ENGI 128

INTRODUCTION TO ENGINEERING SYSTEMS

Lecture 13:
Digital Communications I
"Understand Your Technical World"

Representing Digital Information

The Lowly Bit

10

But they travel in swarms…

Let’s count…

0

Let’s count…

1

Let’s count…

2

Let’s count…

3

Let’s count…

9

We’ve run out of room in the digit. Now what do we do?

Let’s count…

10

We add another digit

Let’s count…

99

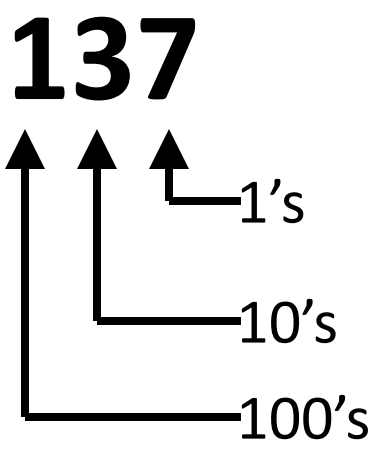
And another…

Let’s count…

100

And another…

Let's count…



We can read off the number by the places
This is called a "base 10" number system, because each digit can take on 10 different values

Let’s count in binary…

0

Let’s count in binary…

1

Let’s count in binary…

1

We’ve run out of room in the digit. Now what do we do?

Let’s count in binary…

10

We add another digit

Let’s count in binary…

11

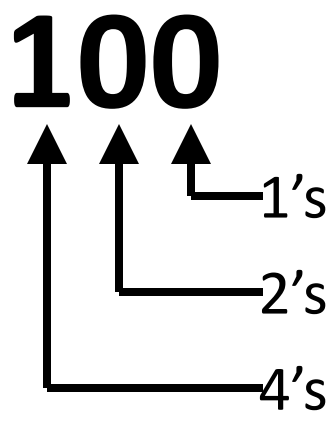
We’re out of room AGAIN?! ok, add another digit

Let’s count in binary…

100

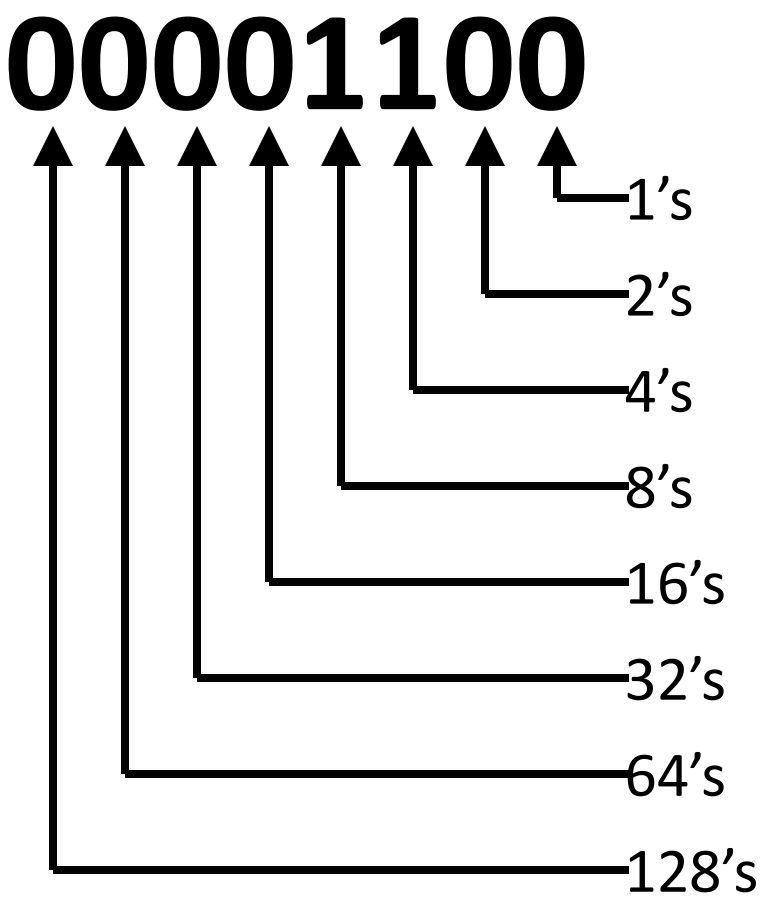
What does this number mean in decimal?

