EECE494:
Computer Bus and SoC Interfacing

USB (Universal Series Bus)

Dr. Charles Kim

Electrical and Computer Engineering

Howard University

Spring 2014

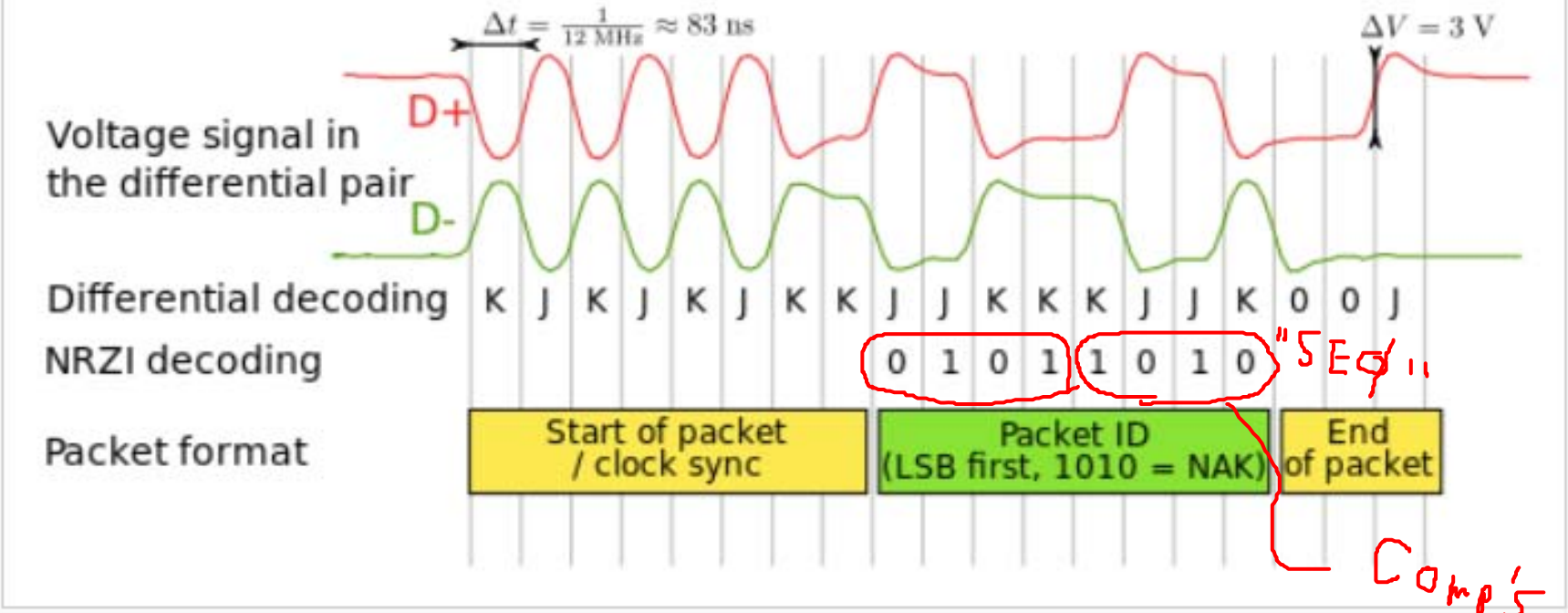
Review - Line Code

- Line Code = “Digital Baseband Modulation”=“Digital Baseband Transmission”
- Used for digital data transport
- Common Line Codes
  - Manchester Code (“Transition Direction”)
  - Non-Return-to-Zero (NRZ)
  - Non-Return-to-Zero, Inverted (NRZI) (Read as “State Change” for “1”)
  - 6b/8b

USB Protocol

- Differential Signaling: D+ and D-
- Toggle between J state and L state
  - J state: D+ High and D- Low
  - K state: D+ Low and D- High
- NRZI Coding – “State change”
  - J →K or K→J : 0
  - J→ J or K→K: 1
- 8-bit Synchronization, 1 bit mark, followed by USB frame, and ended with EOP (end of packet)
- EOP: 2 bit times of SE0 [“single-ended zero”] (D+ and D- both LOW) and 1 bit of J state (See next slide)

Example of USB transmission



Example of a Negative Acknowledge packet transmitted by USB 1.1 Full-speed device when there is no more data to read. It consists of the following fields: clock synchronization byte, type of packet and end of packet. Data packets would have more information between the type of packet and end of packet.

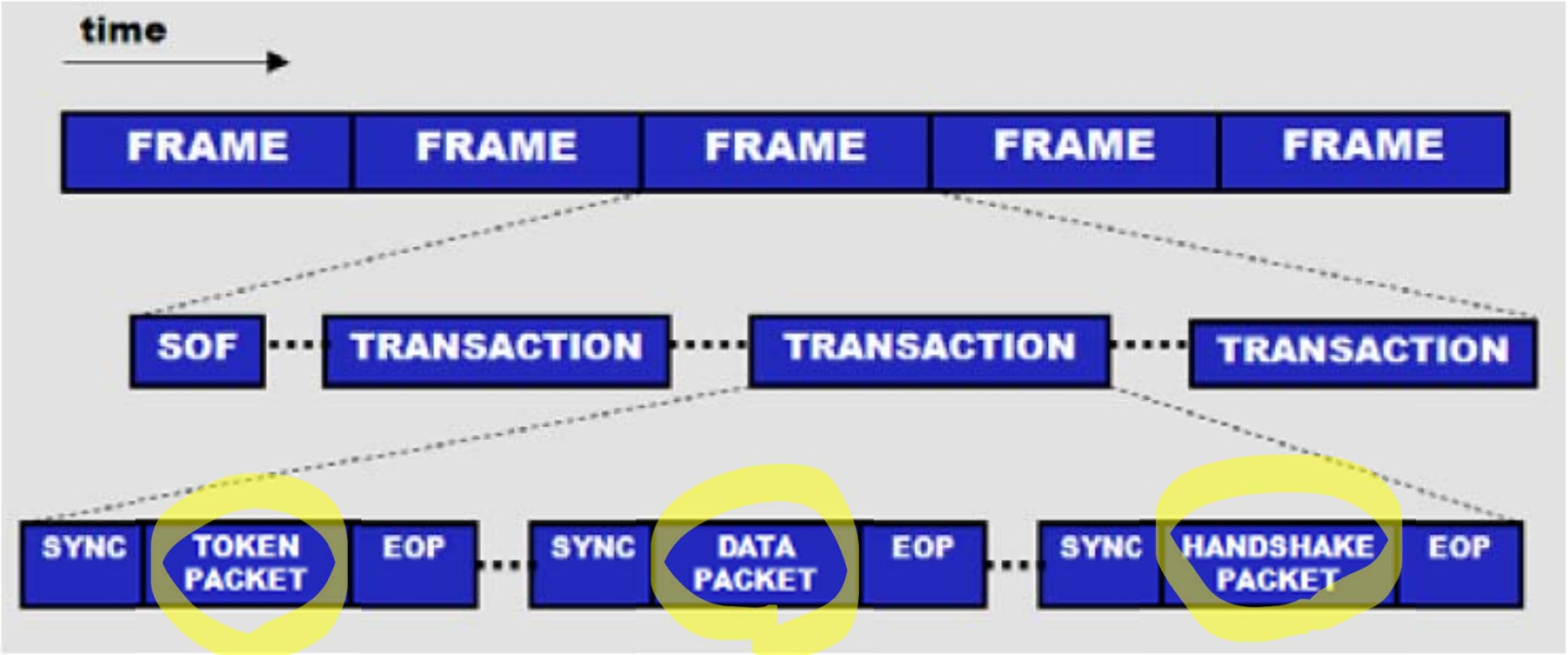
en.wikipedia.org/wiki/USB

USB Eye Diagram

- Electrical testing of D+ and D-
- Signal quality
  - Rise Time
  - Fall Time
  - Undershoot
  - Overshoot
  - D+/D- Line Jitter

