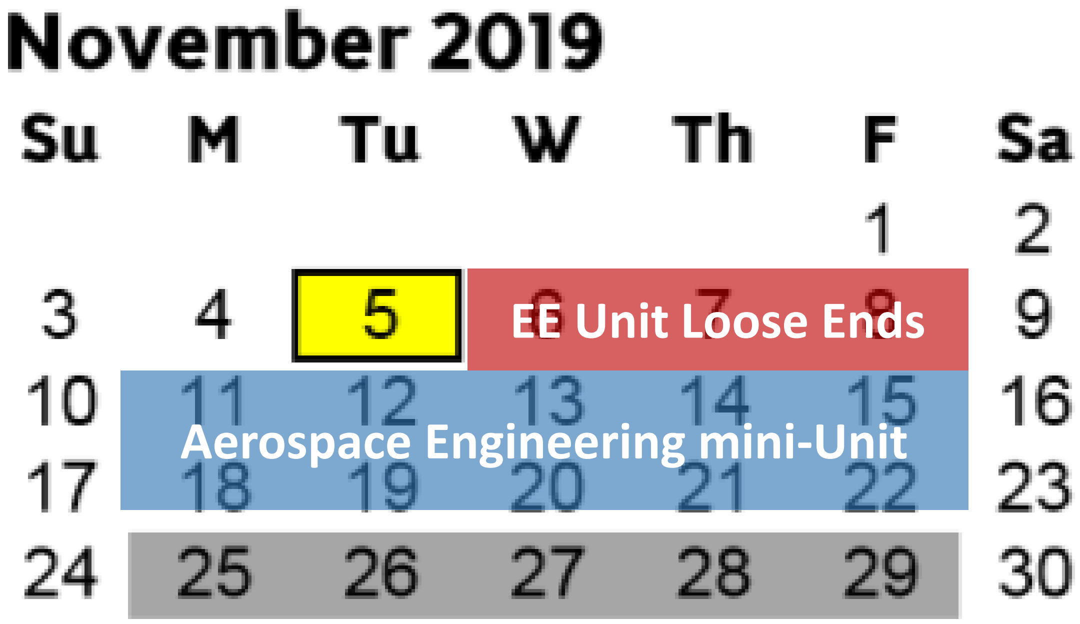
Binary Counting & ASCII Values

Wheeler HS Fall 19

Finishing the Semester

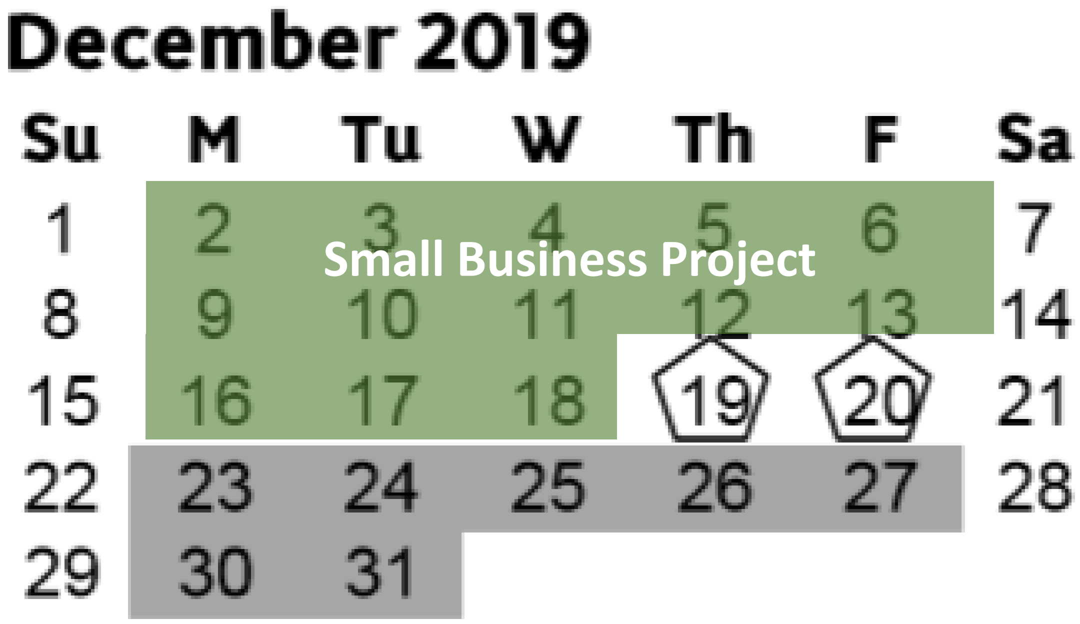


November 2019

Su	M	Tu	W	Th	F	Sa
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

EE Unit Loose Ends

Aerospace Engineering mini-Unit



