:Author: Ayesha Siddiqua, Team Member
:Date: 11/23/2020

= Project: {Arduino Sample Code}

The project seeks to create a wall-following robot using 4DC motors, a sensor, and other hardware components in the Arduino kit. Since it is able to follow a wall, it is also able to detect obstacles.

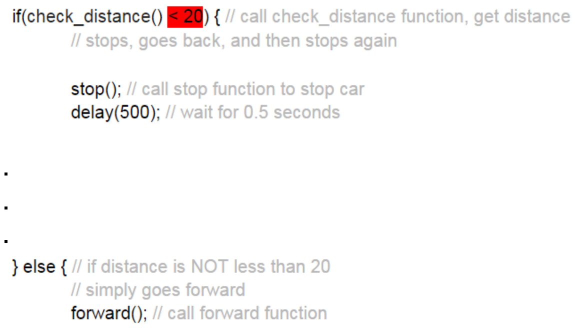
Programming Scheme:

Robot moves in a forward direction. If the sensor detects an object at least 20 in. away, the robot reverses until no obstacles remain 20 in. ahead, and then turns and moves in the direction of no obstacle.

== Code Examples: Examples of usage

The code utilizes 2 types of control statements: if-else statements and while loops.

1. if-else statements: if the condition in the IF part is true, that statements inside it's braces({}) are run. Else, the statements in the ELSE part are run. Both are not run.
    - ➢ example from code: If the distance is less than 20, then statements in if curly braces({}) will run, else the bottom else statements will run.

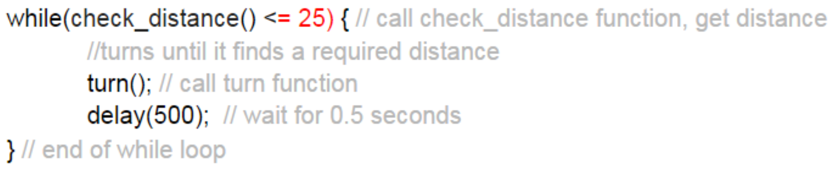


```
if(check_distance() < 20) { // call check_distance function, get distance
        // stops, goes back, and then stops again

        stop(); // call stop function to stop car
        delay(500); // wait for 0.5 seconds

.
.
.
 } else { // if distance is NOT less than 20
        // simply goes forward
        forward(); // call forward function
```

2. while loops will keep iterating until the condition inside the parenthesis is false. Then, the code will continue.
    - ➢ example from code: The below 2 statements will loop as long as no obstacles in view of 25 inches ahead.



```
while(check_distance() <= 25) { // call check_distance function, get distance
        //turns until it finds a required distance
        turn(); // call turn function
        delay(500);  // wait for 0.5 seconds
} // end of while loop
```

The rest of the code can easily be understood with the use of comments for each line.

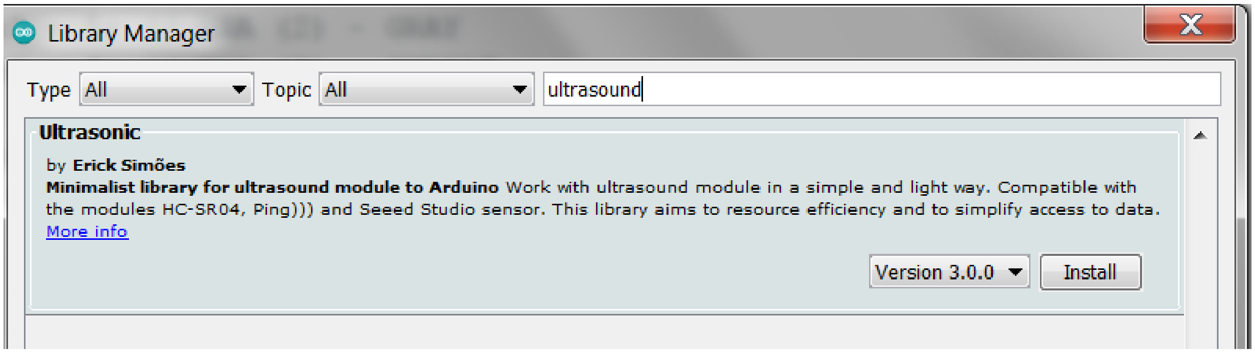
== Step 1: Installation

1. Download Arduino IDE - web or downloadable version
2. Make sure to download all drivers
3. Follow the directions as shown on screen.

== Step 2: Assemble the circuit
Assemble the circuit following the instructions provided in manual

== Step 3: Load the code
Upload the code contained below in this sketch on to your board:

1. Go to Arduino IDE > File > New
2. Copy paste below code onto the new File
3. A library will need to be imported for correct functioning of the ultrasonic sensor. Below are directions to correctly download the file.
    - Use this reference from Arduino site for additional help
    1) On the Arduino IDE menu, click on "Sketch" menu
    2) Next select “Include Library”
    3) Next select “Manage Libraries”
    4) In the search bar, type “ultrasound”
    5) Hit the “Install” button of the below library (latest version)



Type All Topic All ultrasound

Ultrasonic

by **Erick Simões**
Minimalist library for ultrasound module to Arduino Work with ultrasound module in a simple and light way. Compatible with the modules HC-SR04, Ping))) and Seeed Studio sensor. This library aims to resource efficiency and to simplify access to data.