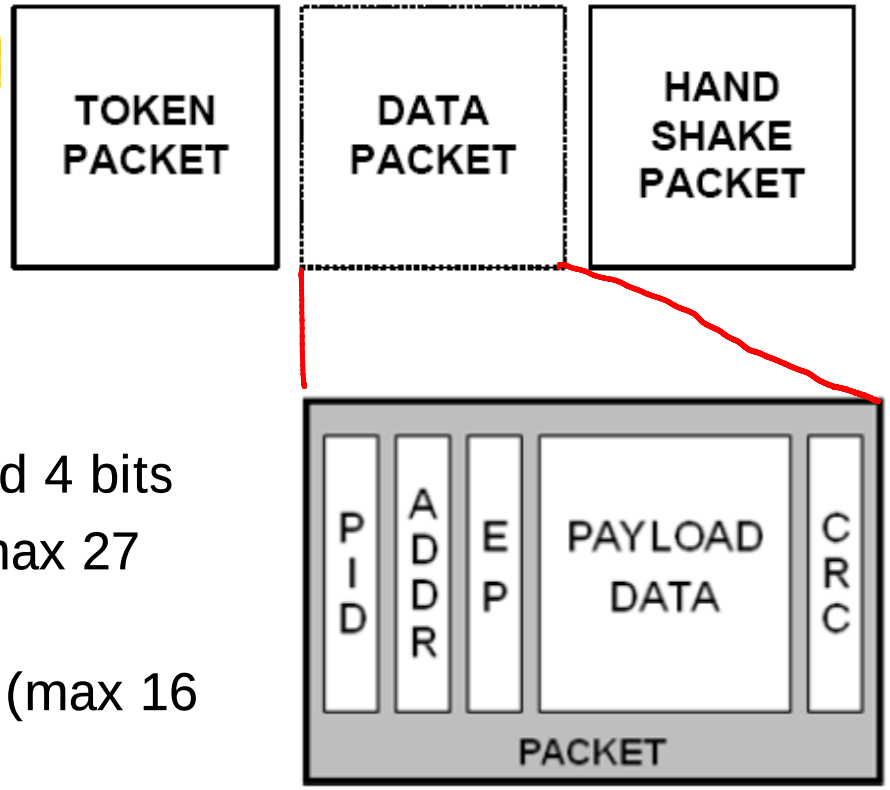
USB Communication

- A series of frames
- Each frame contains:
  - SOF (Start of Frame)
  - Transactions



USB Packet Contents

- Transaction
  - Exchange of packets
    - Sync/Data/End
  - 3 types of packets

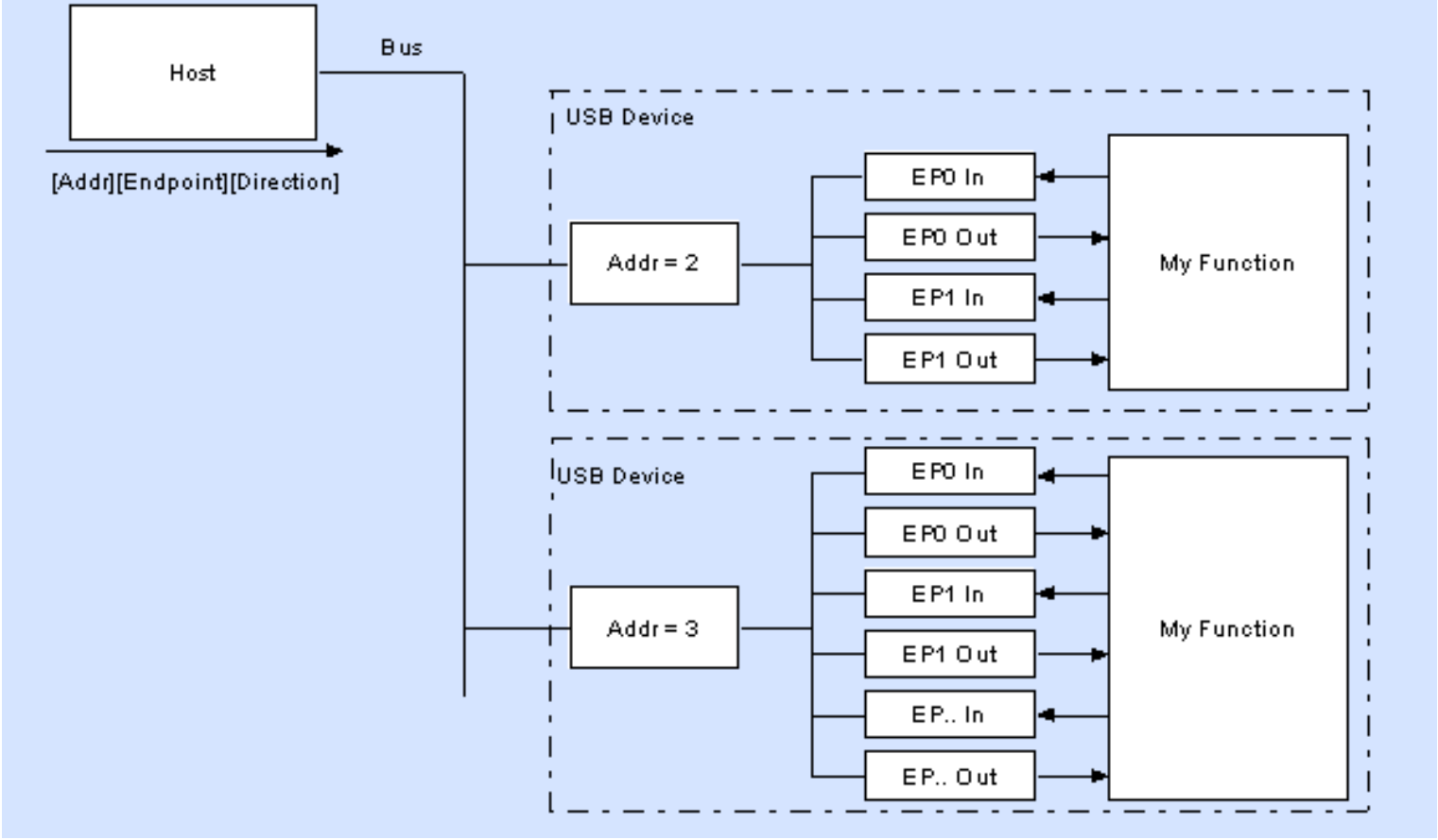


- Information in a packet
  - Packet Identification (PID): 4 and 4 bits
  - Device Addr (optional): 7 bits (max 27 devices)
  - End Point (EP) (optional): 4 bits (max 16 endpoint addresses)
  - Payload Data (optional)(up to 1023 bytes)
  - CRC (optional)

PID (Packet ID): Identification of the type of packets that is being sent

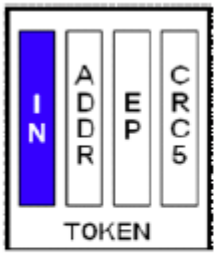
Group	PID Value	Packet Identifier
Token	0001	OUT Token
	1001	IN Token
	0101	SOF Token
	1101	SETUP Token
Data	0011	DATA0
	1011	DATA1
	0111	DATA2
	1111	MDATA
Handshake	0010	ACK Handshake
	1010	NAK Handshake
	1110	STALL Handshake
	0110	NYET (No Response Yet)
Special	1100	PREamble
	1100	ERR
	1000	Split
	0100	Ping

USB Host and Devices

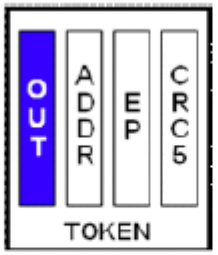


Token Packets

- Always come from the host
- Used to direct traffic on the bus
- The function of the token packet depends on the activity performed
  - IN tokens - to request that devices send data to the host.
  - OUT tokens - to send data from the host.
  - SETUP tokens - to send commands from the host.
  - SOF tokens - to mark time frames.
- With an IN, OUT, and SETUP token packet, there is a 7-bit device address, 4-bit endpoint ID, and 5-bit CRC.

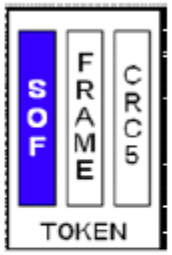


IN Token



OUT Token

SETUP Token

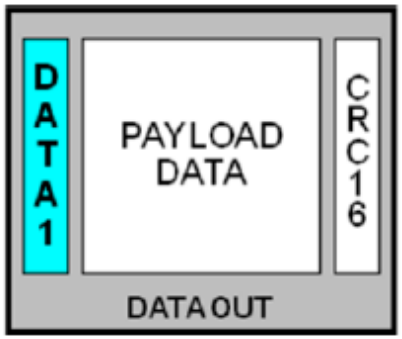


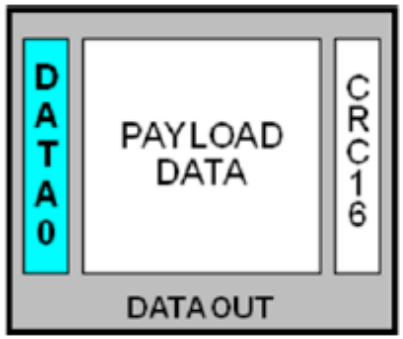
SOF Token

PID Value	Packet Identifier
0001	OUT Token
1001	IN Token
0101	SOF Token
1101	SETUP Token

Data Packet

- Data packets follow IN, OUT, and SETUP token packets.
- The size of the payload data ranges from 0 to 1024 bytes depending on the transfer type.
- The packet ID toggles between DATA0 and DATA1 for each successful data packet transfer, and the packet closes with a 16-bit CRC.
- The receiver of data toggles its data toggle bit when it is able to accept the data and it receives an error-free data packet with the correct identification.
- The sender of data toggles its data toggle bit when it receives a positive acknowledgement from the receiver.
- In this way, the data toggle bits stay synchronized until, for example, a packet with an incorrect identification is received.