December 2019

Su	M	Tu	W	Th	F	Sa
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

Small Business Project

Finishing Electrical Engineering Loose Ends

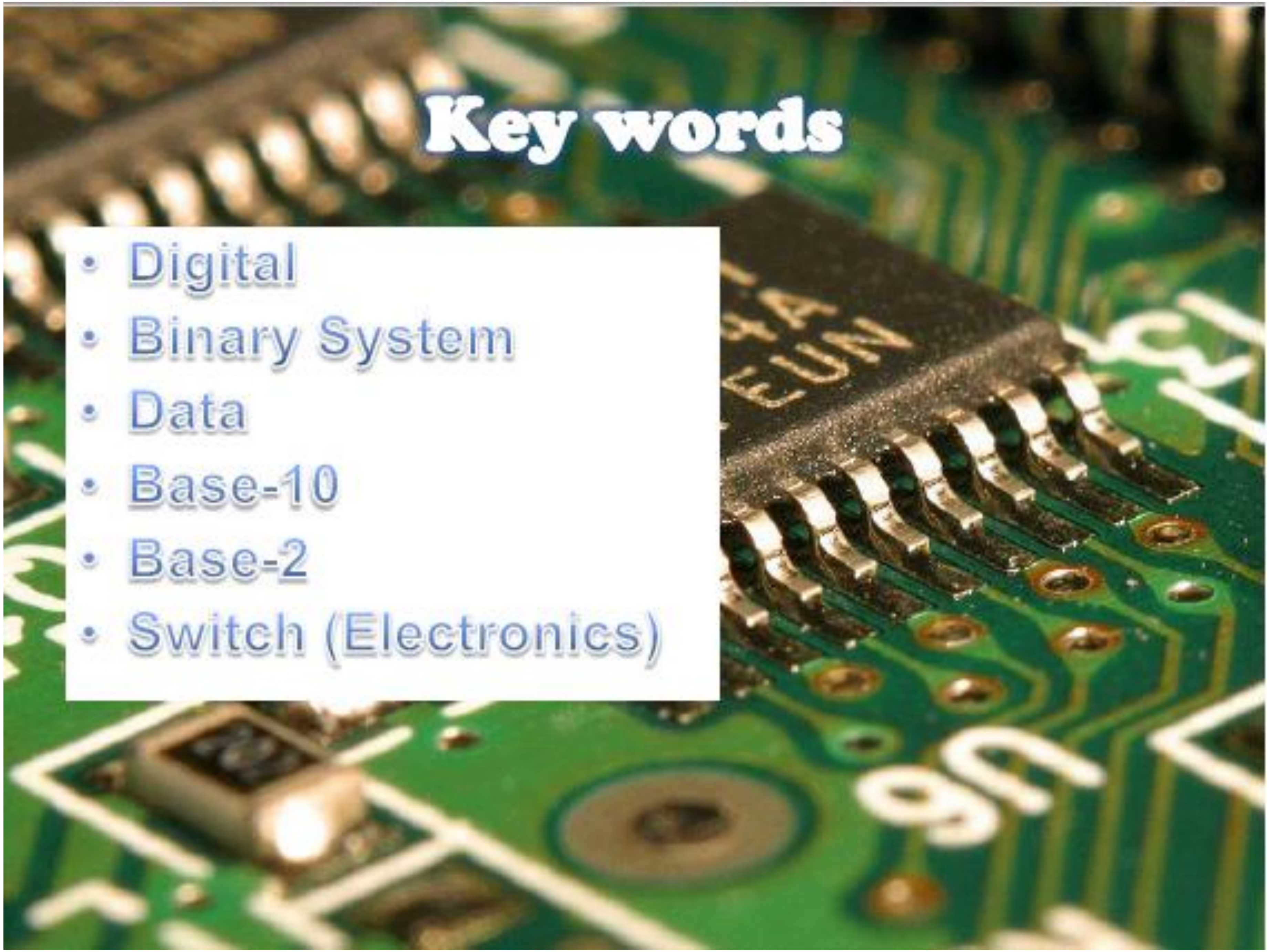
1. Binary and Ascii
    - Notes & worksheet
2. Arduino Project “Presentations”
3. ADC & PWM Challenges

