

# BEFORE WE BEGIN

- Homework files found on github
  - <https://github.com/mcc-robotics/enr161-exercises>
- Homework is due by midnight of day on schedule
- All homework is submitted via github

Programs should include the following header:

```
/* ProgramName
 *
 * Details about the program
 *
 * Author: Geoff Berl
 * Assisted by:
 * Date: 3/1/2019
 */
```

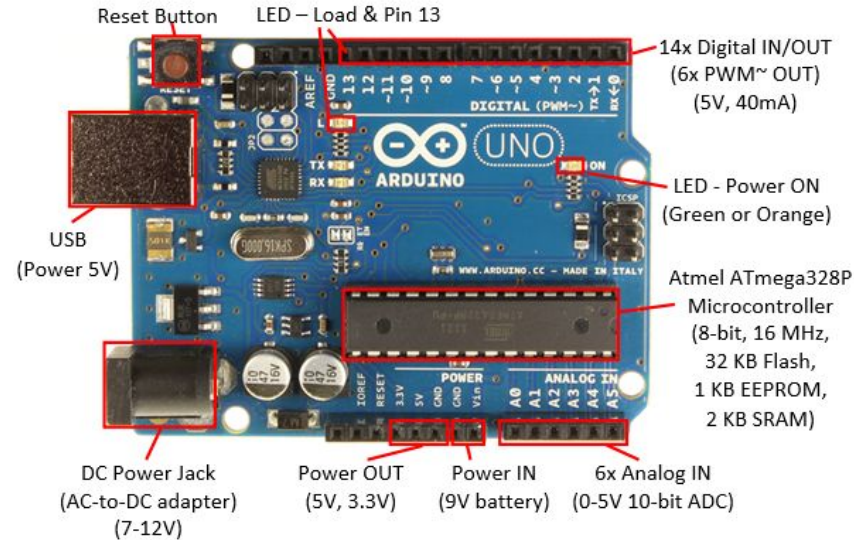
# GETTING STARTED WITH ARDUINO

## **Lesson 1**

How to Write and Compile a Program

# WHAT IS AN ARDUINO?

AN ARDUINO IS A BOARD WITH AN INTEGRATED MICROCONTROLLER FOR PROCESSING CODE



# LESSON OBJECTIVE

Write and compile a program for the Arduino

1. Quick primer on IDEs
2. Fundamental elements of programming
3. Software Requirements
4. Creating an outline
5. Writing the program
6. Compile and Upload

# INTEGRATED DEVELOPMENT ENVIRONMENT

# GETTING STARTED WITH AN IDE

What is an Integrated Development Environment?

Write code with added features

- Syntax highlighting
- Menu option for compiling and/or running
- Error checking
- Formatting and other code-related features

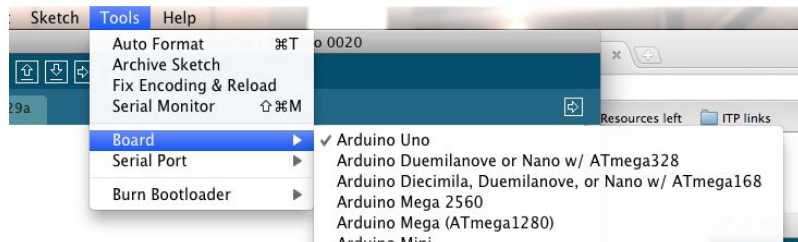


Word processors are to writing as IDEs are to coding

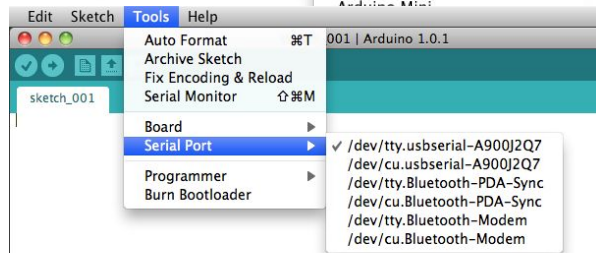
# SETUP YOUR ENVIRONMENT

Three steps to setting up environment

1. Select the right board



2. Select the right port



3. Select the right programmer

# PROGRAMMING FUNDAMENTALS



# PROGRAMMING LANGUAGE FUNDAMENTALS

## Arduino Programming Structure

- Comments
- Functions
- Statements

## Noteworthy Extras

- Syntax
- Libraries
- Code formatting

# COMMENTS

The most highly underutilized element in code  
Comments are free, use them and use them often

