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" Hused 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-

Comparison of the state and L state

✓K state: D+ Low and D- High

∺NRZI Coding – "State change"

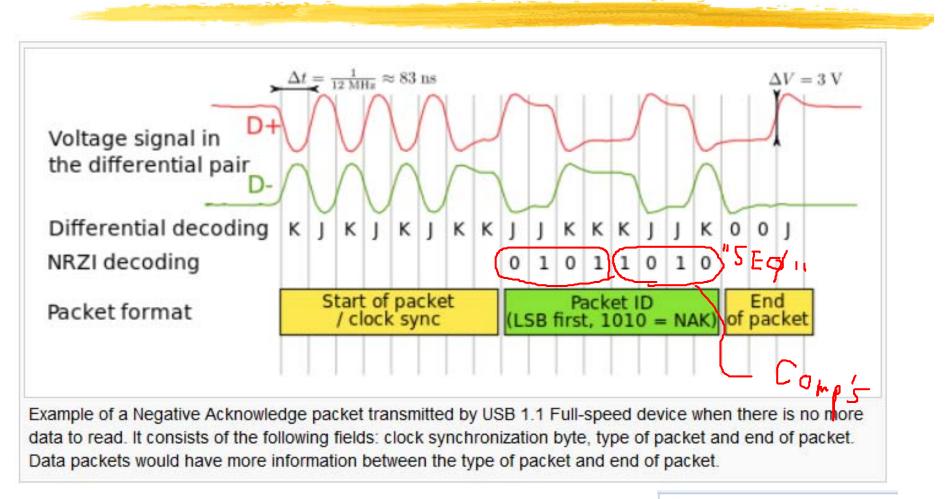
 $\bigtriangleup J \rightarrow K \text{ or } K \rightarrow J : 0$

