Data Buses

We have a need to transmit digital information. How?

Historical

Le Sage 1774: 26 static electric lines between 2 rooms. Completely impractical.

Ronalds 1816: static electric lines with code book. Several miles.

Schilling 1832: 6 wires of data (binary) measured with current meters. 2 control wires (Ground + bell)

= <Letter> + Ground wire

Wires reduced to 2 later .... binary code.

Gauss 1833: Transformers ∃ℇ and binary code.

Morse 1837: Electro mechanical "register" recording device on paper tape
A binary code "Morse code" (but not the Morse code we know)

1844 "WHAT HATH GOD WROUGHT" from DC to Baltimore (44 miles)

Morse Code

Variable length code using binary signal

• — = A = dit dah =

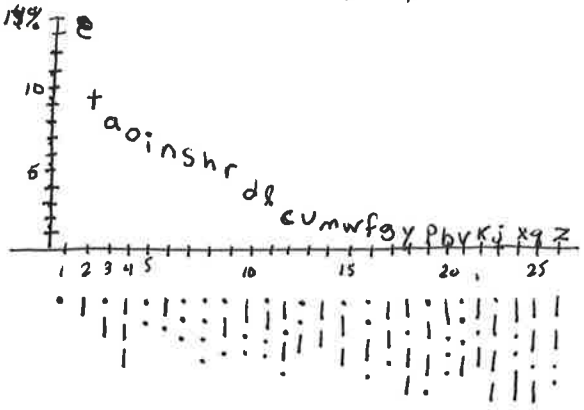
— — • • = Z = dah dah dit dit =

— • = N = dah dit =

— • — • = C = dah dit dah dit =

How do we distinguish between NN and C? Extra space / gaps

Length of code depends on letter frequency



Baudot Code

Fixed length Easier for a machine to understand.

How many bits do we need? States = $base^{bits}$

In binary: $S = 2^{bits} \Rightarrow \log_2 S = bits$

26 letters reqire $\log_2 26 = 4.7$ bits round up to 5

Invented in 1605 by Francis Bacon.

Re-invented by Emile Baudot in 1870s

$2^0 = 1$
$2^1 = 2$
$2^2 = 4$
$2^3 = 8$
$2^4 = 16$
$2^5 = 32$ ←
$2^6 = 64$
$2^7 = 128$
$2^8 = 256$
$2^9 = 512$
$2^{10} = 1024$
$2^{11} = 2048$
$2^{12} = 4096$
$2^{13} = 8192$
$2^{14} = 16384$
$2^{15} = 32768$
$2^{16} = 65536$

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
LETTERS	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
FIGURES	–	?	:	WHO ARE YOU	3	%	@	£	8	BELL	(	)	.	,	9	0
CODE ELEMENTS 1	●	●		●	●	●				●	●					
2	●		●				●		●	●	●	●				●
	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
3			●			●		●	●		●		●	●		●
4		●	●	●		●	●			●	●		●	●	●	
5		●					●	●				●	●		●	●

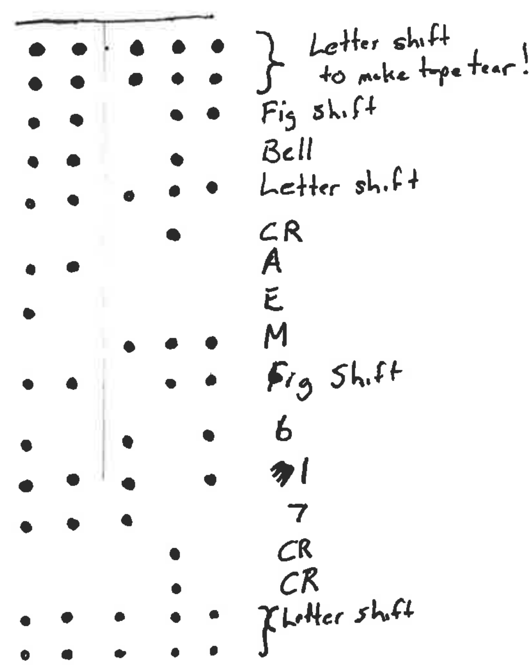
	Q	R	S	T	U	V	W	X	Y	Z	CARRIAGE RETURN	LINE FEED	LETTERS	FIGURES	SPACE	ALL-SPACE NOT IN USE
LETTERS	Q	R	S	T	U	V	W	X	Y	Z	CARRIAGE RETURN	LINE FEED	LETTERS	FIGURES	SPACE	ALL-SPACE NOT IN USE
FIGURES	1	4	'	5	7	=	2	/	6	+						
CODE ELEMENTS 1	●		●		●		●	●	●	●			●	●		
2	●	●			●	●	●					●	●	●		
	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
3	●		●		●	●		●	●				●		●	
4		●				●		●			●		●	●		
5	●			●		●	●	●	●	●			●	●		