Let’s count…



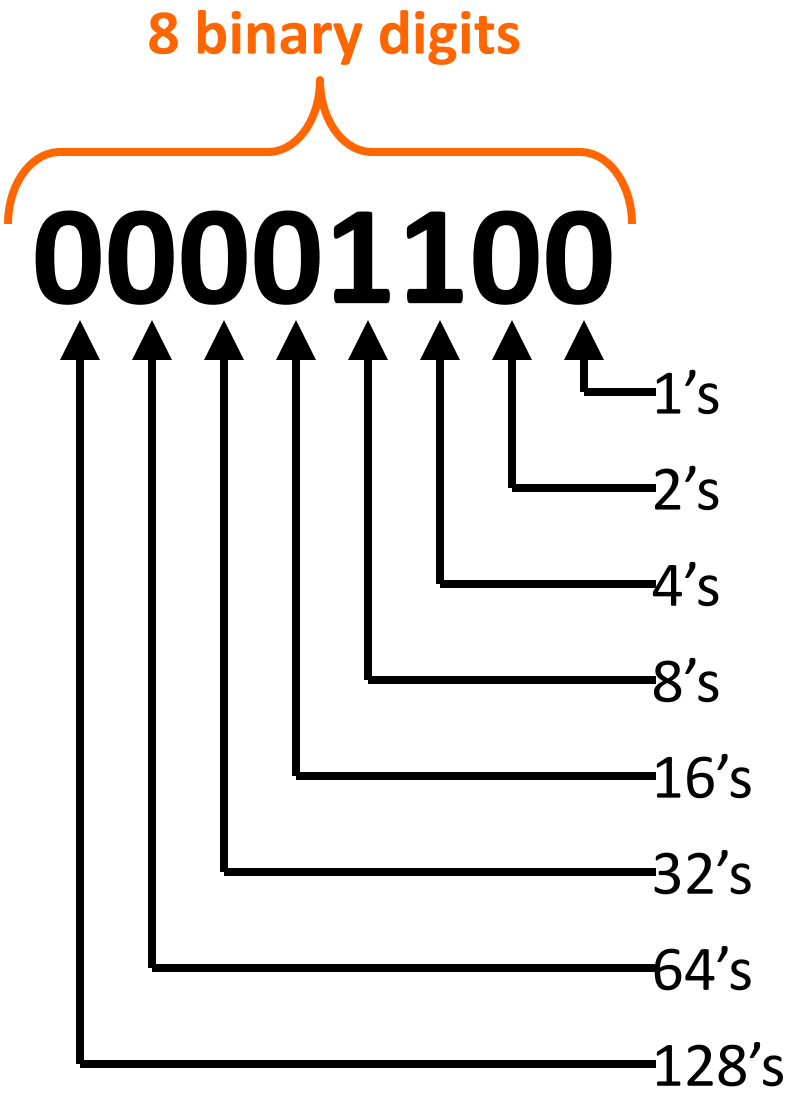
We can read off the number by the places

