

Simple Remote Control

Time required: 60 minutes

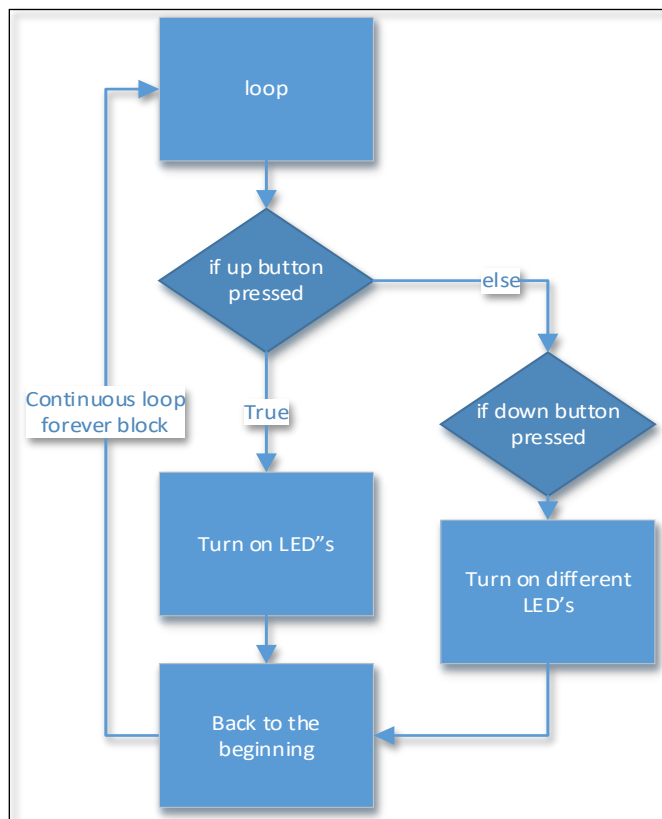
Please read all the directions carefully before beginning the assignment.

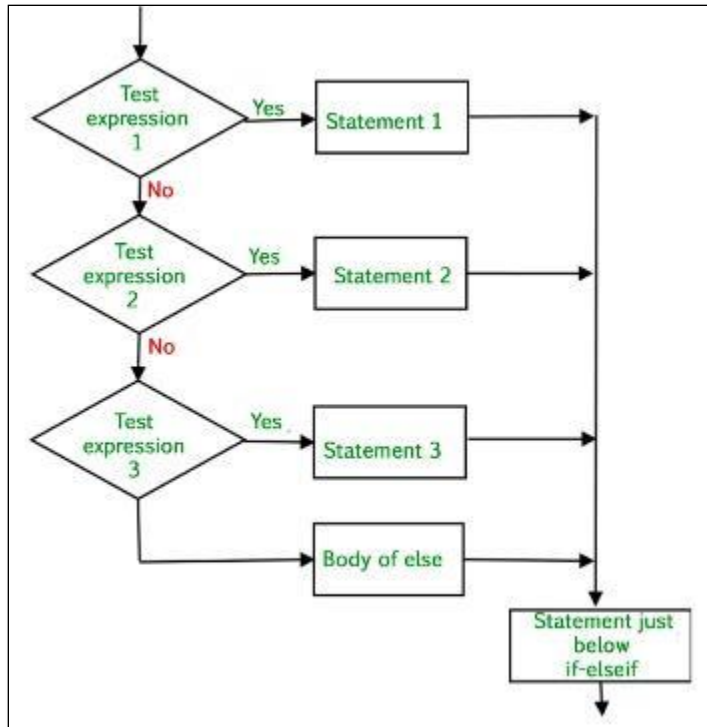
- Comment your code as shown in the tutorials and other code examples.
- Follow all directions carefully and accurately.
- Think of the directions as minimum requirements.

Knowledge Points

If Then Else

If then else extends our decision making. If then else is mutually exclusive. This means that only one of the choices or conditions can be true. You can stack multiple if else statements together.





Debouncing

When you press a button on the remote, it is impossible to press it once. It makes contact several times, bouncing off the contact. It may make contact on one side, then the other, then it settles down. The Arduino scans for each ir code so fast that it can mistake one code for another. Debouncing is putting a tiny delay in between scanning for the ir codes to ensure smooth operation.

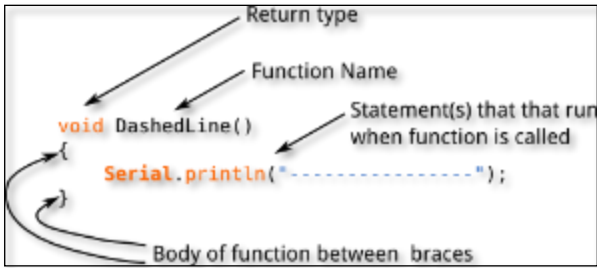
Functions

As our code gets longer and more complex, it can get difficult to follow. Functions allow for reusable and modular code. A function is a code block. It wraps up everything needed to provide a service to the program. You can easily reuse the code in another sketch or the same sketch.

We have used pre written functions, such as **led.setColorAt();** and **delay(500);**. We will start writing our own.

Please go to the following web site to learn more about functions.

