```
  Serial.begin(9600);
}

void loop() {
  // Print every step of the for loop
  for(int count=0;count<10;count++){
    // Print the count to the serial
    Serial.println(count);
```

Arduino: Variable Types, Serial Communication, and Bytes

Eric Burger
Fall 2015

WARNING…

Things are going to get a little technical...

Data in Memory

- Bits and Bytes
  - A bit is a binary digit (0 or 1)
  - A byte is 8 bits
    - 0 to 255 in decimal
    - 0x00 to 0xFF in hexadecimal
- Declaring variables
  - When a variable is created, a certain amount of memory is reserved for that variable
  - For an int, 2 bytes are reserved (0x0000 to 0xFFFF)

Decimal (Base 10)	Binary (Base 2)	Hexadecimal (Base 16)
0	0	0x0 (or 0x00)
1	1	0x1 (or 0x01)
2	10	0x2
3	11	0x3
4	100	0x4
5	101	0x5
6	110	0x6
7	111	0x7
8	1000	0x8
9	1001	0x9
10	1010	0xA
11	1011	0xB
12	1100	0xC
13	1101	0xD
14	1110	0xE
15	1111	0xF
16	10000	0x10

Arduino Variable Types: Numbers

- int :
  - Integer between −32,768 and 32,767
- unsigned int :
  - Integer between 0 and 65,535
- long :
  - Integer between −2,147,483,648 and 2,147,483,647
- unsigned long :
  - Integer between 0 and 4,294,967,295
- double or float :
  - Decimal between $-3.4028235*10^{38}$ and $3.4028235*10^{38}$ with 6 or 7 digitals of precision
    - 0.002014186
    - 2014.186
- Arduino Reference page for full list of number types

Arduino: Arithmetic

- Integer arithmetic generates integers
- Pay attention to order of operations (Send: 0)

```
int r;
r = 9/10*100; // r is 0
r = 9*100/10; // r is 90
r = (9*100)*(1/10); // r is 0
```

- Integer overflow:
  - (0x7FFF + 1 = 0x8000)

```
r = 32767 + 1; // r is -32768
```

- Conversion (Casting) (1)

```
int r;
// Type Casting
r = (float) 9/10*100; // r is 90
// Type Conversion
r = float(9)/10*100;  // r is 90
```

- Avoid float arithmetic (2)
  - Not exact
  - Slower than int math
- Arduino Reference page for details

Other Arduino Variable Types

- void : Null
- boolean : true or false
- char : ASCII Code Chart (3)
  - (0 to 255, 0x00 to 0xFF)

```
int r;
r = int('a'); // r is 97
r = int('2'); // r is 50

char c = 'a';
c++;     // c is 'b'
c = 99; // c is 'c'
```

Character	Decimal	Hexadecimal
!	33	0x21
#	35	0x23
&	38	0x26
0	48	0x30
1	49	0x31
2	50	0x32
A	65	0x41
B	66	0x42
C	67	0x43
a	97	0x61
b	98	0x62
c	99	0x63