 $\bigtriangleup J \rightarrow J \text{ or } K \rightarrow K: 1$

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



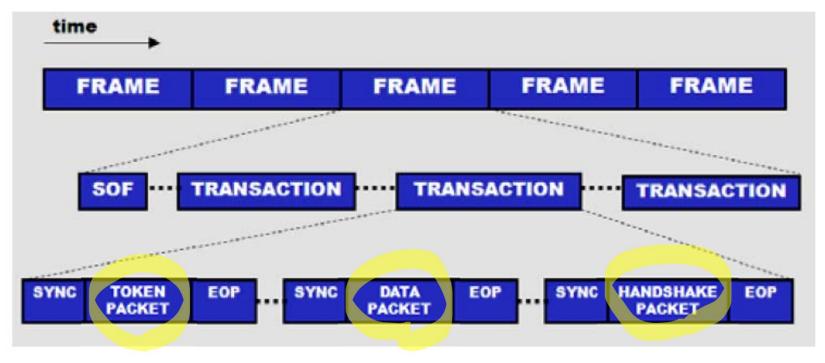
en.wikipedia.org/wiki/USB

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



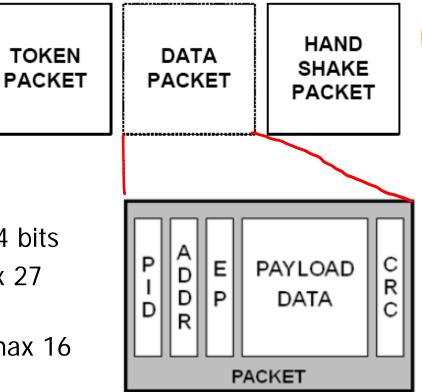
USB Packet Contents

Transaction

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

Information in a packet

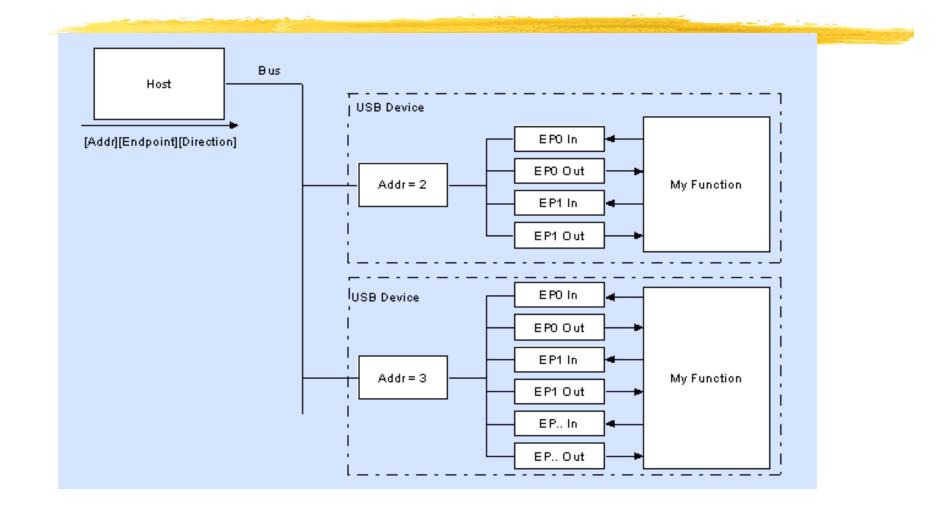
- Packet Identification (PID): 4 and 4 bits
- Device Addr (optional): 7 bits (max 27 devices)
- End Point (EP) (optional): 4 bits (max 16 endpoint addresses)
- Payload Data (optional)(up to 1023 bytes)
- CRC (optional)



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

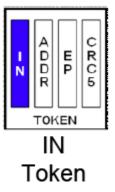
Group	PID Value	Packet Identifier
	0001	OUT Token
Takan	1001	IN Token
Token	0101	SOF Token
	1101	SETUP Token
	0011	DATA0
Data	1011	DATA1
Data	0111	DATA2
	1111	MDATA
	0010	ACK Handshake
Handshake	1010	NAK Handshake
Hanushake	1110	STALL Handshake
	0110	NYET (No Response Yet)
	1100	PREamble
Special	1100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

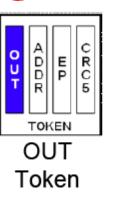
USB Host and Devices



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.







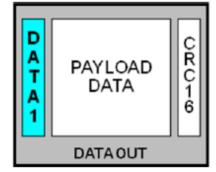
SOF	F R A M E	90 2 0	
т	ЖE	N	
S	O	F	
То	oke	en	

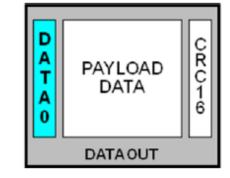
PID Value	Packet Identifier
0001	OUT Token
1001	IN Token
0101	SOF Token
1101	SETUP Token

116:73

Data Packet

- **B** Data packets follow IN, OUT, and SETUP token packets.
- Here the size of the payload data ranges from 0 to 1024 bytes depending on the transfer type.
- Hereica the successful data packet transfer, and the packet closes with a 16-bit CRC.
- Here the formula to the second second
- Here 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.

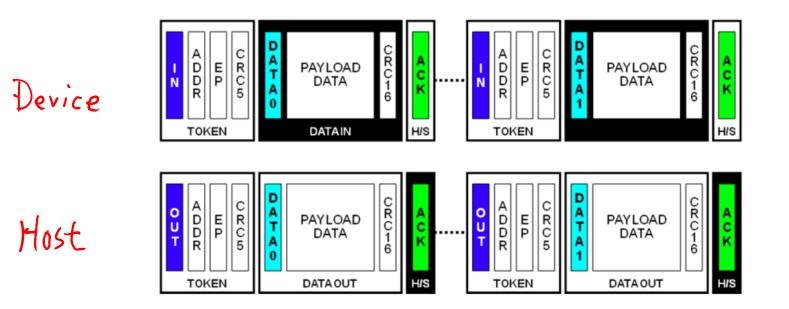




PID Value	Packet Identifier
0011	DATA0
1011	DATA1

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.



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
A	N	ST	N	0010	ACK Handshake
C	A	A	È	1010	NAK Handshake
	ĸ	ť	T	1110	STALL Handshake
H/S	H/S	HIS	HIS	0110	NYET (No Respon

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)

 $\mathbb{N}(629) = 6 \times 10^2 + 2 \times 10^1 + 9 \times 10^0 = 1 \times 2^9 + \dots$

