Interfacing Matlab code with HomeOS code

This document explains how to interface a Matlab code with the HomeOS code. The basic idea is to generate an executable file of the Matlab code and then call that executable file from the HomeOS code with appropriate arguments.

The following 3 steps explain how to generate a Windows executable file of the Matlab code with the help of an example.

1. Run Matlab and create a new file.
2. Copy and paste the following code in the new file and save it with the name `interfacingMatlab.m`. This code simply takes two strings as inputs, concatenates them, and stores the concatenated string in a file named `interfacingMatlabOutput.txt`)

```
function interfacingMatlab(prefix, suffix)
concatenatedString = [prefix, ' ', suffix, '!'];
fid = fopen('interfacingMatlabOutput.txt', 'w');
fprintf(fid, '%s', concatenatedString);
fclose(fid);
```

3. Go to the Matlab command window and execute the following command to generate a Windows executable file from the Matlab file just saved. The executable file will be generated and stored in the Matlab's working directory

```
mcc -m interfacingMatlab.m
```

Now, let's call this Matlab function from the HomeOS code. Before going through the steps below to see how to call the executable generated from Matlab code from the HomeOS code, go through the "Working with the Lab of Things code" videos at `http://www.lab-of-things.com/getstarted.html` to understand HomeOS and how to implement apps, drivers, and scouts etc. in HomeOS. After going through these tutorial videos, follow the steps below.

1. Go to the solution explorer in Visual Studio, right click the folder "Apps", click "Add", and then click "New Project".
2. Name the new project `MatlabInterface` and place it in the "Apps" directory.
3. Follow all the steps as described in the tutorial video, except that this time use the name `MatlabInterface` instead of `Dummy2` everywhere for this App. For example, instead of remaning `Dummy.cs` to `Dummy2.cs`, rename `Dummy.cs` to `MatlabInterface.cs`. Similarly, rename `DummyService.cs` to `MatlabInterfaceService.cs`.
4. After completing all the steps in the video, look for the function `GetReceivedMessages()` in the file named `MatlabInterface.cs` and replace the code of that function with the following code. This code calls the Windows executable file generated from Matlab by passing the string arguments that the Matlab function operates upon. Once the Matlab function completes, it stores its output in a file named `interfacingMatlabOutput.txt`, which this C# code reads and displays in the HTML5 interface of the App.

```
ProcessStartInfo start = new ProcessStartInfo();
start.Arguments = "Hello World";
start.FileName = "<complete path to the executable
file>\\interfacingMatlab.exe";
start.WindowStyle = ProcessWindowStyle.Hidden;
start.CreateNoWindow = true;

try
{
	using (Process proc = Process.Start(start))
	{
		proc.WaitForExit();
		string concatenatedMatlabString =
	System.IO.File.ReadAllText("interfacingMatlabOutput.txt");
		this.receivedMessageListMatlab.Add(concatenatedMatlabString);
	}
}
catch (Exception e)
{
	logger.Log("Error while interfacing Matlab: {0}", e.ToString());
}
```

5. Add the following line at the top of the `MatlabInterface.cs` file
   `using System.Diagnostics;`

6. Search for `List<string> receivedMessageList;` in `MatlabInterface.cs` file and add the following line below it
   `List<string> receivedMessageListMatlab;`

7. Similarly, search for `this.receivedMessageList = new List<string>();` in `MatlabInterface.cs` file and add the following line below it
   `this.receivedMessageListMatlab = new List<string>();`

8. Build the HomeOS platform and run it.
9. Open the HomeOS dashboard, install the Dummy driver and install the associated AppMatlabInterface.
10. Open AppMatlabInterface and click the Update button. This will invoke the executable file that was created earlier from Matlab. The C# code passes “Hello” and “World” as the two string arguments to this executable. The executable concatenates these two strings and stores the result in the file `interfacingMatlabOutput.txt`. This file is located in HomeOS directory under the folder `Hub\output\binaries\Platform`. Once the executable file completes, the C# code automatically reads the result of concatenation from the `interfacingMatlabOutput.txt` file and displays it in the text box in the application AppMatlabInterface.