## Examples

```
// This is a single line comment
```

```
/*  
 * This is a comment block  
 * Comment blocks are used when your comment  
 *   takes up more than one line although  
 *   it is not necessary.  
 */
```

# FUNCTIONS

## Elements of a function

- Return type
- Name
- Parameters

```
return name(Parameters)
```

## Defining a function

- Parameters are optional

```
void loop()
```

## Calling a function

- Only the name and arguments

```
loop()
```



Use the [Arduino Reference](#) to find help with common functions

# STATEMENTS

Statements can include

- Function calls
- Variable declarations
- Variable mutations (arithmetic)

Examples

```
loop();
```

```
int someInteger = 4;
```

```
someInteger = someInteger + 6;
```

# SYNTAX

Set of rules that defines the combinations of symbols that are considered to be a correctly structured document or fragment in that language

- Syntax Aids
  - Color coding for known “elements”
  - Use indentation
    - Tools → Auto Format (or Ctrl+t)
- Common syntactical errors
  - Forgetting semicolon
  - Forgetting parenthesis
  - Forgetting curly braces
  - Improper case

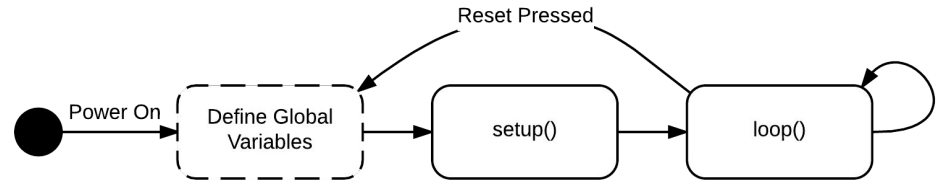
# SOFTWARE REQUIREMENTS

The program shall

- Blink an LED
  - The LED shall be off for one second
  - The LED shall be on for one second
- Utilize variables and constants
  - The constant LED\_BUILTIN shall be used for referencing the LED pin
  - The time for the LED delay shall be stored in a variable

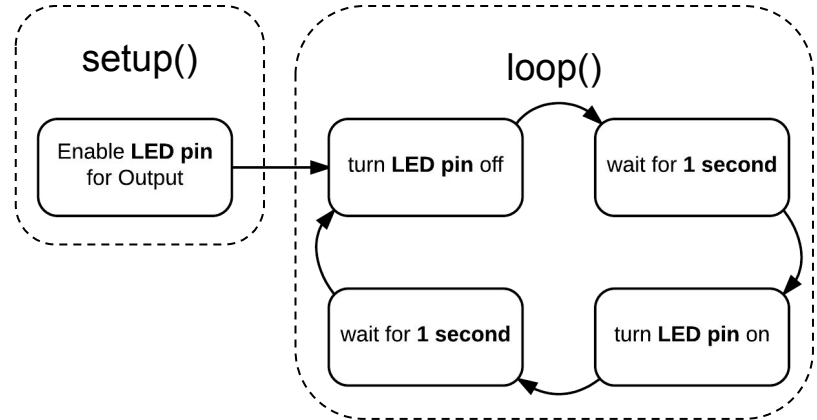
WRITING SOFTWARE

# CREATE AN OUTLINE



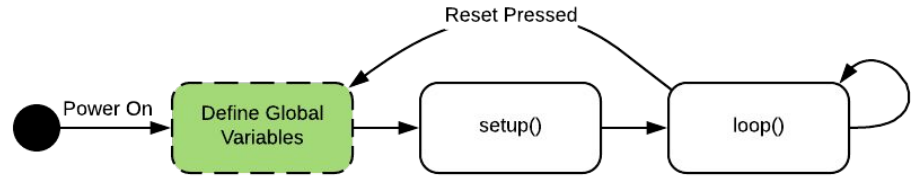
1. Define your ultimate goal
2. Break into sub goals if necessary
3. Break each goal/subgoal into a set of instructions

1. Enable LED pin for output
2. Turn the LED pin off
3. Wait for one second
4. Turn the LED pin on
5. Wait for one second
6. Repeat step 2 through 5



