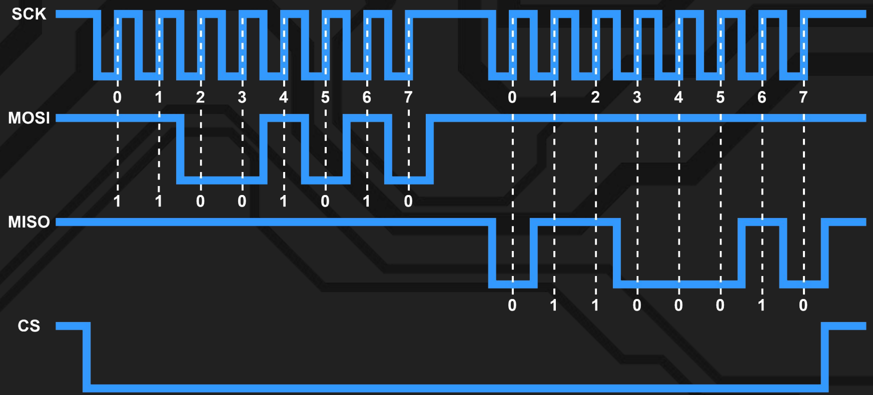
- Very simple implementation with shift register
- Single controller that generates clock
- Data sampled on either rising or falling edge *(Check datasheet)*

Advantages

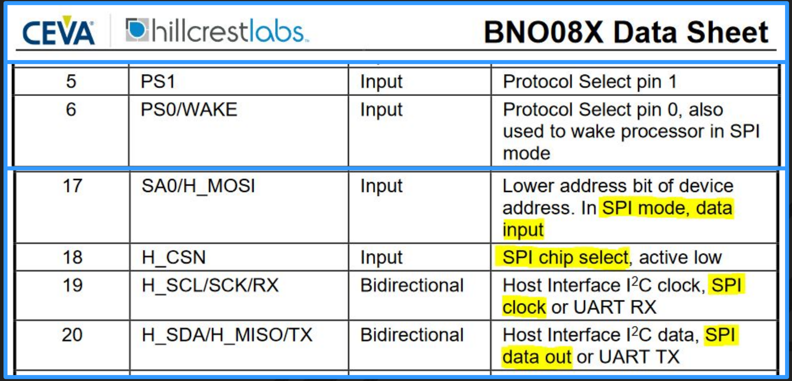
- Can be very fast
- Supports multiple peripherals
- Simpler internal implementation

Disadvantage

- Requires more wires *(CS line per peripheral)*
- The controller must control all communications
- Peripherals can't communicate with each other



SPI: BNO086 Example



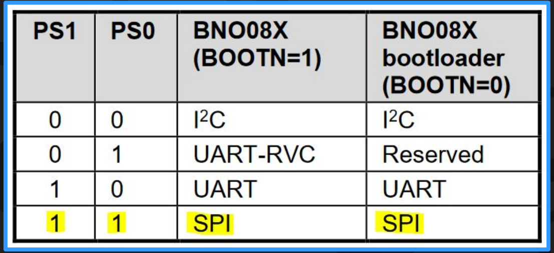
CEVA hillcrestlabs **BNO08X Data Sheet**

5	PS1	Input	Protocol Select pin 1
6	PS0/WAKE	Input	Protocol Select pin 0, also used to wake processor in SPI mode
17	SA0/H_MOSI	Input	Lower address bit of device address. In SPI mode, data input
18	H_CSN	Input	SPI chip select, active low
19	H_SCL/SCK/RX	Bidirectional	Host Interface I^{2}C clock, SPI clock or UART RX
20	H_SDA/H_MISO/TX	Bidirectional	Host Interface I^{2}C data, SPI data out or UART TX

