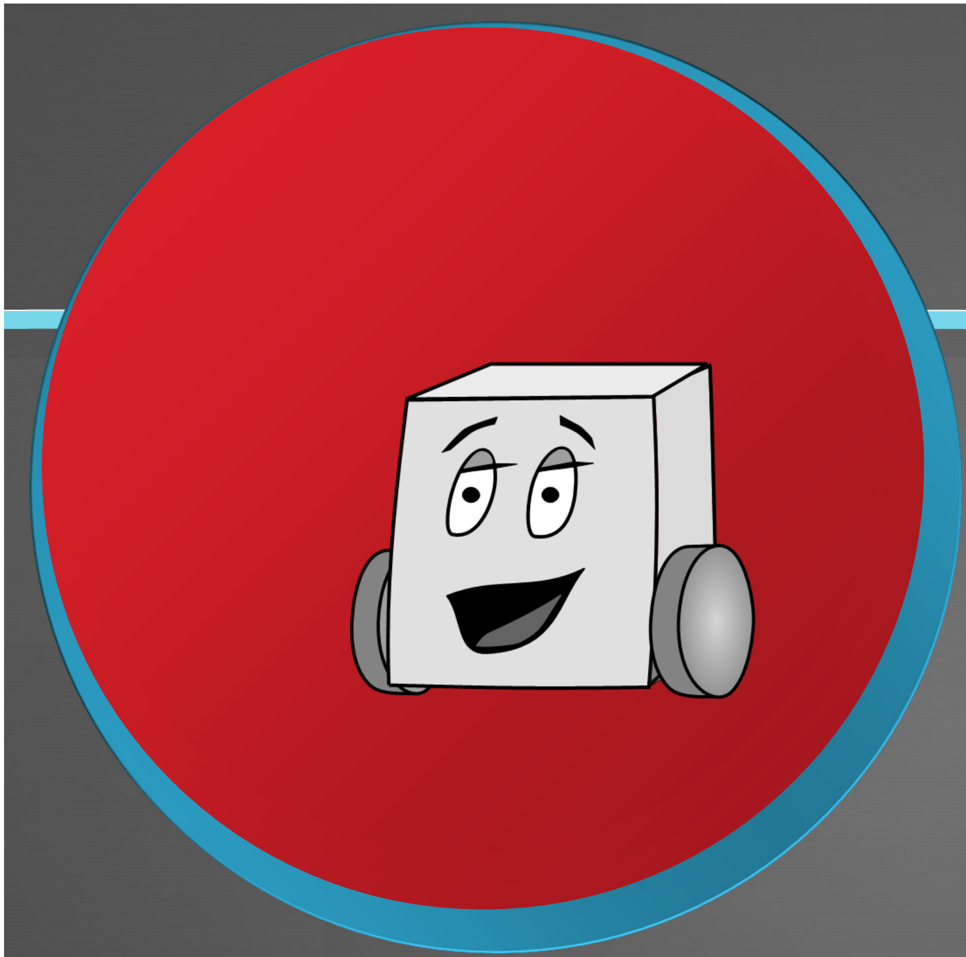


E11 Lecture 4: More C!!!

Profs. David Money Harris & Sarah Harris

Fall 2011

Outline

- Logistics
- Serial Input
- Physical Inputs/Outputs
- Randomness
- Operators
- Control Statements