The Binary System

By Desiree Noland

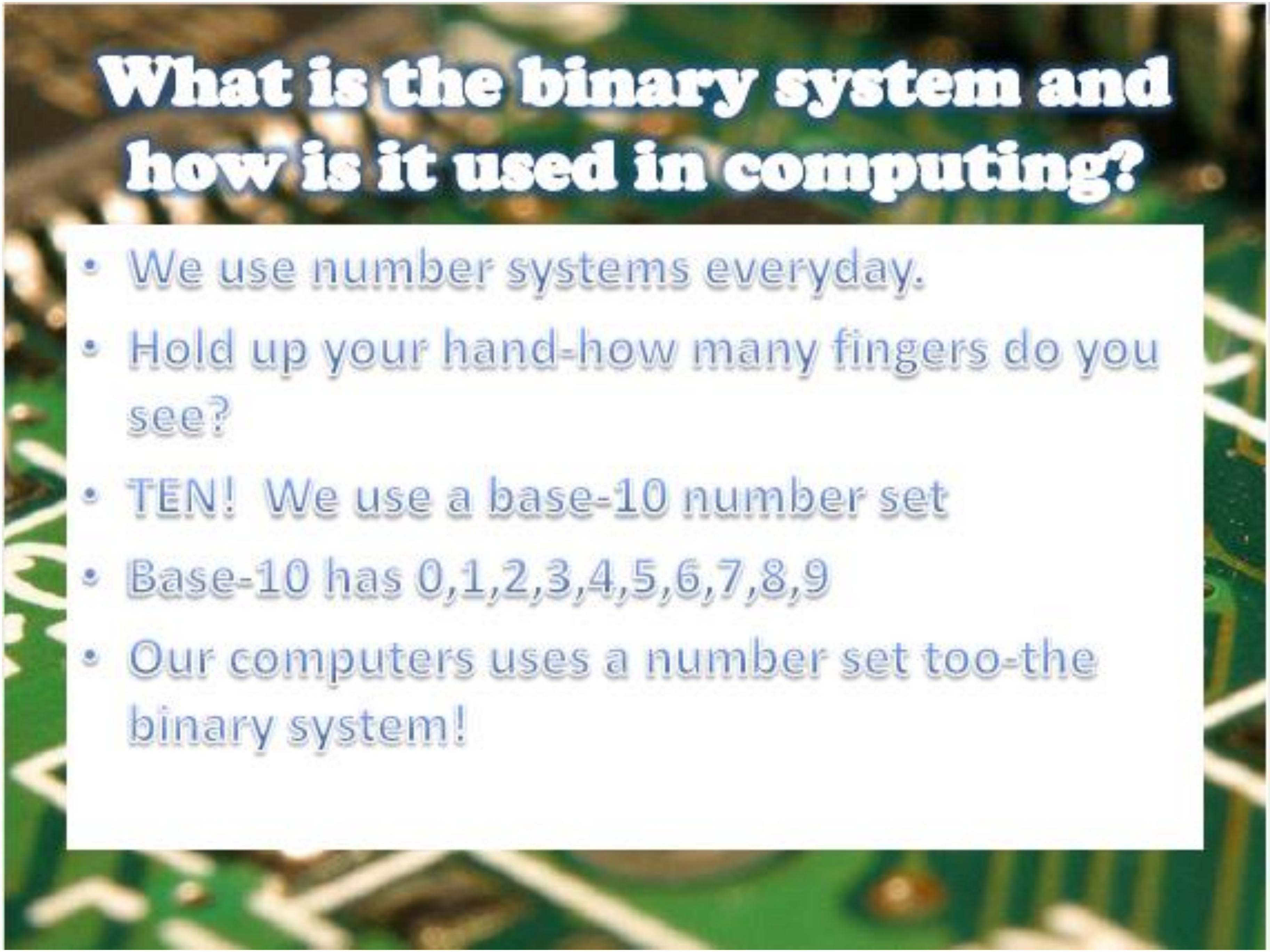
Just how does that computer work???

Warmup: Asides from “ten” how can you symbolically show that there are 10 of an object?



Key words

- Digital
- Binary System
- Data
- Base-10
- Base-2
- Switch (Electronics)



What is the binary system and how is it used in computing?

- We use number systems everyday.
- Hold up your hand-how many fingers do you see?
- TEN! We use a base-10 number set
- Base-10 has 0,1,2,3,4,5,6,7,8,9
- Our computers uses a number set too-the binary system!

Electronics-How do they work?

On

Off

Computers and circuits are in 2 states:

- On
- Off

- This is encoded by the Binary System! The Binary System tells computers and circuits which wires need to be on and which need to be off.

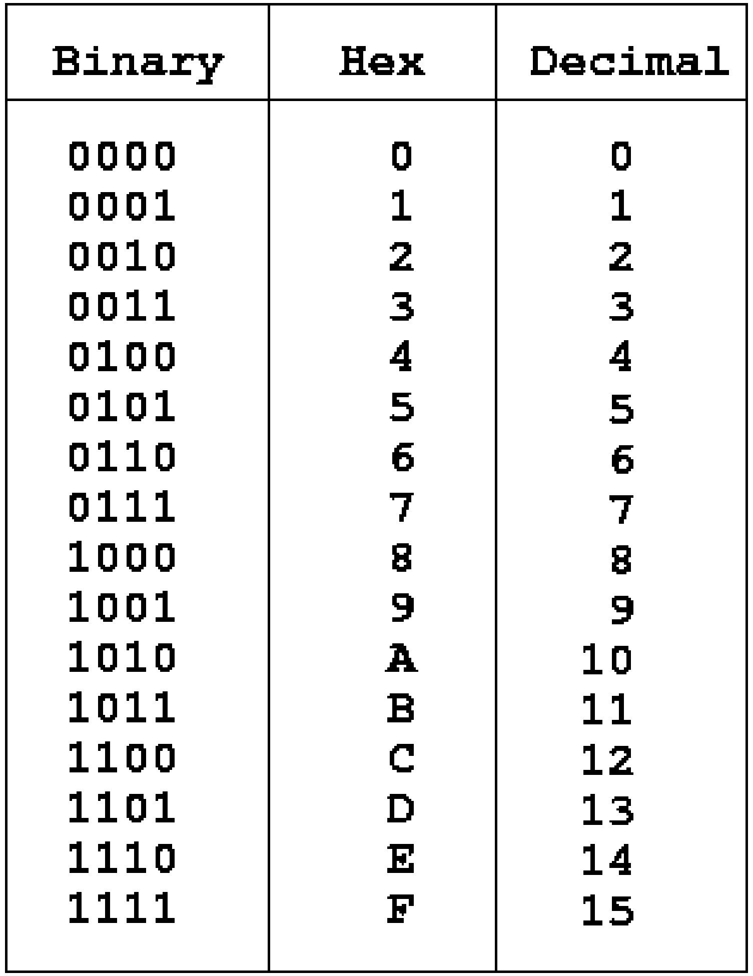
But how does it work???

