

FYS4240

Lab5:

Send IMU data over BLE (Bluetooth Low Energy) and UDP

Introduction

In this lab you will send IMU data from the Arduino board using BLE. On a computer you will use LabVIEW to read and display the data sent over BLE. You will need the BLE dongle **BLED112** and drivers for this, see instructions below. In addition, you will write LabVIEW code to send data over UDP, for instance over Wi-Fi or a cabled Ethernet network to a separate computer. In this lab you will send the data to the *localhost* address (*IP 127.0.0.1*). Localhost refers to the current computer and is used to access the network services that are running on the host via the loopback network interface. In this way we can test network functions on a single computer.

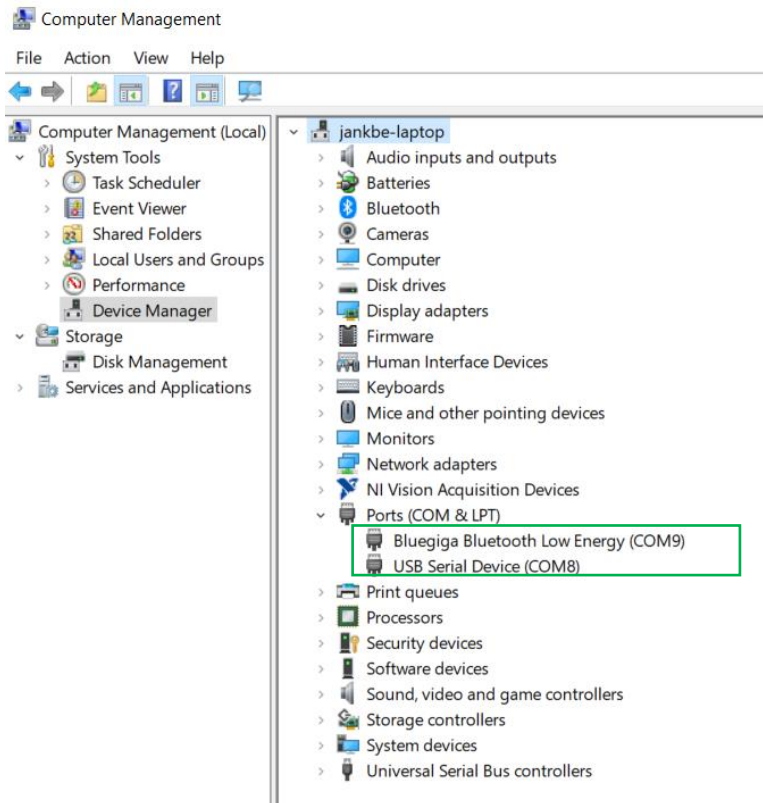
See the BLE information provided in lecture 5 before you start. In the Arduino Cookbook you also find some BLE information on page 536-539. See lecture 3 part 2 for basic information about UDP and the use of UDP in LabVIEW.

Exercise 1 - Install BLE Toolkit for LabVIEW and BLED112 dongle driver

1. Insert the BLED112 BLE dongle in your computer. **Important: you need this specific BLE adapter for this lab!**
2. Follow the instructions on the page [GitHub - MuSAELab/BLE-Toolkit-LabVIEW: BLE \(Bluetooth Low Energy\) Toolkit for LabVIEW](#) to install the **BLE Toolkit for LabVIEW** and the **drivers** for the BLED112.

Exercise 2 – Send IMU data to the computer over BLE

1. Upload the provided sketch called **acc_over_BLE** to the Nano 33 board. This sketch will read accelerometer data (x, y and z), convert the data values to string and send a text string with tab between the values over BLE and to the serial interface over USB (as in previous labs).
2. Make sure that you have the BLED112 BLE dongle inserted in an USB-port.
3. **Open the Serial Monitor in the Arduino IDE.** This is important since the sketch is waiting for a serial port to connect (see the code)! We will use it for debugging/verification!
4. Run the provided program **ble_scan_devices.vi** and verify that you can see the Arduino board and is able to connect to the board.
 - a. You must select the correct COM-port used by the BLE, before you press run. You can verify the COM port used by the Bluegiga BLE dongle by opening the *Device manager*; see figure below.
 - b. You should see the name of the device, *Arduino Nano 33 BLE sense*, as defined in the sketch.
 - c. Copy and paste the *MAC-address* listed for the Arduino Nano 33 BLE, select the type *Public*, and press *Continue*.
 - d. The *Connected* LED should then become green.



