Movement with Functions

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:

functions

We will use functions to modularize our code.

Requirements

Complete and successfully run the program as shown.

Tutorial Assignment

1. Start the Arduino IDE. Create a new sketch called **MovementWFunctions**.
2. Complete and test the program as pictured with the requirements listed.

```
/**
   @file    MovementWFunctions.ino
   @author  William A Loring
   @version V1.0.0
   @date revised 10/06/2018  created: 03/03/17
   @Description: mBot movement with functions
*/
#include <MeMCore.h>          // Include mBot library
MeIR ir;                      // Setup IR Remote object
// Create motor control objects
MeDCMotor MotorL(M1);         // MotorL is Left Motor
MeDCMotor MotorR(M2);         // MotorR is Right Motor
const int MOTOR_POWER = 127;  // Base power setting
const int TIME = 1000;        // Time in milliseconds

void setup() {
  ir.begin(); // Start listening to the ir
}

void loop() {
  // Wait until a remote button is pressed
  if (ir.keyPressed(IR_BUTTON_UP)) {
    Move();
  }
}

```

```
// A function calling other functions
void Move() {
  forward(MOTOR_POWER);
  delay(TIME);
  reverse(MOTOR_POWER);
  delay(TIME);
  forward(MOTOR_POWER);
  delay(TIME);
  leftTurn(MOTOR_POWER);
  delay(TIME);
  reverse(MOTOR_POWER);
  delay(TIME);
  rightTurn(MOTOR_POWER);
  delay(TIME);
  leftTurn(MOTOR_POWER);
  delay(TIME);
  reverse(MOTOR_POWER);
  delay(TIME);
  leftMotor(MOTOR_POWER);
  delay(TIME);
  rightMotor(MOTOR_POWER);
  delay(TIME);
  stop();
}

```

```
// Forward movement function with power argument
void forward(int power) {
  MotorL.run(-power);   // MotorL (Left)  forward is -negative
  MotorR.run(+power);   // MotorR (Right) forward is +positive
}

// Reverse movement function with power argument
void reverse(int power) {
  MotorL.run(+power);   // MotorL (Left)  reverse is +positive
  MotorR.run(-power);   // MotorR (Right) reverse is -negative
}

// Left turn movement function with power argument
void leftTurn(int power) {
  MotorL.run(+power);   // MotorL (Left) backward is +positive
  MotorR.run(+power);   // MotorR (Right) forward is +positive
}

// Right turn movement function with power argument
void rightTurn(int power) {
  MotorL.run(-power);   // MotorL (Left) forward is -negative
  MotorR.run(-power);   // MotorR (Right) backward is -negative
}

// Control just the left motor
void leftMotor(int power) {
  MotorL.run(+power);
}

// Control just the right motor
void rightMotor(int power) {
  MotorR.run(-power);
}

// Stop function
void stop() {
  MotorL.stop();  // Stop MotorL
  MotorR.stop();  // Stop MotorR
}
```

Assignment

Start with your tutorial project and add the following.

1. Add another function with a different remote button that calls a different combination of movements.

Assignment Submission

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