

Name: \_\_\_\_\_

Class: \_\_\_\_\_

**Module 1: Crash Prevention**  
**Lesson 3: Weather Information systems**  
**Programming Activity Using Arduino**  
**Grade 9 - 12**

### Road Weather Information

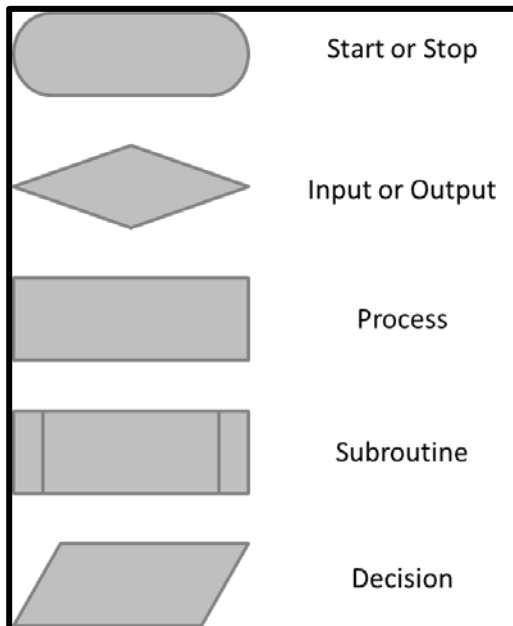
Connected vehicles can communicate with infrastructure (i.e. traffic lights) and other vehicles. One way infrastructure-to-vehicle communication provides relevant information to drivers is through weather information systems. In this activity you will work with your partner to program an Arduino to measure temperature and lighting conditions.

### Hypothesis

Hypothesize the temperature of the room and how much light is present (as a percentage of a bright sunny day).

### Background Information

#### Flow Chart Symbols



#### Arduino Code Examples

```
pinMode(pin, INPUT/OUTPUT);  
  ex: pinMode(13, OUTPUT);  
  
lcd.print(value);  
  ex: lcd.print("Temp  F");  
  
delay(time_ms);  
  ex: delay(2500); // delay of 2.5 sec  
  
analogRead(pin);  
  ex: analogRead(1);  
  
// NOTE: -> commands are CASE-sensitive
```

Note: The photocell (light sensor) is on the Arduino port 0,  
The temperature sensor is on the Arduino port 1.

## Formulas

$$\text{Percentage, \%} = \frac{\text{Part}}{\text{Whole or Maximum Value}} * 100\%$$

$$\text{Temperature, } ^\circ\text{C} = \frac{(\text{Temperature Reading, Volts} - 0.5)}{1024.0}$$

$$\text{Temperature, } ^\circ\text{F} = \text{Temperature, } ^\circ\text{C} * \left(\frac{9.0}{5.0}\right) + 32.0^\circ\text{F}$$

## Materials & Procedure

### Part 1: Analyze the Weather Information Device

First let's explore what the Weather Information Device does:

1. Break into groups of 3 or less.
2. Obtain the following Materials:
  - Weather Information Device
  - Power source (battery and wire)
  - Computer
  - Flash light (*optional*)
3. Power the Weather Information Device by plugging the power source into the bottom of the device.
4. What happens to the light output when you turn out the lights?
5. What happens when you shine light on the device?
6. Where do you think the photocell (light sensor) is located?
7. What happens when you expose the device to a hot temperature?
8. What about cold?

### Part 2: Algorithm Development with Pseudo Code

Now, let's say you're a transportation engineer and you want to program a Weather Information Device on your own, we'll start by developing your own algorithm for the device.

1. Short description:
  - a. Write a brief description of the device (i.e. what will the device do?).
  - b. From the following list, identify the inputs and outputs for the device:
    - i. Photocell (light sensor)
    - ii. LCD screen
    - iii. Temperature sensor
    - iv. Arduino
  - c. Are the inputs analog or digital?