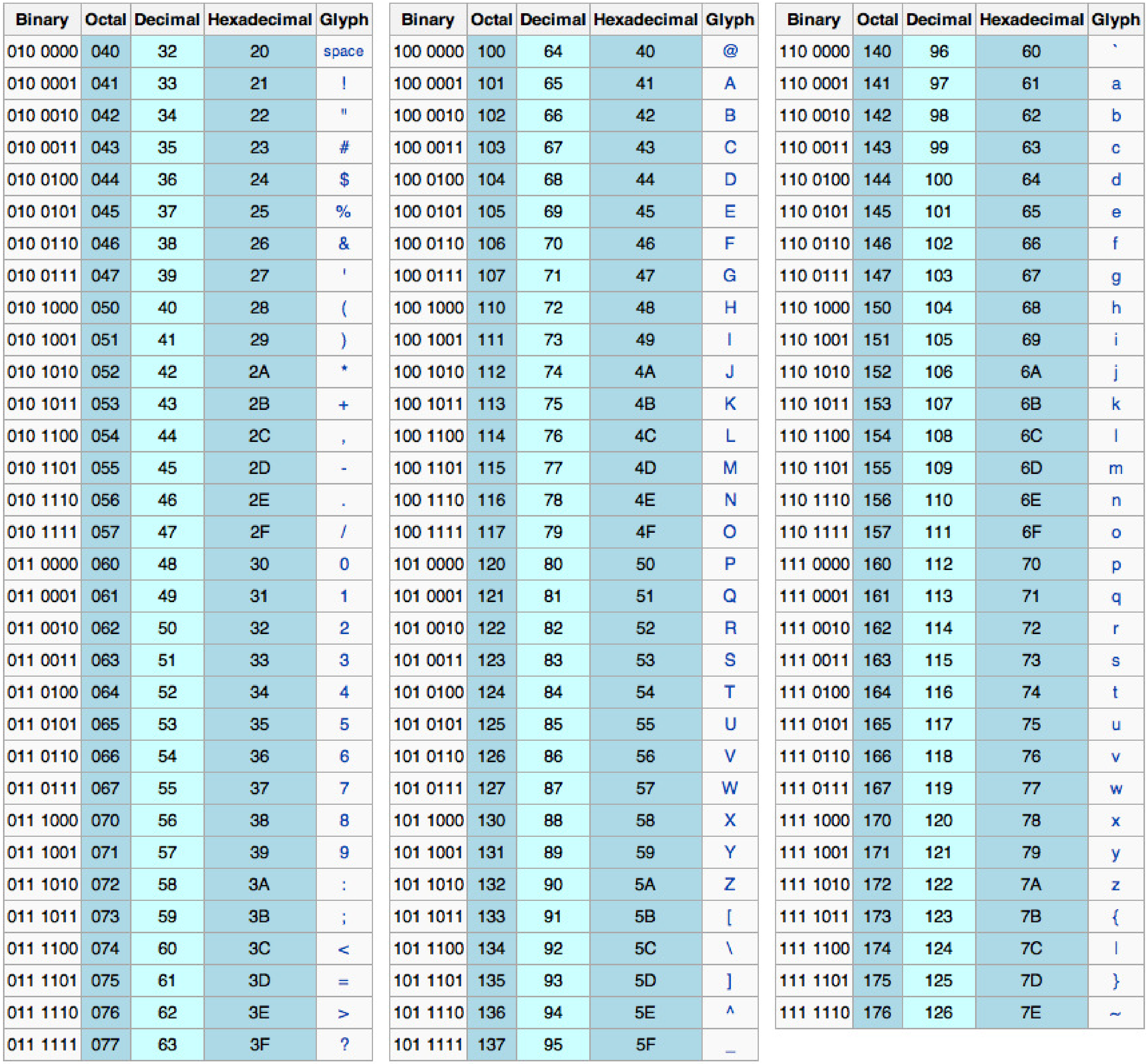
Logistics

- Logistics
  - Tutoring hours: LAC, Saturday, 1-5pm
  - Tutors: Josh Vasquez, Jeremy Usatine
  - Pick up your FTDI cable during tutoring hours
- Running Arduino Software
  - Choose "Arduino Uno" or "Arduino Duemilanove…" under Tools -> Board
- Shop Safety Quiz
  - Congratulations – everyone passed!

Serial Input

```
void setup()
{
  Serial.begin(9600); // opens serial port at 9600 baud
  Serial.println(“Press a key: “);
}

void loop() {
  int pressed = 0; // incoming serial data

  // read user input
  if (Serial.available()) {
    pressed = Serial.read();

    // print result:
    Serial.print("I received: ");
    Serial.println(pressed, BYTE);
  }
}
```

ASCII



Binary	Octal	Decimal	Hexadecimal	Glyph
010 0000	040	32	20	space
010 0001	041	33	21	!
010 0010	042	34	22	"
010 0011	043	35	23	#
010 0100	044	36	24	$
010 0101	045	37	25	%
010 0110	046	38	26	&
010 0111	047	39	27	'
010 1000	050	40	28	(
010 1001	051	41	29	)
010 1010	052	42	2A	*
010 1011	053	43	2B	+
010 1100	054	44	2C	,
010 1101	055	45	2D	-
010 1110	056	46	2E	.
010 1111	057	47	2F	/
011 0000	060	48	30	0
011 0001	061	49	31	1
011 0010	062	50	32	2
011 0011	063	51	33	3
011 0100	064	52	34	4
011 0101	065	53	35	5
011 0110	066	54	36	6
011 0111	067	55	37	7
011 1000	070	56	38	8
011 1001	071	57	39	9
011 1010	072	58	3A	:
011 1011	073	59	3B	;
011 1100	074	60	3C	<
011 1101	075	61	3D	=
011 1110	076	62	3E	>
011 1111	077	63	3F	?
100 0000	100	64	40	@
100 0001	101	65	41	A
100 0010	102	66	42	B
100 0011	103	67	43	C
100 0100	104	68	44	D
100 0101	105	69	45	E
100 0110	106	70	46	F
100 0111	107	71	47	G
100 1000	110	72	48	H
100 1001	111	73	49	I
100 1010	112	74	4A	J
100 1011	113	75	4B	K
100 1100	114	76	4C	L
100 1101	115	77	4D	M
100 1110	116	78	4E	N
100 1111	117	79	4F	O
101 0000	120	80	50	P
101 0001	121	81	51	Q
101 0010	122	82	52	R
101 0011	123	83	53	S
101 0100	124	84	54	T
101 0101	125	85	55	U
101 0110	126	86	56	V
101 0111	127	87	57	W
101 1000	130	88	58	X
101 1001	131	89	59	Y
101 1010	132	90	5A	Z
101 1011	133	91	5B	[
101 1100	134	92	5C	\
101 1101	135	93	5D	]
101 1110	136	94	5E	^
101 1111	137	95	5F	_
110 0000	140	96	60	`
110 0001	141	97	61	a
110 0010	142	98	62	b
110 0011	143	99	63	c
110 0100	144	100	64	d
110 0101	145	101	65	e
110 0110	146	102	66	f
110 0111	147	103	67	g
110 1000	150	104	68	h
110 1001	151	105	69	i
110 1010	152	106	6A	j
110 1011	153	107	6B	k
110 1100	154	108	6C	l
110 1101	155	109	6D	m
110 1110	156	110	6E	n
110 1111	157	111	6F	o
111 0000	160	112	70	p
111 0001	161	113	71	q
111 0010	162	114	72	r
111 0011	163	115	73	s
111 0100	164	116	74	t
111 0101	165	117	75	u
111 0110	166	118	76	v
111 0111	167	119	77	w
111 1000	170	120	78	x
111 1001	171	121	79	y
111 1010	172	122	7A	z
111 1011	173	123	7B	{
111 1100	174	124	7C	|
111 1101	175	125	7D	}
111 1110	176	126	7E	~

Serial Input

```
void setup()
{
  Serial.begin(9600); // opens serial port at 9600 baud
  Serial.println(“Press a key: “);
}

void loop() {
  int pressed = 0; // incoming serial data

  // read user input
  if (Serial.available()) {
    pressed = Serial.read();

    // print result:
    Serial.print("I received: ");
    Serial.println(pressed, DEC);
  }
}
```