Let’s count…

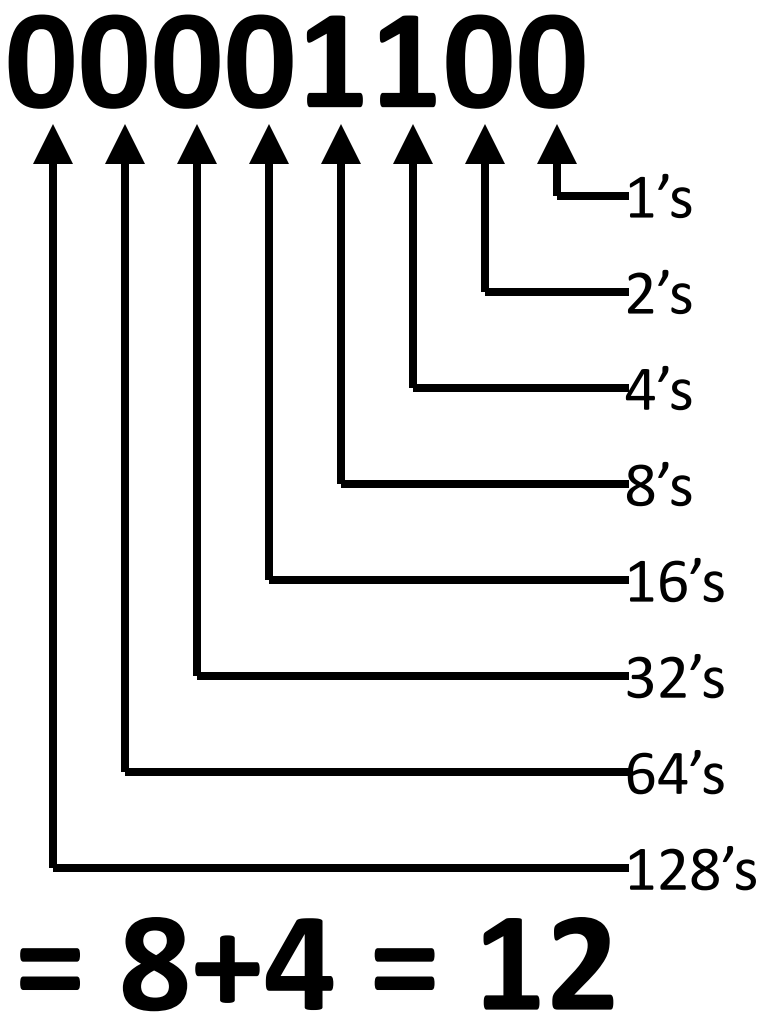


But we need
a lot of places…

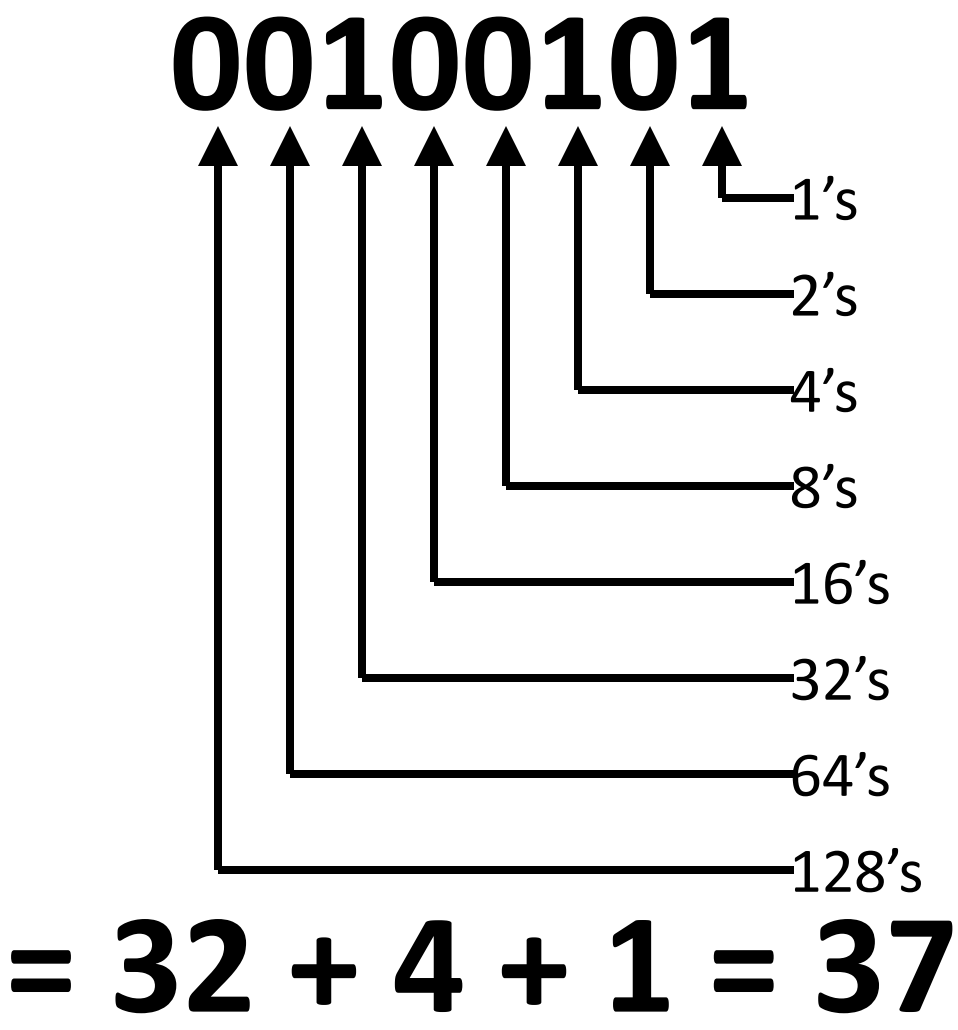
The Byte



How can you read this quickly?

