

Calibrate Movement

Time required: 60 minutes

Please read all the directions carefully before beginning the assignment.

1. Comment your code as shown in the tutorials and other code examples.
2. Follow all directions carefully and accurately.
3. Think of the directions as minimum requirements.

Understanding

Demonstrate understanding of:

variables, constants, if else if

Knowledge Points

We are going to calibrate the robot to go a certain distance and turn accurately. The battery charge can make a difference, make sure the batteries are charged. The surface the robot is traveling on will make a difference. You may want to keep track of the settings for each surface. The sample program settings are for a smooth surface.

NOTE: The mBot is not an accurate robot. As the batteries discharge and the conditions change, it will behave differently. The only things we can change is power and time. Just try to get close.

This program will allow us to calibrate our mBot for the following:

1. Driving straight by adjust the comp variable.
2. Distance by adjusting the time variable to drive 48".
3. Turns by adjusting turnTime360 to have the mBot start and start in approximately the same orientation while making a square.

Requirements

1. Use your mBlock program settings as a starting point.
2. **Create** a program that has a block that move forward 48", does a square turning to the right, and another square turning to the left.
3. Vary the compensation and time in milliseconds to make an exact square.

4. When doing the square, the robot should end where it started.

	Power	90°
Left	127 * 1.1	520
Right	127	

- Use the arrow keys on the remote to drive forward and start the squares.
5. Turning while moving changes the turning rate, you may have to recalibrate your 360TurnTime.

Tutorial Assignment

1. Save the sketch as **CalibrateMovement**
2. Complete and test the program as shown.

```

1  /**
2   @file    CalibrateSquare.ino
3   @author  William A Loring
4   @version v1.0.0
5   @Revised 10/11/2021  Created: 12/10/16
6   @Description: Calibrate distance, driving straight,
7   and turn speed by turning right and left squares
8  */
9  #include <MeMCore.h>  // Include mBot library
10 MeIR ir;              // Setup IR remote object
11 // Create motor control objects
12 MeDCMotor MotorL(M1); // MotorL is Left Motor
13 MeDCMotor MotorR(M2); // MotorR is Right Motor
14 const int POWER = 127; // Base power setting
15 const float COMP = 1.0; // Compensation to make the robot drive straight
16 // Increase COMP .02 at a time if your robot drives to the left
17 // Decrease COMP .02 at a time if your robot drives to the right
18 int lPower = round(POWER * COMP); // Apply compensation to left motor
19 int rPower = POWER;
20 const int DRIVE_TIME = 5400; // Time in milliseconds it takes to go 48"
21 // Increase time if the robot comes up short, decrease if it goes too far
22 const int TURN_TIME = 530; // Time in milliseconds it takes to turn 90 degrees
23 // Increase by 20 at a time if the robots 90 degress is short
24 // Decrease by 20 at a time if the robots 90 degress is too long
25
26 void setup() {
27   ir.begin(); // Start listening to the ir remote
28 }
29
30 void loop() {
31   // Wait until remote button is pressed
32   if (ir.keyPressed(IR_BUTTON_UP)) {
33     forward48();
34   }
35   else if (ir.keyPressed(IR_BUTTON_LEFT)) {
36     leftSquare();
37   }
38   else if (ir.keyPressed(IR_BUTTON_RIGHT)) {
39     rightSquare();
40   }
41 }
42

```

```

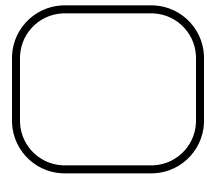
43 // Drive forward 48 inches
44 void forward48() {
45     MotorL.run(-lPower); // MotorL (Left) forward is -negative
46     MotorR.run(+rPower); // MotorR (Right) forward is +positive
47     delay(DRIVE_TIME); // Drive for the amount of time it takes
48     stop();
49 }
50
51 // Right 12 inch square
52 void rightSquare() {
53     // The for loop repeats four times
54     for (int x = 0; x < 4; x++) {
55         forward(); // Drive forward 12"
56         right(); // Turn to the right 90 degrees
57     }
58 }
59
60 // Left 12 inch square
61 void leftSquare() {
62     // The for loop repeats four times
63     for (int x = 0; x < 4; x++) {
64         forward(); // Drive forward 12"
65         left(); // Turn to the left 90 degrees
66     }
67 }
68 // Left turn 90 degrees
69 void left() {
70     MotorL.run(+lPower); // MotorL (Left) backward is +positive
71     MotorR.run(+rPower); // MotorR (Right) forward is +positive
72     delay(TURN_TIME); // Time to turn 90 degrees
73     stop();
74 }
75
76 // Right turn 90 degrees
77 void right() {
78     MotorL.run(-lPower); // MotorL (Left) forward is -negative
79     MotorR.run(-rPower); // MotorR (Right) backward is -negative
80     delay(TURN_TIME); // Time to turn 90 degrees
81     stop();
82 }
83

```

```
84 // Drive forward 12 inches
85 void forward() {
86     MotorL.run(-lPower); // MotorL (Left) forward is -negative
87     MotorR.run(+rPower); // MotorR (Right) forward is +positive
88     delay(DRIVE_TIME / 4); // Time to go 12"
89     stop();
90 }
91
92 // Stop
93 void stop() {
94     MotorL.stop(); // Stop MotorL
95     MotorR.stop(); // Stop MotorR
96 }
```

Extra Credit Challenge Assignment

1. Add a **new** function to calibrate a right car turn.
2. Create a new turning function to trace a right handed square with rounded corners. Combine a differential turn with a 12" straight line and loop it four times.
3. **Comment** your challenge code.



Assignment Submission

- **All students** → Attach finished programs to the assignment in Blackboard.
- **In class assignment submission** → Demonstrate in person.
- **Online submission** → A link to a YouTube video recording showing the assignment placed in the submission area in BlackBoard.