- Base-10 or the decimal system
  - 0,1,2,3,4,5,6,7,8,9
- Base-2 or Binary system:
  - 0,1
- 0=Off and 1=On

Video: Counting in Binary

- https://www.youtube.com/watch?v=zELAfmp3fXY

Decimal (Base 10) vs Binary (Base 2)



Binary	Hex	Decimal
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	A	10
1011	B	11
1100	C	12
1101	D	13
1110	E	14
1111	F	15

Converting from:
Binary (base 2) to Decimal (base 10)

1	0	1	0	0	1	1	1
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1
$1 \cdot 128 +$	$0 \cdot 64 \; +$	$1 \cdot 32 +$	$0 \cdot 16 +$	$0 \cdot 8 +$	$1 \cdot 4 +$	$1 \cdot 2 +$	$1 \cdot 1$

$$= 128 + 32 + 4 + 2 + 1$$
$$= 167$$

We found that:

$10100111_2 = 167_{10}$

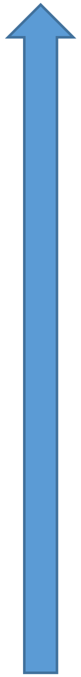
Practice Converting from:
Binary (base 2) to Decimal (base 10)

___	___	___	___	___	___	___	___
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1

$$\_\_ \cdot 128 + \_\_ \cdot 64 \; + \_\_ \cdot 32 + \_\_ \cdot 16 + \_\_ \cdot 8 + \_\_ \cdot 4 + \_\_ \cdot 2 + \_\_ \cdot 1$$

Converting from:
Decimal (base 10) to Binary (base 2)

$167 \div 2$	$= 83$	remainder = 1
$83 \div 2$	$= 41$	remainder = 1
$41 \div 2$	$= 20$	remainder = 1
$20 \div 2$	$= 10$	remainder = 0
$10 \div 2$	$= 5$	remainder = 0
$5 \div 2$	$= 2$	remainder = 1
$2 \div 2$	$= 1$	remainder = 0
$1 \div 2$	$= 0$	remainder = 1



We found that:

$167_{10} = 10100111_2$

Practice Converting from:
Decimal (base 10) to Binary (base 2)