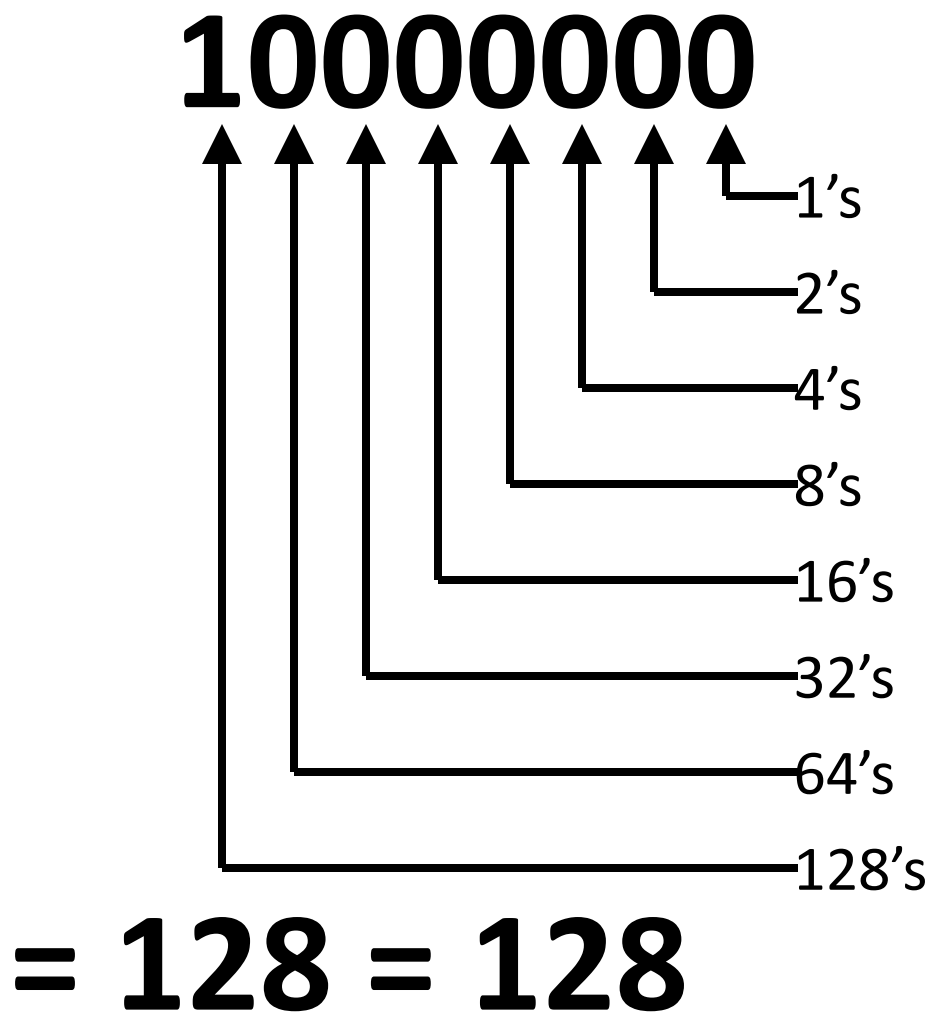


How can you read this quickly?

