Supporting a new device with the Bluetooth Data Explorer

All About the Device

In this example, the device I’m adding is a Govee Smart Thermo-Hygrometer – that means it measures temperature and humidity. Mine is model H5074 (but be warned: these sensors tend to come and go in the marketplace)

Step 1: Power on and run Bluetooth Data Explorer

The device has a little blue tab; pull it and it will start advertising. This is about the last time you’ll need to fiddle with the device. Run the Bluetooth Data explorer program

Graphical user interface

Description automatically generated with medium confidence

Go into settings; in “Show which device” change the setting to “Bluetooth Beacons” and click “Search”.

Graphical user interface, application, Word

Description automatically generated

The Govee sensor will show up; click “Full details” to get as much information as you can from the device. In the Govee case, it’s not much, but we do see the name and the Bluetooth address. From one specific output:

E0:17:54:D0:74:C5 19:21:22.6 -52 IsScannable Govee\_H5074\_74C5

LE General Discoverable Mode+BR/EDR Not Supported

Service UUIDs (complete): Device Information=180A, FEF5, EC88

E0:17:54:D0:74:C5 19:21:22.6 -51 ScanResponse

(is EC88?): Temp=21.53 Hum=50.55% Bat=100% (junk=0) 02

E0:17:54:D0:74:C5 19:21:14.3 -50 ScanResponse

(was apple): Pre=5378 Str=INTELLI\_ROCKS\_HWP Post=74 C5 74 C2

The LE means that it’s a Bluetooth LE device; the General Discoverable mode means it’s ready to be paired and that the user (that’s you) hasn’t done anything special to put it into pairing mode (that would be the limited mode).

From the advertisement, we can conclude that it’s not spitting out data in the advertisement. In theory we can get data by connecting to the device and then poking at service EC88. But the “IsScannable” means that the device will give us more data if we ask; this is turned on automatically by the Bluetooth Device Controller. We get scan responses of two types, each with a manufacturer type. Either the manufacturer claims to be an Apple device (apparently this is required for some kind of Apple interfacing), or manufacturer 0xEC88, which isn’t actually assigned.

The Bluetooth Device Controller needed to be extensively modified to support the Govee. Among other things, the starting Govee advertisement isn’t actually marked as Govee, and the scan response doesn’t have any identifying information, either. Instead we know what the type is based on the name and the bogus manufacturer id.

Now that we’ve updated the code enough to show the advertisements, it’s time to make a specialty page. The Govee is most similar to the Ruvvi Tag, so we will use that as our baseline. For this project, we’re just going to copy the code; I’ve got a bunch more sensors, though, and at the end there will be one nice generic Sensor page that can read in a bunch of different data types.

Make a new Govee class using a copy of the Ruuvi class. Update as needed 😊, and I updated the ‘ParseManufacturerData’ to use the new Govee.Parse method.

For the graphics, I’m updating the RuuviTag graphics page. This includes adding a new UpdatedGoveeAdvertisement in DeviceInformationWrapper that mimics the UpdatedRuuviAdvertisement

Note the horrible way that the wrapper?.BleAdvert.Event(ruuvi.Data) gets called from the CustomizeWrapperFromAdvertisement in BleAdvertisementFormat.cs.