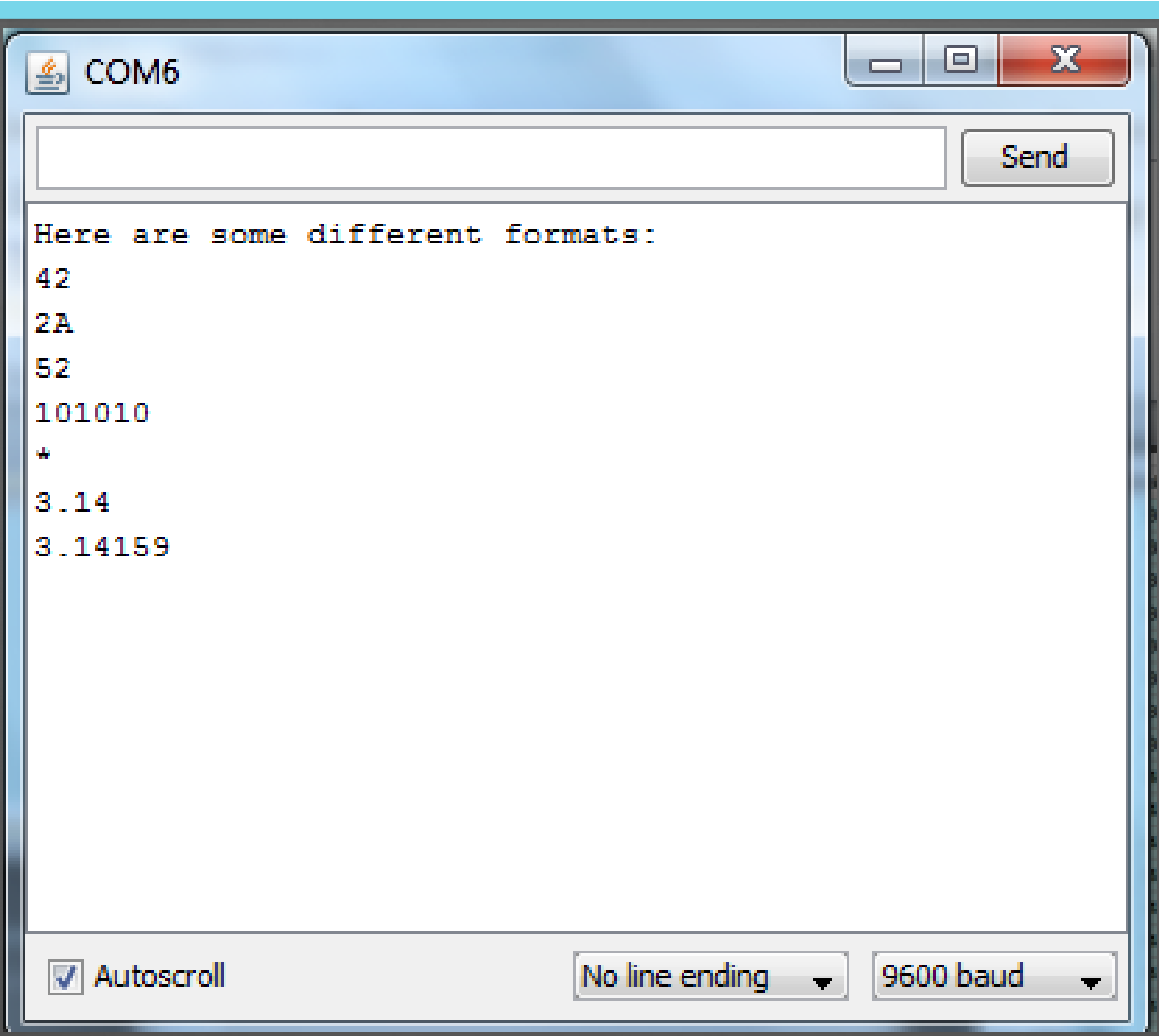
Print Formats

- Print formats
  - Serial.print(*val*, *format*)
    Serial.println(*val*, *format*)
  - *val* is value to print (any data type)
  - *format* is:
    - DEC (decimal)
    - HEX (hexadecimal)
    - OCT (octal)
    - BIN (binary)
    - BYTE (ASCII-interpreted byte)
    - or number of decimal places (for floating point)

Print Format Example

```
void setup()
{
  int x = 42;
  float pi = 3.14159;

  Serial.begin(9600); // opens serial port at 9600 baud
  Serial.println("Here are some different formats: ");

  Serial.println(x, DEC);  // integral formats
  Serial.println(x, HEX);
  Serial.println(x, OCT);
  Serial.println(x, BIN);
  Serial.println(x, BYTE);

  Serial.println(pi, 2);  v// floating point formats
  Serial.println(pi, 5);
}

void loop() {
}
```

Print Format Output



```
Here are some different formats:
42
2A
52
101010
*
3.14
3.14159
```