● INDICATES A MARK ELEMENT (A HOLE PUNCHED IN THE TAPE)
○ INDICATES POSITION OF A SPROCKET HOLE IN THE TAPE

The International Telegraph Alphabet

2 code subsets selected with "Letter shift" and "Figure shift"

AEM617 =



58 symbols in 5 bits

Requires synchronization

Baud

$baud_{binary}$ = bits/second

A 1200 baud modem = 1200 bits/s uncompressed

ASCII 7 bit binary code "American Standard Code for Information Interchange"

$2^7 = 128$ = 95 printable + 33 control codes

[a-z]
[A-Z]
[0-9]
punctuation

{ 010 0000 to 111 1110

NULL
BEL
LF
CR
NAK
ESC
DEL
+ ....

000 0000 to 001 1111 and 111 1111

Very good for English. What about other languages. ü æ 私 T I Z påsk

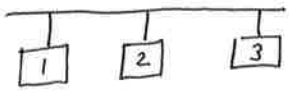
Extended ASCII 8 bits many many codings (i.e maps from bits to characters) "mojibake"

ASCII Common for data output on serial bus

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

Topology

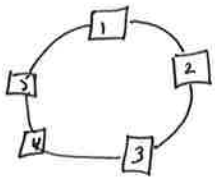


Linear

All listen to all information

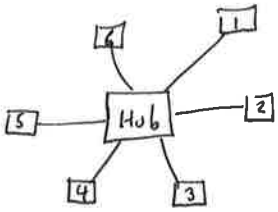
Saturates

Cheap!



Ring

- Pass messages along ring
- Verify topology
- multipath
- latency

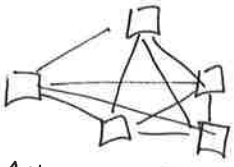


Hub

All messages go through hub

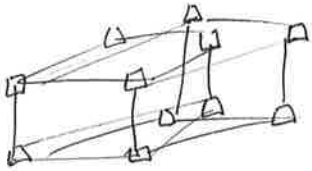
Expensive hub

Single point of failure



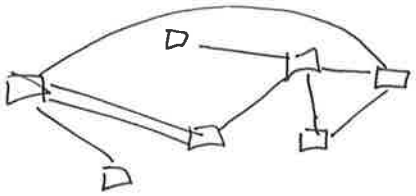
All to all

Low latency



Hypercube

Real not Supercomputers NOT avionics!



Point to Point

weight and complexity.

Ok for small aircraft

Attributes desirable

- Efficiency ≡ $\frac{\text{data bits}}{\text{total bits}}$ for our Boudot AEM 617 ex. $\eta = 35\% = \frac{30}{85}$
- Latency ≡ delay " " " " at 300 baud $\Delta t = 0.28$ s
- Deterministic ≡ a priori behavior
- Integrity ≡ error detection + correction

Not independent

Transmission bus protocols

- Time slot (time division multiplex allocation TDMA): Each unit has a specified time (known beforehand) to transmit.
- Command: A central controller specifies when a unit can communicate (and for how long)
- Token passing: Transmit only when in possession of a single token
- Contention: Transmit, if a collision, wait a random time and try again
- CDMA: (Code division multiple access) known random signal mixed with T_x signal

  think of multiple languages.... Ignore one and listen to another
- ... And others

Experiments!

Error Detection

How can we verify that data was likely sent correctly? Hash

Cyclic Redundancy Check CRC

generate a check value based on a longer bit sequence

$1011101101101 \Rightarrow 1$ (f())

catanate check value to sent sequence

1011011011011

If an error occurs during transmission

1010011011011

The receiver evaluates the bit sequence and verifies the check value

$1010011011011 \Rightarrow 0 \neq 1$ (f(x))

Very common and prudent

CRC1 parity bit (polynomial x+1)

XOR the bit sequence with a sequensially shifted CRC bit pattern

x+1 ⇒ 11

Example

```
1 0 1 1 1 0 1 1 0 1 1 0 1
1 1
---------------------------
0 1 1 1 1 0 1 1 0 1 1 0 1
  1 1
  -----------------------
  0 0 1 1 0 1 1 0 1 1 0 1
    0 1 1
    -------------------
    0 0 0 1 1 0 1 1 0 1
        1 1
        ---------------
        0 0 0 1 1 0 1
            1 1
            -----------
            0 0 0 1 1
                1 1
                -------
                0 0 0 1
                  1 1
                  -----
                  0 1
```

Remainder is check value

Can find a single bit error (could be in the check value!)

