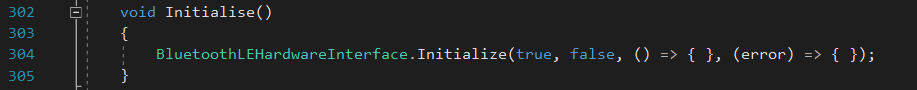
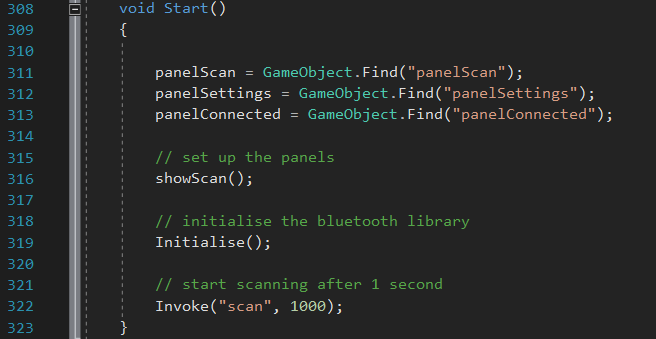
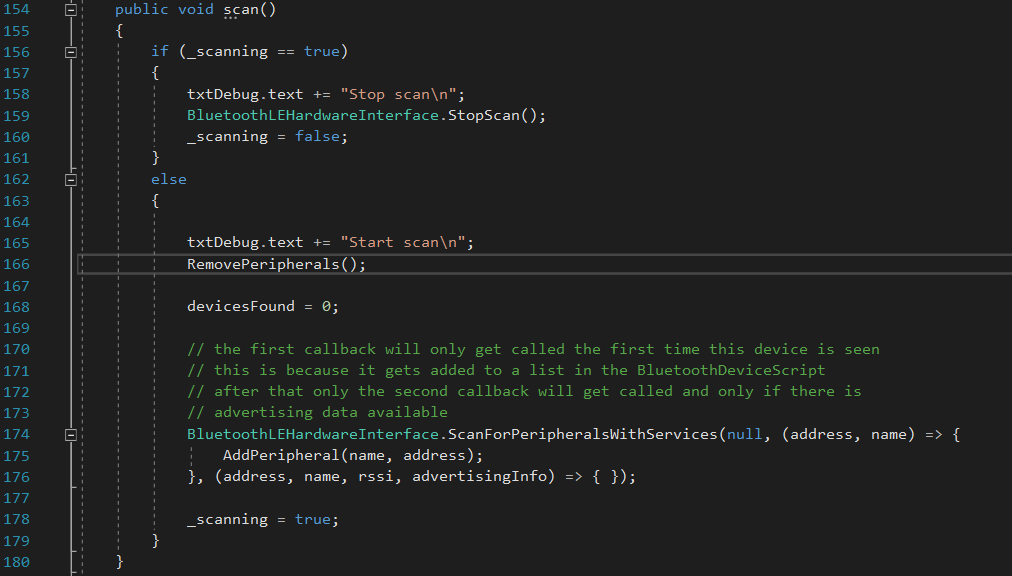
**How this Script Works:**

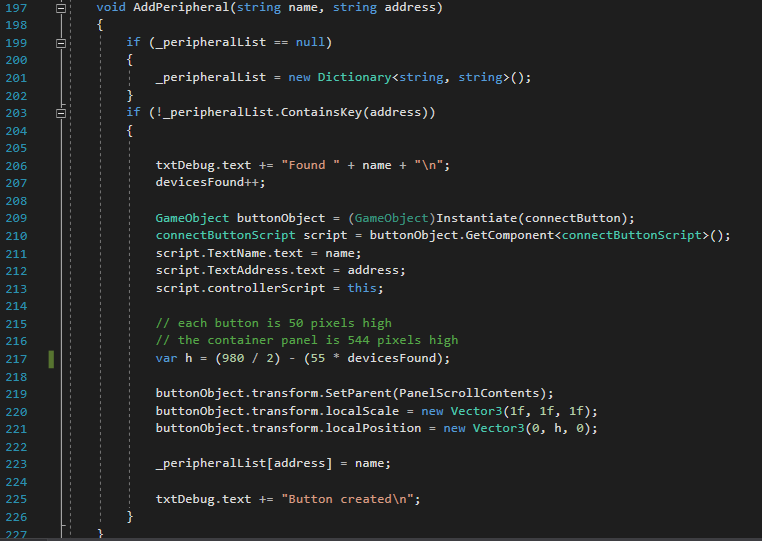
1. The Start() function is recognised by Unity as the position to start the script and is called once at the start. Firstly, it searches for the 3 GameObjects – the panels, and shows the scanning panel first. The BLE module is initialised at line 302 with the initialise() function.
2. The Invoke(“Scan”), then runs the Scan() function after 1 second delay.