☑ Autoscroll

No line ending

9600 baud

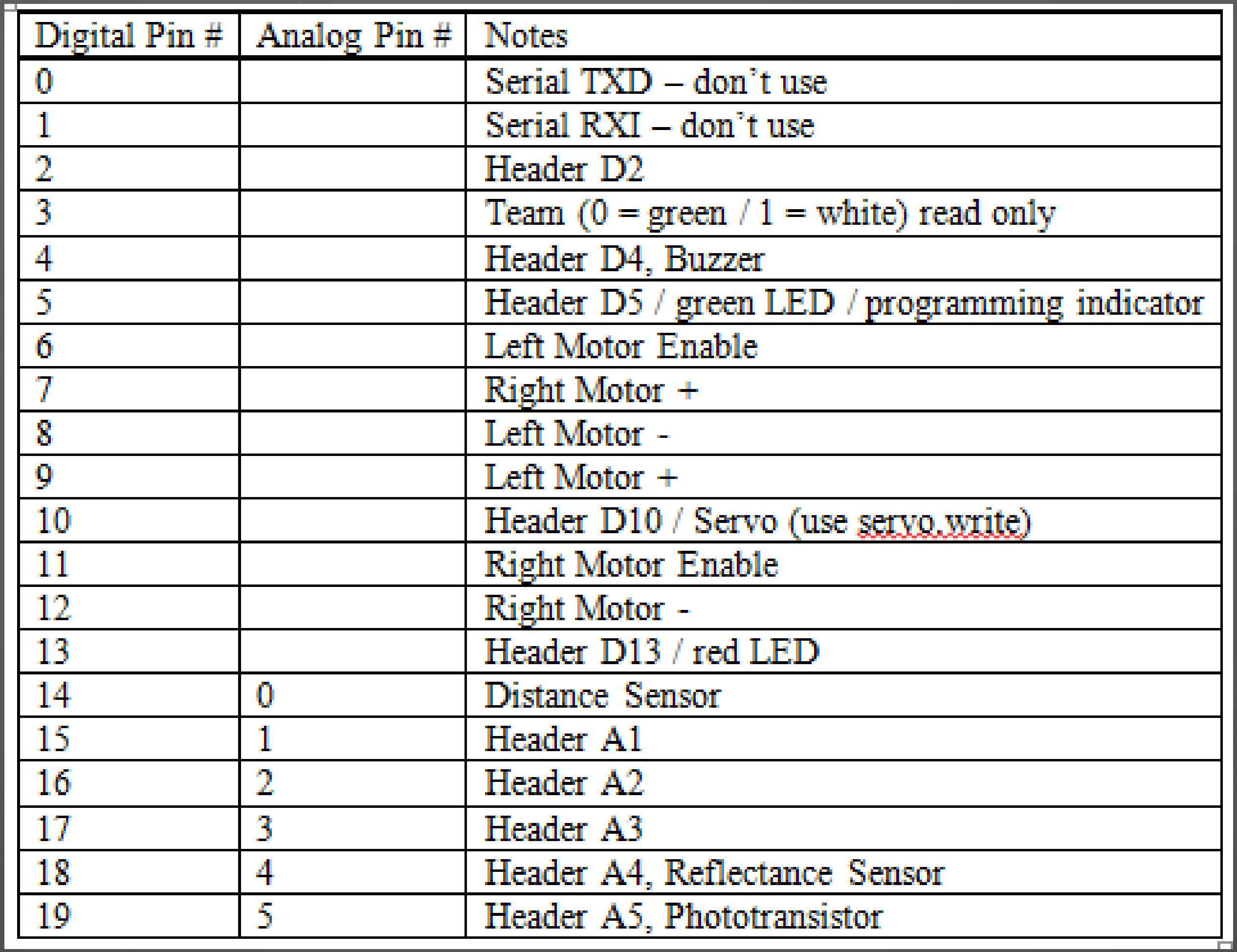
Serial Input

```
void setup()
{
  Serial.begin(9600); // opens serial port at 9600 baud
  Serial.println(“Press a number: “);
}

void loop() {
  int pressed;
  if (Serial.available())
  {
    pressed = Serial.read() – 48;
    Serial.print(“Value pressed is: “);
    Serial.println(pressed);
  }
}
```

Physical Inputs and Outputs

- Setup
  - – pinMode(*pin*, *mode*)
  - – *mode* is either: INPUT or OUTPUT
- Output
  - – digitalWrite(*pin*, *value*)
  - – *value* is either: HIGH or LOW
- Input
  - – digitalRead(*pin*)

Mudduino Pinout



Digital Pin #	Analog Pin #	Notes
0		Serial TXD – don't use
1		Serial RXI – don't use
2		Header D2
3		Team (0 = green / 1 = white) read only
4		Header D4, Buzzer
5		Header D5 / green LED / programming indicator
6		Left Motor Enable
7		Right Motor +
8		Left Motor -
9		Left Motor +
10		Header D10 / Servo (use servo.write)
11		Right Motor Enable
12		Right Motor -
13		Header D13 / red LED
14	0	Distance Sensor
15	1	Header A1
16	2	Header A2
17	3	Header A3
18	4	Header A4, Reflectance Sensor
19	5	Header A5, Phototransistor

Physical Output: LED

```
void setup()
{
  Serial.begin(9600);

  // set LED pin as output
  pinMode(13, OUTPUT); // red LED pin
}

void loop()
{
  Serial.println("Testing LED");

  digitalWrite(13, HIGH); // turn red LED on
  digitalWrite(13, LOW);  // turn red LED off
}
```

