Introduction to the LEGO EV3 Brick

Connecting the Brick

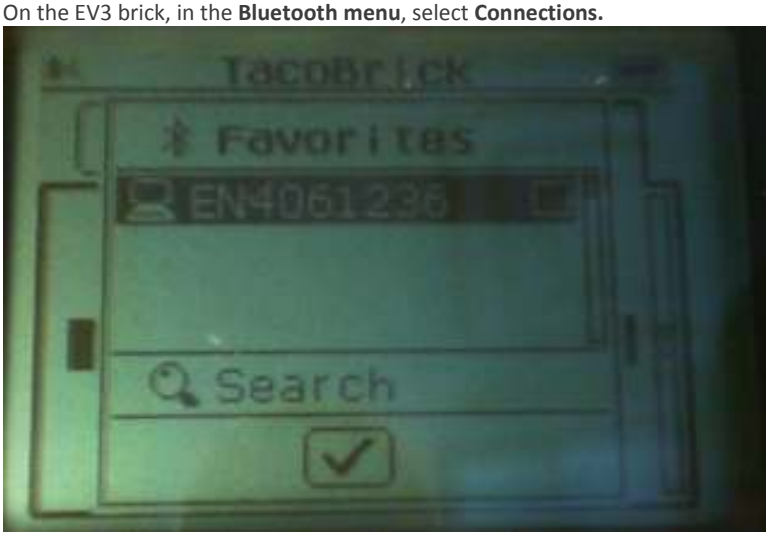
We're going to be communicating with the EV3 Brick over a wireless Bluetooth connection (NOT over the USB cable). This communication requires three components to work together: 1) the LEGO EV3 brick, 2) the PC with its Bluetooth module, 3) the MATLAB application.

Go through the following steps at the beginning of each class:

1. Turn the EV3 brick on, and verify that the small Bluetooth icon is visible in the upper-left corner of the LCD. The icon looks like a capital-B composed of two triangles.
2. If the Bluetooth symbol is not visible, press the right arrow on the EV3 brick three times. Then, select **Bluetooth,** you'll be taken to this menu:

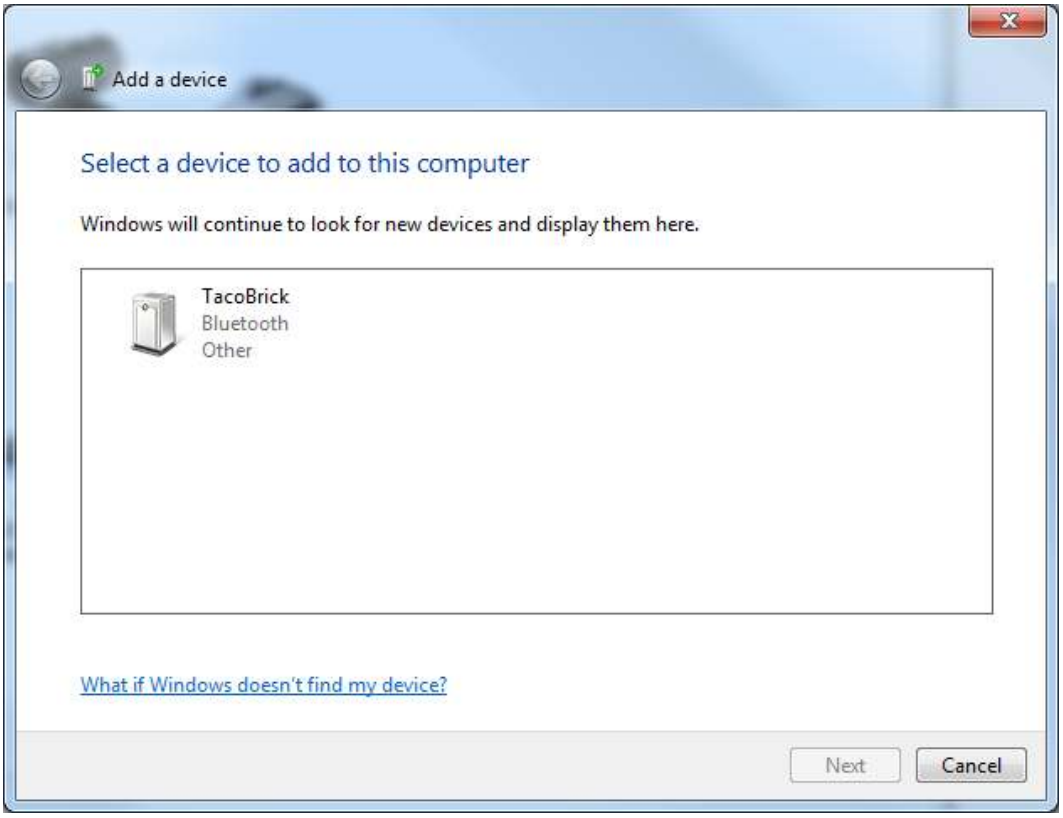
Make sure both "Visibility and "Bluetooth are checked.