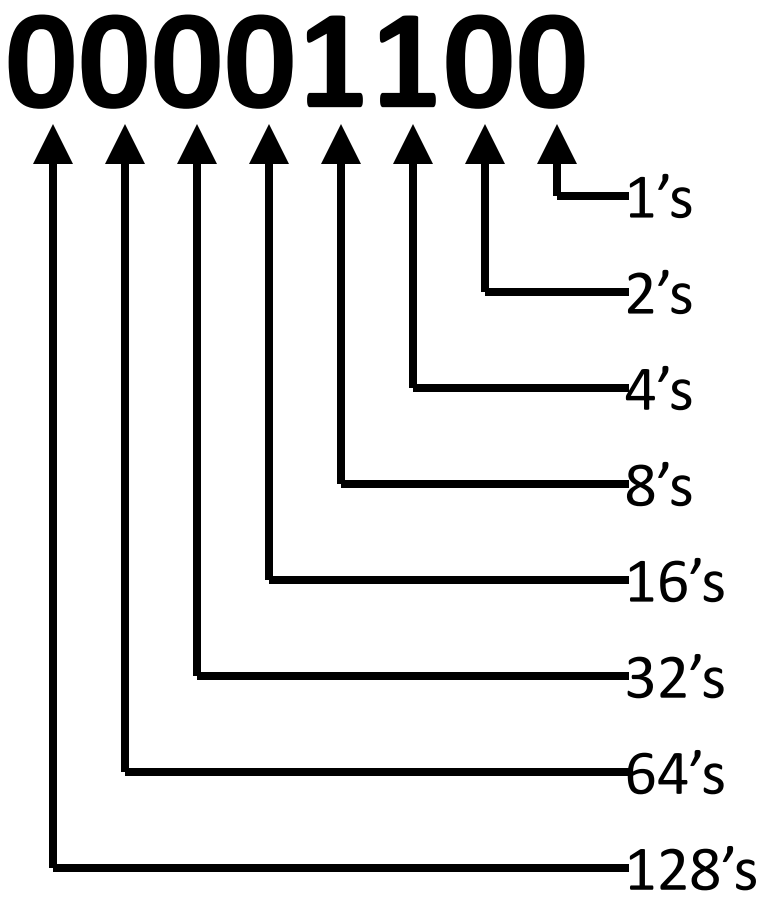


= 32 + 4 + 1 = 37

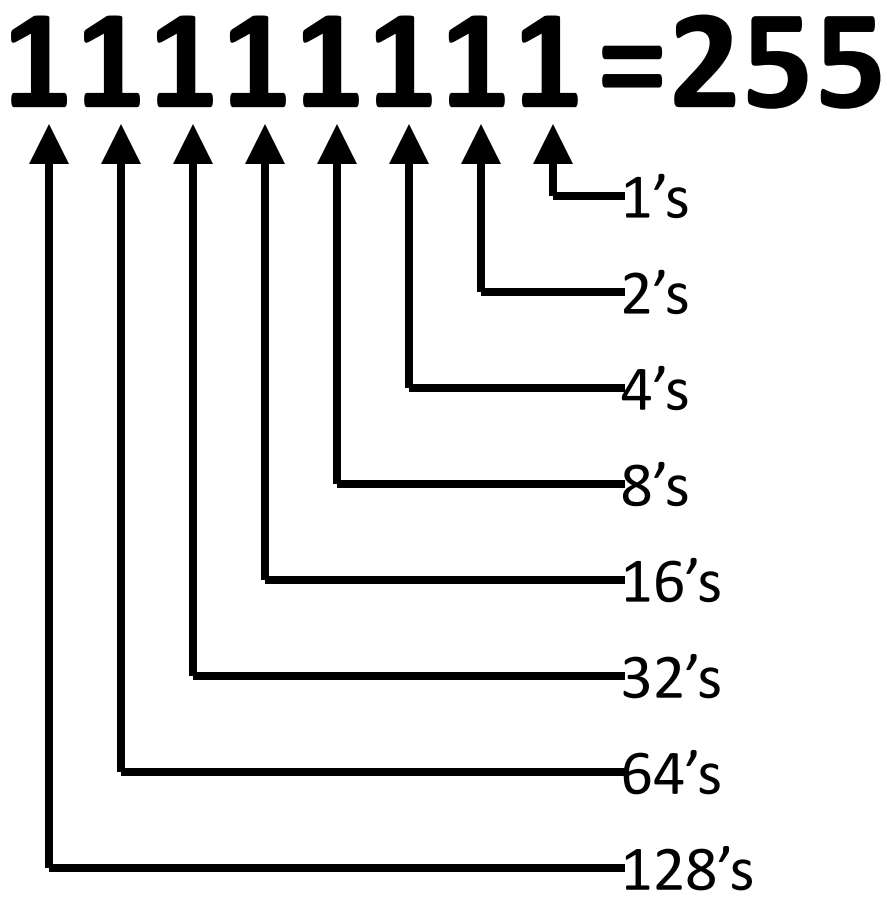
How can you read this quickly?



How Big a Byte Be?



The Biggest, Baddest Byte:



Ok, so you can count, but can you add?

27 = 00011011

+26 = 00011010

[addition on white board]

Ok, fine. You can count and add.

But can you send email?

Huh?

ASCII Table

Easy. Assign a number to each letter.

ASCII value	Character	ASCII value	Character	ASCII value	Character
032	(space)	064	@	096	
033	!	065	A	097	a
034	"	066	B	098	b
035	#	067	C	099	c
036	$	068	D	100	d
037	%	069	E	101	e
038	&	070	F	102	f
039	'	071	G	103	g
040	(	072	H	104	h
041	)	073	I	105	i
042	*	074	J	106	j
043	+	075	K	107	k
044	,	076	L	108	l
045	-	077	M	109	m
046	.	078	N	110	n
047	/	079	O	111	o
048	0	080	P	112	p
049	1	081	Q	113	q
050	2	082	R	114	r
051	3	083	S	115	s
052	4	084	T	116	t
053	5	085	U	117	u
054	6	086	V	118	v
055	7	087	W	119	w
056	8	088	X	120	x
057	9	089	Y	121	y
058	:	090	Z	122	z
059	;	091	[	123	{
060	<	092	\	124	|
061	=	093	]	125	}
062	>	094	^	126	~
063	?	095	_	127	⌂

Let's write some email!

A short email: Only use 4 letters

ASCII value	Character	ASCII value	Character	ASCII value	Character
032	(space)	064	@	096	
033	!	065	A	097	a
034	"	066	B	098	b
035	#	067	C	099	c
036	$	068	D	100	d
037	%	069	E	101	e
038	&	070	F	102	f
039	'	071	G	103	g
040	(	072	H	104	h
041	)	073	I	105	i
042	*	074	J	106	j
043	+	075	K	107	k
044	,	076	L	108	l
045	-	077	M	109	m
046	.	078	N	110	n
047	/	079	O	111	o
048	0	080	P	112	p
049	1	081	Q	113	q
050	2	082	R	114	r
051	3	083	S	115	s
052	4	084	T	116	t
053	5	085	U	117	u
054	6	086	V	118	v
055	7	087	W	119	w
056	8	088	X	120	x
057	9	089	Y	121	y
058	:	090	Z	122	z
059	;	091	[	123	{
060	<	092	\	124	|
061	=	093	]	125	}
062	>	094	^	126	~
063	?	095	_	127	⌂

Let's write some email!

