



Distance Gun

Instructor Guide

Level 3

1. Explanation For Loop

Instructor guide: Explain the components of `for(int i = 0; i < number; i++){}` and their significance.

```
for(int i = 0; i < number; i++){
```

```
    // Code 1
```

```
}
```

 is a statement that allows the code inside to be executed a specific number of times.

Explain the following:

- **Counter** `int i = 0;` This statement tells the program that we are tracking a variable `i` that starts at 0.
- **Condition** `i < number;` This statement has the program check if the variable `i` is less than the `number` that was specified. Every time the loop runs, the program checks this condition.
- **Increment** `i++` This statement increments the variable `i` by a certain amount. In this case every time the loop runs we increase `i` by 1.

2. Complete Efficient Code Challenge:

Student Hint: Make sure only values that are not 0 are calculated into the average. Also make sure to check the average reading is equal to 0 at the end instead of an individual reading.

Instructor guide: This challenge will teach the builder to use new concepts to make their readings more accurate and efficient. A for loop will be used to take a number of readings and calculate the average. The average will be calculated by adding all the values that are 0 and counting all the values that are not 0. By dividing the first value by the second the average reading will be found. Additionally, the builder will check if the final value equals 0. If that happens to be the case, "Object Not Found" will be displayed on the LCD screen.

Notes:

CODE The following code is a sample solution for Level 3.

```
#include <LiquidCrystal_I2C.h>
#include <Wire.h>
#include <UltraDistSensor.h>

UltraDistSensor scanner;
LiquidCrystal_I2C lcd(ADDRESS, 20, 4);
float reading;
```

```
void setup() {  
  
    lcd.init();  
    lcd.backlight();  
    scanner.attach(9, 10);  
  
}  
  
void loop() {  
  
    lcd.clear();  
    lcd.setCursor(0, 0);  
    lcd.print("Distance");  
  
    float distances = 0;  
    float numTimes = 0;  
  
    for(int i = 0; i < 10; i++){  
        reading = scanner.distanceInInch();  
        if(distance != 0.0){  
            distances += distance;  
            numTimes++;  
        }  
    }  
  
    int averageDistance = distances/numTimes;  
  
    if(averageDistance != 0.0){  
        lcd.print(averageDistance);  
    } else {  
        lcd.setCursor(0, 1);  
        lcd.print("Object Not Found");  
    }  
  
    delay(500);  
  
}
```

Notes:

3. Add Button Module to Distance Gun

Student Hint: Use the build legend on the back of the map

Instructor guide: Connect the pieces together using the build legend on the back of the map. connect the module to the specified signal pin. Be sure to connect the power pins of the Button Module to 3.3V and GND. For further explanation please see the build video for this kit located in your instructor portal.

IMPORTANT: THE VCC PIN ON THE LCD SHOULD BE CONNECTED TO 3.3V and not 5V.

Notes:

4. Upload Test Code: Button Module

Instructor guide: Once the test code is uploaded it will print whether it detected a reading or not in the serial monitor. It will say print two different statements: “Pressed” or “Not Pressed” depending on the reading detected. If these readings do not match up with the physical state of the button press then the button is not wired up properly. If this happens to be the case then rewire the button module and make sure the power and signal cables are connected properly.

How to Perform the Test: The instructor will upload to the code through the USB cable. The user will press the button on and off at different speeds and delays to test if the readings on the serial monitor match up to it. To open the serial monitor click the magnifying glass in the upper right hand corner of the Arduino IDE.

IMPORTANT: MAKE SURE THE BUTTON PIN VARIABLE IN THE CODE IS THE SAME AS THE PIN THE SIGNAL WIRE IS CONNECTED TO.

CODE Copy and paste or type the following into the Arduino IDE and upload to the Arduino Board. You may also visit: www.BuildARobotWorkshop.com/-----

```
int buttonpin = 2;
int val = 0;

void setup() {

    pinMode(buttonpin, INPUT);
    Serial.begin(9600);

}
```

```
void loop() {  
  
    val = digitalRead(buttonpin);  
  
    if(val == HIGH){  
        Serial.println("Pressed");  
    } else {  
        Serial.println("Not Pressed");  
    }  
  
    delay(1000);  
  
}
```

Notes:

POSSIBLE PROBLEMS AND ANSWERS

Problem: Readings are not accurate

Possible Solutions:

1. Values that are 0 are being added to the variable for the sum of the distances measured.
2. Values that are 0 are being added to the variable for the number of readings.

Problem: Button is not detected presses accurately

Possible Solutions:

1. The VCC pin is connected to 5V and not 3.3V
2. The GND pin is not connected to GND properly.
3. The pin that the signal wire connects to, does not match the button pin in the test code.