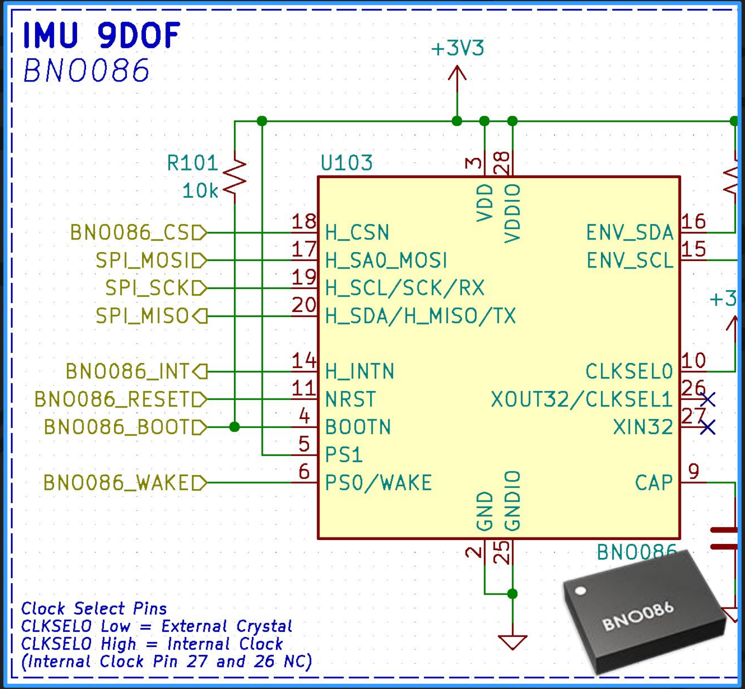
Schematic

Pin Descriptions

SPI Implementation



PS1	PS0	BNO08X (BOOTN=1)	BNO08X bootloader (BOOTN=0)
0	0	I^{2}C	I^{2}C
0	1	UART-RVC	Reserved
1	0	UART	UART
1	1	SPI	SPI



Clock Select Pins
CLKSEL0 Low = External Crystal
CLKSEL0 High = Internal Clock
(Internal Clock Pin 27 and 26 NC)

"Single chip 9 axis sensor with embedded sensor fusion that enables rapid development of sensor-enabled robotics, AR, VR, and IoT devices." - CEVA

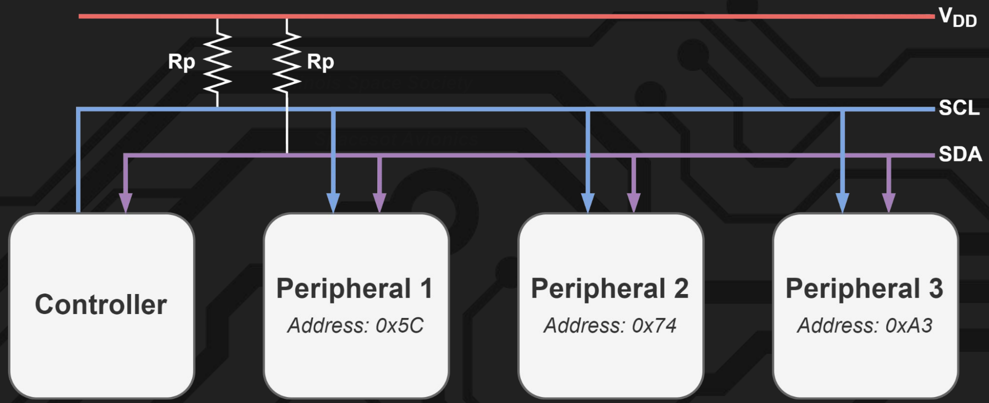
I²C: Inter-Integrated Circuit

Protocol Signals

- SDA: Bidirectional data
- SCL: Clock

Modes

- Original: 100kHz
- Fast-mode: 400kHz
- Fast-mode plus: 1MHz
- High-speed mode: 3.4MHz
- Ultra-fast mode: 5MHz
- Custom speeds: 0kHz to 5MHz



Pull-Up Resistors

- Open drain, pull up resistors required for all signals
- 4.7kΩ is good starting point
  - Adjust down as needed
  - Datasheet may have recommendations

Multiple Controllers

- Multiple controllers can share a bus (SDA/SCLK) and talk to all peripherals
- Controllers can't talk to each other
- Controllers must take turns sharing bus