A short email: Only use 4 letters

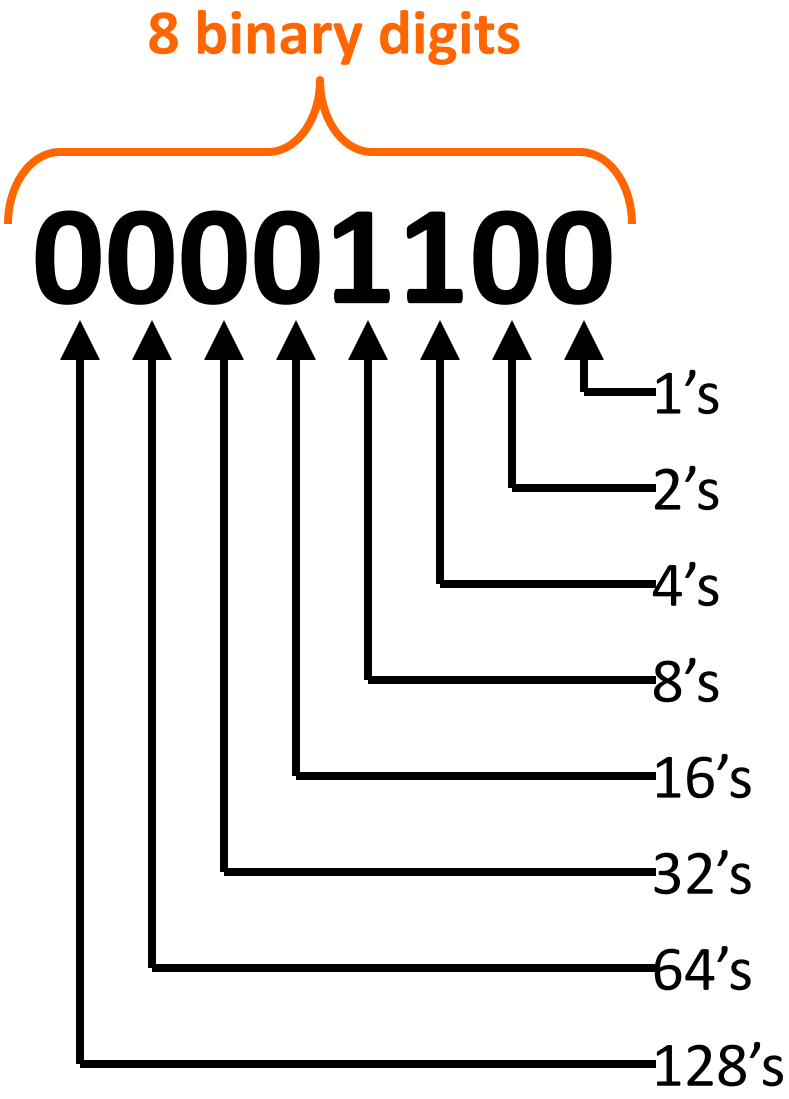
ASCII value	Character	Control character	ASCII value	Character	ASCII value	Character	ASCII value	Character
000	(null)	NUL	032	(space)	064	@	096	
001	☺	SOH	033	!	065	A	097	a
002	☻	STX	034	"	066	B	098	b
003	♥	ETX	035	#	067	C	099	c
004	♦	EOT	036	$	068	D	100	d
005	♣	ENQ	037	%	069	E	101	e
006	♠	ACK	038	&	070	F	102	f
007	(beep)	BEL	039	'	071	G	103	g
008	◘	BS	040	(	072	H	104	h
009	(tab)	HT	041	)	073	I	105	i
010	(line feed)	LF	042	*	074	J	106	j
011	(home)	VT	043	+	075	K	107	k
012	(form feed)	FF	044	'	076	L	108	l
013	(carriage return)	CR	045	-	077	M	109	m
014	♫	SO	046	.	078	N	110	n
015	☼	SI	047	/	079	O	111	o
016	►	DLE	048	0	080	P	112	p
017	◄	DC1	049	1	081	Q	113	q
018	↕	DC2	050	2	082	R	114	r
019	!!	DC3	051	3	083	S	115	s
020	π	DC4	052	4	084	T	116	t
021	§	NAK	053	5	085	U	117	u
022	▬	SYN	054	6	086	V	118	v
023	↨	ETB	055	7	087	W	119	w
024	↑	CAN	056	8	088	X	120	x
025	↓	EM	057	9	089	Y	121	y
026	→	SUB	058	:	090	Z	122	z
027	←	ESC	059	;	091	[	123	{
028	(cursor right)	FS	060	<	092	\	124	|
029	(cursor left)	GS	061	=	093	]	125	}
030	(cursor up)	RS	062	>	094	^	126	~
031	(cursor down)	US	063	?	095	_	127	⌂

Ok, The email's done

Now how do we send it from robot to robot?