 $\square D(25) = 2 \times 10^{1} + 5 \times 10^{0} = 1 \times 2^{4} + \dots$

₩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

Bata packets

⊡Up to 1023 bytes

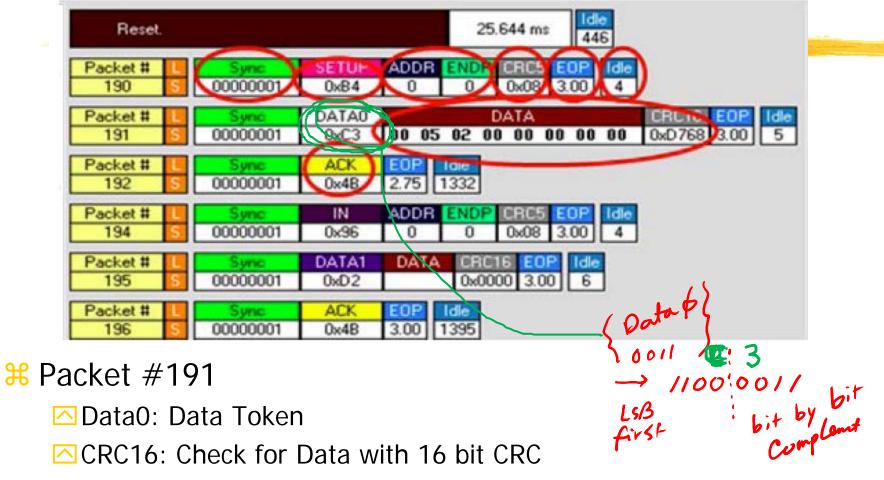
△16-bit CRC protection

USB Analyzer Output Example - A

-	Reset.					25.644 ms	Idle 446			
	Packet #		Sync	SETUR		ENDP GREA	EOP Ide			
	190		00000001	0x84	0	0 0x08	3.00 4			
	Packet #	L	Sync	DATA0		DATA		CRUICE		
	191	S	00000001	0xC3	00 05	02 00 00 00	0 00 00	0xD768 3.	00 5	
	Packet #		Sync	ACK	EOP	laie				
	192	S	00000001	0x4B		1332				
	Packet #		Sync	IN	ADDR	ENDP CRCS	EOP Idle			
	194	s	00000001	0x96	0	and the second se	3.00 4			
	Dealer #		0		DATA					
	Packet # 195	5	Sync 00000001	DATA1 0xD2	DATA	0x0000 3.00	Idle 6			
			- 4.2	1000	-					
	Packet # 196	-	Sync 00000001	ACK 0x4B		Idle 1395	/			
	136	2	00000001	UX4D	3.00	1335		setup		
		<u>л</u> 1	00				() I	ORI	1	
ЪЧ	acket -	Ŧ	90						1	
_			•					1011	0100	Δ
Ľ	≤L/S: L	_OW	/ Speec				LSB		Col	1,.1
			•				LSB First		col bit b	y b ⁽¹⁾
	≤Toker	ר P	acket				·		١	1
		_		-						

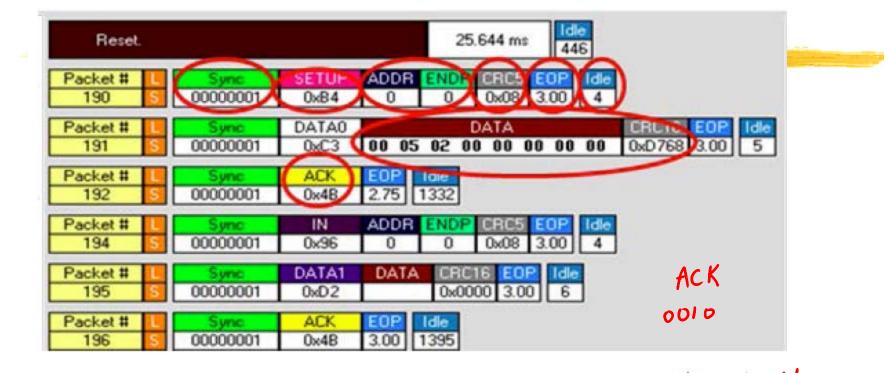
- CRC5: 5 bit CRC
- ► EOP: End of Packet