___ ÷ 2 = _____ remainder =

___ ÷ 2 = _____ remainder =

___ ÷ 2 = _____ remainder =

___ ÷ 2 = _____ remainder =

___ ÷ 2 = _____ remainder =

___ ÷ 2 = _____ remainder =

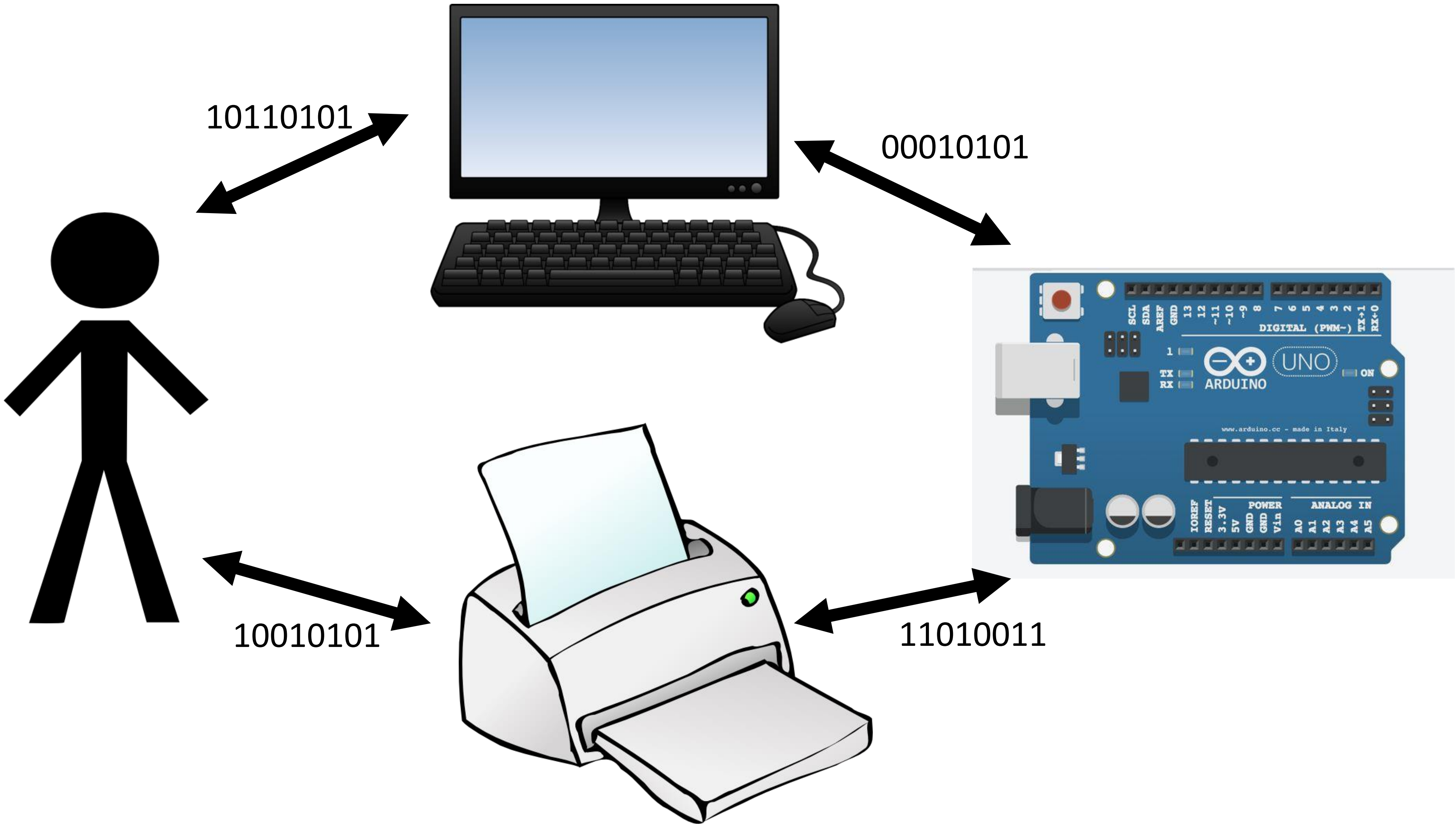
___ ÷ 2 = _____ remainder =

___ ÷ 2 = _____ remainder =

Another Way to Convert between binary and decimal

- Google it! (type “convert from ____ to ____”)
- https://www.binaryhexconverter.com/binary-to-decimal-converter

