Project 21 – Novastoma Documentation for XIAO nRF52840 and Nordic SDK

Nick Urch – nurch@ncsu.edu

1. **Set up Nordic SDK and the toolchain in VS Code.**
    a. For a step-by-step guide on setting up the Nordic SDK and the toolchain in VS Code, refer to the tutorial video available at https://www.nordicsemi.com/Products/Development-software/nRF-Connect-SDK/GetStarted#infotabs
    b. To begin, download the code from GitHub and save it to your device. Ensure that the folder you save the code to does not contain any spaces. Then, in the nRF extension tab in VS Code, under the "Welcome" tab, click open an existing application. Next, navigate to the folder where you saved the code, and click ok to open it in VS Code.
    c. In the "Application" tab, click "Build all configurations." This will build the application and prepare it for upload.
2. **Reprogramming the XIAO nRF52840 board.**
    a. A small reset button is next to the USB-C connection. Press it twice to put the board into bootloader mode.
    b. Windows will ask you what you want to do when this device is found. You can click to open File Explorer automatically.
    c. In Windows, this device will appear as a storage device with its own drive, similar to a USB storage device. It will be named "XIAO-SENSE" even though this is a non-sense board.
    d. Open the folder where you saved the application and navigate to the "build\zephyr" directory. There will be a file named "zephyr.uf2". Either copy/paste or drag/drop this file into the XIAO-SENSE.
    e. The bootloader will download the code and reset the board. The new application will then run.
3. **Files included.**
    a. Prj.config – This is the configuration file for the application. This includes all config settings needed for the MCU's operation of devices, such as Bluetooth, GPIO, ADC, and Flash memory.
    b. Main.c – The main program that will run on the MCU.
    c. Battery.h and Battery.c – The header and c files for the battery helper.
        i. Marcus Alexander Tjomsaas wrote this code, and the GitHub is located at: https://github.com/Tjoms99/xiao_sense_nrf52840_battery_lib
        ii. You can visit the GitHub above for more information on the battery helper.
    d. CMakeLists.txt – The make files for the application.
    e. Nova_nRF.overlay – This overlay file maps the GPIO pins on the board. This is not used only for the Arduino framework.
4. **Code walkthrough – main.c.**
    a. Run through of main.c file, leaving out trivial or obvious code.
    b. Lines 1-15: #include files for the Zephyr RTOS and Bluetooth operation.
    c. Line 18: Register the log module for USB serial debugging and information.

d. Lines 20-33: User-defined data for bag waste level, commands, buffering, and time-stamp.

e. Lines 46-59: Definition for GPIO pins used to power the three sensors and read data. Also, the device pointers for the GPIO port drivers.

f. Lines 62-64: Preparing the battery helper.

g. Lines 68-113: Macros to build the GATT Service and Characteristics. These form the 128-bit UUID for the above. The client-characteristic configuration callbacks are also defined here. These allow a user to subscribe to notification events. Below are the GATT Characteristic callbacks for reading and writing to the three characteristics: Data, Battery, and Command.

h. Lines 116-133: Macro to build the Service and Characteristics with the appropriate UUIDs and permissions.

i. Lines 136-145: Setting up the data to send while Bluetooth is advertising.

j. Lines 149-267: Bluetooth callbacks for various events.

    i. Connected – When a Bluetooth connection has been established.

    ii. Disconnected – When a Bluetooth connection has been severed. The reason for disconnection is logged. The reasons can be found in the Bluetooth datasheet https://www.bluetooth.com/specifications/specs/core-specification-5-3/

    iii. Mtu_updated – Information log for when the byte size of the packets has changed.

    iv. On_le_param_updated – When the connection parameters have changed.

    v. ****_ccc_cfg_changed – When a connection has subscribed/unsubscribed to the characteristic.

    vi. ****_read_characteristic_cb – When a connection reads from the characteristic.

    vii. ****_write_characteristic_cb – When a connection writes to the characteristic.

k. Lines 270-292: Initialize Bluetooth and start advertising.

l. Lines 295-319: Initialize GPIO.

m. Lines 322-326: Calling the battery helper handler.

n. Lines 329-398: User functions.