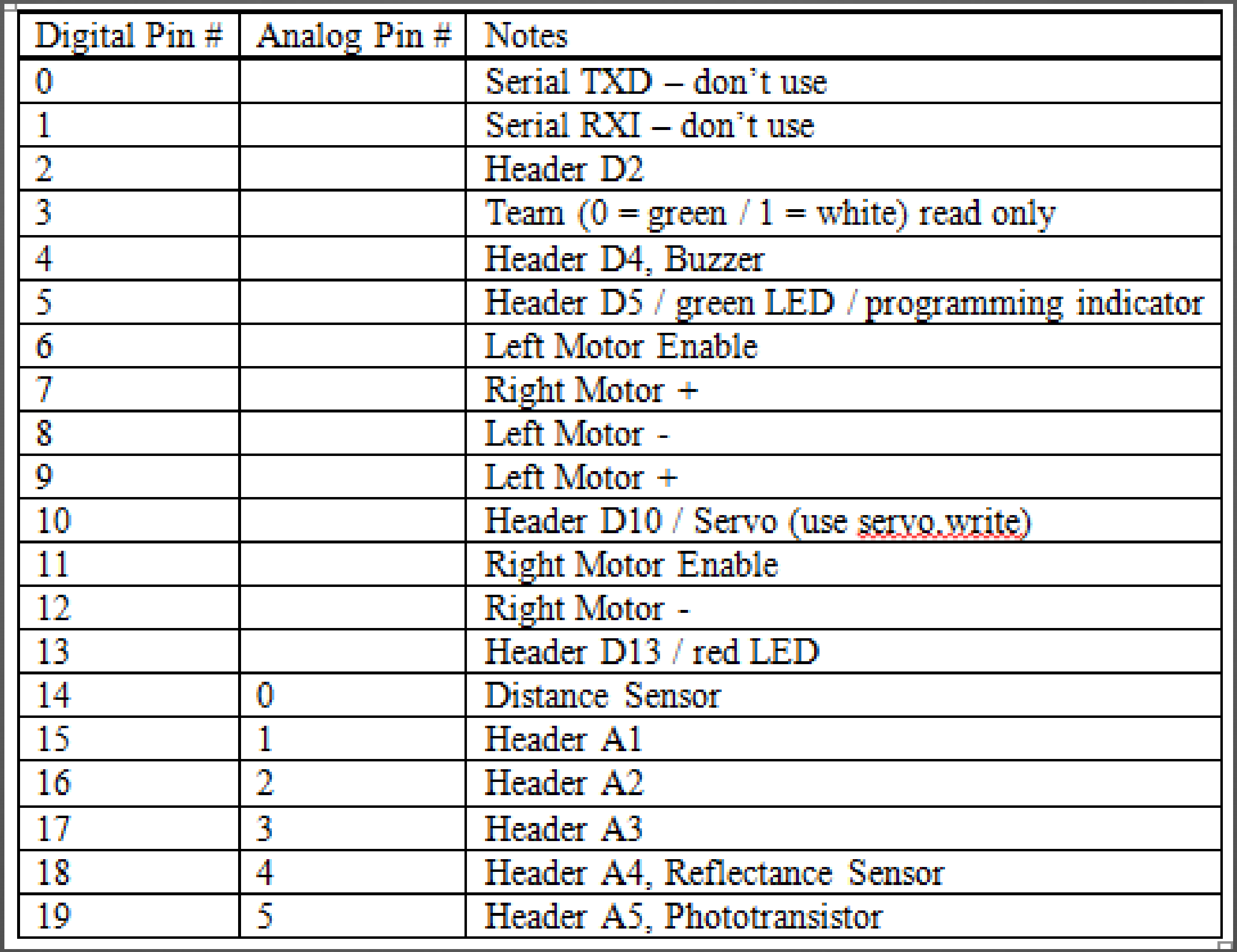
Physical Output: LED

```
void setup()
{
  Serial.begin(9600);

  // set LED pin as output
  pinMode(13, OUTPUT); // red LED pin
}

void loop()
{
  Serial.println("Testing LED");

  digitalWrite(13, HIGH); // turn red LED on
  delay(200);   // delay 200 ms

  digitalWrite(13, LOW);  // turn red LED off
  delay(200);   // delay 200 ms
}
```

Physical Output: Speaker

```
void setup()
{
  Serial.begin(9600);
  // set speaker pin as output
  pinMode(4, OUTPUT);  // speaker pin
 }

void loop()
{
  Serial.println("Testing speaker");

  tone(4, 440); // write tone of 440 Hz to speaker
  delay(200);   // delay 200 ms

  noTone(4);    // turn the speaker (pin 4) off
  delay(200);   // delay 200 ms
}
```

Mudduino Pinout



Digital Pin #	Analog Pin #	Notes
0		Serial TXD – don’t use
1		Serial RXI – don’t use
2		Header D2
3		Team (0 = green / 1 = white) read only
4		Header D4, Buzzer
5		Header D5 / green LED / programming indicator
6		Left Motor Enable
7		Right Motor +
8		Left Motor -
9		Left Motor +
10		Header D10 / Servo (use servo.write)
11		Right Motor Enable
12		Right Motor -
13		Header D13 / red LED
14	0	Distance Sensor
15	1	Header A1
16	2	Header A2
17	3	Header A3
18	4	Header A4, Reflectance Sensor
19	5	Header A5, Phototransistor

Physical Input: Analog Port

```
void setup()
{
  Serial.begin(9600);
  pinMode(14, INPUT);   // set digital pin 14
                        // (analog pin 0) as input
}

void loop()
{
  int randNum;

  Serial.print("Pin 0: ");
  randNum = analogRead(0);
  Serial.println(randNum);
  delay(800);
}
```

#define

Makes the program more consistent – no magic numbers!

#define

Makes the program more consistent – no magic numbers!

So instead of …

```
void setup()
{
  Serial.begin(9600);
  pinMode(13, OUTPUT); // red LED pin
}

void loop()
{
  Serial.println("Testing LED");
  digitalWrite(13, HIGH); // turn red LED on
  delay(200);
  digitalWrite(13, LOW);  // turn red LED off
  delay(200);
}
```

#define

Makes the program more consistent – no magic numbers!

We have…

```
#define REDLED 13

void setup()
{
  Serial.begin(9600);
  pinMode(REDLED, OUTPUT); // red LED pin
}

void loop()
{
  Serial.println("Testing LED");
  digitalWrite(REDLED, HIGH); // turn red LED on
  delay(200);
  digitalWrite(REDLED, LOW);  // turn red LED off
  delay(200);
}
```

Randomness

```
void setup()
{
  Serial.begin(9600);
  Serial.println("Here are some random numbers between 0 and 43.");

}

void loop()
{
  int randNum;

  randNum = random(0, 43);
  Serial.println(randNum);
  delay(1000);
}
```

Your turn!

Write a program that repeatedly plays a random tone (between 200 and 500 Hz) to the speaker for 800 ms. The speaker should then turn off for ½ a second.