- d. Are there any existing systems the device will integrate?
  - e. Include some brief descriptive notes about the device.
2. Draw a block diagram of the device including power, the microcontroller (Arduino), temperature sensor, photocell (light sensor) and LCD screen.
3. Create a logic flow chart for a program to read the temperature and lighting condition where the temperature and lighting condition is displayed on an LCD screen in degrees F and a percentage of the possible lighting conditions. Assume that the temperature will be read into the microcontroller in degrees C and need to be converted to degrees F.
4. Edit the logic chart to include an “if statement” to select a colored LED or the color of the LCD screen to match with the temperature conditions (e.g. if the temperature is HOT, the LCD screen turns red).
  - a. Where should the “if statement” go?
  - b. Decide what temperature condition you are going to look for.
    - i. For instance if the temperature is above a certain point that you define, the temperature is HOT, or below a certain point that you define the temperature is COLD.
  - c. There may be a number of places it could be included, can you identify where the “if statement” should NOT go?
5. Using the logic flow chart identify which sections will go in the loop section of the Arduino code. (Which parts will we need the microcontroller to process more than once?)
6. From the logic flow chart and the Arduino code examples above, write the pseudo code for the Weather Information Device. Be sure to include the two necessary pieces of Arduino code.
  - a. `void setup ()`

```

{
}

```
  - b. `void loop()`

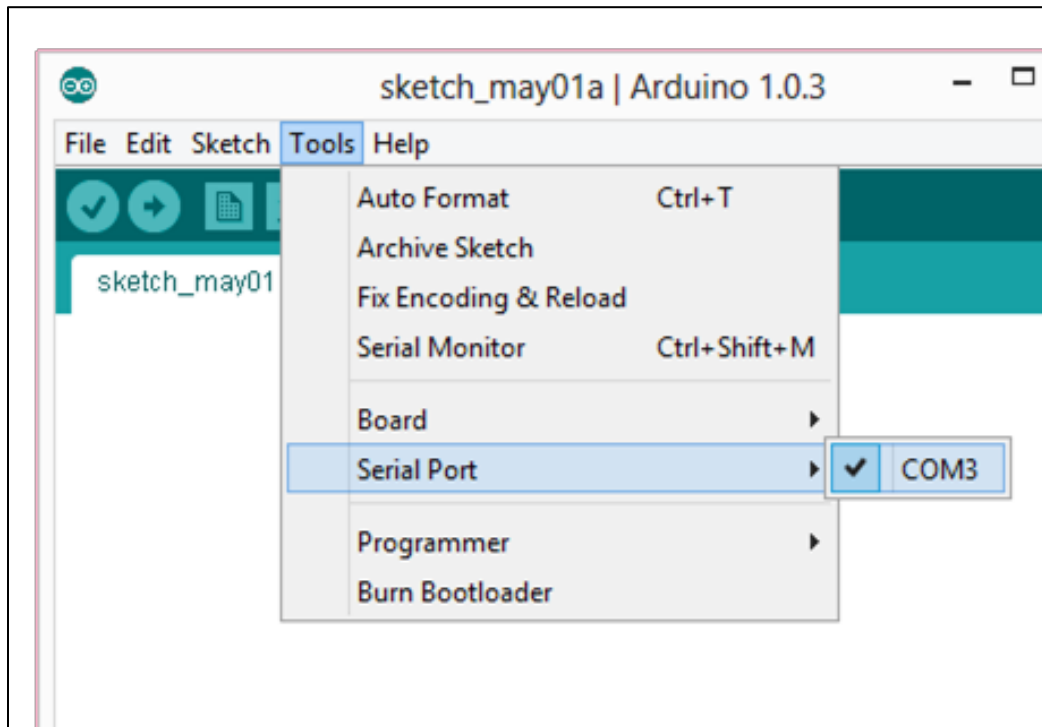
```

{
}

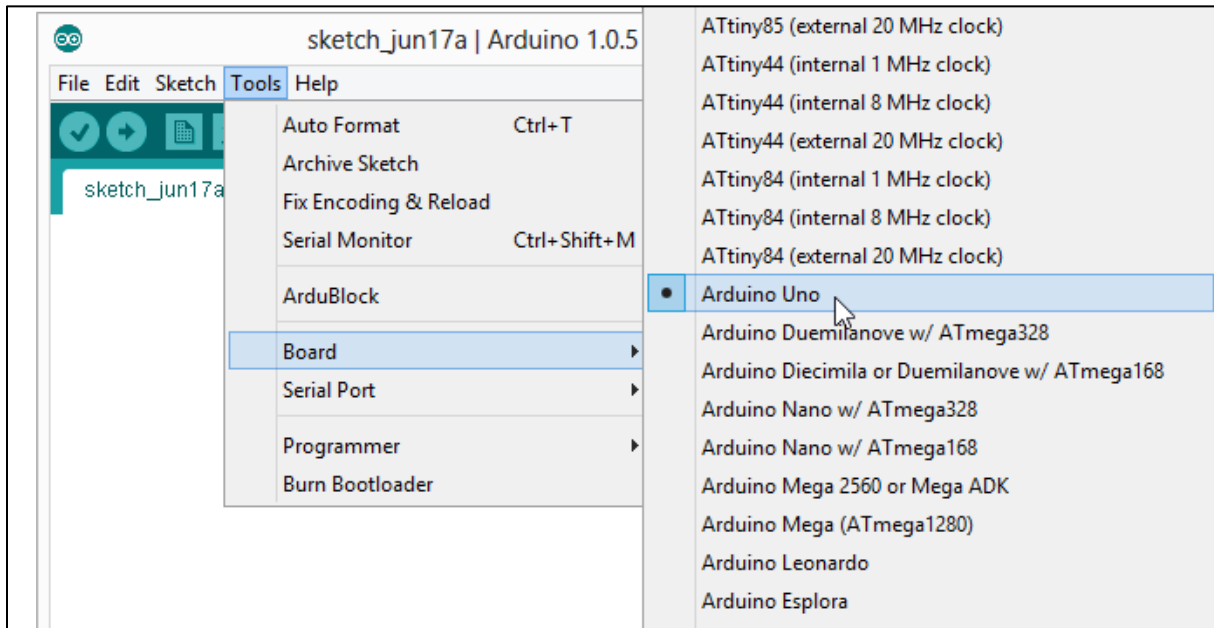
```