The point of binary: Communication



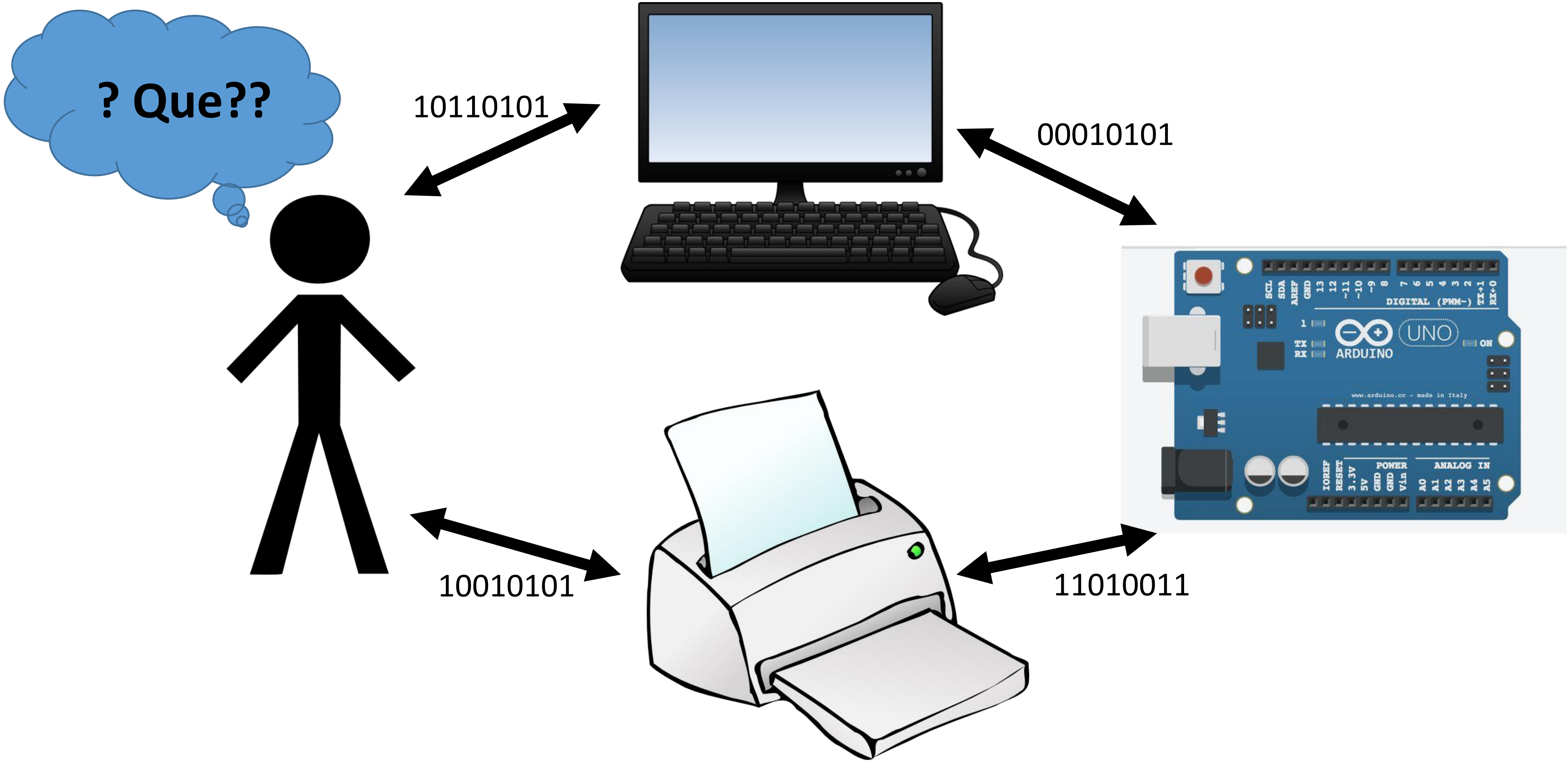
10110101

00010101

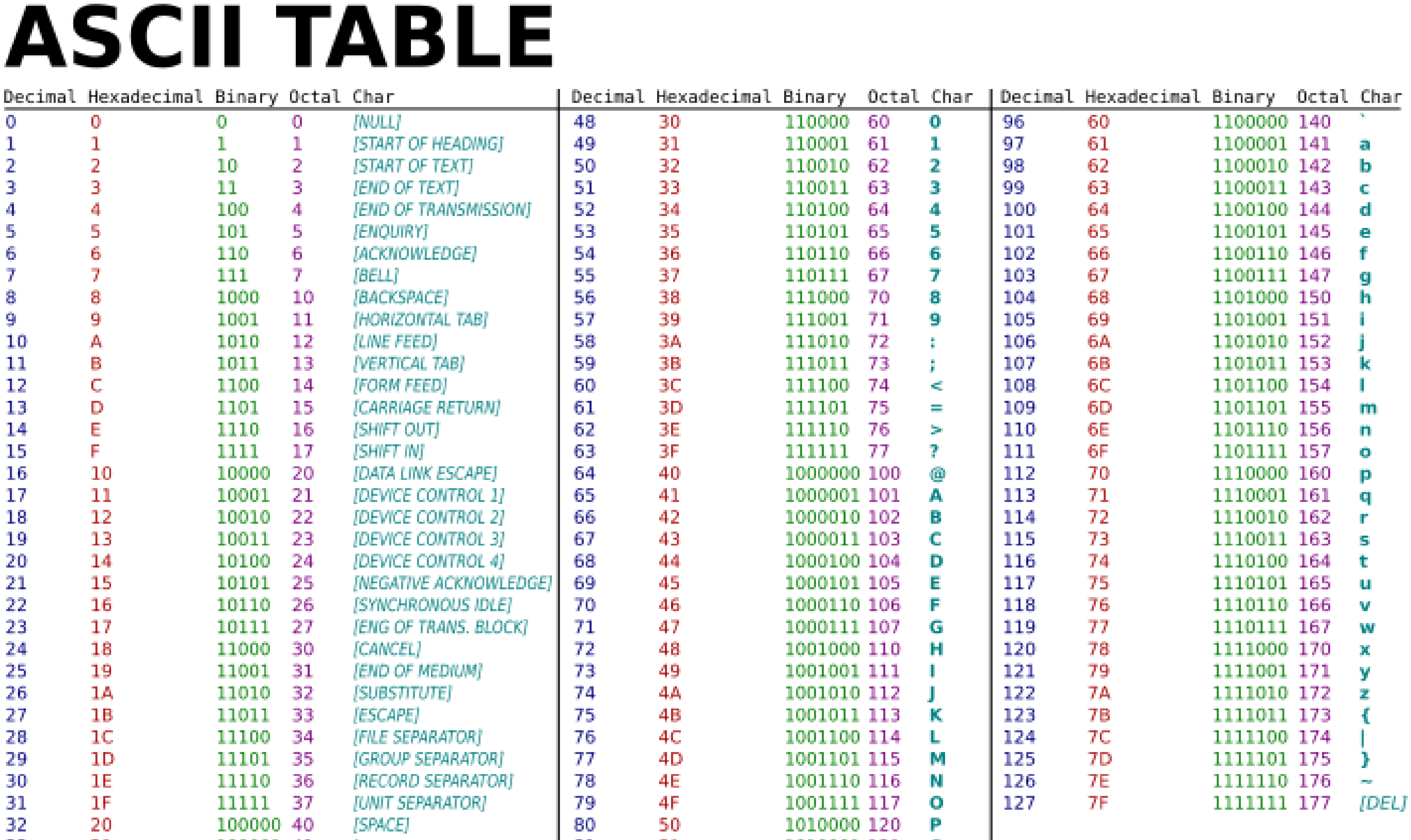
10010101

11010011

Ascii: To help those of us who aren’t fluent in binary



ASCII Characters – A way to numerically represent letters



ASCII TABLE

Decimal	Hexadecimal	Binary	Octal	Char	Decimal	Hexadecimal	Binary	Octal	Char	Decimal	Hexadecimal	Binary	Octal	Char
0	0	0	0	[NULL]	48	30	110000	60	0	96	60	1100000	140	`
1	1	1	1	[START OF HEADING]	49	31	110001	61	1	97	61	1100001	141	a
2	2	10	2	[START OF TEXT]	50	32	110010	62	2	98	62	1100010	142	b
3	3	11	3	[END OF TEXT]	51	33	110011	63	3	99	63	1100011	143	c
4	4	100	4	[END OF TRANSMISSION]	52	34	110100	64	4	100	64	1100100	144	d
5	5	101	5	[ENQUIRY]	53	35	110101	65	5	101	65	1100101	145	e
6	6	110	6	[ACKNOWLEDGE]	54	36	110110	66	6	102	66	1100110	146	f
7	7	111	7	[BELL]	55	37	110111	67	7	103	67	1100111	147	g
8	8	1000	10	[BACKSPACE]	56	38	111000	70	8	104	68	1101000	150	h
9	9	1001	11	[HORIZONTAL TAB]	57	39	111001	71	9	105	69	1101001	151	i
10	A	1010	12	[LINE FEED]	58	3A	111010	72	:	106	6A	1101010	152	j
11	B	1011	13	[VERTICAL TAB]	59	3B	111011	73	;	107	6B	1101011	153	k
12	C	1100	14	[FORM FEED]	60	3C	111100	74	<	108	6C	1101100	154	l
13	D	1101	15	[CARRIAGE RETURN]	61	3D	111101	75	=	109	6D	1101101	155	m
