Nordic Semiconductor nRF51822 development

Fra Robolab-wiki

Indholdsfortegnelse

Nordic Semiconductor nRF51822

The nRF51822 (https://www.nordicsemi.com/eng/Products/Bluetooth-R-low-energy/nRF51822) is a Bluetooth 4.0 Low Energy (http://en.wikipedia.org/wiki/Bluetooth_low_energy) (a.k.a. Bluetooth Smart) SoC (http://en.wikipedia.org/wiki/System_on_a_chip) that incorporates a 2.4GHz radio, some I/O and a Cortex M0 (http://en.wikipedia.org/wiki/ARM_Cortex-M#Cortex-M0) 32bit ARM core in a single chip. The nRF51822-EK (https://www.nordicsemi.com/eng/Products/Bluetooth-R-low-energy/nRF51822-Evaluation-Kit) evaluation kit is a cheap easy way to get started with software development. A key feature in Nordic's solution is that the Bluetooth Smart protocol stack and runtime is completely seperated from the user application code, which makes development much easier. This is in contrast to e.g. Texas Instruments CC2540 (http://www.ti.com /product/cc2540) SoC, where the user application runs in the same micro OS as the BT protocolstack, which gives a more complicated development environment.

Here is an overview of Nordics Bluetooth Low Energy related products (https://www.nordicsemi.com /eng/Products/Bluetooth-R-low-energy)

Nordic's application note nAN-29 provides instructions for setting up Eclipse with CDT + CodeSourcery's gcc ARM compiler + Nordic's nRF51 series SDK, to develop and debug nRF51 embedded applications using free tools. It is possible, with a few easy tweaks, to develop low energy Bluetooth Smart applications on the nRF51822 SoC with this setup. For Bluetooth Smart applications, such as low energy heart rate monitors, blood glucose meters, proximity sensors, and so on, the nRF51822 has to run Nordics S110 Bluetooth Smart protocol stack. The embedded user application then interacts with the S110 protocol stack through high level API calls.

Setting the open source development environment

To setup Eclipse, Code Sourcery ARM gcc compiler and Segger debugger for developing nRF51 embedded applications ***without*** the S110 Bluetooth protocol stack:

- Folow the instructions in application note nAN-29 (https://www.nordicsemi.com/eng/nordic/download_resource/16124/3/27704629) and setup the Eclipse IDE, CodeSourcery compiler, Segger debugger, nRF51 SDK, and confirm that you can compile, flash and run the blinky example project on the nRF51822-EK (https://www.nordicsemi.com/eng/Products/Bluetooth-R-low-energy/nRF51822-Evaluation-Kit) Evaluation Kit.

Note: In order for Eclipse's indexer to see the whole API, be sure to på tick-marks in 'Add to all configurations' and 'Add to all languages' when adding include paths.

To enable Bluetooth Smart communication in your nRF51 embedded application:

- Download and extract S110 SoftDevice (https://www.nordicsemi.com/eng/nordic/download_resource/16681/4/84747796) -we are interessted in the .hex file which is the ROM image of the Bluetooth Stack driver that needs to be flashed the nRF51822 later.
- Install nRFgo Studio (https://www.nordicsemi.com/eng/nordic/download_resource/14959/28/95085502) -this is used to flash the S110 SoftDevice .hex file later.
- Folow the instructions in the support reply from Nordic, using the files from this ZIP archiveFil:Nrf51-3.0-gcc special nordic reply.zip
  - Makefile.common should replace the one in Source/templates/gcc.
  - Remove usage of __current_sp() in ble_error_log.c

```
//p_stack    =   (uint32_t*) __current_sp();
p_stack = (uint32_t*) ROM_ADDRESS_START;
```

  - Replace ble_sdk_util in Include/ble/
  - app_timer.c in Source/ant interferes with the app_timer.c used in ble. For now, just remove it.
  - replace Makefile in ble_app_hrs with the Makefile from the ZIP archive and apply instructions in nAN-29 for on the ble_app_hre example project.
- Folow the instructions in the nRF51822-EK User Guide (https://www.nordicsemi.com/eng/nordic/download_resource/14735/1/10831829) on how to flash the S110 hex file
- On iPhone/iPad with Bluetooth 4.0 support: Install the nRF iOS Demo App from Apple's App Store and see that the HRM function can connect to the PCA
  - The XCode project + source code for the iOS Demo App is available here (https://www.nordicsemi.com/eng/Products/ANT/nRFready-iOS-Demo-App)

Step by step - using ble_app_template

- copy Board/nrf6310/ble/ble_app_template to Board/pca10001/ble/ble_app_template
-

Other insteresting

- Apple's Bluetooth Accessory Design Guidelines (https://developer.apple.com/hardwaredrivers/BluetoothDesignGuidelines.pdf) has a section on Bluetooth Low Energy starting on page 17.

- developer.bluetooth.org's getting started pages (http://developer.bluetooth.org/DevelopmentResources/Pages/Getting-Started.aspx) . Be sure to read custom profiles development (http://developer.bluetooth.org/DevelopmentResources/Pages/Custom-Profile-Development.aspx) to get an overview of Profiles, Services and Characteristics and how they relate.
- bluetooth.org's specification documents and list of adopted Bluetooth Profiles, Services, Protocols and Transports here (https://www.bluetooth.org/Technical/Specifications/adopted.htm) . Here you can find the latest v4.0 Core Specification, 2300 pages long :-)
- EE Times has an older but interesting two-part article series on BLE. Part 1 (http://www.eetimes.com/design/communications-design/4217866/Bluetooth-4-0--An-introduction-to-Bluetooth-Low-Energy-Part-I) and Part 2 (http://eetimes.com/design/communications-design/4218319/Bluetooth-4-0--An-introduction-to-Bluetooth-Low-Energy-Part-II)
- Wahoo Fitness has a comprehensive list of iOS apps using their API here (http://www.wahoofitness.com/Apps/Apps.asp)
- Samsung has released a beta of their BTLE SDK here (http://developer.samsung.com/ble)

iOS app signing and distribution

- Apple's App Distribution Guide (https://developer.apple.com/library/ios/#documentation/IDEs/Conceptual/AppDistributionGuide/Introduction/Introduction.html)

- Denne side blev senest ændret 24. maj 2013 08:33.