USB Analyzer Output Example - B

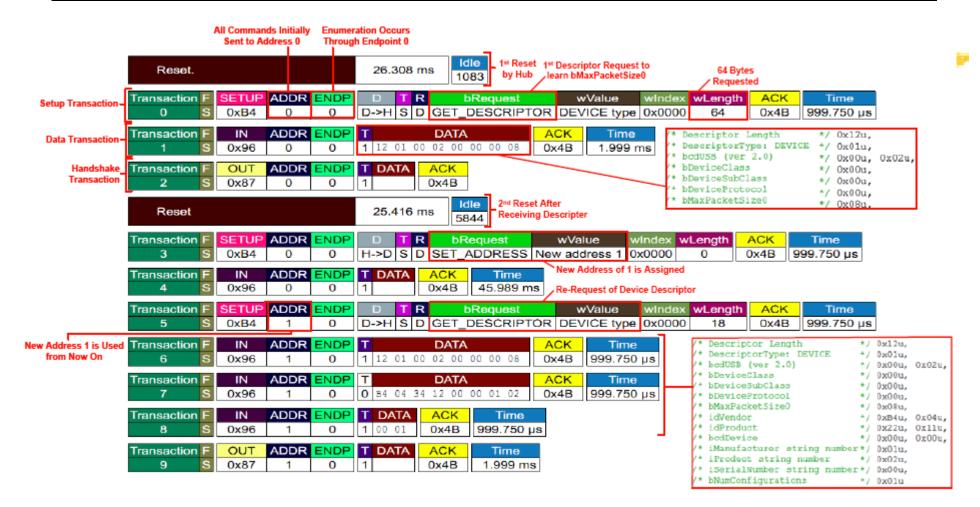


☐ Idle: Time between the current packet and the previous packet

USB Analyzer Output Example - C



Here Hacket #192
ACK: Successful indication of Data packet



19

Trar	nsaction F 10 S	OxB4	ADDR 1	ENDP 0		T R S D GET	bRequest DESCRIPT	OR CO	wValue NFIGURATIO	N type		wLength 9	ACK 0x4B	
	Packet # 165	F Sy S 0000		ETUP A	ADDR E		C5 EOP Id 17 2.75 2		equest for Configura or more details on S)	
Setup Transaction Packets (These 3 Packets Comprise – Transaction #251)	Packet # 166	F Sy S 0000		DATAO 0xC3 8	0 06 00	DATA 02 00 00		016 EOF						
	Packet # 167	F Sy S 0000			EOP Id 3.00 117	lle 798								
Trar	11 S	IN 0x96	ADDR 1	ENDP 0	T 1 09 02	DA1 2 19 00 0	A 1 01 00 80	ACK 0x4B	Time 999.750 μs	/* /* /*	Descripto wTotalLer		TG */ 0x0 */ 0x1	2u, 9u, 0x00u,
Trar	12 F	IN 0x96	ADDR 1	ENDP 0	T DAT/ 0 32	A ACK 0x4B		s			iConfigur	rationValue	*/ 0x0 */ 0x0 */ 0x0 */ 0x8	lu, Du,
Trar	saction F 13 S	OUT 0x87	ADDR 1	ENDP 0	T DAT/ 1	A ACK 0x4B		5		/*	bMaxPover		*/ 0x3	