14	E	1110	16	[SHIFT OUT]	62	3E	111110	76	>	110	6E	1101110	156	n
15	F	1111	17	[SHIFT IN]	63	3F	111111	77	?	111	6F	1101111	157	o
16	10	10000	20	[DATA LINK ESCAPE]	64	40	1000000	100	@	112	70	1110000	160	p
17	11	10001	21	[DEVICE CONTROL 1]	65	41	1000001	101	A	113	71	1110001	161	q
18	12	10010	22	[DEVICE CONTROL 2]	66	42	1000010	102	B	114	72	1110010	162	r
19	13	10011	23	[DEVICE CONTROL 3]	67	43	1000011	103	C	115	73	1110011	163	s
20	14	10100	24	[DEVICE CONTROL 4]	68	44	1000100	104	D	116	74	1110100	164	t
21	15	10101	25	[NEGATIVE ACKNOWLEDGE]	69	45	1000101	105	E	117	75	1110101	165	u
22	16	10110	26	[SYNCHRONOUS IDLE]	70	46	1000110	106	F	118	76	1110110	166	v
23	17	10111	27	[ENG OF TRANS. BLOCK]	71	47	1000111	107	G	119	77	1110111	167	w
24	18	11000	30	[CANCEL]	72	48	1001000	110	H	120	78	1111000	170	x
25	19	11001	31	[END OF MEDIUM]	73	49	1001001	111	I	121	79	1111001	171	y
26	1A	11010	32	[SUBSTITUTE]	74	4A	1001010	112	J	122	7A	1111010	172	z
27	1B	11011	33	[ESCAPE]	75	4B	1001011	113	K	123	7B	1111011	173	{
28	1C	11100	34	[FILE SEPARATOR]	76	4C	1001100	114	L	124	7C	1111100	174	|
29	1D	11101	35	[GROUP SEPARATOR]	77	4D	1001101	115	M	125	7D	1111101	175	}
30	1E	11110	36	[RECORD SEPARATOR]	78	4E	1001110	116	N	126	7E	1111110	176	~
31	1F	11111	37	[UNIT SEPARATOR]	79	4F	1001111	117	O	127	7F	1111111	177	[DEL]
32	20	100000	40	[SPACE]	80	50	1010000	120	P					