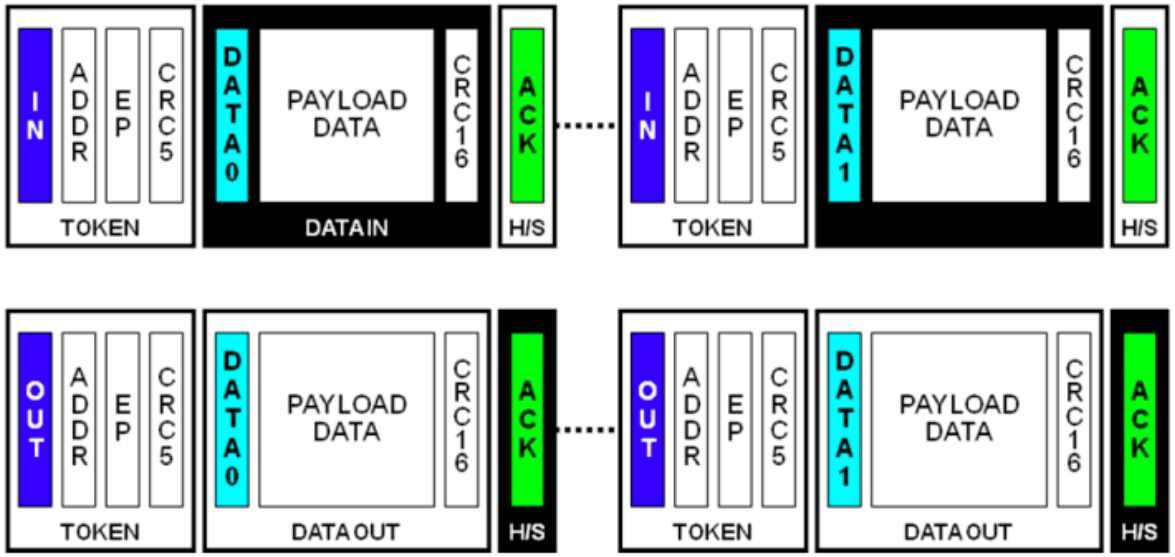
PID Value	Packet Identifier
0011	DATA0
1011	DATA1

Data Packets – Data Toggle Bit

- An example where the data toggle is used is if an ACK is sent but not received.
  - the sender updates the data toggle from '1' to '0' but the receiver does not.
  - The receiver remains at '1'.
  - This causes the host and device to be out of sync on the next data stage, which indicates an error.
- White boxes represent the transaction is coming from the host and black boxes represent the transaction is coming from the device.

Device

Host



Handshake Packets

- Handshake packets conclude each transaction.
- Each handshake includes an 8-bit packet ID and is sent by the receiver of the transaction.
  - ACK: Acknowledge successful completion
  - NAK: Negative acknowledgment
  - Stall: Error indication sent be a device
  - NYET: Indicates the device is “Not ready to receive” another data packet (High Speed only)

PID Value	Packet Identifier
0010	ACK Handshake
1010	NAK Handshake
1110	STALL Handshake
0110	NYET (No Respons

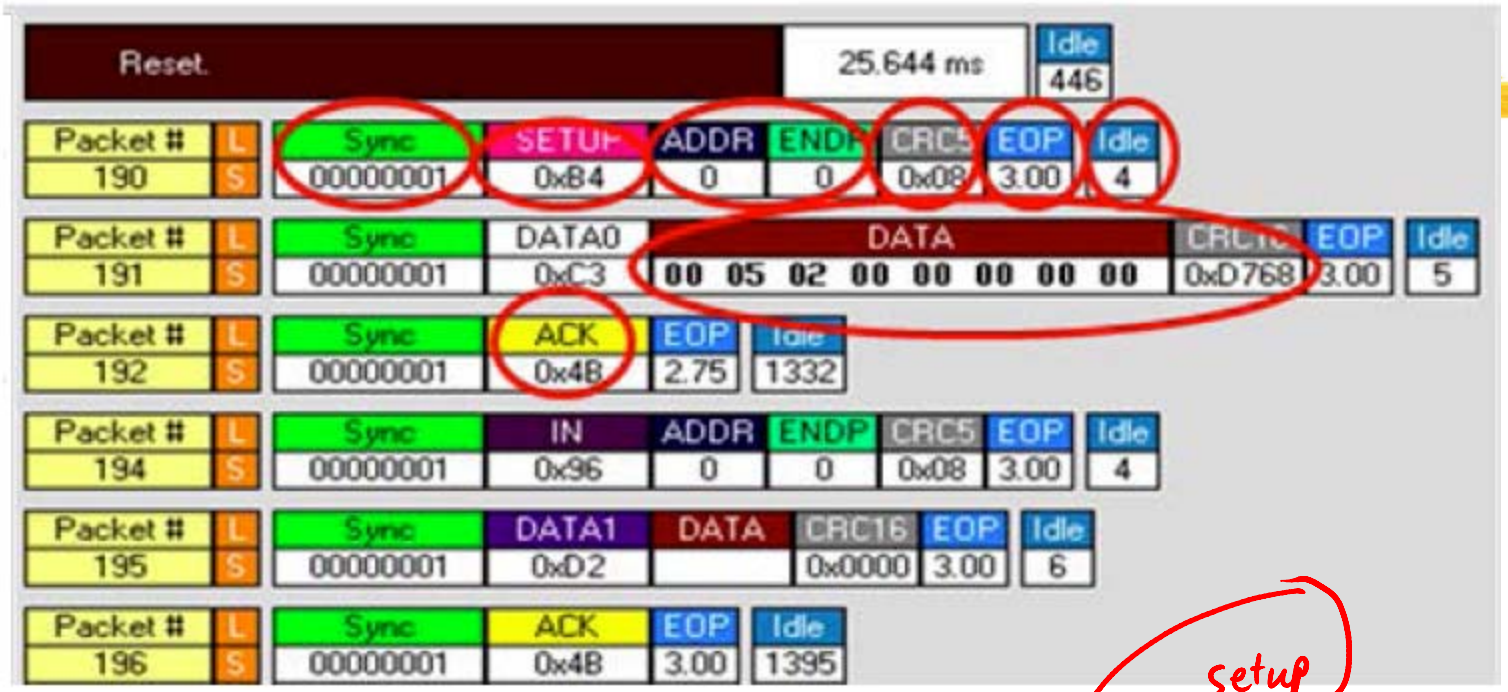
CRC – Background 1

- Integer Division: Dividend (N) divider by divisor (D)
- N(629)/D(25) = Q (25) – R(4)
  - See, [N-R]/D= Q (25) – R(0)
- Send N (629) and R(4) to a target
- Receiver, for data integrity, re-computes R' and check against the received R

CRC in USB

- Polynomial Division: Dividend (N) divider by divisor (D)
  - N(629)=$6*10^2+2*10^1+9*10^0$.=$1*2^9$+.....
  - D(25)=$2*10^1+5*10^0$ = $1*2^4$+.....
- 4-bit PID (Packet ID) field
  - Simple bitwise inversion of the PID is enough
  - No CRC protection
- Token Packets
  - Protected region is only 11 bits
  - 5-bit CRC protection
- Data packets
  - Up to 1023 bytes
  - 16-bit CRC protection