Binary Representation

The Byte



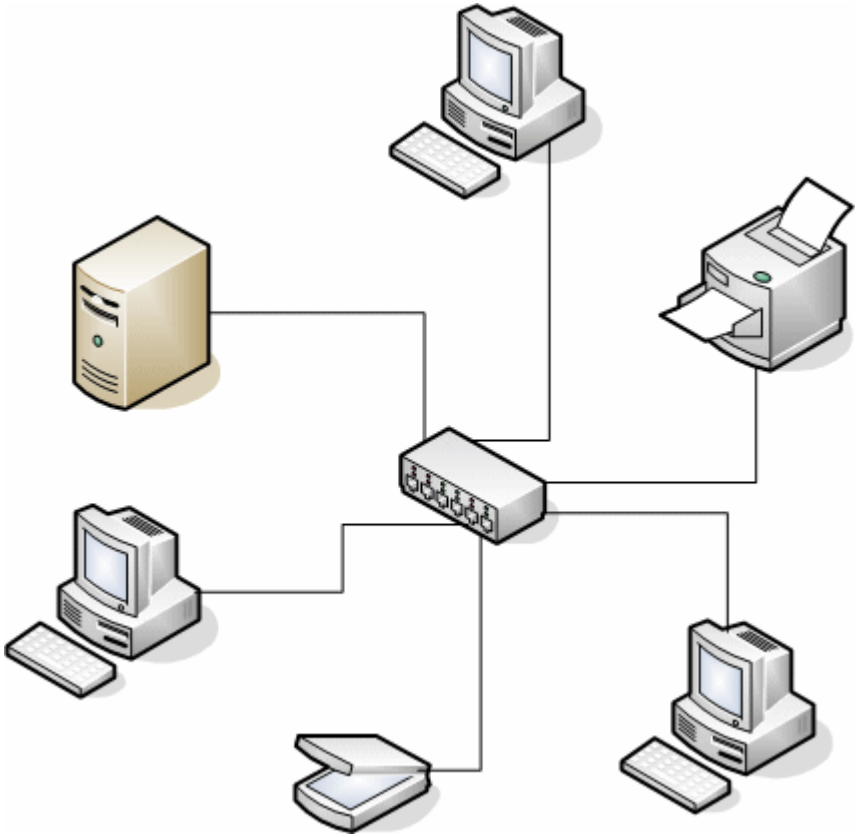
ASCII Table

Easy. Assign a number to each letter.

ASCII value	Character	ASCII value	Character	ASCII value	Character
032	(space)	064	@	096	
033	!	065	A	097	a
034	"	066	B	098	b
035	#	067	C	099	c
036	$	068	D	100	d
037	%	069	E	101	e
038	&	070	F	102	f
039	'	071	G	103	g
040	(	072	H	104	h
041	)	073	I	105	i
042	*	074	J	106	j
043	+	075	K	107	k
044	,	076	L	108	l
045	-	077	M	109	m
046	.	078	N	110	n
047	/	079	O	111	o
048	0	080	P	112	p
049	1	081	Q	113	q
050	2	082	R	114	r
051	3	083	S	115	s
052	4	084	T	116	t
053	5	085	U	117	u
054	6	086	V	118	v
055	7	087	W	119	w
056	8	088	X	120	x
057	9	089	Y	121	y
058	:	090	Z	122	z
059	;	091	[	123	{
060	<	092	\	124	|
061	=	093	]	125	}
062	>	094	^	126	~
063	?	095	_	127	⌂

Digital Communications

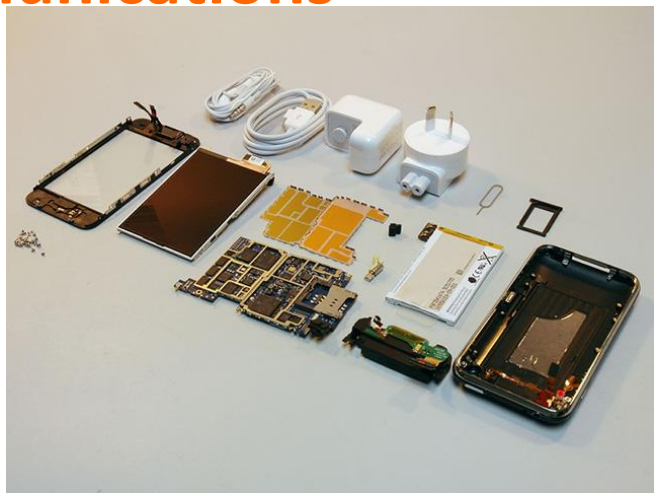
We have digital communication Computers are networked



