

This video illustrates, from start to finish, creation of Bluetooth low energy functionality.

The hardware is a “RedBear BLE Nano – 2”.
Red Bear is now part of particle.io. They make many kinds of these powerful little devices.

It has a Nordic nRF52832 ARM cortex MF4
Micro-controller and “DAP link” com port
debugger.

Code is developed with the C++ Arduino / Genuino IDE

First setup the Arduino plug in for RedBear Nano-2 hardware.

With some helpful instructions from Red Bear's Github site you can get this to install.

This is just a snippet of the steps, but they do work.

Once the support package is installed, it becomes one of the choices you can select when you code a project “Arduino style”.

I selected the "BLE_SimplePeripheral" sketch in "Examples for BLE_Nano2" hardware as my starting point.

My goal to show you code that can send/receive data between the NANO-2 and an Android / Moto G phone.

A modified “BLE_SimplePeripheral “ sketch file 320 lines does the job.

(There is pre-written library code backing this up.)

Next are slides discussing the code:

(Code slide 1)

I only had to modify or add 6 lines of code,
But I also added about 42 lines of comments to assist in our understanding of how this sketch file works.

On Line 22, I changed the name as it will appear on my cell phone to “BLE_Periph_JMK” to identify my device.

On Line 43-46 comments were added to clarify the jargon:

A “characteristic” refers to data that is sent over the wireless channel.
This can be of different types

- 1) A Write Characteristic is sent by the cell phone to the IOT device.
- 2) A Read Characteristic is sent by IOT device to cell phone upon request.
- 3) A Notify Characteristic is sent by IOT device to cell phone automatically.

(Code slide 2)

A very important function is "gattServerWriteCallback(..)"

This is an event handler (a call back function) that occurs when the BLE nano receives data from cell phone.

This occurs when data sent by the cell Phone is received by the Nano.

A lines 184 and 185 I added code to assign the data in the Write characteristic#1 also to the Read characteristic#2

That way user of the cell phone can latter request a read of this data to verify that whatever was sent, was received.

(This is known as a Loop Back test.)

(Code Slide 3)

taskHandle(void) is the callback for the 1 second time task.

At line 196: I Eliminated a Serial.println call and
At line 218-19 I added instead an LED blink, instead.
It replaces debug output with a blinking led to indicate this task is
working. This makes reading other debug output easier.

I also put in a long comment on how I located the port for the LED

Asside:

I also added comments pointing out where this task updates
custom Notify characteristic #3, and also where it loads a new simulated
heart beat rate every second.

Heart rate simulation is typical BLE sample code.

(Code Slide 4)

Finally,

The "loop()" is where non event related code gets put to run any Arduino application.

For this demo sketch all the action happens in the event related code.

So we just wait for events.

Now lets talk about my test setup for Android G phone.

For convenience and easy screen shot capture, I remotely controlled my cell phone over a USB cable with a free PC program called “Visor”.

I used the "nrFConnect" App provided by Nordic Semiconductor to test my Nano – 2 code. Learn more about this app from Nordic Semi... Or - follow these steps to press the buttons to see the "nrFConnect" App in action.

- First look at on the air BLE devices to find ours. "BLE_Periph_JMK" shows you it is there.
- Then Press the Connect button.

This opens up the list of custom Characteristics.

Click on the Up arrow for the one with WRITE property.

Select data type "Text", type in "Joe".

Go to "advanced" select "Command".

Finally save as load choice "Joe1"

(Slide 13 and 14)

Press Load, Click on Send

So what happens ?

On your cell phone the WRITE Characteristic
Indicates "Joe" was written to the Nano.

(But this does not prove it was really received.)

We can verify by reading back the READ Characteristic from the Nano by clicking on the down arrow as shown.

Once this is done the value "Joe" sent earlier now appears as the value of the READ characteristic

This works for "Mike" as well. *(Try it if you like.)*

For more info and source code for the modified Sketch, see my git hub site listed bottom of this slide.

This breaks the ice for future IOT / BLE Adventures !