USB Analyzer Output Example - A

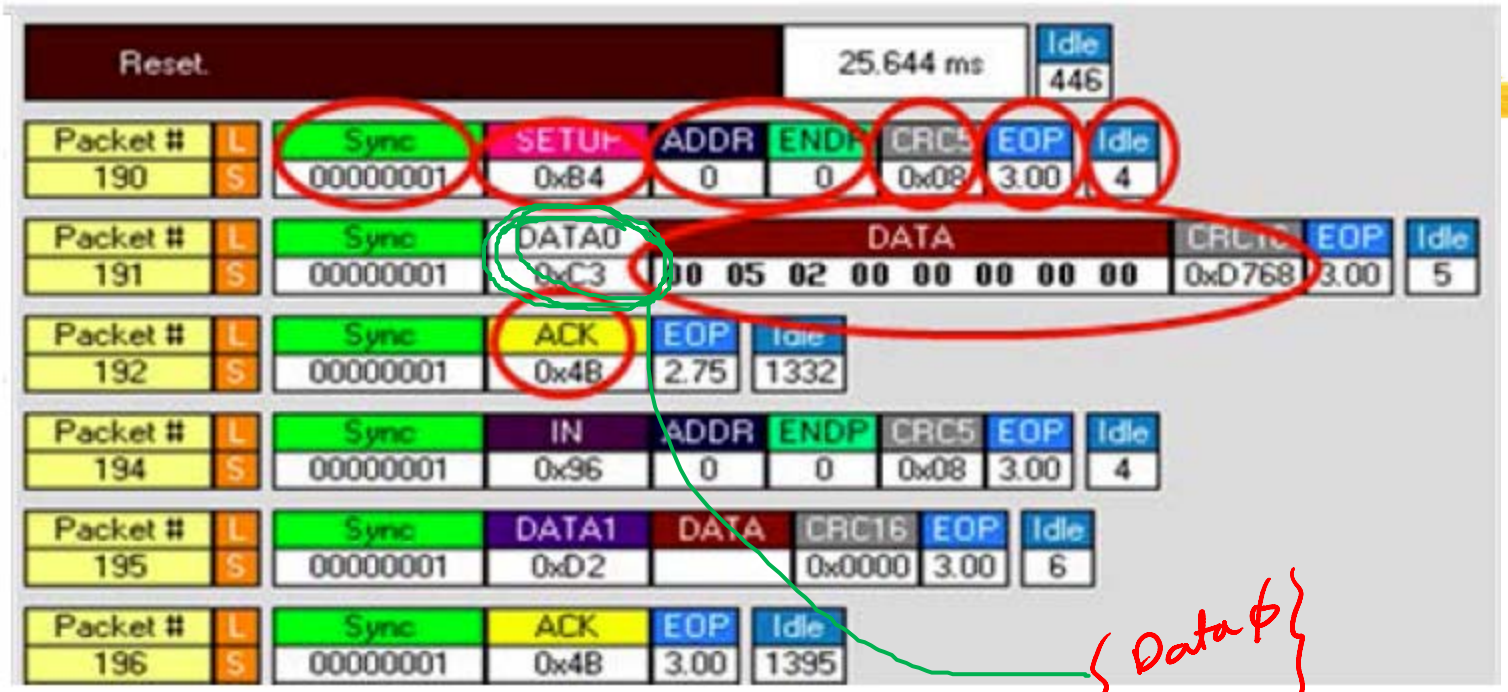


Reset						25.644 ms	Idle 446	
Packet # 190	L/S	Sync 00000001	SETUP 0x84	ADDR 0	ENDP 0	CRC5 0x08	EOP 3.00	Idle 4
Packet # 191	L/S	Sync 00000001	DATA0 0xC3	DATA 00 05 02 00 00 00 00 00		CRC16 0xD768	EOP 3.00	Idle 5
Packet # 192	L/S	Sync 00000001	ACK 0x4B	EOP 2.75	Idle 1332			
Packet # 194	L/S	Sync 00000001	IN 0x96	ADDR 0	ENDP 0	CRC5 0x08	EOP 3.00	Idle 4
Packet # 195	L/S	Sync 00000001	DATA1 0xD2	DATA	CRC16 0x0000	EOP 3.00	Idle 6	
Packet # 196	L/S	Sync 00000001	ACK 0x4B	EOP 3.00	Idle 1395			

setup
1101 B, 4
→ 1011 0100
LSB First
comp bit by bit

- Packet #190
  - L/S: Low Speed
  - Token Packet
  - CRC5: 5 bit CRC
  - EOP: End of Packet

USB Analyzer Output Example - B



Data 0

0011 → 3

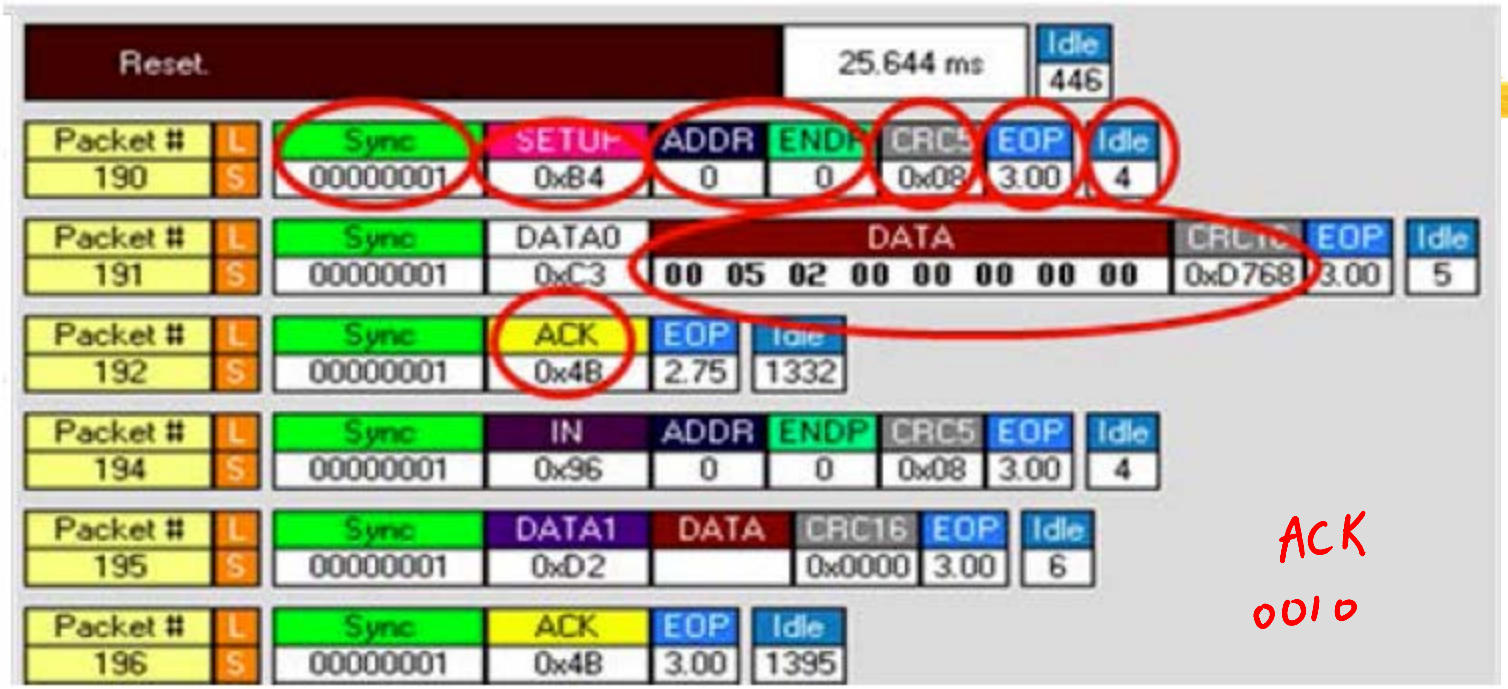
→ 1100 0011

LSB first

bit by bit Complement

- Packet #191
  - Data0: Data Token
  - CRC16: Check for Data with 16 bit CRC
  - Idle: Time between the current packet and the previous packet