3. On the EV3 brick, in the **Bluetooth menu**, select **Connections.**



Select each of the contacts and delete them until there are none left.

4. In Windows, open **Control Panel** > **Devices and Printers** to check whether the EV3 brick (by name) has already been added to Devices.
5. If the EV3 brick is not among the devices, click **Add a device**.
6. In the **Add a device** wizard that opens, select your EV3 brick, and click **Next.**



7. The LCD on the EV3 brick will ask if you want to connect. Select the check mark, and then the EV3 will display the default passkey, 1234, promptly press the center button on the EV3 brick.
8. Enter the same passkey, 1234, at the prompt in Windows.
9. When you see confirmation that the device has been added, click **Close**.
10. **IMPORTANT**: **Restart the EV3 brick after adding the device.**

After the Brick has been connected over Bluetooth, you can execute the following command to connect it in MATLAB:

```
brick = ConnectBrick(brickname);
```

Where brickname is the name of your EV3 Brick in single quotes. You can find this name at the top of the screen on your EV3 brick.

For example, my brick is "TacoBrick" so I would execute:

```
brick = ConnectBrick('TacoBrick');
```

This setup process needs to be done at the beginning of each lab session, or if you run into problems with your brick. Do not run this command multiple times in one session!

Testing the Brick Connection

Start a new script file in MATLAB, and copy the following code into the script. Save it and execute it.

```
%% Example 1: Play Tone and Get Battery Level

% Play tone with frequency 800Hz and duration of 500ms.
brick.playTone(100, 800, 500);

% Get current battery level.
v = brick.GetBattVoltage()
```

The Brick should beep and a number should appear in the MATLAB command window.

Also note that comments (in green) are always preceded by a "%" symbol.

Create ConnectToEV3 Script

To make it easier to connect each lab day, you should create a MATLAB script file named **ConnectToEV3.m**
The script should:

- Issue commands to connect to your team's EV3 Brick.
- Test the connection by instructing the LEGO EV3 brick to emit a tone and to display the battery's voltage level.

Switches

You will need at least one EV3 switch sensor (pictured below). Connect the Switch sensor to **Port 1** of the EV3.

To read the state of the switch sensor try the following in the Command Window:

```
brick.TouchPressed(1)
```

Now, hold down the orange button on the switch sensor, and while holding it down run the same **brick.TouchPressed(1)** command again. What output did you get this time?

Start a new script file named **Beeper.m**. Be sure that the switch sensor is plugged into **Port 1.** Then type the following script commands and run it:

```
display('Push the button.')

while brick.TouchPressed(1) == 0
  brick.playTone(100, 300, 500);
  pause(0.75);
end

display('Done!')
```

Make sure you understand how the above program works.

Start a new script file named **BeepSwitch**. Write a program that does the following when run:

1. Display the message 'PUSH button to start the tone.'
2. Wait for the switch to be pressed.
3. When pressed, play a tone, and display the message 'RELEASE button to turn tone OFF.' The tone should sound like it's playing continuously.
4. Wait for the switch to be released.
5. When released, the tone should stop playing.

Music Tempo

Start a new script, called **MusicTempo.m**

Connect 1 Switch to your EV3 Brick.