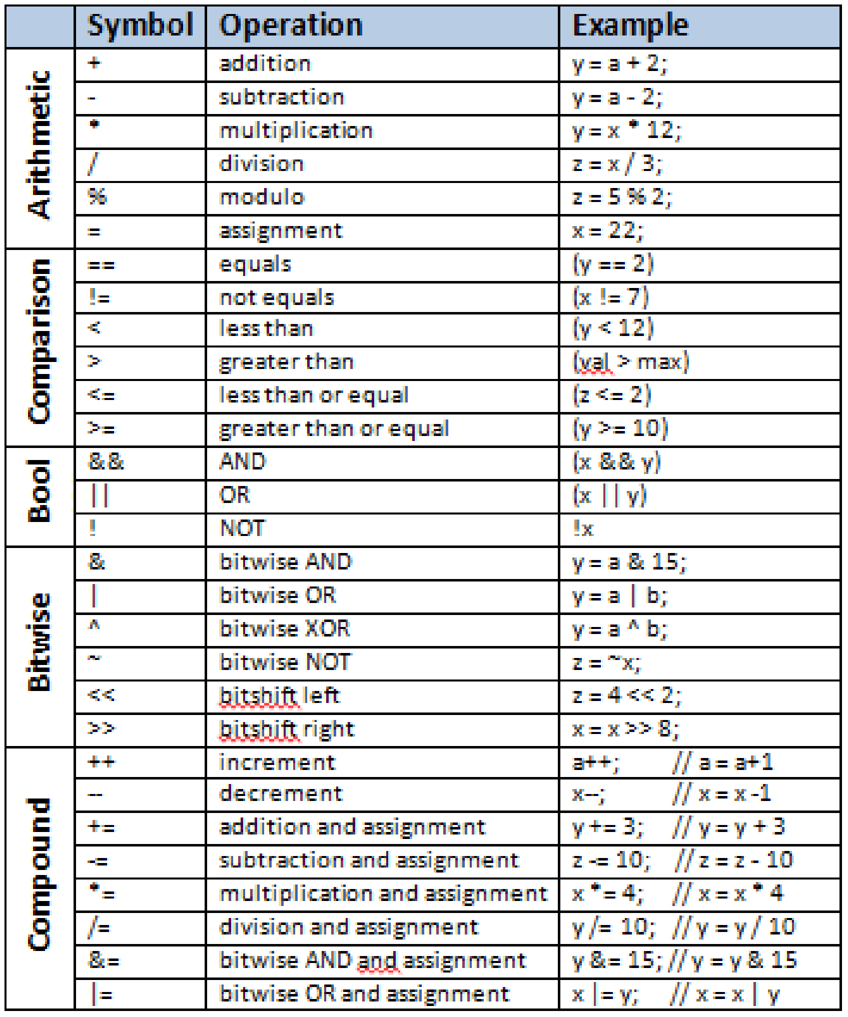
Music?

```
#define SPEAKER 4

void setup()
{
  Serial.begin(9600);
  // set speaker pin as output
  pinMode(SPEAKER, OUTPUT);   // speaker pin
}

void loop()
{
  int randNum = random(200, 501);

  Serial.println("Writing random tones to speaker");
  tone(SPEAKER, randNum); // write tone to speaker
  delay(800);             // tone lasts 800 ms
  noTone(SPEAKER);        // turn the speaker (pin 4) off
  delay(500);             // speaker is off for 500 ms
}
```

Even more random…

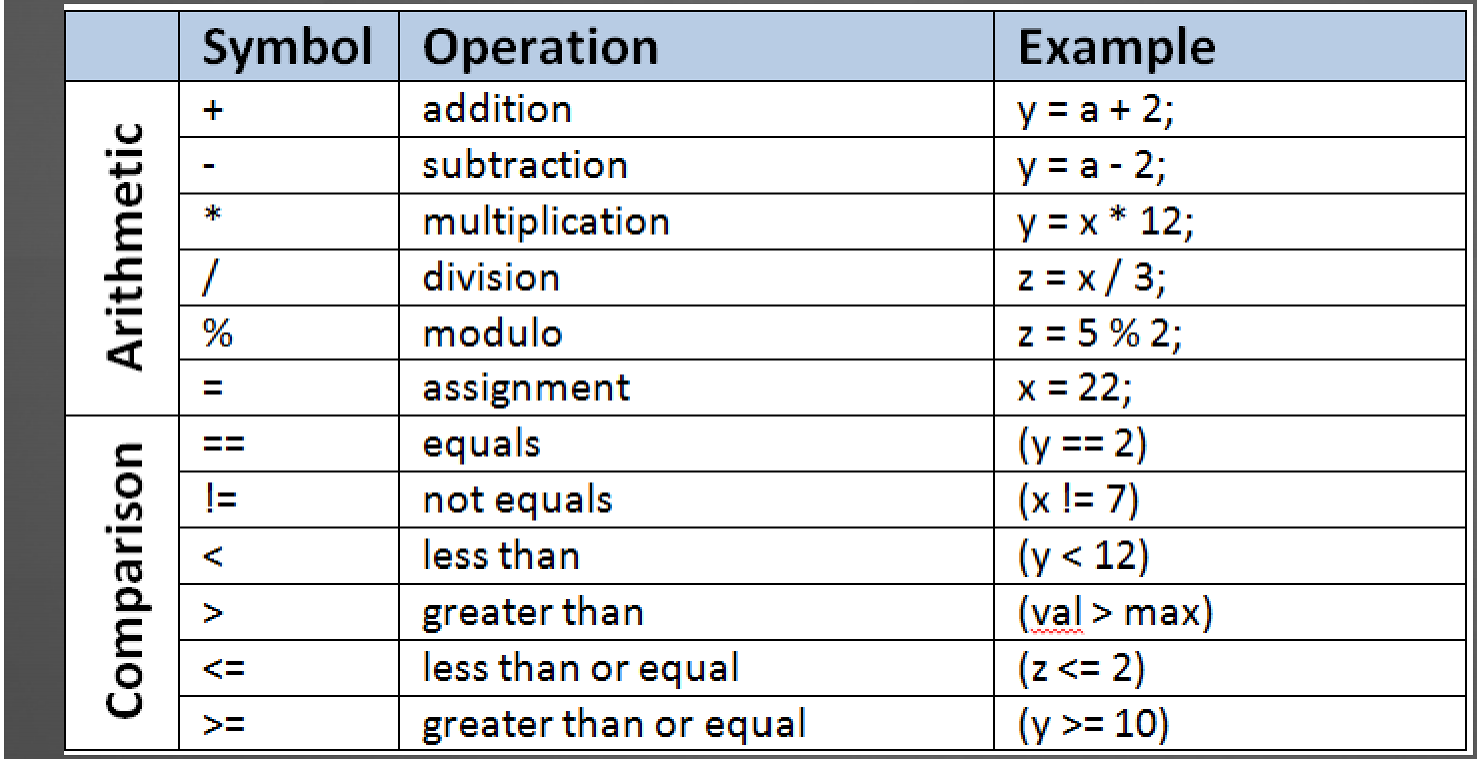
```
void setup()
{
  int randSeed;

  Serial.begin(9600);
  Serial.println("Here are some random numbers between 0 and 43.");
  pinMode(14, INPUT);   // set digital pin 14
                        // (analog pin 0) as input
  randSeed = analogRead(0);  // read random value from port 0
  Serial.print("Random seed is: "); Serial.println(randSeed);
  randomSeed(randSeed);     // seed the random number generator
}
void loop()
{
  int randNum = random(0, 43);  // set the random number
  Serial.println(randNum);      // print the random number
  delay(1000);
}
```