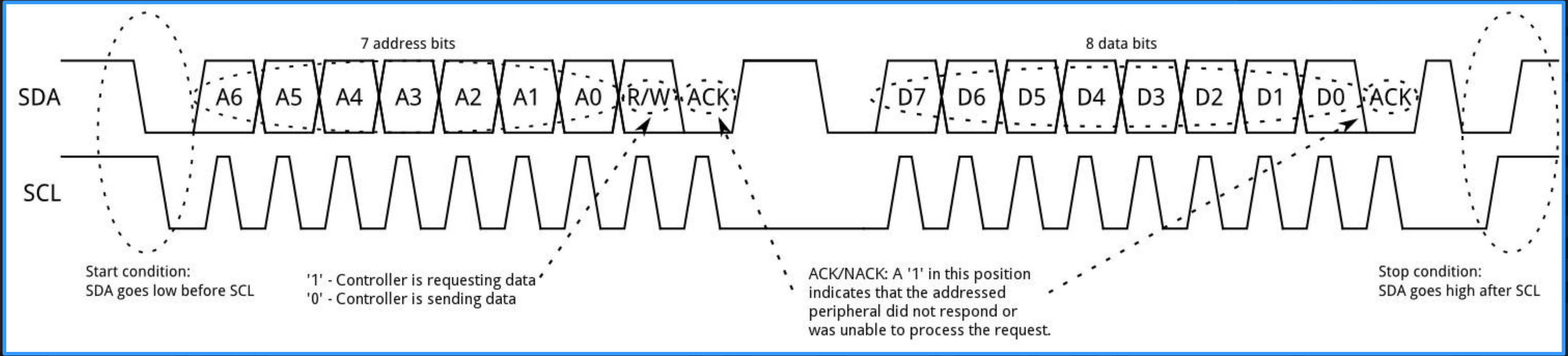
Addresses

- 7-bit *(112 Devices)*
- 10-bit *(1008 Devices)*
- 10b & 7b devices can coexist

SMBus

- System Management Bus
- I2C but more strict timing
- 10kHz to 100KHz only

I^{2}C: Inter-Integrated Circuit



Start condition:
SDA goes low before SCL

'1' - Controller is requesting data
'0' - Controller is sending data

ACK/NACK: A '1' in this position indicates that the addressed peripheral did not respond or was unable to process the request.

Stop condition:
SDA goes high after SCL

Start Condition

- Controller starts address frame by leaving SCL high and pulls SDA low
- All peripherals on alert
- Controller that pulls SDA low first wins in shared bus

7-bit Address Frame

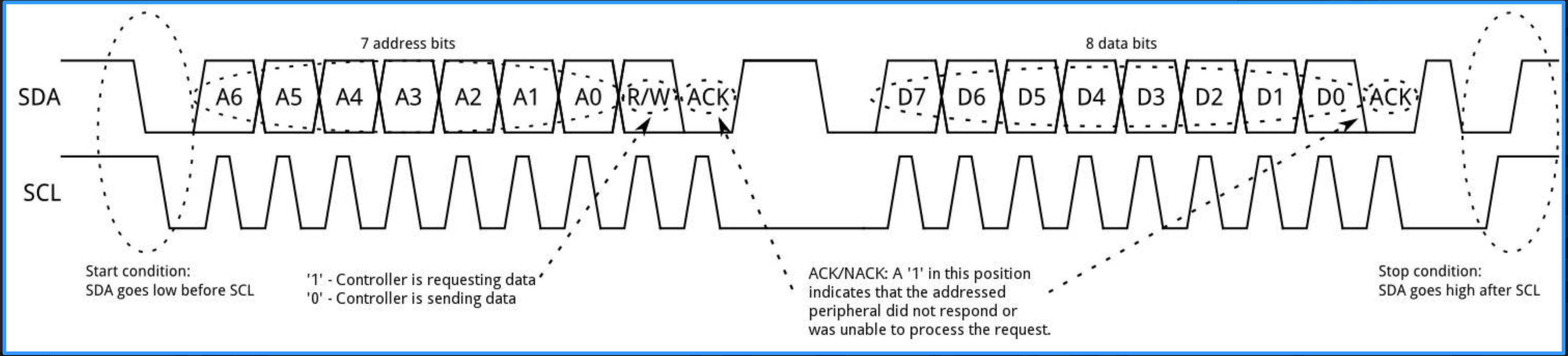
- Always sent first in new communication
- 7-bit Address *(MSB first)*
- Last bit is Read/Write *(1 = Read, 0 = Write)*

Data Frame

- Occur after address frames
- Controller continues sending clock pulses at regular intervals
- Number of data frames arbitrary

Slide Designer: Peter Giannetos

I^{2}C: Inter-Integrated Circuit



Start condition:
SDA goes low before SCL

'1' - Controller is requesting data
'0' - Controller is sending data

ACK/NACK: A '1' in this position indicates that the addressed peripheral did not respond or was unable to process the request.