We're surrounded by Digital Communications

Ethernet

USB



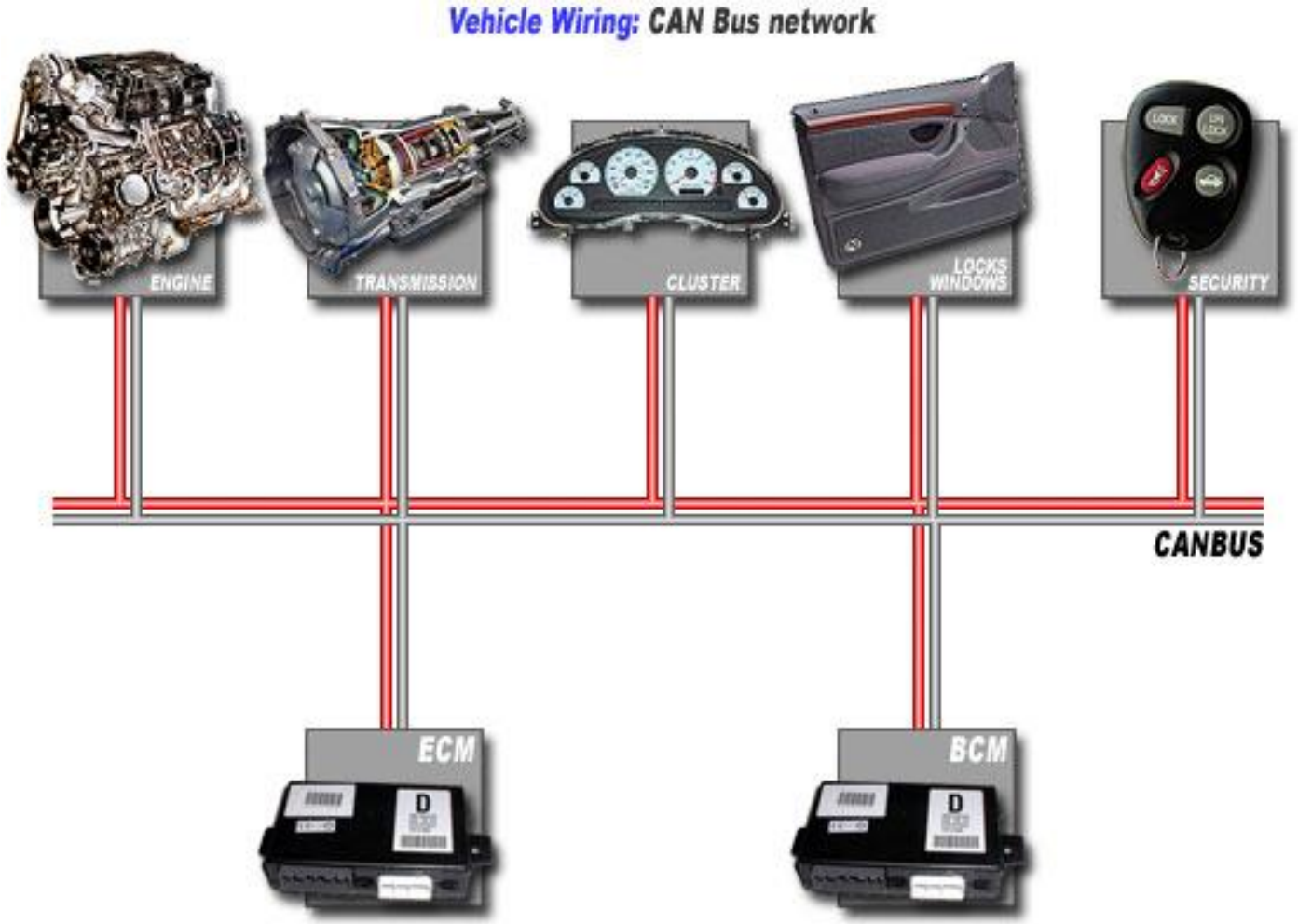
3G and WiFi
(This is an iPhone)

WiFi

Remote Control

r-one robot

CAN Bus


Vehicle Wiring: CAN Bus network
ENGINE
TRANSMISSION
CLUSTER
LOCKS
WINDOWS
SECURITY
CANBUS
ECM
BCM

Physical Representation of Digital Data

Ethernet
Voltage

USB
Voltage

3G and WiFi
radio waves

WiFi
radio waves

Remote Control
IR Light

r-one robot
IR Light and radio

Bandwidth

This is a measure of the amount of bits we can transmit over a *channel*

- Its units are BPS, Bits Per Second

Anybody know some popular bandwidths?

- Ethernet = {10/100/1000} mbps
- Wi-Fi (802.11g) = 54 mbps
- USB = 480 mbps
- r-one radio: 2 mbps, IR: 1.25 kbps

Bandwidth

Ethernet
1000 mbps

USB
480 mbps

3G and WiFi
384 kbps/54 mbps

WiFi
54 mbps

Remote Control
1.25 kbps

r-one robot
2 mbps/1.25 kbps


That’s a lot of bits!

If we're sending so many bits so quickly, how do we tell them apart?

Telling one bit from another

A continuous stream of bits is hard to deal with.

Imagine a computer network: You might have many questions about these bits:

- Are these bits for you?
- Where did these bits come from?
- What do these bits mean?
- Are these bits error-free?

The Packet

A packet is a chunk of data with a well-defined beginning, end, and structure

A packet has four parts:

- Some kind of **start indication** that tells the network that a packet is starting
- Some kind of **header** that tells the network what the packet is, where it is from, and where it is going
- Some kind of **data**. That's kind of the point of the packet…
- Some kind of **error detection** to check the validity of the packet.