Write a program that does the following when run:

1. Wait for the switch to be pressed.
2. When pressed, measure the time that the switch is held down.
3. When the switch is released, the EV3 should play a short song, using the switch-time as a base frequency for the notes.
    a. For Example, if you press the switch for a long time, the song should play slowly, with the duration of each note lengthened. If you press and release the switch very quickly, the song should play quickly, with each note being short.

Motors

You will need two EV3 Motor actuators (pictured below). Connect one motor to **Port A** and the other motor to **Port B** of the EV3.

Once you have the motors connected, execute the following commands ***one at a time*** in the Command Window:

```
brick.MoveMotor('A', 50);
brick.StopMotor('A');
```

Now, execute these following commands ***one at a time*** in the Command Window:

```
brick.MoveMotor('B', 100);
brick.StopMotor('B');
```

We can control motors by changing the parameters that we send to the MoveMotor function. The first parameter ('A' or 'B' in these examples) is the port that connects the motor we want to control. The second parameter is a number that represents the speed of the motor. This number can be anything between -100 and 100.

Start a new script file named **MotorTest**. Be sure that the motors are plugged into **Ports A** & **B**. Then type the following script commands and run it:

```
brick.MoveMotor('A', 50);
pause(5); % pause the script for 5 second
brick.MoveMotor('A', -50);
pause(5); % pause the script for 5 second
brick.StopMotor('A');
```

You should read more about how to use the **StopMotor** command in the online documentation.

In a new script called **TwoMotors**, write a program that does the following when run:

1. Both motors start running at the same time and at 100 power, but in opposite directions for 5 seconds
2. Then, both motors stop for 1 second
3. Then motor A starts running at 50 power
4. 3 seconds later motor B starts running at 50 power
5. 5 seconds after motor B starts running, both motors stop.

Precise Angle Motor Control

Make sure that the motors are still connected to **Ports A** & **B**, then execute the following commands ***one at a time*** in the Command Window:

```
brick.ResetMotorAngle('A');

brick.MoveMotorAngleRel('A', 25, 90, 'Brake');
brick.MoveMotorAngleRel('A', 25, 90, 'Brake');
brick.MoveMotorAngleRel('A', 25, 90, 'Brake');
brick.MoveMotorAngleRel('A', 25, 90, 'Brake');

brick.MoveMotorAngleRel('A', 25, 180, 'Brake');
brick.MoveMotorAngleRel('A', 25, 180, 'Brake');

brick.MoveMotorAngleRel('A', 25, 360, 'Brake');
brick.MoveMotorAngleRel('A', 25, 360, 'Brake');
```

You should read more about the **MoveMotorAngleRel** function in the online documentation.

Note that if you try to move the motors in 'Brake' mode, they will strongly resist. If you try to force them, you can break the motors and the EV3 Brick. To release the motors, use the following command:

```
brick.StopMotor('A', 'Coast');  % Stops one motor in unlocked state
```

or

```
brick.StopAllMotors('Coast');   % Stops all motors in unlocked state
```

You can use either of these commands in a script or the command window.

Start a new script file named **AngleTest**. Be sure that the motors are still connected to **Ports A** & **B**. Then type the following script commands and run it:

```
Brick.ResetMotorAngle('A');
brick.MoveMotorAngleRel('A', 50, 360, 'Brake');
brick.WaitForMotor('A');

brick.MoveMotorAngleRel('A', 50, -180, 'Brake');
brick.WaitForMotor('A');
```

Run the script a few times to see what it does. Now remove (or comment out) the **brick.WaitForMotor('A');** commands, and run the script again.

In a new script called **BackAndForth**, Write a program that does the following when run:

1. Wait for the switch sensor to be pressed, then both motors A and B make a ½ turn in opposite directions (A – clockwise, B – counter clockwise).
2. Wait for the switch sensor to be released, then both motors make a ½ turn in the other direction (A – counter clockwise, B –clockwise). If you attach bars to the motors, the motion of steps 1 and 2 should look like a person waving their arms back and forth.
3. When the switch has been pressed and released 10 times, the script stops.

Hint: Remember that you can "nest" a **while** loop inside of another **while** loop.
Hint: How will you count the number of time the switch has been pressed and released?