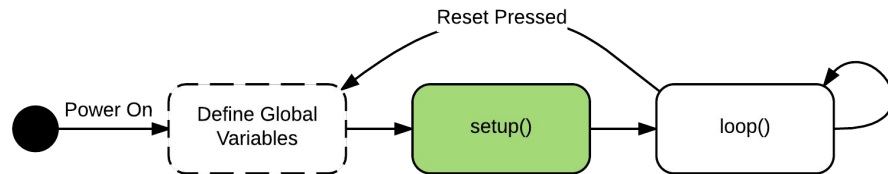


# DEFINE GLOBAL VARIABLES



```
// Number of milliseconds to hold the LED on and off
int ledDelayMS = 1000;
int ledPin = 13;           // The LED pin
```

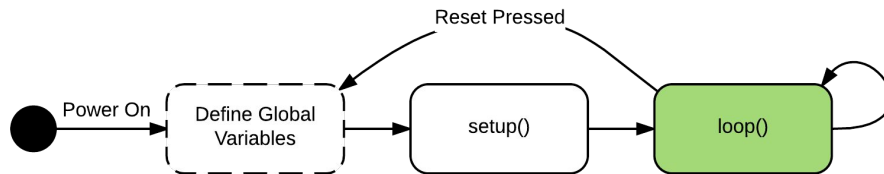
# THE 'SETUP()'



```
/*
 * setup()
 * Called one time, used for initializing program elements
 */
void setup() {
    // Enable LED pin for output
    pinMode(LED_BUILTIN, OUTPUT);
}
```

- If this was `pinMode(13, 1)` would that make sense?
- Other options
  - a. `pinMode(ledPin, OUTPUT);` // Using our own variable
  - b. `pinMode(LED_PIN, OUTPUT);` // Using our own constant

# THE 'LOOP()'



```
/*
 * loop()
 * Represents the main program,
 * Blink an LED using the global variable time delay
 */
void loop() {
    // Turn the LED pin off
    digitalWrite(LED_BUILTIN, LOW);

    // Wait for ledDelayMS milliseconds
    delay(ledDelayMS);

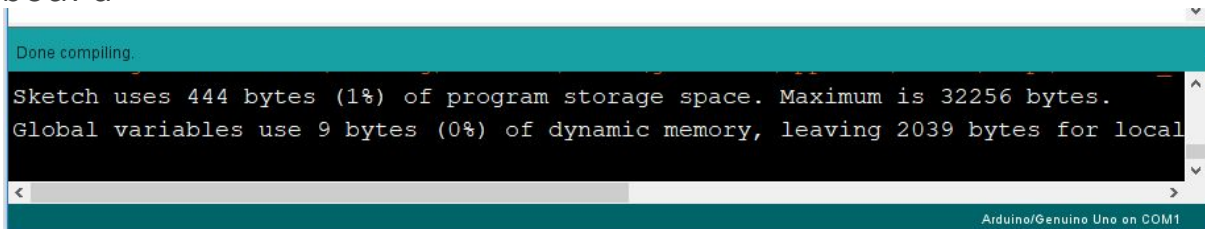
    // Turn the LED pin on
    digitalWrite(LED_BUILTIN, HIGH);

    // Wait for ledDelayMS milliseconds
    delay(ledDelayMS);
}
```

# VERIFY, COMPILE, AND UPLOAD

## Verify/Compile

- Compiles the code
- Checks for errors
- Can be done without a board

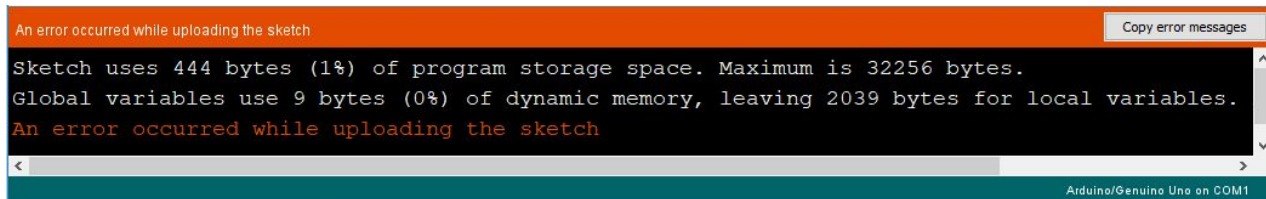


A screenshot of the bottom status bar of the Arduino IDE. The top part is a teal bar with the text "Done compiling." in white. Below it is a black area with white text: "Sketch uses 444 bytes (1%) of program storage space. Maximum is 32256 bytes. Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables." At the bottom right, it says "Arduino/Genuino Uno on COM1".

```
Done compiling.  
Sketch uses 444 bytes (1%) of program storage space. Maximum is 32256 bytes.  
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local  
variables.  
Arduino/Genuino Uno on COM1
```