Stop condition:
SDA goes high after SCL

Acknowledge

- 9th bit of every frame
- Error checking
- NACK (1) = Not Acknowledged
- ACK (0) = Acknowledged

Acknowledge Steps

1. Receiving device given control of SDA
2. If receiving device doesn't pull SDA low before the 9th CLK pulse then transmission halts

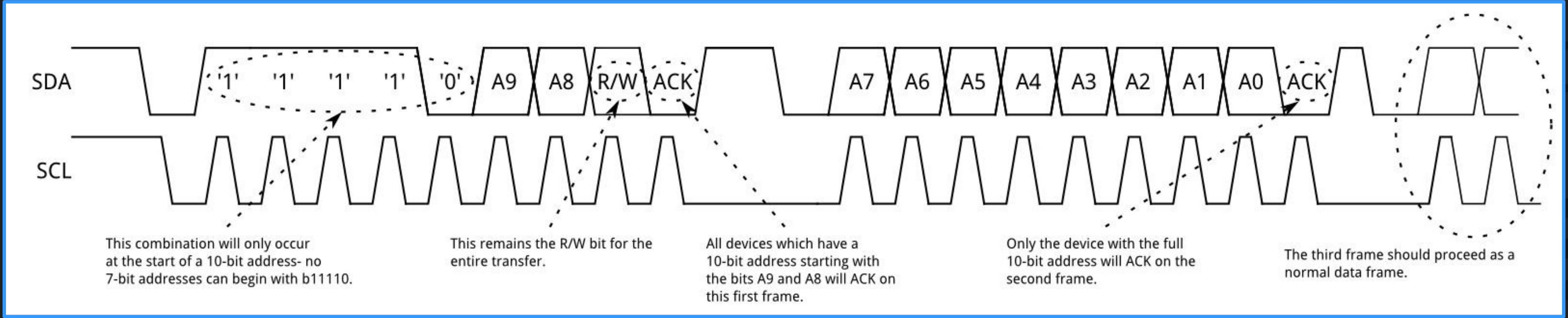
(Failure response decided by system code)

Stop Condition

- SDA goes high after SCL
- During normal data writing operation, the value on SDA should not change when SCL is high, to avoid false stop conditions.

Slide Designer: Peter Giannetos

I²C: Inter-Integrated Circuit



This combination will only occur at the start of a 10-bit address- no 7-bit addresses can begin with b11110.

This remains the R/W bit for the entire transfer.

All devices which have a 10-bit address starting with the bits A9 and A8 will ACK on this first frame.

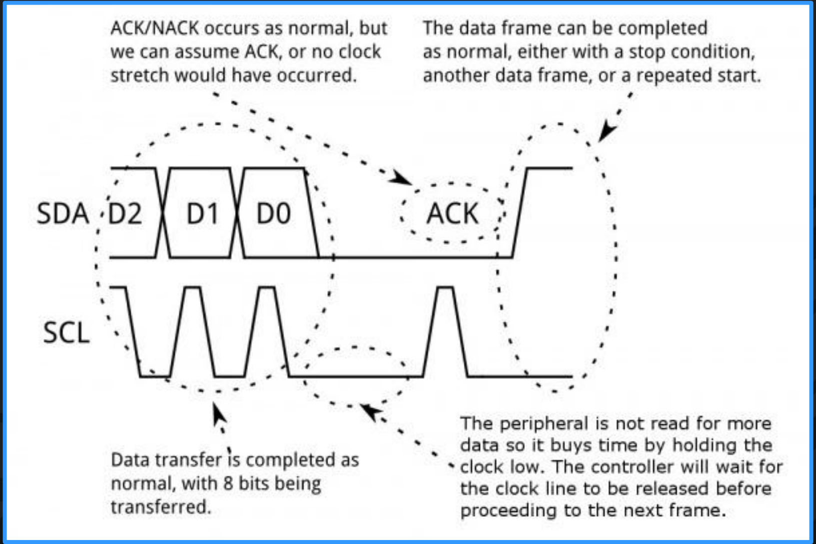
Only the device with the full 10-bit address will ACK on the second frame.

The third frame should proceed as a normal data frame.

10-bit Address Frames

- 1st Frame
  - "11110" - Specific flag for 10-bit addresses
  - A9/A8 - Address bits
  - Read/Write bit *(1 = Read, 0 = Write)*
  - Acknowledge bit from all devices that match A9 and A8 address bits
- 2nd Frame
  - A7…A0 - Address bits
  - Acknowledge bit from specific device