USB Analyzer Output Example - C

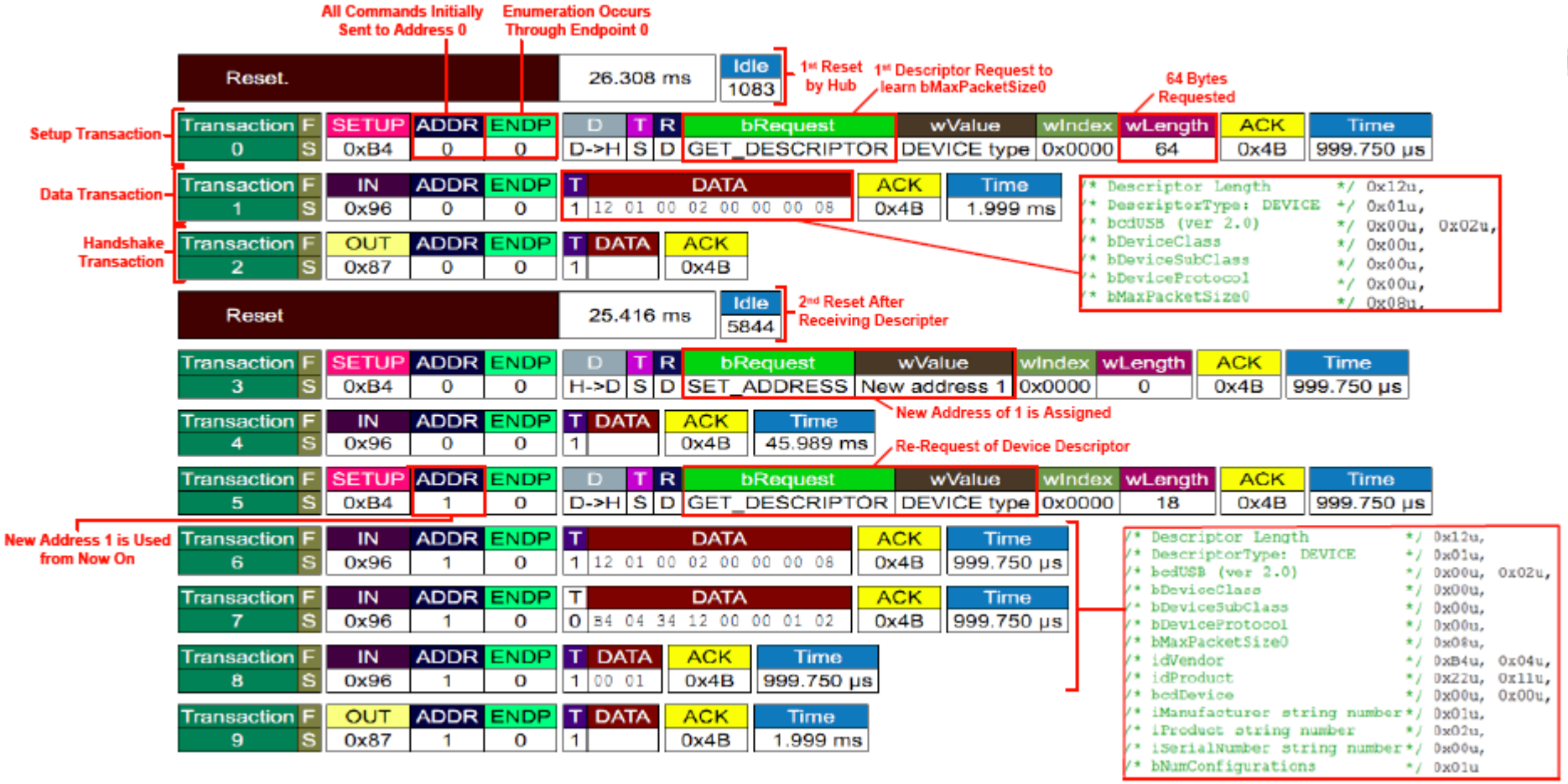


- Packet #192

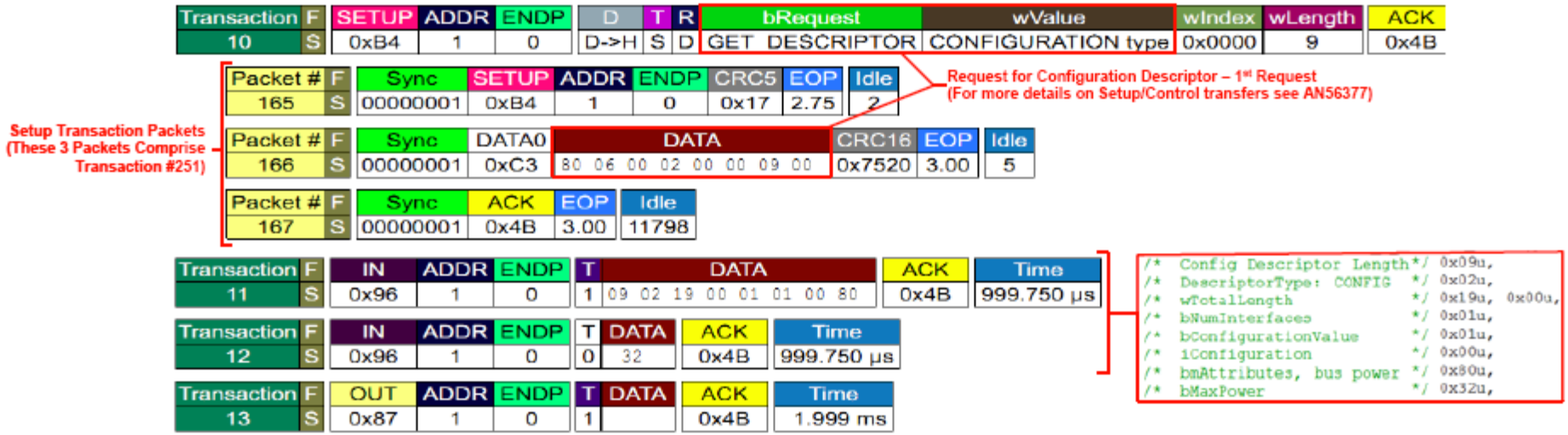
→ 0100 1011
4 B

  - ACK: Successful indication of Data packet

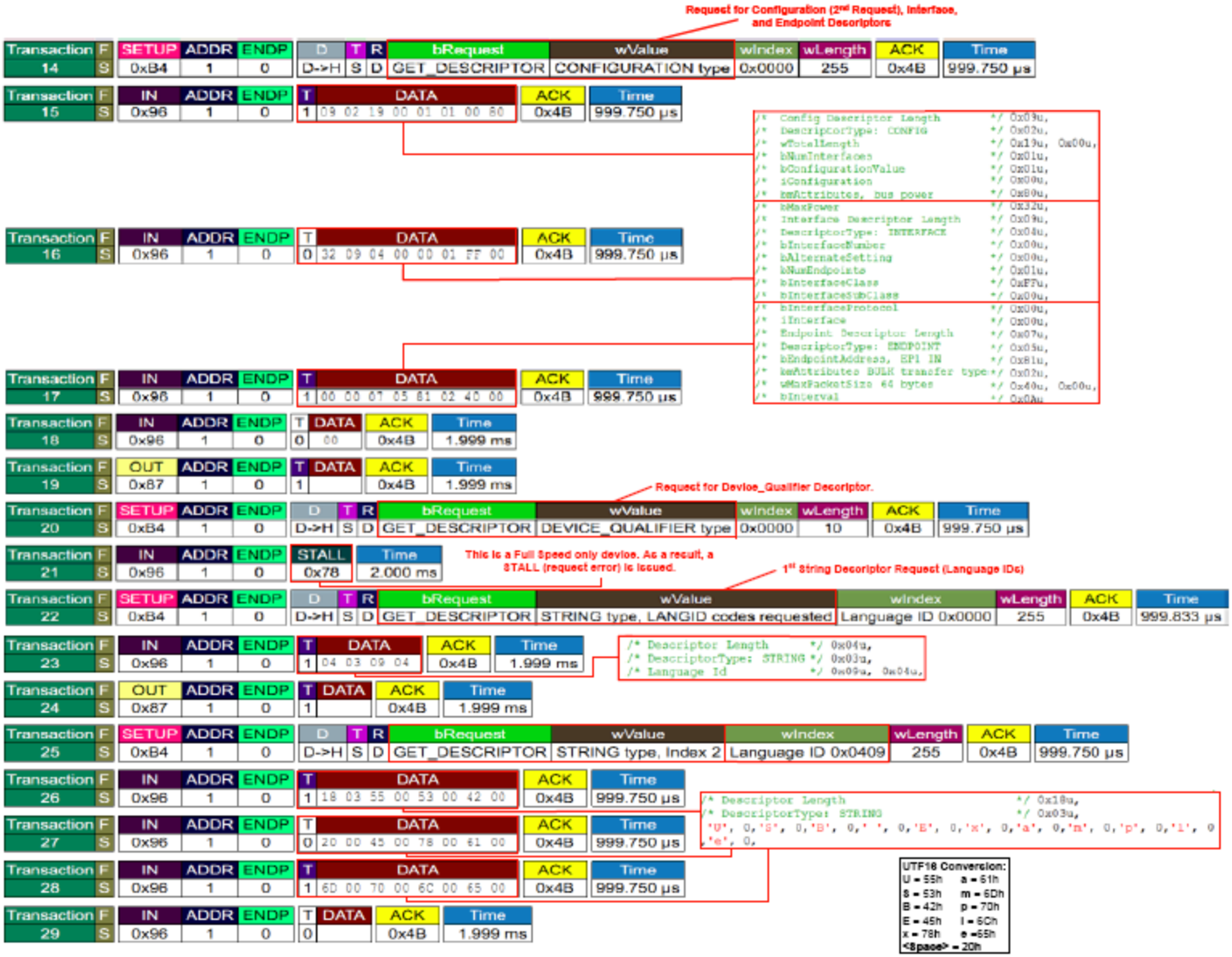
Bus Analyzer Capture - 1



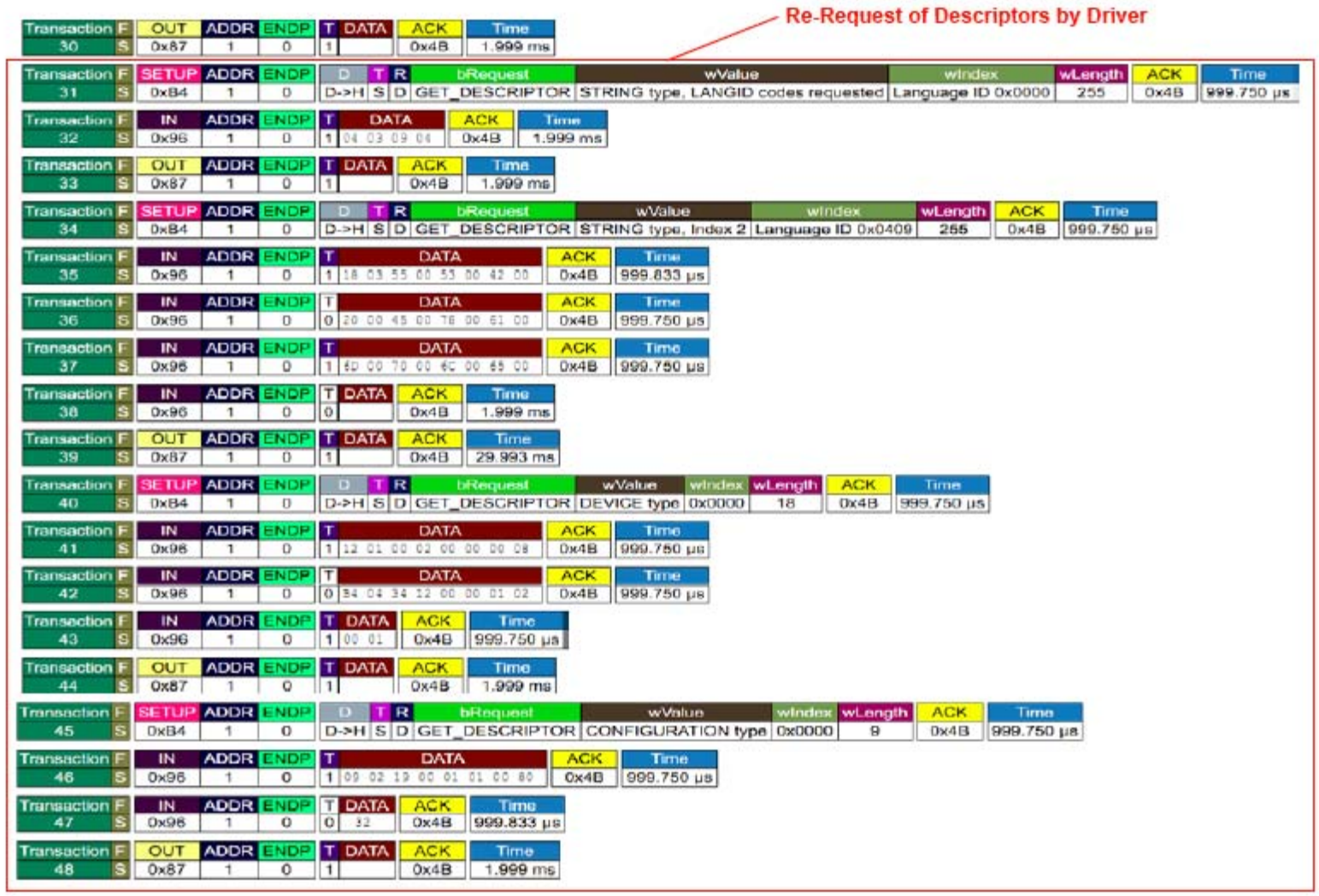
Bus Analyzer Capture - 2



Bus Analyzer Capture - 3



Bus Analyzer Capture - 4



Transaction	F	OUT	ADDR	ENDP	T	DATA	ACK	Time
30	S	0x87	1	0	1		0x4B	1.999 ms

Transaction	F	SETUP	ADDR	ENDP	D	T	R	bRequest	wValue	windex	wLength	ACK	Time
31	S	0xB4	1	0	D->H	S	D	GET_DESCRIPTOR	STRING type, LANGID codes requested	Language ID 0x0000	255	0x4B	999.750 µs

Transaction	F	IN	ADDR	ENDP	T	DATA	ACK	Time
32	S	0x96	1	0	1	04 03 09 04	0x4B	1.999 ms

Transaction	F	OUT	ADDR	ENDP	T	DATA	ACK	Time
33	S	0x87	1	0	1		0x4B	1.999 ms

Transaction	F	SETUP	ADDR	ENDP	D	T	R	bRequest	wValue	windex	wLength	ACK	Time
34	S	0xB4	1	0	D->H	S	D	GET_DESCRIPTOR	STRING type, Index 2	Language ID 0x0409	255	0x4B	999.750 µs

Transaction	F	IN	ADDR	ENDP	T	DATA	ACK	Time
35	S	0x96	1	0	1	18 03 55 00 53 00 42 00	0x4B	999.833 µs