## Upload

- Ready for deployment
- Compiles the code
- Uploads the compiled code to the microcontroller



A screenshot of the bottom status bar of the Arduino IDE. The top part is an orange bar with the text "An error occurred while uploading the sketch" in white. To the right of this bar is a button labeled "Copy error messages". Below the orange bar is a black area with white text: "Sketch uses 444 bytes (1%) of program storage space. Maximum is 32256 bytes. Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables." At the bottom, it says "An error occurred while uploading the sketch" in orange text. At the bottom right, it says "Arduino/Genuino Uno on COM1".

```
An error occurred while uploading the sketch  
Sketch uses 444 bytes (1%) of program storage space. Maximum is 32256 bytes.  
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables.  
An error occurred while uploading the sketch  
Arduino/Genuino Uno on COM1
```

# PARTING THOUGHTS

1. Mind your cases, 'LOW' is not the same as 'low'.
2. Lines of code end with a semicolon ';'.
3. Functions are encapsulated with curly braces '{' and '}'.
4. Most typos are due to odd parenthesis, odd curly braces or missing semicolons.
5. Reference the Arduino website to learn more

# Arduino Programming Cheat Sheet

Geoff Berl | October 18, 2018

## Structure

/\* Every arduino program **must** contain the following \*/

```
void setup() {  
  // Runs once at the beginning  
}
```

```
void loop() {  
  // Runs continuously  
}
```

## Comments

```
// Single line comment  
/* Multi line  
comment */
```

## Control Structures

```
if (x < 5) {  
  // If x < 5, do this  
} else {  
  // Otherwise, do this  
}  
  
for (int i = 0; i < 100; i++) {  
  // do this; start at i=0 and while i < 100  
}  
  
while(x < 6) {  
  // while x < 6, do this  
}
```

## Math Operators

```
= // assignment  
+ // addition  
- // subtraction  
* // multiplication  
/ // division  
% // modulus  
++ // Increment  
-- // Decrement  
+= // Compound addition  
-= // Compound subtraction
```

## Logical Operators

```
== // equal to  
!= // not equal to  
< // less than  
> // greater than  
<= // less than or equal to  
>= // greater than or equal to  
&& // boolean AND  
|| // boolean OR  
! // boolean NOT
```

## Math Functions

```
// Limit a value within a range  
char result = constrain(val, min, max);  
/* Change from one range to another, this  
example changes a value within 0-->1023  
to 0-->255. (Useful for reading an analog  
pin 0-->1023 and outputting to PWM  
0-->255) */  
char result = map(val, 0, 1023, 0, 255);
```

## Digital I/O

```
// Sets the mode of a pin to an input or output  
pinMode(pin, INPUT/OUTPUT);  
// Write a HIGH or LOW value to a digital pin  
digitalWrite(pin, HIGH/LOW);  
// Read the HIGH or LOW value of a digital pin  
boolean value = digitalRead(pin);
```

## Analog I/O

```
// Sets the mode of a pin to an input or output  
pinMode(pin, INPUT/OUTPUT);  
// Write a HIGH or LOW value to an analog pin  
analogWrite(pin, 0-->255);  
// Read the 0-->1023 value of an analog pin  
int value = analogRead(pin);
```

## Constants

HIGH / LOW  
INPUT / OUTPUT  
PI  
LED\_BUILTIN  
true / false

## Data Types

```
void  
boolean 0, 1, false, true  
char 'a' OR -128 --> 127  
unsigned char 0 --> 255  
int -32,768 --> 32,767  
unsigned int 0 --> 65,535  
long -2,147,483,648 --> 2,147,483,647  
unsigned long 0 --> 4,294,967,295  
float -3.4028235E+38 --> 3.4028235E+38
```

## Time

```
// Pause the program for msDelay milliseconds  
delay(msDelay);  
// Pause the program for usDelay microseconds  
delayMicroseconds(usDelay);  
// Obtain the number of ms since the board turned on  
long currentMSTime = millis();  
// Obtain the number of us since the board turned on  
long currentUSTime = micros();
```

# HOMework

Homework examples will consist of examples from

- The Arduino IDE
- Various Libraries (Also the IDE)
- Github <https://github.com/mcc-robotics/enr161-exercises>
  - Easiest method is to download if you are using your own PC

Clone or download ▾

# QUESTIONS?