Request for Configuration (2nd Request), Interface,

		and Endpoint Descriptors	
Transaction F	SETUP ADDR ENDR 0xB4 1 0	D T R bRequest wValue windex wLength ACK Time D=>H S D GET_DESCRIPTOR CONFIGURATION type 0x0000 255 0x4B 999.750 µs	
Transaction F 15 S	IN ADDR ENDR 0x96 1 0		
Transaction F 16 S	IN ADDR END 0x96 1 0	P DATA ACK Time /* NMaxPower */ 0x04u, 0 32 0.9 0.4 0.0 0.1 FF 0.0 399.750 µs /* binterfacebraiter */ 0x04u, /* binterfacebraiter */ 0x04u, /* binterfacebraiter */ 0x04u,	
Transaction F 17 S Transaction F 18 S	IN ADDR ENDR 0x98 1 0 IN ADDR ENDR 0x96 1 0	1 00 00 07 05 61 02 40 00 0x4B 999.750 µs /* binterval */ 0x0Au	
Transaction F 19 S	OUT ADDR ENDP 0x87 1 0	1 0x4B 1.999 ms Request for Device_Qualifier Descriptor.	
Transaction F 20 S	0xB4 1 0	D T R bRequest wValue wIndex wLength ACK Time D->H S D GET_DESCRIPTOR DEVICE_QUALIFIER type 0x0000 10 0x4B 999.750 μs	
Transaction F 21 S	INADDRENDP0x9610	STALL Time This is a Full Speed only device. As a result, a 0x78 2.000 ms STALL (request error) is issued. 1 st String Descriptor Request (Language IDs)	
Transaction F 22 S	SETUPADDRENDP0xB410	D T D D Request wValue windex wLength ACK Time D->H S D GET_DESCRIPTOR STRING type, LANGID codes requested Language ID 0x0000 255 0x4B 999.833 μs	
Transaction F 23 S	IN ADDR ENDI 0x96 1 0	T DATA ACK Time /* Descriptor Length */ 0x04u, 1 04 03 09 04 0x4B 1.999 ms /* DescriptorType: STRING */ 0x03u, /* Language Id */ 0x04u, /* Language Id */ 0x04u,	
Transaction F 24 S	OUT ADDR END 0x87 1 0	T DATA ACK Time 1 0x4B 1.999 ms	
Transaction F 25 S	SETUPADDRENDR0xB410	P D T R bRequest wValue wIndex wLength ACK Time D->H S D GET_DESCRIPTOR STRING type, Index 2 Language ID 0x0409 255 0x4B 999.750 μs	
Transaction F 26 S	IN ADDR ENDI 0x96 1 0	1 18 03 55 00 53 00 42 00 0x48 999.750 µs /* Descriptor Length */ 0x184,	
Transaction F 27 S	IN ADDR END 0x96 1 0	P T DATA ACK Time '' DescriptorType: STRING '/ Ox03u, 0 20 00 45 00 70 0, 'B', 0, 'B', 0, ' , 0, 'E', 0, 'x', 0, 'a', 0, 'n', 0, 'p', 0, '1', 0 ''e', 0, ''e', 0, ''e', 0, ''e', 0, ''e', 0,	
Transaction F 28 S	IN ADDR ENDI 0x96 1 0	1 6D 00 70 00 6C 00 65 00 0x4B 999.750 μs S= 53h m = 61h	
Transaction F 29 S	IN ADDR ENDF 0x96 1 0	P T DATA ACK Time B = 42h p = 70h 0 0x4B 1.999 ms E = 45h 1 = 50h 2 0 0x4B 1.999 ms 45h 1 = 65h 2	21