My Name in binary

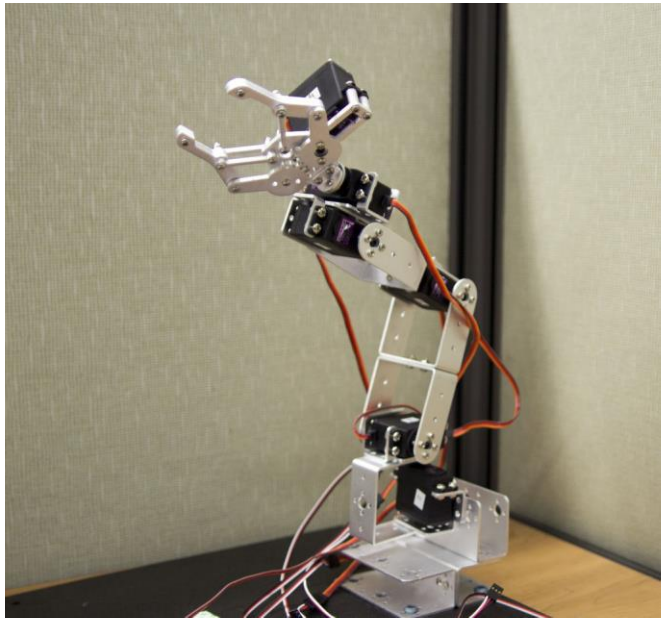
Character		Decimal		Binary
M	->	77	->	01001101
a	->	97	->	01100001
r	->	114	->	01110010
s	->	115	->	01110011
[space]	->	32	->	00100000
B	->	66	->	01000010
e	->	101	->	01100101
r	->	114	->	01110010

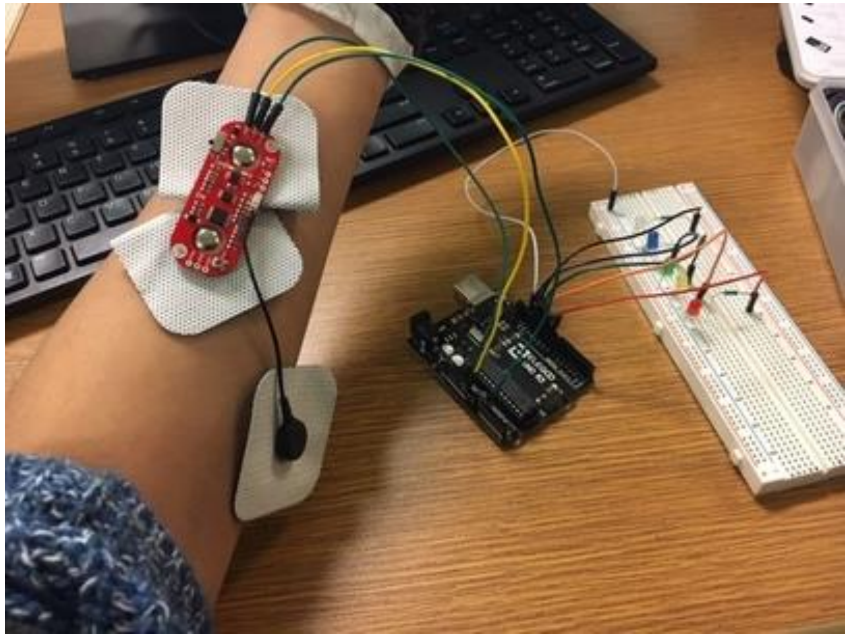
My Name written in Binary

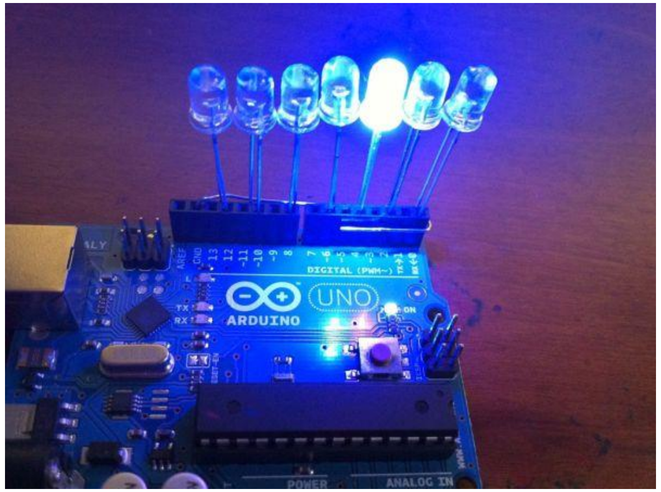
01001101, 01100001, 01110010, 01110011, 00100000, 01000010, 01100101, 01110010, 01110111, 01100001, 01101110, 01100111, 01100101, 01110010

Binary Applications on Arduino

- Functions Arduino uses to control other devices (~DigitalRead and DigitalWrite)
  - AnalogWrite()
    - Using PWM (Pulse Width Modulation)
  - AnalogRead()
    - Using ADC (Analog to Digital Conversion)







Your Task for Today

1. Complete Binary/Ascii Worksheet
2. Give Arduino Project Presentations and make sure documentation turned in
3. Get ahead: ADC/PWM challenges
   - We’ll discuss this more in depth tomorrow but I have the notes on my blog already and some of you already are familiar with this
4. Historical Technology Project (we will have a laptop cart tomorrow)

Closing

- What is the point of binary?

- Another counting system is Hexadecimal (base 16 as opposed to binary base 2 or decimal base 10). What do you think is the advantage of Hexadecimal

- Why do we regularly use a base 10 counting system as opposed to base 2 or base 16 or another base?