### Part 3: Programming

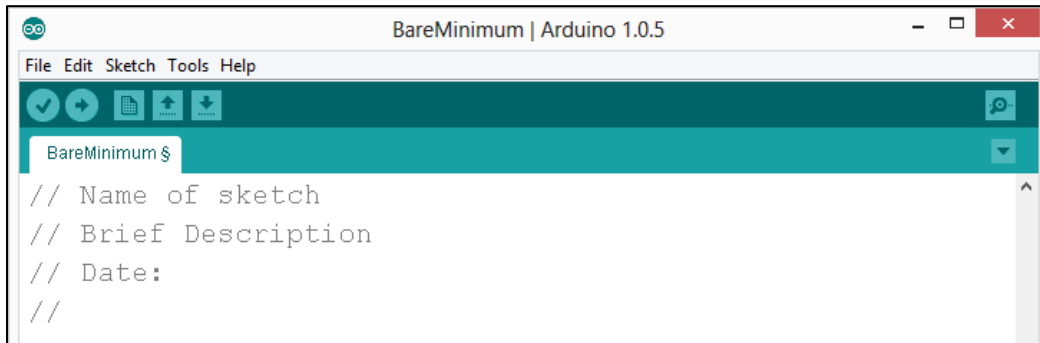
1. Connect Computer to USB port on the Weather Information device.
2. Open the Arduino program:
3. Find the Arduino connection on the computer: Tools → Serial Port  
Your computer communicates to the Arduino microcontroller via a serial port → through a USB-Serial adapter. Check to make sure that the drivers are properly installed.



4. Double-check that the proper board is selected under the Tools→Board menu. (The Weather Information Device has an Arduino Uno inside of it.)



5. Just like any work you do – put your name and date at the top in a comment along with a brief description:



6. Enter the two required functions / methods / routines as shown above and provided below:

```
void setup()
{
    // runs once <- this is a comment
}
void loop()
{
    // repeats
}
```

7. Because we will be outputting to the LCD screen on the Arduino, we will need to include the LCD library, tell the Arduino which ports the LCD is connected on and initialize the LCD screen. To do that enter the following above the setup step:

```
#include <LiquidCrystal.h>
```

```
LiquidCrystal lcd(7, 8, 9, 10, 11, 12);
```

Also, we will need to include the initialize statement for the LCD screen in the setup() routine since it only needs to run once. Enter the following statement in your setup() routine:

```
lcd.begin(16, 2);
```

This code lets the Arduino know that the LCD has 16 characters in length and 2 rows.