<https://startingelectronics.org/software/arduino/learn-to-program-course/15-functions/>



Sample function

```

18 // Loop forever function
19 void loop() {
20     // Call function
21     simpleFunction();
22 }
23
24 void simpleFunction() {
25     // Put the code here
26 }
  
```

Requirements

The robot will move in the direction of the arrow keys on the remote, then stop when the keys are released.

IR Remote Constants for mBot

The following table is a reference for the constants for reading the IR remote.

IR_BUTTON_A	IR_BUTTON_SETTING	IR_BUTTON_0
IR_BUTTON_B	IR_BUTTON_LEFT	IR_BUTTON_1
IR_BUTTON_C	IR_BUTTON_RIGHT	IR_BUTTON_2
IR_BUTTON_D	IR_BUTTON_UP	IR_BUTTON_3
IR_BUTTON_E	IR_BUTTON_DOWN	IR_BUTTON_4
IR_BUTTON_F		IR_BUTTON_5
		IR_BUTTON_6
		IR_BUTTON_7
		IR_BUTTON_8
		IR_BUTTON_9

Tutorial Assignment

1. Start the Arduino IDE. Save the sketch as **SimpleRemoteControl**.
2. Include **notes.h** in the **SimpleRemoteControl** folder.
3. Complete and test the program as pictured with the requirements listed.

```
1  /*-----
2   File      SimpleRemoteControl.ino
3   Author    William A Loring
4   Version   V1.0.0
5   Date revised: 02/05/20 created: 12/9/17
6   Description: Simple remote control program
7   -----
8   #include <MeMCore.h> // Include mBot library
9   #include "notes.h"    // Include notes.h to easily play notes
10  MeIR ir;              // Create ir remote object
11  MeBuzzer buzzer;      // Setup buzzer object
12  MeRGBLed led(0, 30);  // Setup the onboard LED object
13  MeDCMotor MotorL(M1); // MotorL object is Left Motor
14  MeDCMotor MotorR(M2); // MotorR object is Right Motor
15  const int MOTOR_POWER = 127; // Base power setting of 50%
16  const int DEBOUNCE = 20;    // Debounce delay for smooth IR Operation
17
18  void setup() { // Setup function runs once
19    ir.begin();   // Start listening to the remote
20    led.setpin(13); // Set pin for led access
21    initialize(); // Call startup function, mBot announces it is ready!
22  }
23
24  void loop() { // Loop forever
25    remote();
26  }
```

```

28 □ /*-----
29   Move the mBot in the direction of the IR remote keys
30   -----*/
31 □ void remote() {
32   // Determine which remote button was pressed
33 □ if (ir.keyPressed(IR_BUTTON_UP)) {
34     delay(DEBOUNCE);           // Delay or debounce for smooth IR operation
35     MotorL.run(-MOTOR_POWER);   // MotorL (Left) forward is -negative
36     MotorR.run(+MOTOR_POWER);   // MotorR (Right) forward is +positive
37   } else if (ir.keyPressed(IR_BUTTON_DOWN)) {
38     delay(DEBOUNCE);           // Delay or debounce for smooth IR operation
39     MotorL.run(+MOTOR_POWER);   // MotorL (Left) reverse is +positive
40     MotorR.run(-MOTOR_POWER);   // MotorR (Right) reverse is -negative
41   } else if (ir.keyPressed(IR_BUTTON_LEFT)) {
42     delay(DEBOUNCE);           // Delay or debounce for smooth IR operation
43     MotorL.run(+MOTOR_POWER);   // MotorL (Left) reverse is +positive
44     MotorR.run(+MOTOR_POWER);   // MotorR (Right) forward is +positive
45   } else if (ir.keyPressed(IR_BUTTON_RIGHT)) {
46     delay(DEBOUNCE);           // Delay or debounce for smooth IR operation
47     MotorL.run(-MOTOR_POWER);   // MotorL (Left) forward is -negative
48     MotorR.run(-MOTOR_POWER);   // MotorR (Right) reverse is -negative
49   } else {
50     MotorL.stop(); // Stop MotorL
51     MotorR.stop(); // Stop MotorR
52   }
53 }

```

```

55 ▢ /*-----
56   Announce to the world that the mighty mBot is ready to go!
57   -----*/
58 ▢ void initialize() {
59   // Play initialization notes and lights to announce mBot is ready
60   delay(200);           // Debounce startup sound
61   led.setColor(40, 0, 0); // Set both LED to Red
62   led.show();           // Use .show() to make new color take effect.
63   playNote(noteC4, HN);
64   led.setColor(0, 40, 0); // Set both LED to Green
65   led.show();           // Use .show() to make new color take effect.
66   playNote(noteD4, HN);
67   led.setColor(0, 0, 40); // Set both LED to Blue
68   led.show();           // Use .show() to make new color take effect.
69   playNote(noteE4, HN);
70   led.setColor(0, 0, 0); // Set both LED off
71   led.show();           // Use .show() to make new color take effect.
72 }
73
74 ▢ /*-----
75   This custom function takes two parameters, note and duration to make playing songs easier.
76   Each of the notes have been #defined in the notes.h file. The notes are broken down by
77   octave and sharp (s) / flat (b).
78   -----*/
79 ▢ void playNote(int note, int duration) {
80   buzzer.tone(note, duration);
81 }

```

Assignment

Start with your tutorial project and add the following.

- Use LED's to indicate direction and movement.

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.