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
Welcome to Teensy 3.2
32 Bit Arduino-Compatible Microcontroller

To begin using Teensy, please visit the website & click Getting Started.
www.pjrc.com/teensy

Digital Pins
- digitalRead
- digitalWrite
- pinMode

PWM Pins
- analogWrite
- analogWriteRes

Analog Pins
- analogRead
- analogReference
- analogReadRes

Touch Sense Pins
- touchRead

Serial Ports
- Serial1
- Serial2
- Serial3

I²C Port
- Wire Library

SPI Port
- SPI Library

I/O
- Vin (3.6 to 6.0 volts)
- 3.3V (250 mA max)
- AGND
- GND
- 3.3V
- VBat

All digital pins have interrupt capability.

Teensy 3.2 Pinout
//John Wright 2017
//January 18, 2017

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

Gravity Line Tracking Sensor for Arduino

Product Code: RB-Dfr-40 by DFRobot

- Line tracking sensor to guide robot by telling white and black via TTL signal
- Includes high quality Opto interrupter for improved sensitivity
- Supply voltage: 3.3V to 5V
- Interface: Digital

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
}
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

[Image of Flame Sensor]

https://youtu.be/P8fgrlDGHE8
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.print("Inches = ");
    Serial.println(Echo1.Inches);
    Serial.println(" ");
    delay(500);
}
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