Transaction	F	IN	ADDR	ENDP	T	DATA	ACK	Time
36	S	0x96	1	0	0	20 00 45 00 76 00 61 00	0x4B	999.750 µs

Transaction	F	IN	ADDR	ENDP	T	DATA	ACK	Time
37	S	0x96	1	0	1	6D 00 70 00 6C 00 65 00	0x4B	999.750 µs

Transaction	F	IN	ADDR	ENDP	T	DATA	ACK	Time
38	S	0x96	1	0	0		0x4B	1.999 ms

Transaction	F	OUT	ADDR	ENDP	T	DATA	ACK	Time
39	S	0x87	1	0	1		0x4B	29.993 ms

Transaction	F	SETUP	ADDR	ENDP	D	T	R	bRequest	wValue	windex	wLength	ACK	Time
40	S	0xB4	1	0	D->H	S	D	GET_DESCRIPTOR	DEVICE type	0x0000	18	0x4B	999.750 µs

Transaction	F	IN	ADDR	ENDP	T	DATA	ACK	Time
41	S	0x96	1	0	1	12 01 00 02 00 00 00 08	0x4B	999.750 µs

Transaction	F	IN	ADDR	ENDP	T	DATA	ACK	Time
42	S	0x96	1	0	0	B4 04 34 12 00 00 01 02	0x4B	999.750 µs

Transaction	F	IN	ADDR	ENDP	T	DATA	ACK	Time
43	S	0x96	1	0	1	00 01	0x4B	999.750 µs

Transaction	F	OUT	ADDR	ENDP	T	DATA	ACK	Time
44	S	0x87	1	0	1		0x4B	1.999 ms

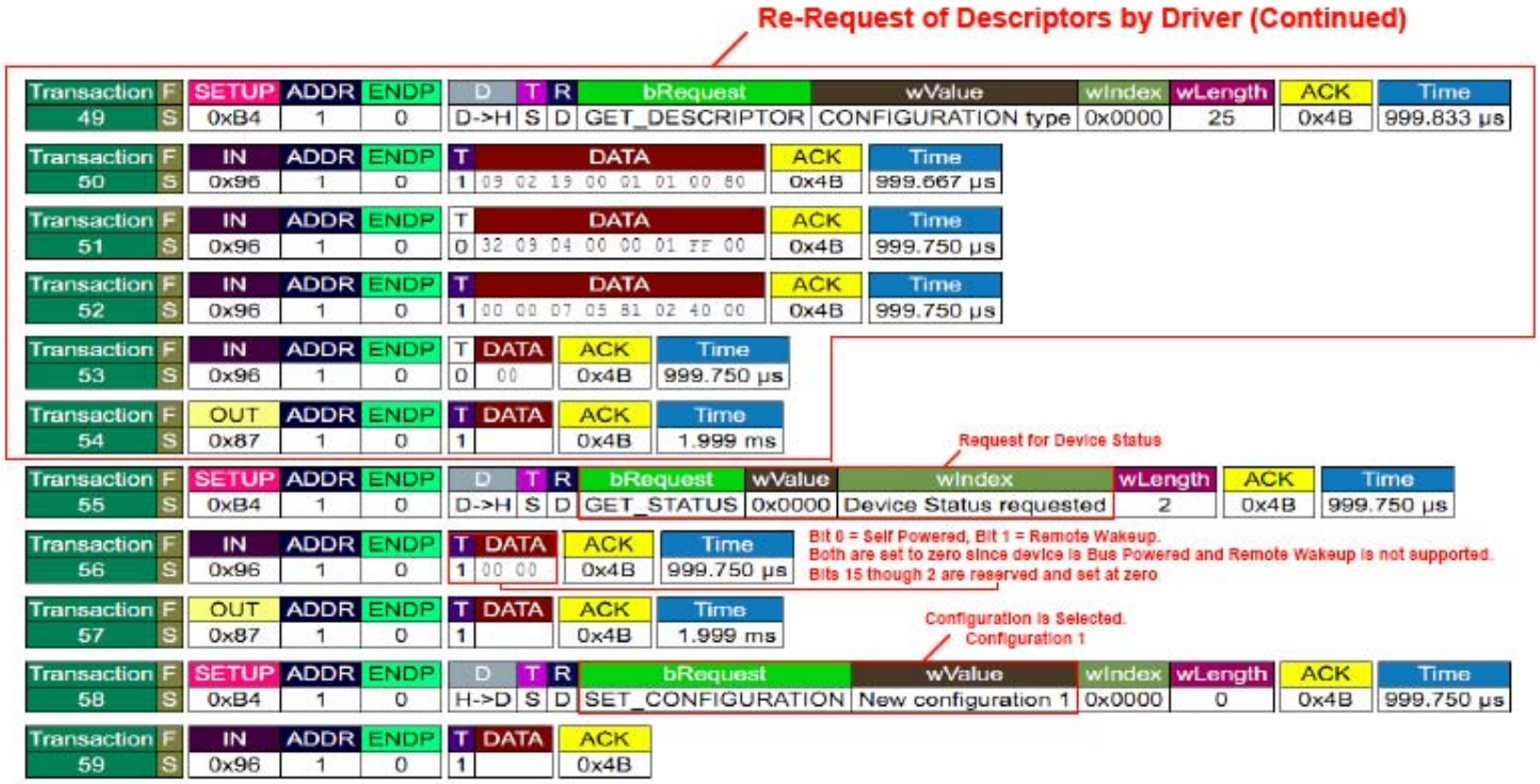
Transaction	F	SETUP	ADDR	ENDP	D	T	R	bRequest	wValue	windex	wLength	ACK	Time
45	S	0xB4	1	0	D->H	S	D	GET_DESCRIPTOR	CONFIGURATION type	0x0000	9	0x4B	999.750 µs