CRC1(1011101101101) = 1

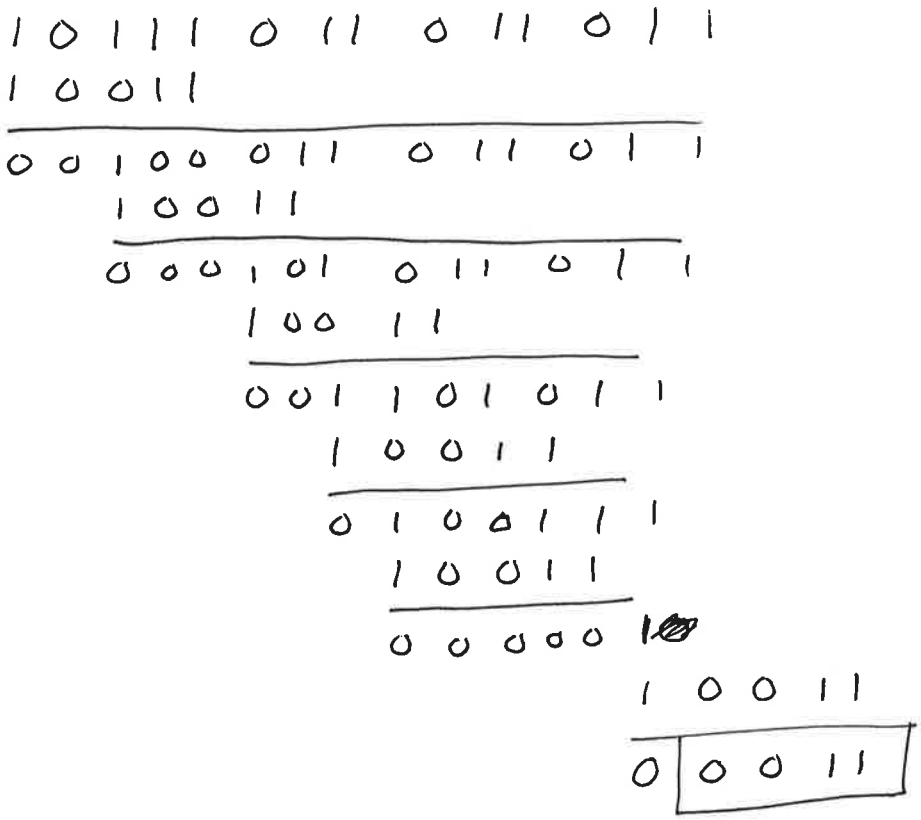
"Parity bit" counts odd/even # bits!

CRC5-USB

$x^5 + x^2 + 1 \Rightarrow x^5 + 0x^4 + 0x^3 + x^2 + 1$

1 0 0 1 1

Ex:



CRC32

Very common in internet applications

$$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

0x 04C11DB7

In general, you can detect errors with CRC but not correct.

Unless you have redundant information

Send multiple times and perform an operation to find and remove errors

NMEA

GPS output standard
ASCII serial output

"NMEA 0183"
proprietary and costs money to get the official standard

Structure

82 character messages

- $ start delimiter
- * Checksum delimiter
- <LF> Line feed

Message:
- 2 character source
- 3 character type of message "Sentence"
- message text separated by commas

XOR of ASCII between $ and * (2 byte)

Common Sentences

DD.xxxx°
DD MM' SS"
DD MM.MMM'

GPGGA = Fix data (i.e. location and time and quality)

DGPS data

$GPGGA,154323.000,3312.9535,N,08732.6477,W,2,06,1.16,128.6,M,-28.6,M,0000,0000*56

- Fix data
- UTC time: 3:42 pm and 23 seconds in Greenwich
- 33° 12.9535' N latitude
- 87° 32.6477' W longitude
- type of fix: 2 = Differential GPS
- # sat tracked
- Dilution of precision "accuracy"
- Altitude Meters
- geoid height above ellipsoid
- Checksum

GPGSA = Satellites tracked and DOP

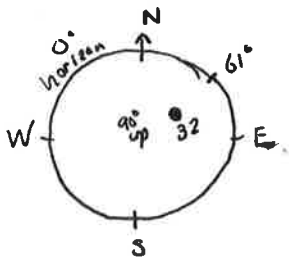
$GPGSA,A,3,12,22,10,25,03,32,,,,,,,1.47,1.16,0.90*08