I²C: Inter-Integrated Circuit

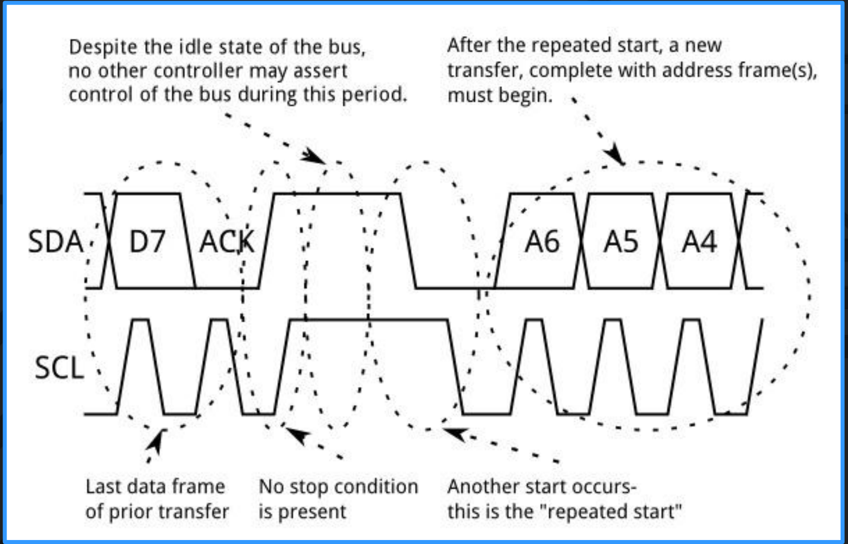


ACK/NACK occurs as normal, but we can assume ACK, or no clock stretch would have occurred.

The data frame can be completed as normal, either with a stop condition, another data frame, or a repeated start.

Data transfer is completed as normal, with 8 bits being transferred.

The peripheral is not read for more data so it buys time by holding the clock low. The controller will wait for the clock line to be released before proceeding to the next frame.



Despite the idle state of the bus, no other controller may assert control of the bus during this period.

After the repeated start, a new transfer, complete with address frame(s), must begin.

Last data frame of prior transfer

No stop condition is present

Another start occurs- this is the "repeated start"

Clock Stretching

- Current controller controls clock
- Peripheral can pull clock low to delay the controller sending more data

Repeated Start Condition

- Controller needs to continuously send data larger than 7 bits
- Don't send stop condition between frames
- Resends start condition

I^{2}C: MAX-M10S Example

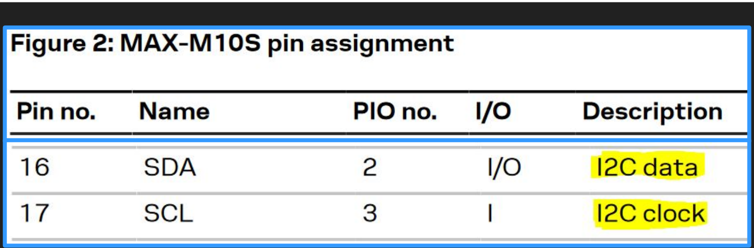
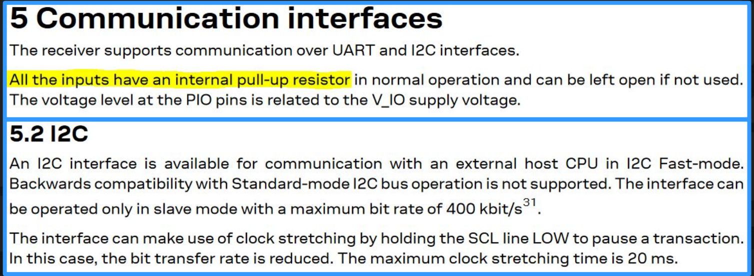


Figure 2: MAX-M10S pin assignment

Pin no.	Name	PIO no.	I/O	Description
16	SDA	2	I/O	I2C data
17	SCL	3	I	I2C clock



5 Communication interfaces

The receiver supports communication over UART and I2C interfaces.

All the inputs have an internal pull-up resistor in normal operation and can be left open if not used. The voltage level at the PIO pins is related to the V_IO supply voltage.