Transaction	F	IN	ADDR	ENDP	T	DATA	ACK	Time
46	S	0x96	1	0	1	09 02 19 00 01 01 00 80	0x4B	999.750 µs

Transaction	F	IN	ADDR	ENDP	T	DATA	ACK	Time
47	S	0x96	1	0	0	32	0x4B	999.833 µs

Transaction	F	OUT	ADDR	ENDP	T	DATA	ACK	Time
48	S	0x87	1	0	1		0x4B	1.999 ms

Bus Analyzer Capture - 5



Transaction	F	PID	ADDR	ENDP	D	T	R	bRequest	wValue	wIndex	wLength	ACK	Time
49	S	0xB4	1	0	D->H	S	D	GET_DESCRIPTOR	CONFIGURATION type	0x0000	25	0x4B	999.833 µs

Transaction	F	PID	ADDR	ENDP	T	DATA	ACK	Time
50	S	0x96	1	0	1	09 02 19 00 01 01 00 80	0x4B	999.667 µs
51	S	0x96	1	0	0	32 09 04 00 00 01 FF 00	0x4B	999.750 µs
52	S	0x96	1	0	1	00 00 07 05 81 02 40 00	0x4B	999.750 µs
53	S	0x96	1	0	0	00	0x4B	999.750 µs
54	S	0x87	1	0	1		0x4B	1.999 ms

Transaction	F	PID	ADDR	ENDP	D	T	R	bRequest	wValue	wIndex	wLength	ACK	Time
55	S	0xB4	1	0	D->H	S	D	GET_STATUS	0x0000	Device Status requested	2	0x4B	999.750 µs

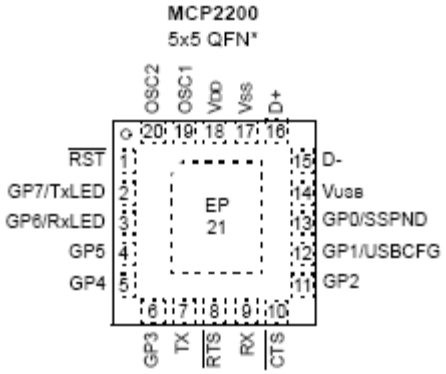
Transaction	F	PID	ADDR	ENDP	T	DATA	ACK	Time
56	S	0x96	1	0	1	00 00	0x4B	999.750 µs
57	S	0x87	1	0	1		0x4B	1.999 ms

Transaction	F	PID	ADDR	ENDP	D	T	R	bRequest	wValue	wIndex	wLength	ACK	Time
58	S	0xB4	1	0	H->D	S	D	SET_CONFIGURATION	New configuration 1	0x0000	0	0x4B	999.750 µs

Transaction	F	PID	ADDR	ENDP	T	DATA	ACK
59	S	0x96	1	0	1		0x4B

Device is Ready for Use!

USB-Serial

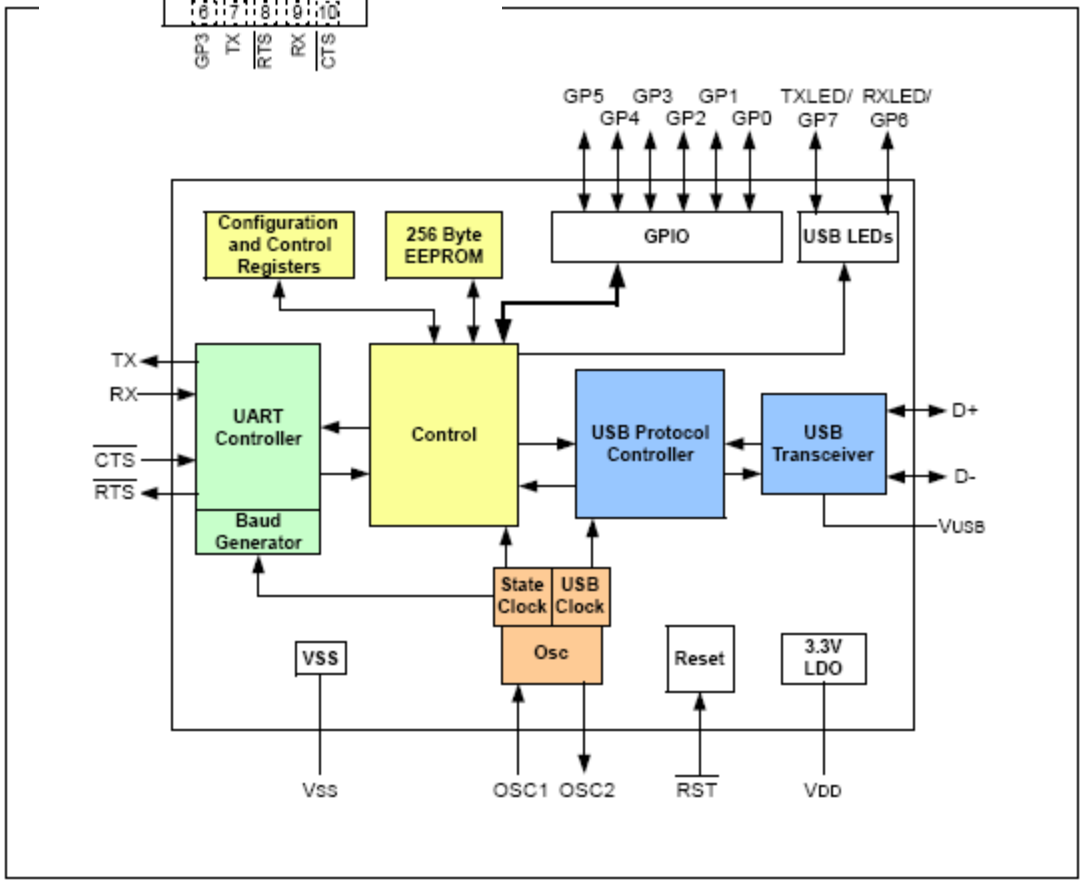


MICROCHIP

MCP2200

USB 2.0 to UART Protocol Converter with GPIO

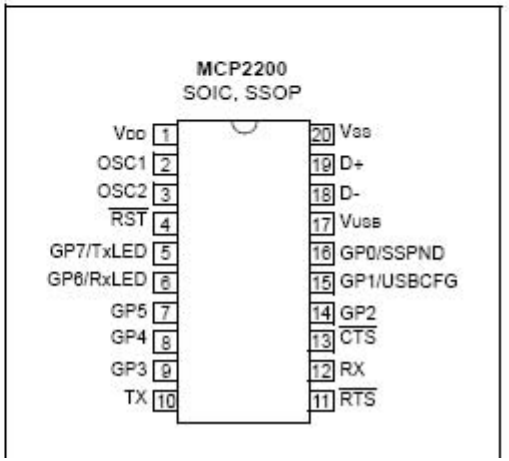
Bl



Package Types

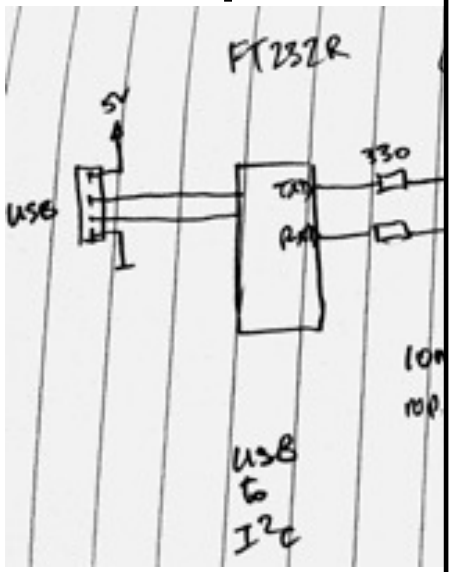
The device will be offered in the following packages:

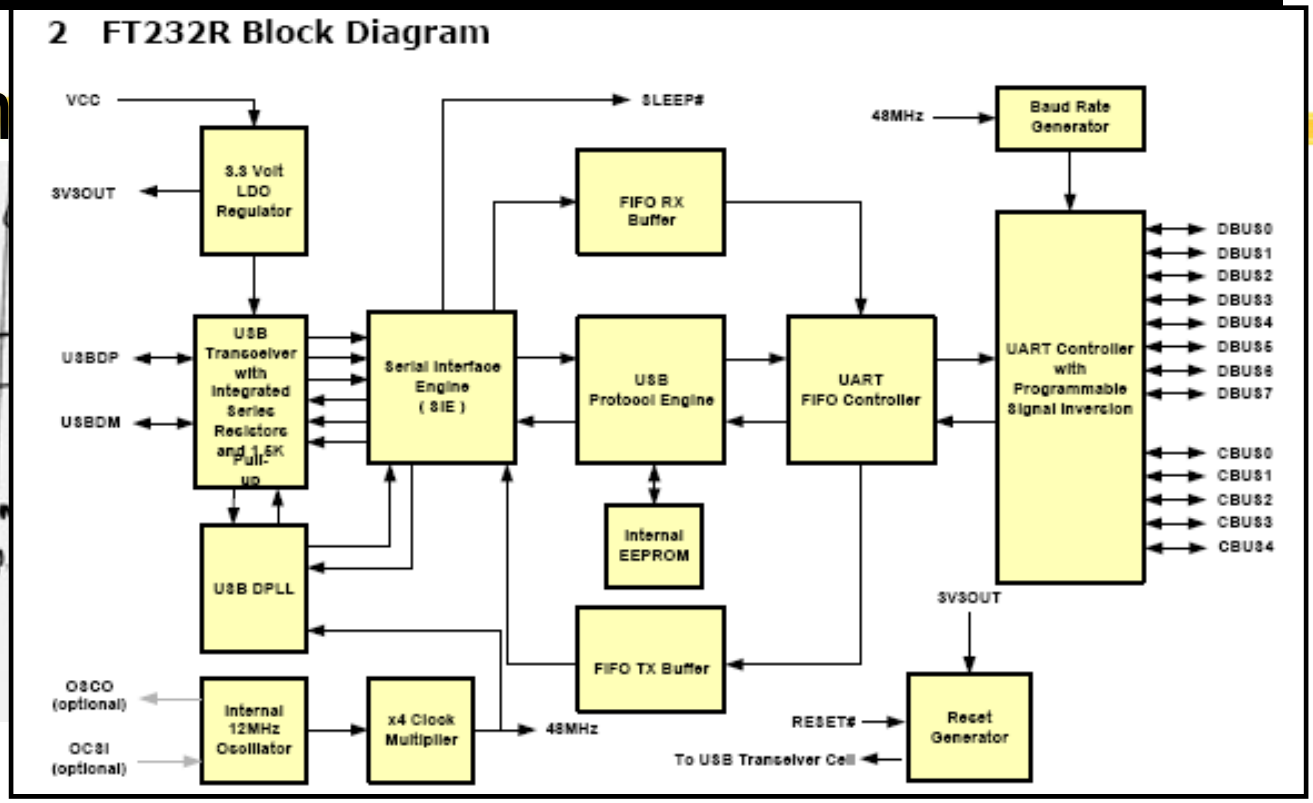
- 20-lead QFN (5 x 5 mm)
- 20-lead SOIC
- 20-lead SSOP

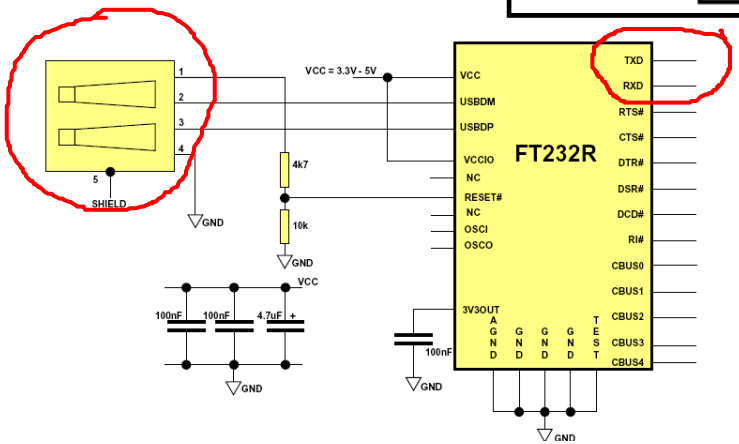


USB-Serial

- ftdichip.com

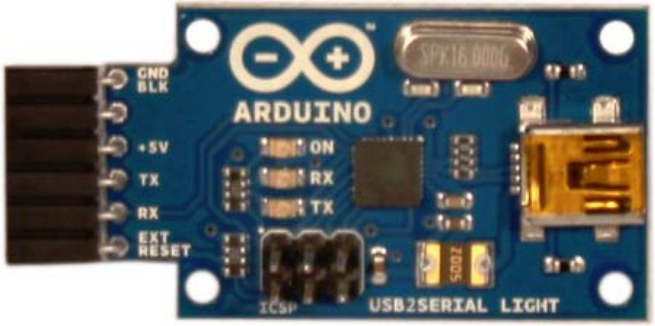




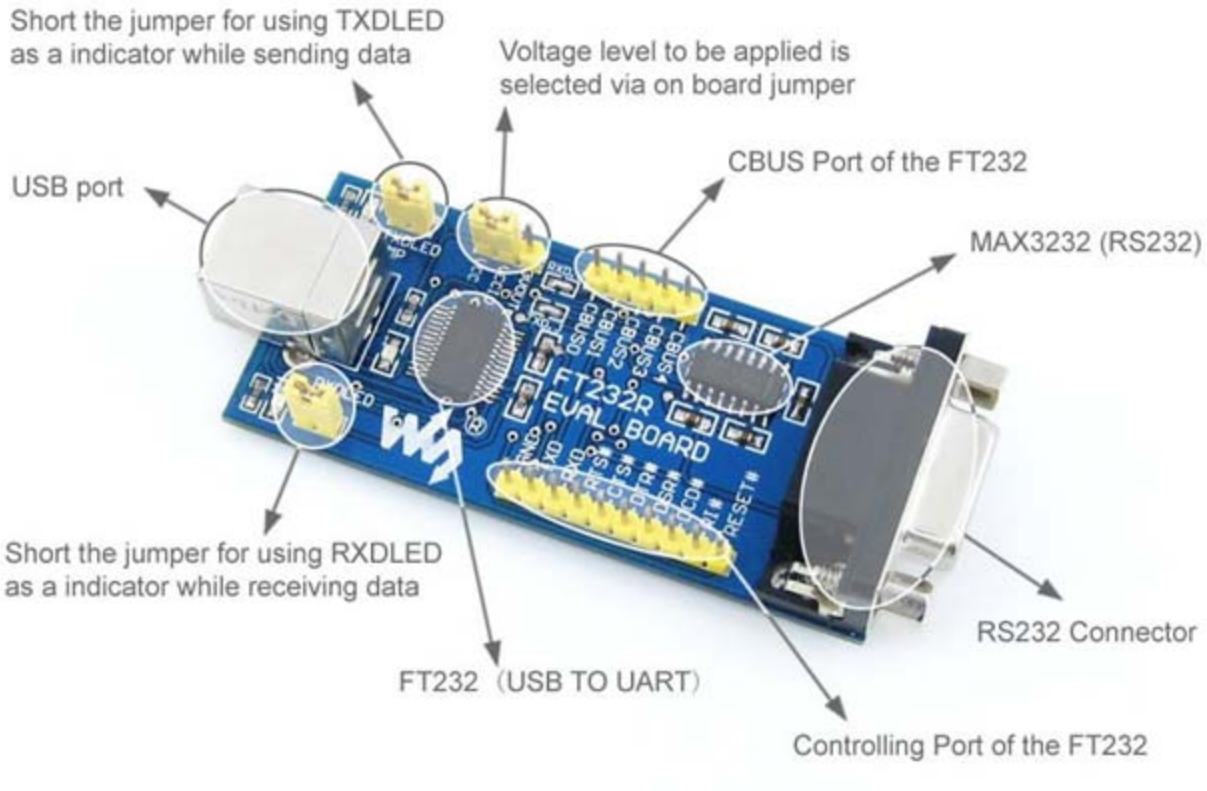


FTDI Chip

USB-Serial


ARDUINO
GND
BLK
+5V
TX
RX
EXT
RESET
ON
RX
TX
ICSP
USB2SERIAL LIGHT


USB to Serial FTDI
Chip Based RoHS Adapter


Short the jumper for using TXDLED
as a indicator while sending data
Voltage level to be applied is
selected via on board jumper
USB port
CBUS Port of the FT232
MAX3232 (RS232)
FT232R
EVAL BOARD
Short the jumper for using RXDLED
as a indicator while receiving data
RS232 Connector
FT232 (USB TO UART)
Controlling Port of the FT232


FT232RL
USB
PWR LED

References and Acknowledgment

- Many screen shots of this lecture note are from "***USB 101: An Introduction to Universal Serial Bus 2.0***" from AN57294 (written by Robert Murphy) from Cypress Semiconductor, San Jose, CA