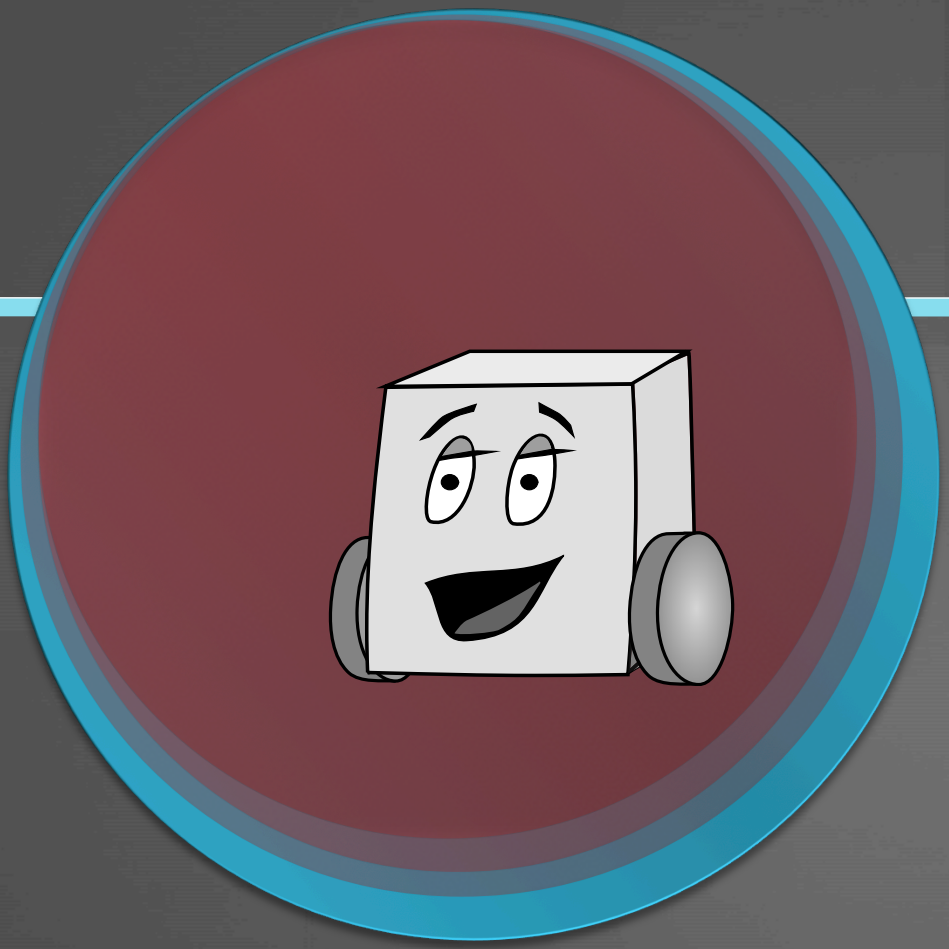

E11 Lecture 4: More C!!!

Prof David Money Harris

Fall 2014

Outline

- Analog Inputs
- Randomness
- Operators
- Control Statements

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);   // D14/A0 as input
}

void loop()
{
  int randNum;

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

Physical Output: Analog (PWM)

```
void setup()
{
  Serial.begin(9600);
  pinMode(5, OUTPUT);   // D5 (green LED)
}

void loop()
{
  analogWrite(5, 0);  // 0 = off
  delay (500);
  analogWrite(5, 127); // 127 = half (2.5V)
  delay(500);
  analogWrite(5, 255); // 255 = full (5 V)
  delay(500);
}
```

#define

Makes the program easier to read and keep up to date

- no magic numbers!

#define

Makes the program easier to read and keep up to date

- 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 easier to read and keep up to date

- 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);
}
```

Pseudo-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);
}
```

Pseudo-randomness

- What happens if you run the program again?
- Random number seed

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);

  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
}
```

Seeding the Random Numbers

```
void setup()
{
  long randSeed;

  Serial.begin(9600);
  Serial.println(”Press a key to begin.");
  while (Serial.available() == 0); // wait until a key is pressed
  randomSeed(micros()); // Seed the random number generator with time
}
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

Operators Example

```
int z, x = 14; int y = 43;  // x = 1110, y = 101011

z = y / x;
z = y % x;
z = x && y;
z = x && 0;
z = x || y;
z = x || 0;
z = x & y;
z = x | y;
z = x ^ y;
z = x << 2;
z = y >> 3;
x += 2;
y &= 15;
```

Control Statements

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

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!");
}
```

switch / case Statement

```
switch (var) {
  case 0:
    Serial.prinln(“Nice choice!”);
    break;
  case 1:
    Serial.prinln(“I wouldn’t have done that!”);
    break;
  default:
    Serial.println(“You pressed an invalid number”);
}
```

while Statement

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

do / while 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;
```

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.

Your turn!

```
switch(choice) {
   case 0: break;
   case 1:
      digitalWrite(REDLED, HIGH); // turn red LED on
      delay(300);
      break;
   case 2:
      digitalWrite(REDLED, HIGH); // turn red LED on
      delay(800);
      break;
   case 3:
      digitalWrite(REDLED, HIGH); // turn red LED on
      delay(2000);
      break;
   default:
 }

 if (choice) {
   digitalWrite(REDLED, LOW);  // turn red LED off
   delay(500);
 }
```