5.2 I2C

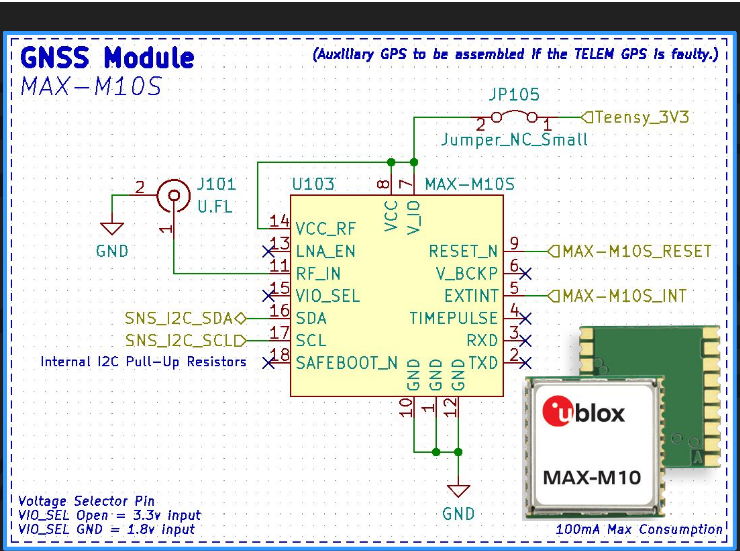
An I2C interface is available for communication with an external host CPU in I2C Fast-mode. Backwards compatibility with Standard-mode I2C bus operation is not supported. The interface can be operated only in slave mode with a maximum bit rate of 400 kbit/s[31].

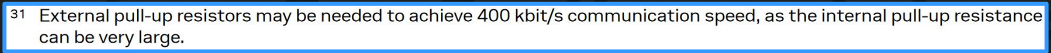
The interface can make use of clock stretching by holding the SCL line LOW to pause a transaction. In this case, the bit transfer rate is reduced. The maximum clock stretching time is 20 ms.

Schematic

Pin Descriptions

I2C Implementation





[31] External pull-up resistors may be needed to achieve 400 kbit/s communication speed, as the internal pull-up resistance can be very large.

Global Navigation Satellite System *(GNSS)* = Global Positioning System *(GPS, US Version)*

"Ultra-low-power GNSS receiver for high-performance asset-tracking devices. Less than 25 mW power consumption without compromising GNSS performance. Maximum position availability with concurrent reception of 4 GNSS." - ublox

SD/MMC: Secure Digital / Multi-Media Card

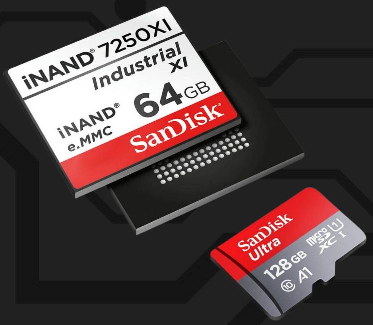
MicroSD Card

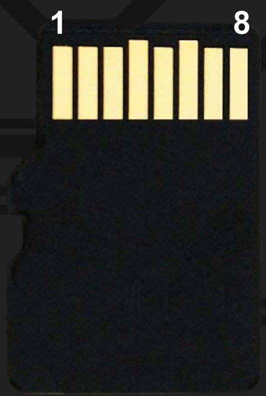
- Can either be used with SD or SPI interfaces
- Sometimes slots come with card detect pins
  - Can be left floating if unused

eMMC

- Embedded multi-media card
- Like a MicroSD card but soldered in place
- May have up to 8 data lanes

Pin	SD	SPI
1	DAT2	X
2	CD/DAT3	CS
3	CMD	DI/MOSI
4	VDD	VDD
5	CLK	SCLK
6	VSS	GND
7	DAT0	DO/MISO
8	DAT1	X





Slide Designer: Peter Giannetos

PWM: Pulse Width Modulation

Protocol Signals

- PWM: Pulses

Characteristics

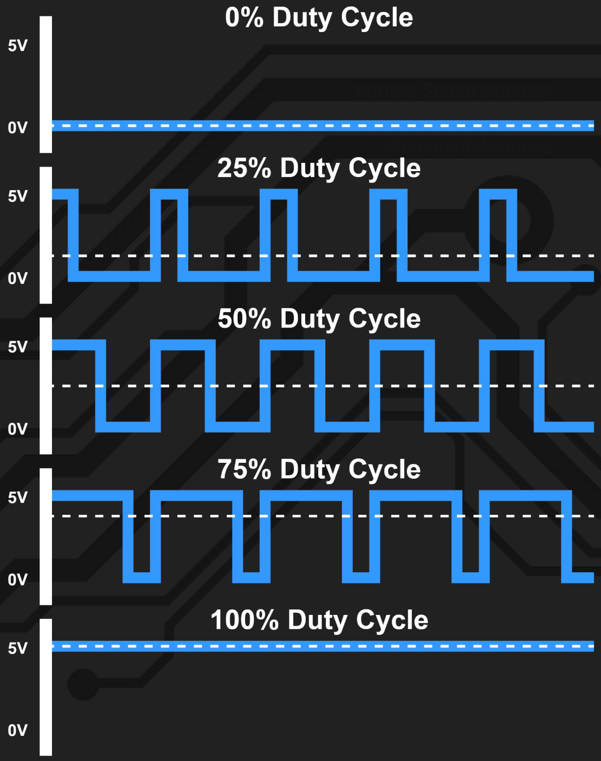
- Frequency: Rate of oscillation
- Duty Cycle: Percentage of "ON" time