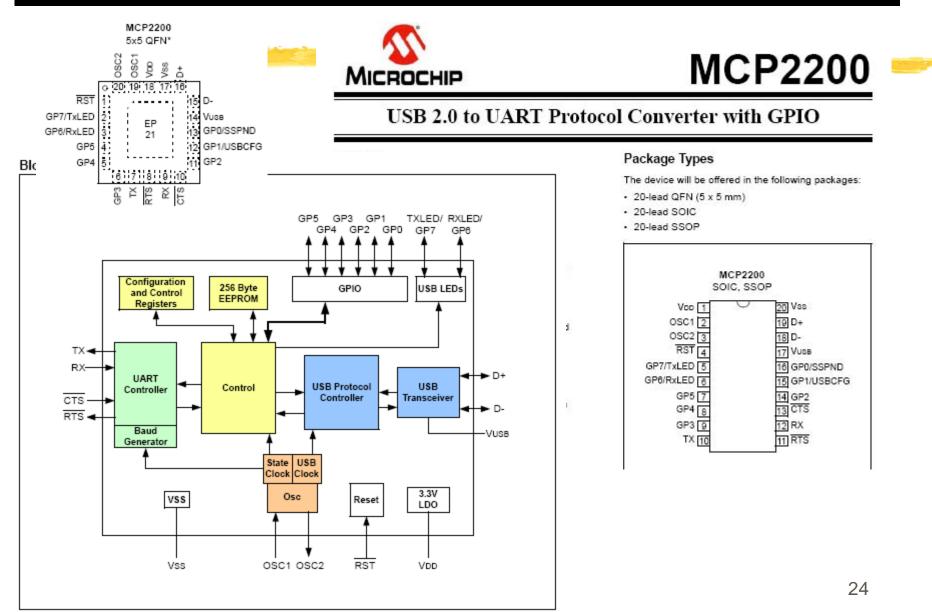
ansaction F	OUT 0x87	ADDR	ENDP 0	T DA	and the second second	KAB	Time 1.999 m			/	Re-	Request	of Des	criptors	by Driv	er	
ansaction F 31 S	SETUP 0x84	ADDR 1	ENDP 0	Comparison and the owner where	T R S D		Request DESCRIPT		RING typ	wValu e, LANGI	the second s	quested L	winc anguage l	and the state of the second	wLength 255	ACK 0x4B	Time 999.750 µs
ansaction F 32 S	IN 0x96	ADDR 1	ENDP 0	and the second se	DATA 03 09 1			lime 999 ms									
ransaction F 33 S	OUT 0x87	ADDR 1	ENDP 0	T DA	and the second se	ACK 0x4B	Time 1.999 m	3									
ansaction F 34 S	Ox84	ADDR 1	ENDP 0	D D->H	T R S D (A COLUMN TWO IS NOT	Request DESCRIPT	OR STR	wVal RING typ		the second se	Index e ID 0x040	wLengt	h ACK 0x4B	Time 999.750	the second s	
ansaction F 35 S	IN 0x98	ADDR 1	ENDP 0			DATA	00 42 00	ACK 0x4B	Tim 999.83								
anstaction F 36 S	IN 0x96	ADDR 1	ENDP 0			DATA 00 TE	00 61 00	ACK 0x4B	999.75								
ansaction F 37 S	IN 0x96	ADDR	ENDP 0	Т 1 бр		DATA	00 65 00	ACK 0x4B	Tim 999.75								
ransaction F 38 S	IN 0x96	ADDR 1	ENDP 0	T DA		KAB	Time 1.999 m	3									
ransaction F 39 S	OUT 0x87	ADDR 1	ENDP 0	T DA	_	XCK 1x4B	Time 29.993 m	5									
ansaction F 40 S	DxB4	ADDR 1	ENDP 0	and the owner where the party of	TR SD		Request	the second se	wValue VIGE typ		wLength 18		Time 999.750 µ	5			
ransaction F 41 S	IN 0x96	ADDR 1	ENDP 0			DATA	00 00 08	ACK 0x4B	100 999.75	Contract of the local division of the local							
ansaction F 42 S	IN 0x96	ADDR 1	ENDP 0	_		DATA 12 00	00 01 02	ACK 0x4B	999.75	and the second se							
ansaction F 43 S	IN 0x96	ADDR	ENDP 0	T DA 1 00		ACK 0x4B	Time 999.750	a									
ansaction F 44 S	OUT 0x87	ADDR 1	ENDP 0	T DA		x4B	Time 1.999 m	3									
ansaction E 45 S	DxB4	ADDR 1	ENDP 0		T R S D		DESCRIP	OR CO		alue RATION I§		wLengt 0 9	ACK 0x4B	Time 999.750			
ansaction E 46 S	IN 0x96	ADDR 1	ENDP 0	T 1 09	02 19	DATA 00 01	01 00 80	ACK 0x4B	the second se	the second s							
ansaction F 47 S	IN 0x96	ADDR 1	ENDP 0	T DA 0 31		ACK 0x4B	Time 999.833	IS									
48 S	OUT 0x87	ADDR 1	ENDP 0	T DA 1	and the second se	ACK 0x4B	Time 1.999 m	s									

Transactic	n F	SETUP	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
Transactio	n F	IN	ADDR	ENDP	T DATA ACK Time 1 03 02 19 00 01 01 00 80 0x4B 999.667 µs
50	S	0x96	1	0	
fransactio	n F	IN	ADDR	ENDP	T DATA ACK Time 0 32 03 04 00 01 FF 00 0x4B 999.750 μs
51	S	0x96	1	0	
Fransactio	n F	IN	ADDR	ENDP	T DATA ACK Time 1 00 00 07 05 81 02 40 00 0x4B 999.750 μs
52	S	0x96	1	0	
fransactio	n F	IN	ADDR	ENDP	T DATA ACK Time 0 00 0x4B 999.750 μs
53	S	0x96	1	0	
Fransactic	n F	OUT	ADDR	ENDP	T DATA ACK Time 1 0x4B 1.999 ms Request for Device Status
54	S	0x87	1	0	
Fransactic	n F	SETUP	ADDR	ENDP	D T R bRequest wValue windex wLength ACK Time D->H S D GET_STATUS 0x0000 Device Status requested 2 0x4B 999.750 µs
55	S	0xB4	1	0	
Fransactio	n F	IN	ADDR	ENDP	TOATA ACK Time Bit 0 = Self Powered, Bit 1 = Remote Wakeup. 1 00 00 0x4B 999.750 µs Bits 15 though 2 are reserved and set at zero.
56	S	0x96	1	0	
Fransactio	n F	OUT	ADDR	ENDP	T DATA ACK Time Configuration is Selected. 1 0x4B 1.999 ms Configuration 1
57	S	0x87	1	0	
Fransactio	n F	SETUP	ADDR	ENDP	D T R bRequest wValue wIndex wLength ACK Time H->D S D SET_CONFIGURATION New configuration 1 0x0000 0 0x4B 999.750
58	S	0xB4	1	0	
Fransactio	n F	IN	ADDR	ENDP	T DATA ACK
59	S	0x96	1	0	1 0x4B