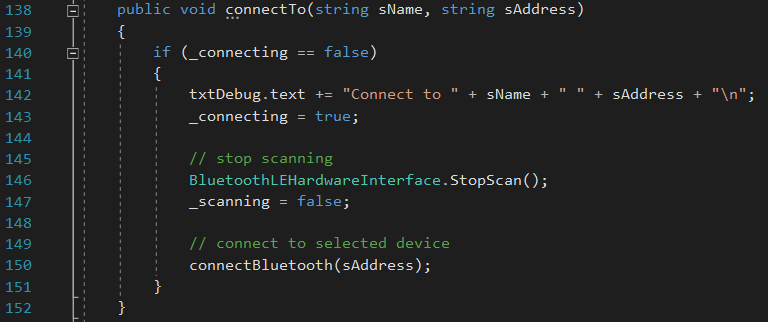
1. The scan function runs (line 154), and will remove any old peripherals, then add new peripherals:



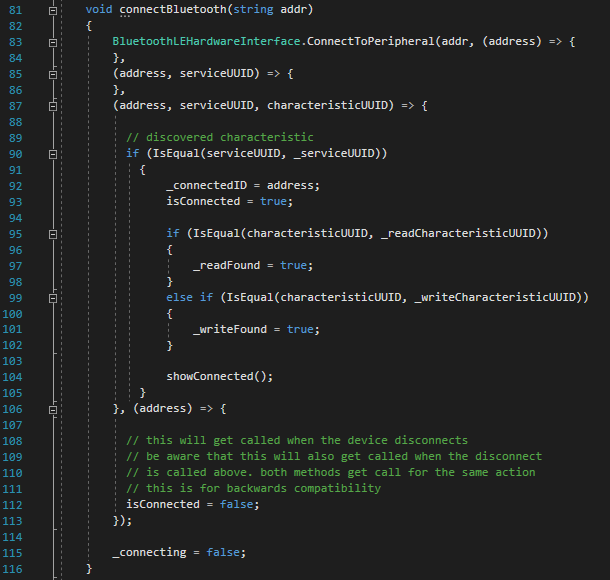
1. The addPeripheral() function creates a new button game object in the scroll panel, which has a behaviour that fills in the global variables: TextName = The name of the BLE device, and TextAddress = the device address. These are variables found within the addPeripheral() class when scanning.



1. Once a button has been created, if it is clicked (the function is within the connectButtonScript.cs), the connectTo() function is run. This stops the BLE plugin from scanning, then runs the connectBluetooth() function.



1. Using the connectBluetooth() function, the BLE plugin will connect to the peripheral, if the characteristics are equal to those specified at the start of the script. First it will check the serviceUUID, and if equal, set the connected state to True. Then the two sub checks are the read and write abilities. If these also match, then both abilities will occur. If the serviceUUID does not match, the device will not connect, and nothing will happen at the button click.



1. Upon successful connection, the showConnected() function will run, which sets the connected panel visibility to On. With this, we can subscribe to the data coming in and out from the device. The readfound and writefound parameters must both be true for this main Update() function to work. These are set in instruction 6). The main Update() function will now listen for data and display it when it sees it. I am unsure yet if the send data button function will work, although I doubt it.