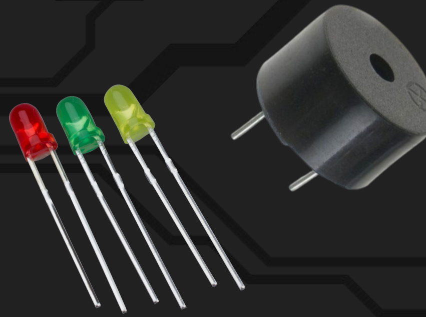
Example: An LED with a 20% duty cycle at 1 Hz will be obvious that it's turning on and off to your eyes, meanwhile, a 20% duty cycle at 100 Hz or above will just look dimmer

Usages

- Servo Motors
- Piezo Buzzers
- LED Dimmers
- *And much more!*

PWM emulates an analog signal by controlling the average power output. Faster frequencies produce a better analog appearing signal.

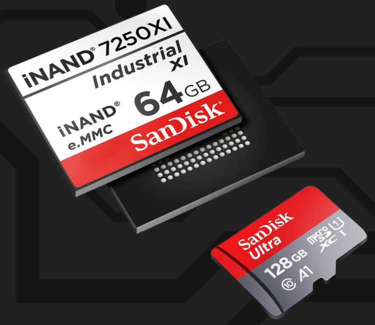


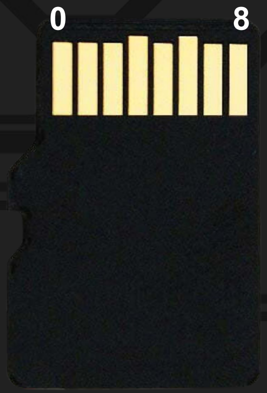


SD/MMC: Secure Digital / Multi-Media Card

Speed Classes?

Pin	SD	SPI
1	DAT2	X
2	CD/DAT3	CS
3	CMD	DI/MOSI
4	VDD	VDD
5	CLK	SCLK
6	VSS	GND
7	DAT0	DO/MISO
8	DAT1	X





Slide Designer: Peter Giannetos

Balanced and Differential

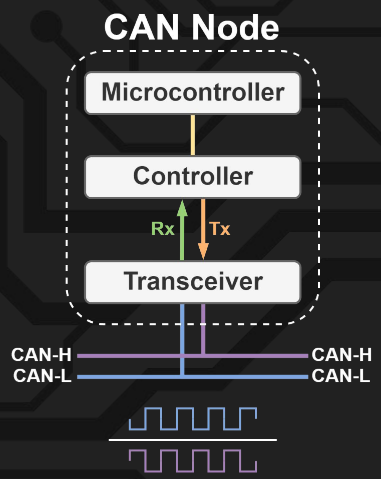
1. Text
2. https://www.allaboutcircuits.com/technical-articles/the-why-and-how-of-differential-signaling/
3. https://forum.headphones.com/t/single-ended-balanced-differential-what-it-means/3571

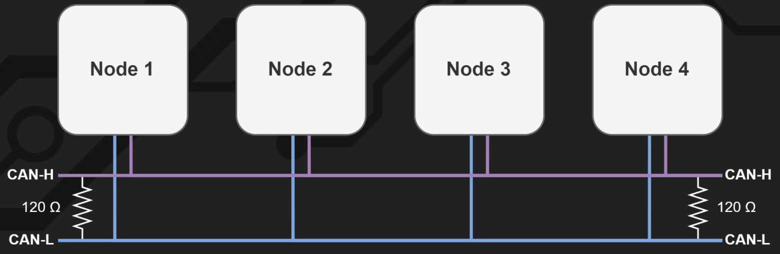
Slide Designer: Peter Giannetos

CAN

Controller

Transceiver





Signal Names

Names

- SDO – Serial Data Out. An output signal on a device where data is sent out to another SPI device.
- SDI – Serial Data In. An input signal on a device where data is received from another SPI device.
- CS – Chip Select. Activated by the controller to initiate communication with a given peripheral.
- PICO (peripheral in / controller out). For devices that can be either a controller or a peripheral; the signal on which the device sends output when acting as the controller, and receives input when acting as the peripheral.
- POCI (peripheral out / controller in). For devices that can be either a controller or a peripheral; the signal on which the device receives input when acting as the controller, and sends output when acting as the peripheral.
- SDIO – Serial Data In/Out. A bi-directional serial sign
-
- SCK – Serial Clock. The clock for the bus generated by the controller.
-
- MOSI – Master Out Slave In
- MISO – Master In Slave Out
- SS – Slave Select
- MOMI Master Out Master In
- SOSI Slave Out Slave In