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

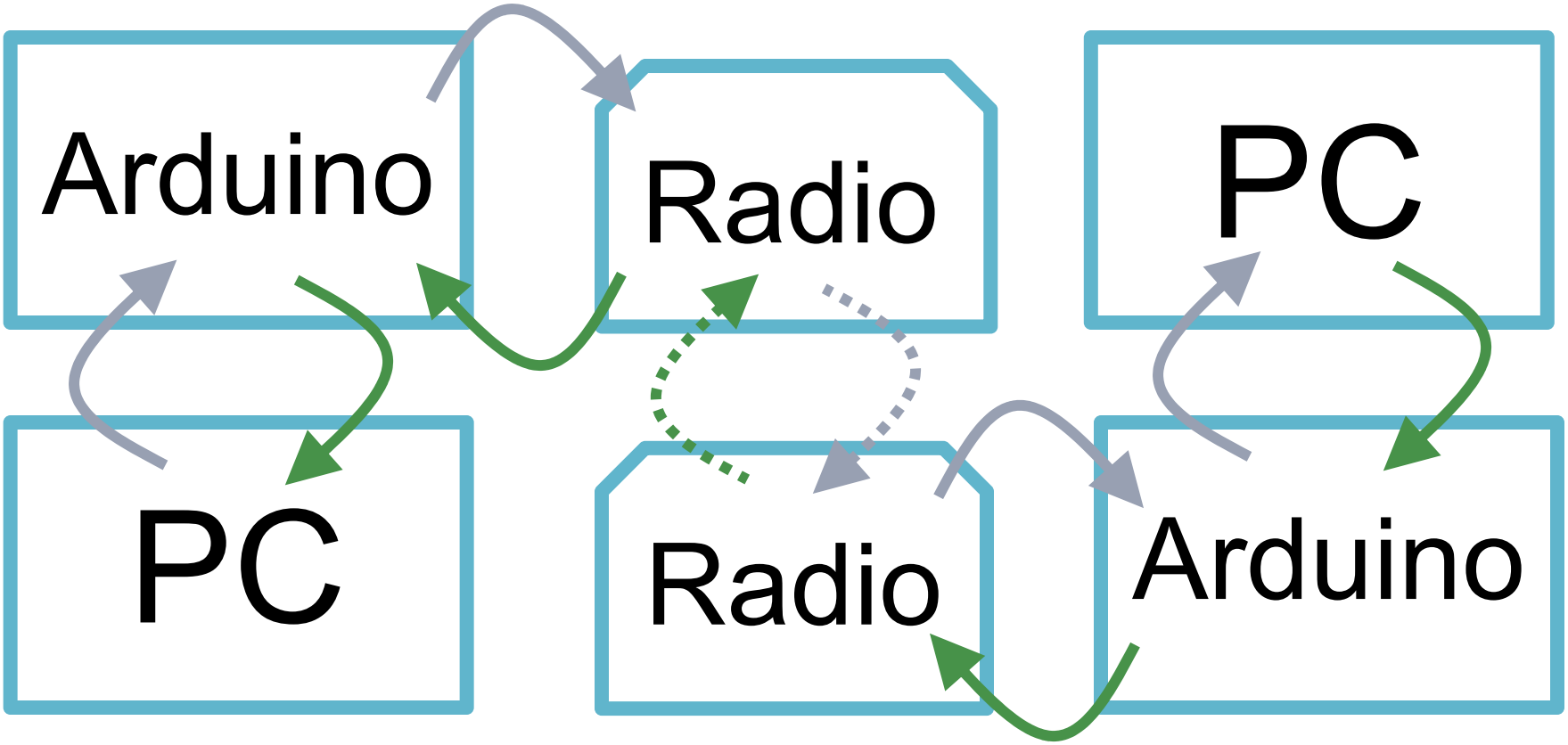
Source: www.LookupTables.com

Variables in Memory

Boolean	1	true = 0x01, false = 0x00		
Char	1	See Hx values in chart at http://www.asciitable.com/		
	Bytes	Min Hexadecimal Value	Median Hexadecimal Value	Max Hexadecimal Value
Byte	1	0x00	0x7F	0xFF
Integer or Short	2	0x8000	0x0000	0x7FFF
Unsigned Integer	2	0x0000	0x7FFF	0xFFFF
Long	4	0x80000000	0x00000000	0x7FFFFFFF
Unsigned Long	4	0x00000000	0x80000000	0xFFFFFFFF
Double or Float	4	0xFF7FFFFF	0x00000000	0x7F7FFFFF

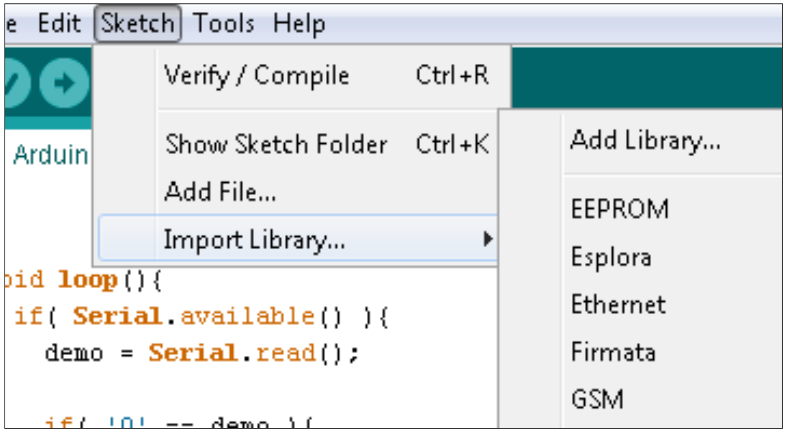
Building a Simple System

- Why learn bytes?
- Wireless communication with Xbee radios



Adding Libraries

- Libraries extend the functionality of Arduino environment.
- Incorporate code written by other developers.
- Syntax:
  - Starts with “#include”
- Filename:
  - The filename of the library is placed between “< >”
  - All libraries are “.h” files
- Advice: Check examples




```
#include <SoftwareSerial.h>
#include <Servo.h>

// RX=>DOUT, TX=>DIN
SoftwareSerial mySerial(10, 1
// Create a servo object to c
Servo myservo;
```

Exercises

- Work with a partner to complete Part 2 exercises.