Operators



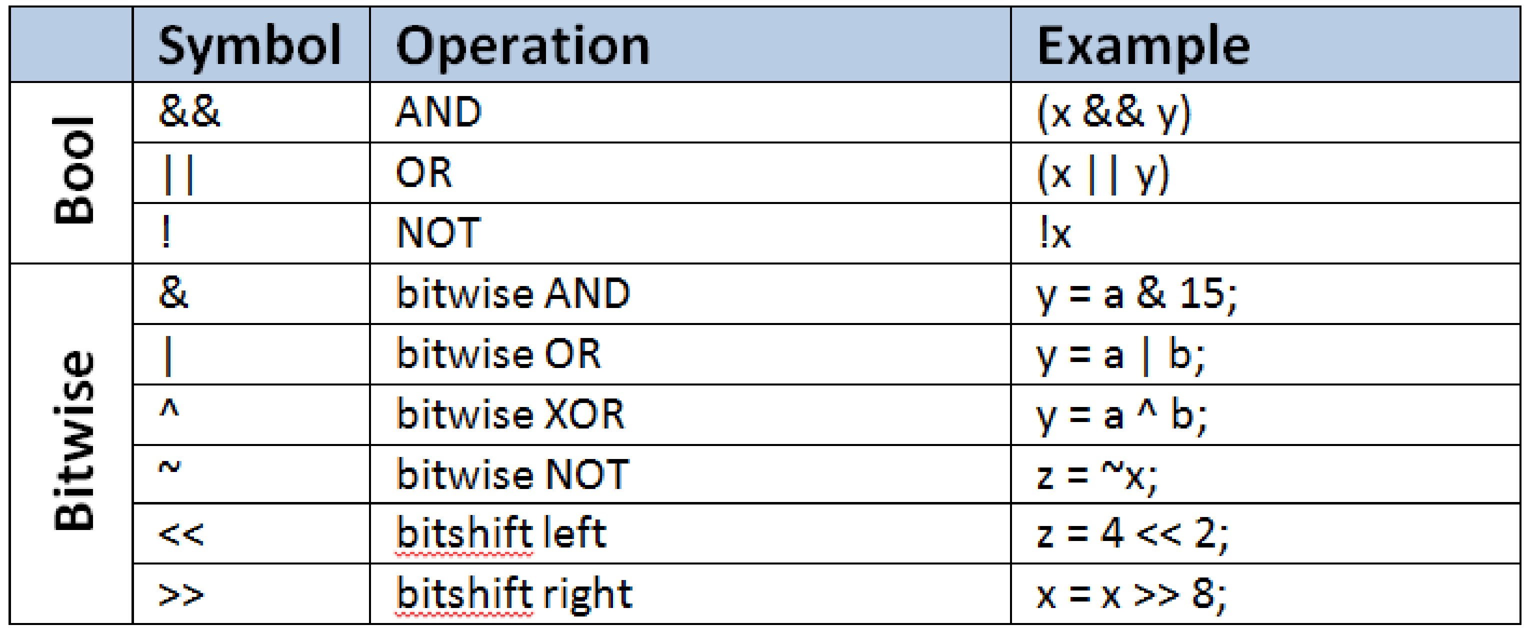
	Symbol	Operation	Example
Arithmetic	+	addition	y = a + 2;
	-	subtraction	y = a - 2;
	*	multiplication	y = x * 12;
	/	division	z = x / 3;
	%	modulo	z = 5 % 2;
	=	assignment	x = 22;
Comparison	==	equals	(y == 2)
	!=	not equals	(x != 7)
	<	less than	(y < 12)
	>	greater than	(val > max)
	<=	less than or equal	(z <= 2)
	>=	greater than or equal	(y >= 10)
Bool	&&	AND	(x && y)
	||	OR	(x || y)
	!	NOT	!x
Bitwise	&	bitwise AND	y = a & 15;
	|	bitwise OR	y = a | b;
	^	bitwise XOR	y = a ^ b;
	~	bitwise NOT	z = ~x;
	<<	bitshift left	z = 4 << 2;
	>>	bitshift right	x = x >> 8;
Compound	++	increment	a++; // a = a+1
	–	decrement	x–; // x = x -1
	+=	addition and assignment	y += 3; // y = y + 3
	-=	subtraction and assignment	z -= 10; // z = z - 10
	*=	multiplication and assignment	x *= 4; // x = x * 4
	/=	division and assignment	y /= 10; // y = y / 10
	&=	bitwise AND and assignment	y &= 15; // y = y & 15
	|=	bitwise OR and assignment	x |= y; // x = x | y