    6) Now, Verify and Upload the code to your Arduino board!
    7) Your car is ready to function

Code below pasted for reference

```
#include <Ultrasonic.h> //importing a library for Arduino-Sensor compatibility
Ultrasonic ultrasonic(13, 12); // trig = pin 12; echo = pin 13

int in1 = 6; //YELLOW - motor 1 forward
int in2 = 7; //BROWN - motor 1 reverse
int in3 = 8; //ORANGE - motor 2 forward
int in4 = 9; //WHITE - motor 2 reverse
int in5 = 2; //GRAY - motor 3 forward
int in6 = 3; //PURPLE - motor 3 reverse
int in8 = 4; //GREEN - motor 4 forward
int in7 = 5; //BLUE - motor 4 reverse

void setup() { //start of setup function
        Serial.begin(9600); //begins serial monitor
        pinMode(in1, OUTPUT); // set in1 as OUTPUT pin
        pinMode(in2, OUTPUT); // set in2 as OUTPUT pin
        pinMode(in3, OUTPUT); // set in3 as OUTPUT pin
        pinMode(in4, OUTPUT); // set in4 as OUTPUT pin
        pinMode(in5, OUTPUT); // set in5 as OUTPUT pin
        pinMode(in6, OUTPUT); // set in6 as OUTPUT pin
        pinMode(in7, OUTPUT); // set in7 as OUTPUT pin
        pinMode(in8, OUTPUT); // set in8 as OUTPUT pin
} // end of setup function

/*loop function will go on repeatedly until power source depletes*/
void loop() { // start of loop function
        /*if part will execute/run only if the condition in parentheses is true (i.e. distance is < 20)*/
        if(check_distance() < 20) { // call check_distance function, get distance
                // stops, goes back, and then stops again

                stop(); // call stop function to stop car
                delay(500); // wait for 0.5 seconds

                reverse(); // call reverse function to reverse car
                delay(150); // wait for 0.15 seconds

                stop(); // call stop function to stop car
                 delay(500); // wait for 0.5 seconds

                /*while loop will keep looping until condition in parenthesis is false*/
                while(check_distance() <= 25) { // call check_distance function, get distance
                        //turns until it finds a required distance
                        turn(); // call turn function
                        delay(500);  // wait for 0.5 seconds
                } // end of while loop

                stop(); // call stop function to stop car
                delay(1000); // wait for 1 second
        } else { // if distance is NOT less than 20
                // simply goes forward
                forward(); // call forward function
        } // end of else
} // end of loop function
```

```
int check_distance() { //Checking for distance and returning
        int distance_in_inches = ultrasonic.read(); // Get in. from sensor’s read function(library)
        return distance_in_inches; // return inches to caller
} // end check_distance function

int stop() { // start of stop function
        digitalWrite(in1, LOW); // motor 1 no power to forward
        digitalWrite(in2, LOW); // motor 1 no power to reverse
        digitalWrite(in3, LOW); // motor 2 no power to forward
        digitalWrite(in4, LOW); // motor 2 no power to reverse
        digitalWrite(in5, LOW); // motor 3 no power to forward
        digitalWrite(in6, LOW); // motor 3 no power to reverse
        digitalWrite(in7, LOW); // motor 4 no power to forward
        digitalWrite(in8, LOW); // motor 4 no power to reverse
} // end stop function

int reverse() { // start of reverse function
        digitalWrite(in1, LOW); // motor 1 forward “off”
        digitalWrite(in2, HIGH); // motor 1 reverse “on”
        digitalWrite(in3, LOW); // motor 2 forward “off”
        digitalWrite(in4, HIGH); // motor 2 reverse “on”
        digitalWrite(in5, LOW); // motor 3 forward “off”
        digitalWrite(in6, HIGH); // motor 3 reverse “on”
        digitalWrite(in7, LOW); // motor 4 forward “off”
        digitalWrite(in8, HIGH); // motor 4 reverse “on”
} // end reverse function

int forward() { //start of forward function
        digitalWrite(in1, HIGH); // motor 1 forward “on”
        digitalWrite(in2, LOW); // motor 1 reverse “off”
        digitalWrite(in3, HIGH); // motor 2 forward “on”
        digitalWrite(in4, LOW); // motor 2 reverse “off”
        digitalWrite(in5, HIGH); // motor 3 forward “on”
        digitalWrite(in6, LOW); // motor 3 reverse “off”
        digitalWrite(in7, HIGH); // motor 4 forward “on”
        digitalWrite(in8, LOW); // motor 4 reverse “off”
} // end forward function

int turn() { // start of turn function
        digitalWrite(in1, HIGH); // motor 1 forward HIGH
        digitalWrite(in2, LOW); // motor 1 reverse LOW
        digitalWrite(in3, LOW); // motor 2 forward LOW
        digitalWrite(in4, HIGH); // motor 2 reverse HIGH
        digitalWrite(in5, LOW); // motor 3 forward LOW
        digitalWrite(in6, HIGH); // motor 3 reverse HIGH
        digitalWrite(in7, HIGH); // motor 4 forward HIGH
        digitalWrite(in8, LOW); // motor 4 reverse LOW
} // end turning function
```