Slide Designer: Peter Giannetos

USB

Protocol Signals
5V: Voltage supply
+D: Data positive
-D: Data negative
ID: On-The-Go (OTG) identification
GND: Ground

Pro/Cons
Simple to implement
Power and data
Slower than other USB types
Special PCB routing considerations (Differential pair)

USB OTG ID
Some devices have flexible dual roles
Negotiates between two devices to determine the host and peripheral device

Note: Electrostatic Discharge (ESD) chips are commonly used to protect USB connectors and devices when connecting and disconnecting.

Bit Banging

Bit-Banging

- A "term of art" for any method of data transmission that employs software as a substitute for dedicated hardware to generate transmitted signals or process received signals

Further Reading

1i101i10, and Big6Big6. “Difference between Open Drain and Push-Pull Modes.” *Electrical Engineering Stack Exchange*, 18 May 2022, electronics.stackexchange.com/questions/620150/difference-between-open-drain-and-push-pull-modes. Accessed 29 Dec. 2023.

bri_huang. “Logic Levels.” *SparkFun*, learn.sparkfun.com/tutorials/logic-levels/all. Accessed 29 Dec. 2023.

jimblom. “Serial Communication.” *SparkFun*, learn.sparkfun.com/tutorials/serial-communication. Accessed 29 Dec. 2023.

JordanDee. “Pulse Width Modulation.” *SparkFun*, learn.sparkfun.com/tutorials/pulse-width-modulation. Accessed 29 Dec. 2023.

MikeGrusin. “Serial Peripheral Interface.” *SparkFun*, learn.sparkfun.com/tutorials/serial-peripheral-interface-spi. Accessed 29 Dec. 2023.

Seidle, Nathan. “A Redefintion of SPI Signal Names.” *SparkFun*, www.sparkfun.com/spi_signal_names. Accessed 29 Dec. 2023.

SFUptownMaker. “I2C.” *SparkFun*, learn.sparkfun.com/tutorials/i2c/all. Accessed 29 Dec. 2023.

Stoyanov, Yasen. “Open Drain Output vs. Push-Pull Output.” *Open4Tech*, 18 May 2019, open4tech.com/open-drain-output-vs-push-pull-output/.

Valdez, Jonathan, and Jared Becker. “Understanding the I2C Bus.” *Texas Instruments*, www.ti.com/lit/an/slva704/slva704.pdf. Accessed 29 Dec. 2023.

Slide Designer: Peter Giannetos

Further Reading

Works Cited:

1. https://learn.sparkfun.com/tutorials/logic-levels/all
2. https://electronics.stackexchange.com/questions/620150/difference-between-open-drain-and-push-pull-modes
3. https://open4tech.com/open-drain-output-vs-push-pull-output/
4. https://www.sparkfun.com/spi_signal_names
5. https://learn.sparkfun.com/tutorials/i2c/all
6. https://www.ti.com/lit/an/slva704/slva704.pdf
7. https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi
8. https://learn.sparkfun.com/tutorials/serial-communication
9. https://learn.sparkfun.com/tutorials/pulse-width-modulation

10. https://www.ti.com/lit/an/sloa101b/sloa101b.pdfhttp://www.electroniccircuitsdesign.com/pinout/sd-microsd-card-pinout.html
11. https://www.circuitstate.com/tutorials/interfacing-catalex-micro-sd-card-module-with-arduino/
12. https://www.kingston.com/unitedkingdom/en/blog/personal-storage/memory-card-speed-classes
13. https://www.integralmemory.com/articles/memory-cards-understanding-the-labels/
14. https://www.banggood.com/fr/SPT-Servo-SPT70HV-320-70KG-Large-Torque-Metal-Gear-Digital-Servo-For-RC-Robot-RC-Car-p-1577512.html?cur_warehouse=CN
15. https://www.pcboard.ca/minipiezo-buzzer

Slide Designer: Peter Giannetos