Arithmetic and Comparison



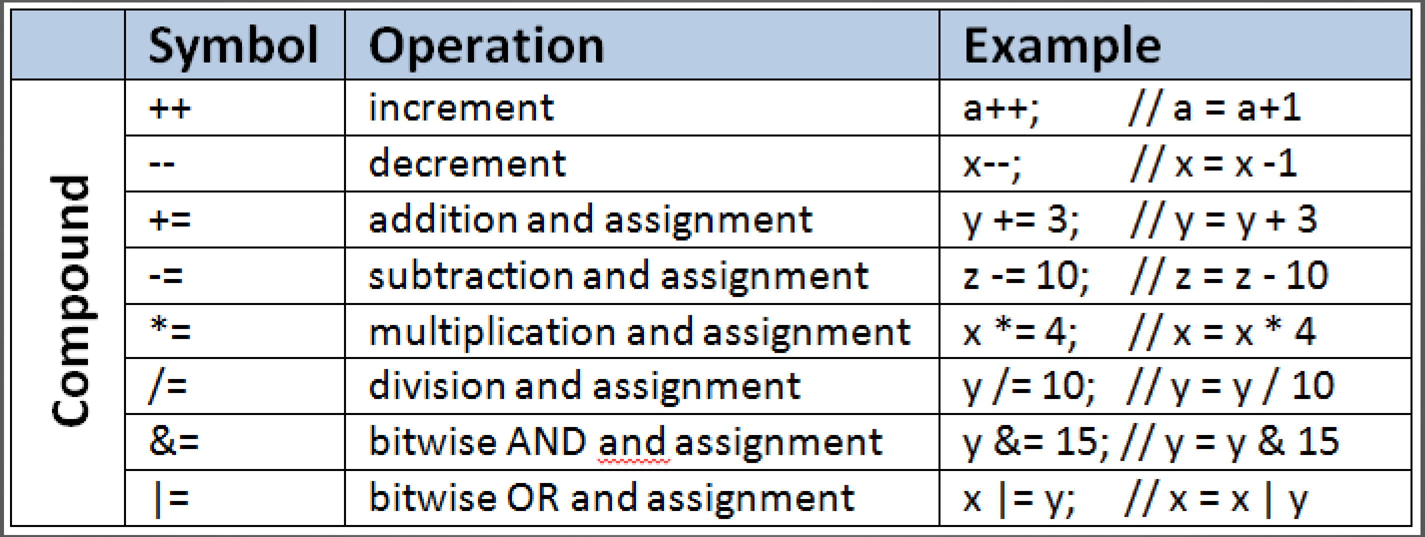
	Symbol	Operation	Example
Arithmetic	+	addition	y = a + 2;
	-	subtraction	y = a - 2;
	*	multiplication	y = x * 12;
	/	division	z = x / 3;
	%	modulo	z = 5 % 2;
	=	assignment	x = 22;
Comparison	==	equals	(y == 2)
	!=	not equals	(x != 7)
	<	less than	(y < 12)
	>	greater than	(val > max)
	<=	less than or equal	(z <= 2)
	>=	greater than or equal	(y >= 10)

Boolean and Bitwise



	Symbol	Operation	Example
Bool	&&	AND	(x && y)
	||	OR	(x || y)
	!	NOT	!x
Bitwise	&	bitwise AND	y = a & 15;
	|	bitwise OR	y = a | b;
	^	bitwise XOR	y = a ^ b;
	~	bitwise NOT	z = ~x;
	<<	bitshift left	z = 4 << 2;
	>>	bitshift right	x = x >> 8;

Compound Operations



	Symbol	Operation	Example
Compound	++	increment	a++; // a = a+1
	--	decrement	x--; // x = x -1
	+=	addition and assignment	y += 3; // y = y + 3
	-=	subtraction and assignment	z -= 10; // z = z - 10
	*=	multiplication and assignment	x *= 4; // x = x * 4
	/=	division and assignment	y /= 10; // y = y / 10
	&=	bitwise AND and assignment	y &= 15; // y = y & 15
	|=	bitwise OR and assignment	x |= y; // x = x | y

Control Statements

- if
- if / else
- while
- do
- for
- switch / case

if Statement

```
if (i == 25) {
  Serial.println(“You guessed the magic number!”);
}
y = 42;
```

if / else Statement

```
if (i == 25) {
  Serial.println(“You guessed the magic number!”);
}
else {
  Serial.println(“Try again!”);
}
```

while Statement

```
int x = 1;
while (x < 1000) {
  Serial.println(x);
  x = x*2;
}
```

do Statement

```
int x = 0;

do {
  delay(100);  // delay 100 ms between readings
  x = analogRead(0);
} while (x < 300);
```

for Loop

```
for (initialization; condition; loop operation)
  loop body
```

- `initialization`: executes before the loop begins
- `condition`: is tested at the beginning of each iteration
- `loop operation`: executes at the end of each iteration
- `loop body`: executes each time the condition is met

for Loop

```
int i;
int x = 1;

for (i = 2; i < 10; i++)
  x = x * i;
```

switch / case Statement

```
switch (var) {
  case 0:
    ...     // do something
    break;
  case 1:
    ...     // do something else
    break;
  default:
    Serial.println(“You pressed an invalid number”);
}
```

Your turn!

Write a program that turns on an LED for a length of time depending on a user input of 1, 2, or 3. The choices correspond to LED on times of 300, 800, or 2000 ms. The LED should then turn off for at least ½ a second until the next user input.

Assume you already have the user input: `int choice;` `choice` is 0 if there is no user input.