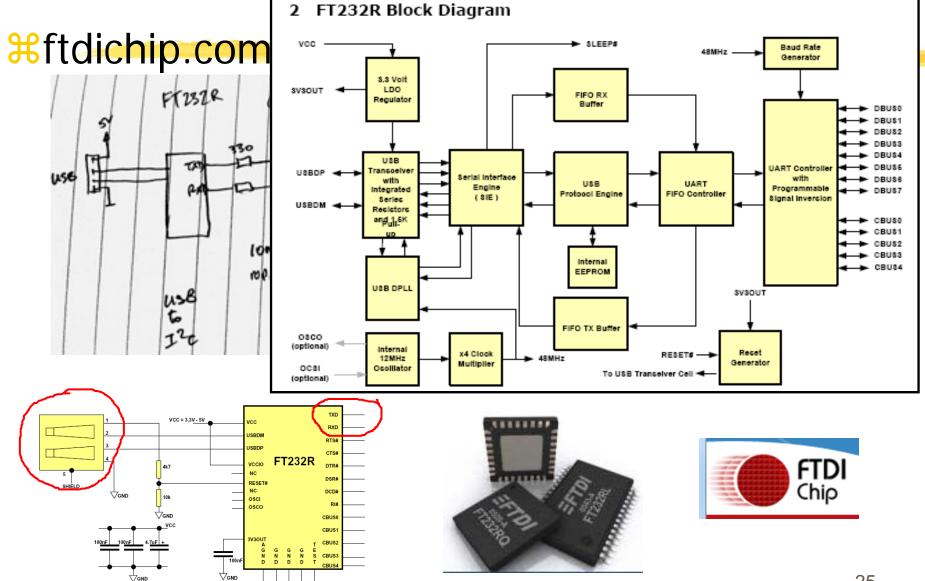
Re-Request of Descriptors by Driver (Continued)

Device is Ready for Use!

USB-Serial

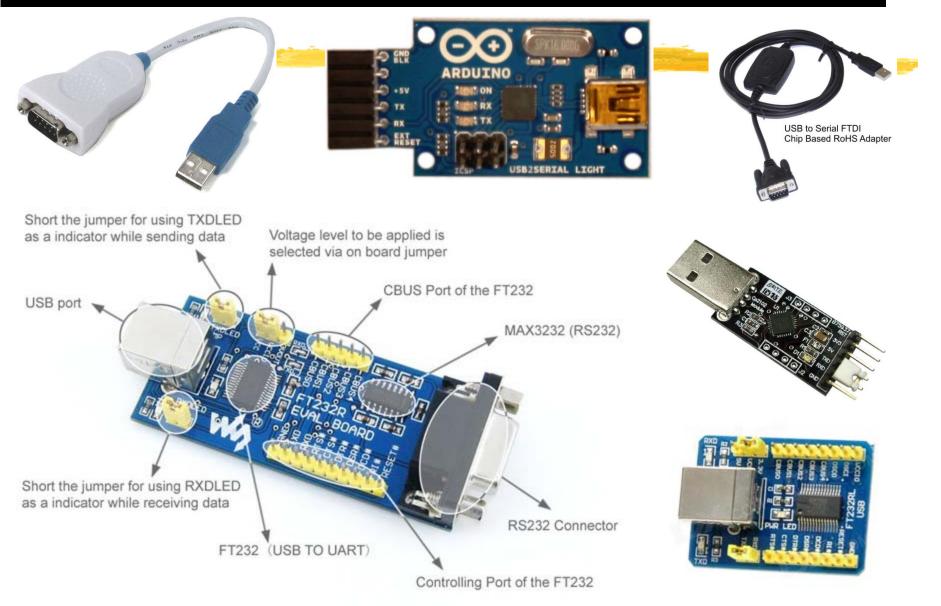


USB-Serial



ḋgnd

USB-Serial



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