C++ Code Snippets

PART I: Inputs for Arduino IDE/Teensy 3.2

John R. Wright, Jr., PhD, CSTM, CLSSGB, CSCE, F.ATMAE
AENG 467, Mobile Robotics
int SharpIR = A0;  // Set variable as integer & tell where connected on board
int val = 0;  // Set variable as integer and set to zero

void setup() {
  pinMode(SharpIR, INPUT);  // Set the pin direction to input
  Serial.begin(9600);  // Establish serial baud rate
  while(!Serial);  // Wait until good serial connection is established
}

void loop() {
  val = analogRead(SharpIR);  // Read value from sensor
  Serial.println(val);  // Print to monitor
  delay(200);  // Wait 200ms
Line Tracking Sensor

//John Wright 2017
//January 18, 2017
//Same code as used with SharpIR

int LineTrackS1 = A0;  // Where sensor is connected on board
int val = 0;  // Set val to zero

void setup() {
  pinMode(LineTrackS1, INPUT);  // Set the pin direction to input
  Serial.begin(9600);  // Establish serial baud rate
  while(!Serial);  // Wait until good serial connection is established
}

void loop() {
  val = analogRead(LineTrackS1);  // Read value from sensor
  Serial.println(val);  // Print to monitor
  delay(200);  // Wait 200ms
}
Line Tracking Sensor

https://youtu.be/UiAZhpYzYKs
Flame Sensor w/LED
(input controlling an output)

//John Wright 2017
//January 18, 2017
//Same code as used with SharpIR

int FlameS1 = A0;                  // Where sensor is connected on board
int val = 0;                       // Set val to zero

void setup() {
  pinMode(FlameS1, INPUT);        // Set the pin direction to input
  pinMode(13, OUTPUT);            // Set the direction of pin 13 to output
  Serial.begin(9600);             // Establish serial baud rate
  while(!Serial);                 // Wait until good serial connection is established
}
Flame Sensor w/LED
(input controlling an output)

```cpp
void loop() {
  val = digitalRead(FlameS1);  // Read value from sensor
  Serial.println(val);  // Print to monitor
  delay(200);  // Wait 200ms
  if (val < 1) {
    digitalWrite(13, HIGH);  // Turns on onboard LED if flame detected
    delay(5000);  // Wait 5 sec so we can see the detection
  }
  else {
    digitalWrite(13, LOW);  // Turns off on-board LED if flame not detected
  }
}
```
Flame Sensor

https://youtu.be/P8fgrlDGHE8
What is a Library?

“In the C++ programming language, the C++ Standard Library is a collection of classes and functions, which are written in the core language and part of the C++ ISO Standard itself.

The C++ Standard Library provides several generic containers, functions to utilize and manipulate these containers, function objects, generic strings and streams (including interactive and file I/O), support for some language features, and functions for everyday tasks such as finding the square root of a number.

What is a Library?

The C++ Standard Library also incorporates 18 headers of the ISO C90 C standard library ending with ".h", but their use is deprecated.

No other headers in the C++ Standard Library end in ".h".

Features of the C++ Standard Library are declared within the std namespace.”

What is a Header File?

Think of both like this (Disclaimer: this is a really high-level analogy ;) ..

- The header is a phone number you can call, while...
- ...the library is the actual person you can reach there!

It's the fundamental difference between "interface" and "implementation"; the interface (header) tells you how to call some functionality (without knowing how it works), while the implementation (library) is the actual functionality.

Note: The concept is so fundamental, because it allows you flexibility: you can have the same header for different libraries (i.e. the functionality is exactly called in the same way), and each library may implement the functionality in a different way. By keeping the same interface, you can replace the libraries without changing your code.

And: you can change the implementation of the library without breaking the calling code!

4 Pin Sonar

4 Pin Sonar


1) Download and install library onto computer under the Arduino Library Folder

2) Link Library in Arduino

   SKETCH, IMPORT LIBRARY
#include <HCSR04.h>

// Code & Library from Patton Robotics
// Must get library file from Patton Robotics and install - point to folder on your computer
// Sketch, Import Library

HCSR04 Echo1(7,8); // New instance of the class
                    // HCSR04(int EchoPin, int TrigPin)

void setup() {
    Serial.begin(9600); // Launch Serial
}
void loop() {
    Echo1.ReadEchoCM(); // Get Data in Centimeters
    delay(10); // Give a chance to establish a new low on the trigger
                // delay likely not needed, I just play it safe
    Echo1.ReadEchoInches(); // Get Data in Inches
    Serial.print("CM = ");
    Serial.println(Echo1.CMs);
    Serial.println(" ");
    delay(500);
}
4 Pin Sonar

CAUTION!!!!

- Pin colors do not reflect + & -
  - Red and Black may mean different pins
  - Be very careful with the wiring!
  - Two pins will be signals (Trig & Echo)
#include <Ping.h> // Library function for Ping Sonar
Ping ping = Ping(0); // Tells us what input the Ping Sonar is wired to

void setup() {
  pinMode(13, OUTPUT); // Sets pin 13 to use as an output for on board LED
  Serial.begin(115200); // Sets baud rate for the serial connection
}
void loop() {
  ping.fire();  // Pulses Ping Sonar
  Serial.print(ping.inches());  // Prints output/result of Ping Sonar to Screen
  Serial.println();  // Sets a return so data scrolls down instead of across the
                     // screen
  delay(100);  // Delays 100ms

  if (ping.inches() < 10) {
    digitalWrite(13, HIGH);  // Turns on on-board LED if object less than 10 inches away
  }
  else {
    digitalWrite(13, LOW);  // Turns off the on-board LED if object >= 10 inches away
  }
}
Good Luck! Time to “Code Hard!”

https://www.youtube.com/watch?v=b-CroEWwaTk