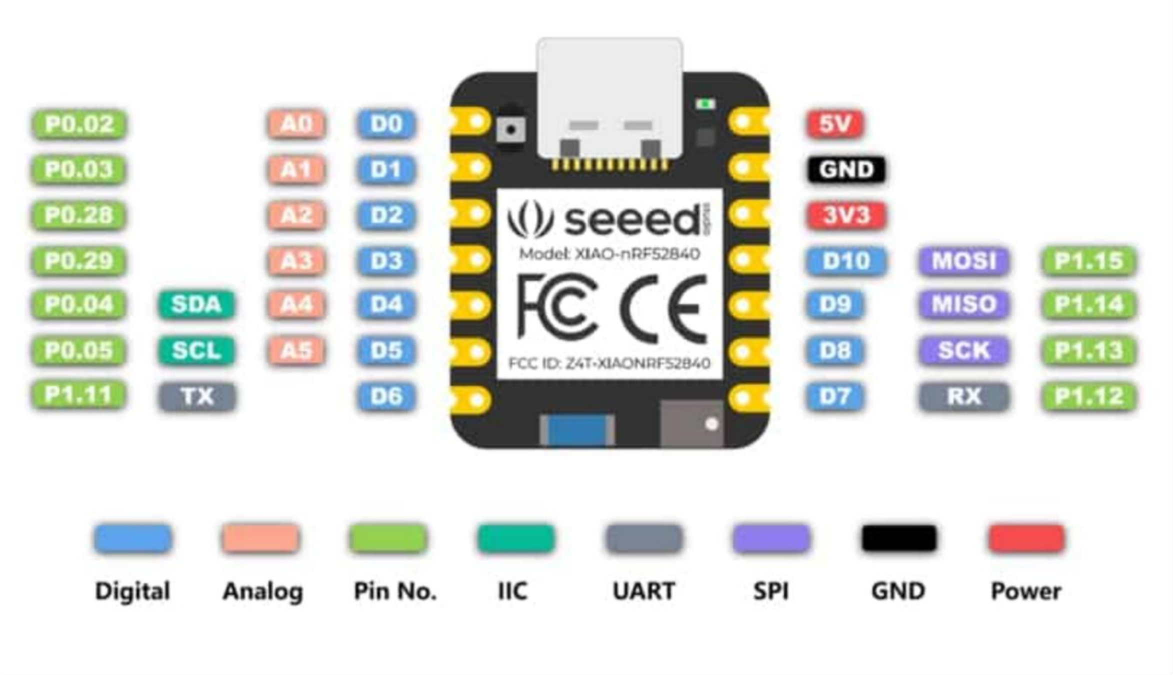
    i. Get_Bag_Level: This function powers the sensors through the GPIO and then reads their levels. After the sensors are read, they are powered back down to save the battery.

    ii. Create_Outgoing_Data: This function builds a string with the timestamp as characters [0-9] and the waste level as character [10].

    iii. Send_Buffer: This function will send out the buffered data from oldest to newest order.

o. Lines 401-481: The main loop of the application. This will initialize all the peripherals and ready the Bluetooth. In the for(;;) loop, the program will get the waste level of the bag, check for Bluetooth and send out the data if connected. If not connected and a connection is lost, the data is buffered. The thread will then sleep for a specified amount of time.

5. Zephyr Documentation can be found at: https://www.zephyrproject.org/
6. More information about the XIAO nRF52840 can be found at: https://www.seeedstudio.com/Seeed-XIAO-BLE-nRF52840-p-5201.html
7. XIAO nRF52840 Pinout.



   a. Note: The bottom pins marked P1.11 (Tx) and P1.12 (Rx) did not seem to work when trying to power or read from the sensors. These may be used for Serial or UART communication, but I did not look into the problem; I just avoided using them.

8. I can be reached at nurch@ncsu.edu for any questions. If, after graduation, that e-mail is deactivated, you can e-mail me at nmu0901@yahoo.com or message me on Discord at nicku252.