- Auto
- 3D fix
- Satellites #s (stay tuned for what the #s mean)
- Dop = $\sqrt{HDop^2 + VDop^2}$
- Checksum

GPGSV = Satellites in View (detailed data)

$GPGSV,4,1,13,31,79,309,20,14,62,013,17,32,51,061,21,51,45,213,35*7D

- # Sentences needed
- Sentence #
- Satellites in View
- Satellite #32, 51° elevation, 61° azimuth, SNR = 21 signal to noise ratio dB
- Checksum



GPRMC = position, time, velocity

$GPRMC,154323.000,A,3312.9535,N,08732.6477,W,0.53,223.54,060318,,,D*76

- time: 1543 Zulu, 23 seconds
- Activ.
- latitude + longitude: 33° 12.9537' N, 87° 32.6477' W
- Speed kts
- track deg
- date: 6th March 2018
- ??

Ex: Where is this GPS? # Satellites and location in sky? Time?

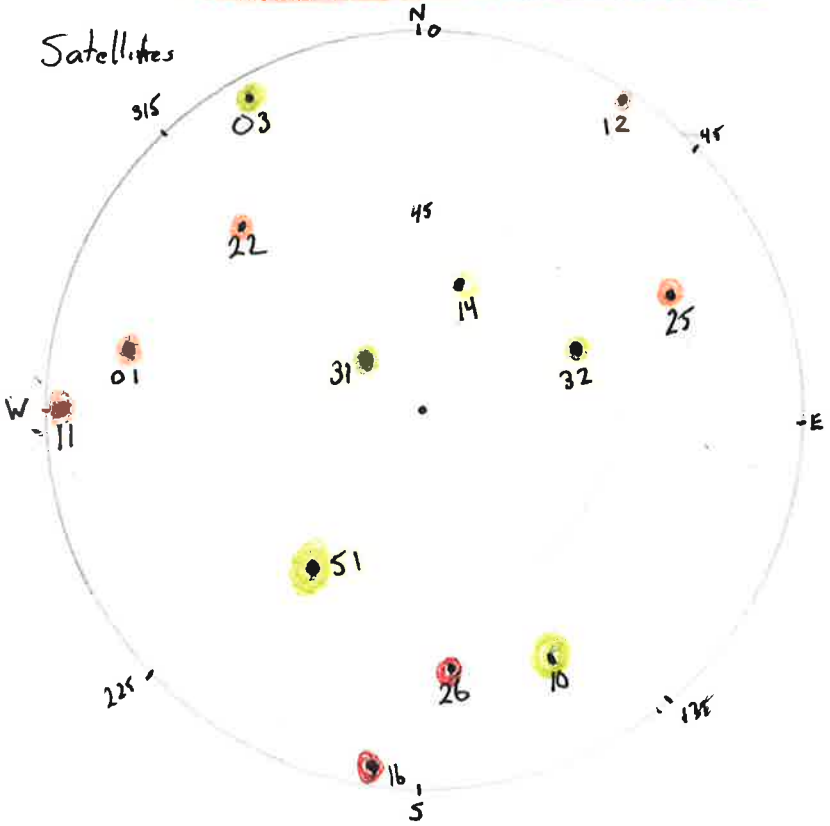
$GPGGA,154323.000,3312.9535,N,08732.6477,W,2,06,1.16,128.6,M,-28.6,M,0000,0000*56
$GPGSA,A,3,12,22,10,25,03,32,,,,,,,1.47,1.16,0.90*08
$GPGSV,4,1,13,31,79,309,20,14,62,013,17,32,51,061,21,51,45,213,35*7D
$GPGSV,4,2,13,22,35,312,15,25,34,061,22,10,34,148,32,26,29,173,*78 ← Error?
$GPGSV,4,3,13,01,22,285,19,03,17,318,26,12,05,034,17,11,05,270,19*7E
$GPGSV,4,4,13,16,03,188,*4E
$GPRMC,154323.000,A,3312.9535,N,08732.6477,W,0.53,223.54,060318,,,D*76
$GPVTG,223.54,T,,M,0.53,N,0.99,K,D*3C

Time: 1543 Zulu + 23 s on 6th March 2018

Location: 33° 12.9535' N 87° 32.6477 W

Nearly stationary 0.53 kts

Satellites



Location shows that precision signal is possibly bouncing off of neighboring building?