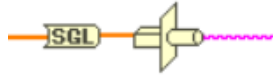
5. Run the LabVIEW program **ReadPlot_BLE.vi**, and verify that signals plotted from the accelerometer are as expected.
 - a. Select the COM port for the BLED112 USB receiver, and Run the VI.
 - i. After *Scanning* is finished, the *Connected* LED and the *Reading Data* LED should turn green after a while. When the LabVIEW program is connected over BLE the orange LED on the Arduino board should light up. In the IDE Serial Monitor you should now see that accelerometer values are transmitted also over the serial interface.
 - ii. The code inside the Sub-VI called *BLE_setup.vi* is similar to the code used in the program *ble_scan_devices.vi*, except that now a search is used to automatically find the MAC address of the Nano 33 board. In addition, indicators are reset/initialized at the start of the program.
 - b. Verify that the accelerometer values sent over BLE are identical to the values sent over the serial USB cable connection (using the Serial Monitor).
 - c. What is the frequency of the BLE read and plot loop? Any comments?

Hint: if you have problems, try to close the entire LabVIEW development environment and open it and the file again! Because if you get a program “crash” you could lose the proper connection to the Arduino card. Verify that you actually see the Arduino Nano 33 in the device list!

Exercise 3 – Distribute (stream) the IMU data to another computer using UDP

1. Save the program as **ReadPlot_BLE_send_UDP.vi**, and add distribution of data over UDP using a producer-consumer structure:

- a. In the while loop “Read and plot BLE data”, add the accelerometer data to a queue called UDP_send.
 - i. You can add either the SGL array data, or the string data.
- b. Create a new while loop called UDP_send, and read data out of the queue.
 - i. If you added the SGL array data (*BLE data*) you can convert this back to a string in the following way:



- c. Open a UDP connection (outside the UDP loop), write data to UDP (inside the loop) and close the UDP connection (outside the loop) when the program is finished (stop button is pressed).
 - i. Write to the *localhost* address
 - ii. Write to the same port as used in the provided **DataReceiver.vi** program.
 - iii. Hint: A UDP sender-receiver example can be found under *Help-Find Examples*. Search for UDP, and look at the example “simple UDP”.
2. Open the provided program **DataReceiver.vi** (on the same computer) and verify that signals from the accelerometer are transmitted and received as expected over UDP. (Localhost/IP 127.0.0.0 is used to allow UDP transmission to the same computer). **Remember that the Serial Monitor in the Arduino IDE must be open. This is important since the sketch is waiting for a serial port to connect.**
- a. Take a screen shot showing data plotted in the DataReceiver.vi program, to show correct behavior of your program.

Optional: Exercise 4 – BLE only

1. Connect the Arduino board to a power bank or a wall power outlet close to the computer. In order to have power on the board, but without a USB connection to the computer.
 - a. You now need to remove the code in the sketch that is sending data over serial and waiting for serial to be ready, since we now do not have a serial connection (USB)! Save this new sketch as **acc_over_BLE_noSerial**
 - i. If you do not remove or comment out the *while (!Serial)* your program will forever wait for a serial connection ...
 - b. Upload the new sketch to the Nano 33 board.
 - c. Verify that you still get accelerometer data plotted on the PC (in LabVIEW), without the USB cable connected.
 - d. Approximately, how far away from your computer did you manage to communicate over BLE? Probably you experienced that you lost connection at a distance closer than 10 meters?

What to hand in:

- The answer to the question in Exercise 2.5c.

- The program **ReadPlot_BLE_send_UDP.vi** you made in Exercise 3.
- The screen shot in Exercise 3.2a.
- Optional: The answer to the question in Exercise 4.1d and the sketch **acc_over_BLE_noSerial**