8. Determine whether your initializing statements belong in the setup() routine or the loop() routine. Place them there.
9. What variables do we need to declare? (*Look at your flowchart & pseudocode.*)

10. Determine whether your sensor read statements belong in the setup() routine or the loop() routine. Place them there.
  - a. What data type should we assign to the sensor values; integer or float?
  - b. Remember: the photocell (light sensor) is on the Arduino port 0, the temperature sensor is on the Arduino port 1.
11. What calculations do you need to perform on your inputs?
  - a. Before converting the temperature from °C to °F we will have to convert it from a reading to a voltage using the following: **tempVolts** = [your temperature variable name] \* **5.0 / 1024.0**; (*Note: if we do not use the decimals, the data type will change to integer.*)
  - b. To convert from the voltage to °C we will also need to perform a conversion using formula in the formulas section. (*Note: if we do not use the decimals, the data type will change to integer.*)
  - c. Now, you can convert the temperature from °C to °F.
  - d. Remember to calculate the percentage of light. (***The maximum light value is 1000.***)
12. Determine whether your LCD output statements belong in the setup() routine or the loop() routine. Place them there.
  - a. Remember to include quotations around any text that is a string.
  - b. An easy way to change the location of the LCD output is by using the command: `lcd.setCursor(CHAR, ROW)`; where CHAR is the number of the character and ROW is the number of the row. Since the Arduino starts counting at 0, when 1 is used for the ROW in the command, you are actually referencing the second row.
13. So that we can read the output on the LCD a delay is typically used, the delay is given to the Arduino in milliseconds. Setting a delay will delay how often the sensors are read, calculated and the results are output. A delay of at least 1 second will make the output readable, but you may want to experiment with this.
14. When complete, verify your code using the Arduino IDE (click the check mark).
15. If your code didn't produce any errors on the bottom, upload the code to the Arduino (click the right pointing arrow) and see if it works. If it produced errors look for missing parenthesis and semicolons.
16. If it does not work or produces an error, hypothesize why it is not working, double check to see if you missed any important steps.

#### Part 4: Extra Credit

If you were successful at having the device display the temperature and percent lighting, see if you can add code to change the color of the screen for a given condition (i.e. turn red when hot). To do this, you will use the modified flow chart you made in Part 2.

The RED LED on the LCD is wired to port or pin 6, the GREEN LED is on port 5 and the BLUE LED is wired to port 3. The LED's are what "backlight" the LCD screen. Outputting a 0 value to the LED will turn the LED on, conversely outputting 255 to an LED will keep the LED off. These outputs are analog outputs that require the `analogWrite(PIN, VALUE);` command.

In programming if statements allow you to tell the microcontroller or computer to do something if a certain condition is true. Hence, they are really if, then do statements and the syntax varies from language to language. But, in Arduino programming, if statements do not use semicolons and look like this:

```
if (CONDITION)
{
  \\ insert what you want to happen here if the condition is true (i.e. turn the RED LED on)
}
```

Can you program the LCD to change color? How would you program the LCD to turn purple?

#### Conclusion

Based on the results, form a conclusion as to whether your hypothesis was supported or rejected and explain.

#### Analysis questions

1. List three ways your code could be improved.
  - a. Are there ways to reduce the number of lines (make it more efficient)?
  - b. Are there ways you could make your code more readable/user friendly?
  - c. What are some other ways your code could be improved?
2. Name one thing that you learned about programming that you did not previously know.
3. What are some weather-related warning messages that infrastructure-to-vehicle systems could send to a driver?
  - a. Name at least one temperature specific warning,
  - b. Name at least one light specific warning.
4. How can infrastructure-to-vehicle communication help to prevent weather-related crashes?
5. What else can sensors detect that might provide useful information to drivers?
  - Name one thing that future iterations of connected vehicles might sense to:
    1. Increase safety
    2. Increase awareness

## **Closure**

In the real world smartphones are capable of providing relevant weather information. However, using these devices while driving is extremely dangerous and they lack the capability to warn you about weather that you will encounter en route to your destination. Real-time weather information systems and infrastructure-to-vehicle communication systems require a large amount of programming to enable communication. Some transportation engineers program these devices, but all of them have a basic understanding of programming and hardware.