

n-Blocks

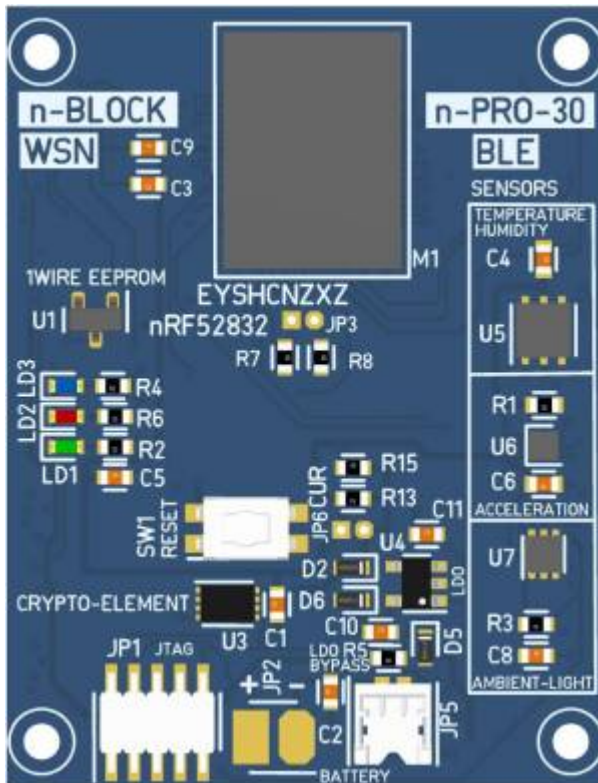
n-PRO-30

Table of Contents

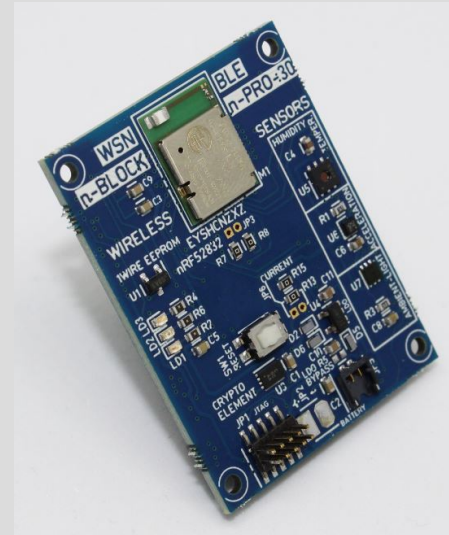
Overview	1
MCU Features	2
n-PRO-30 Features	3
Board Pinout	3
Getting started	8
Blinky using Nordic SDK	8
Blinky with mbed online compiler	10
Flash Memory Programming	10
Programming with Nordic nRF52-DK board	10
USE-CASE examples	11
Related articles in this Wiki	12

n-PRO-30

n-PRO-30 is an mbed-enabled Bluetooth Low Energy development board from the n-Blocks family, in the standard [n-Blocks pro form factor](#).



n-PRO-30



n-PRO-30

License	GPL 2.0
Status	Tested-Blink
Buy at:	
Categories	
Hardware repo	Bitbucket
Firmware repo	

Overview

n-PRO-30 is a simple Bluetooth Low Energy board based on Nordic nRF52832. It consists of temperature/humidity sensor, acceleration sensor and ambient light sensor which can be used to collect data and transmit it over bluetooth. The board supports the standard Nordic Software Development Tool-chain using Keil, IAR and GCC. It also supports ARM mbed tool-chain for rapid prototyping and development using mbed's IDE and tool-chain with an extensive range of open-source software libraries.

MCU Features

- 2.4 GHz transceiver
 - -96 dBm sensitivity in Bluetooth® Low Energy mode
 - Supported data rates: 1 Mbps, 2 Mbps Bluetooth Low Energy mode
 - -20 to +4 dBm TX power, configurable in 4 dB steps
 - On-chip balun (single-ended RF)
 - 5.3 mA peak current in TX (0 dBm)
 - 5.4 mA peak current in RX
 - RSSI (1 dB resolution)
- ARM® Cortex®-M4 32-bit processor with FPU, 64 MHz
 - 215 EEMBC CoreMark score running from flash memory
 - 58 µA/MHz running from flash memory
 - 51.6 µA/MHz running from RAM
- Flexible power management
 - 1.7 V–3.6 V supply voltage range
 - Fully automatic LDO and DC/DC regulator system
 - Fast wake-up using 64 MHz internal oscillator
 - 0.3 µA at 3 V in System OFF mode
 - 0.7 µA at 3 V in System OFF mode with full 64 kB RAM retention
 - 1.9 µA at 3 V in System ON mode, no RAM retention, wake on RTC
- Memory
 - 512 kB flash/64 kB RAM
 - 256 kB flash/32 kB RAM
- Nordic SoftDevice ready
- Support for concurrent multi-protocol
- Type 2 near field communication (NFC-A) tag with wakeup-on-field and touch-to-pair capabilities
- 12-bit, 200 ksps ADC - 8 configurable channels with programmable gain
- 64 level comparator
- 15 level low power comparator with wakeup from System OFF mode
- Temperature sensor
- 32 general purpose I/O pins
- 3x 4-channel pulse width modulator (PWM) unit with EasyDMA
- Digital microphone interface (PDM)
- 5x 32-bit timer with counter mode
- Up to 3x SPI master/slave with EasyDMA
- Up to 2x I2C compatible 2-wire master/slave
- I2S with EasyDMA
- UART (CTS/RTS) with EasyDMA
- Programmable peripheral interconnect (PPI)
- Quadrature decoder (QDEC)
- AES HW encryption with EasyDMA
- Autonomous peripheral operation without CPU intervention using PPI and EasyDMA

- 3x real-time counter (RTC)
- Single crystal operation

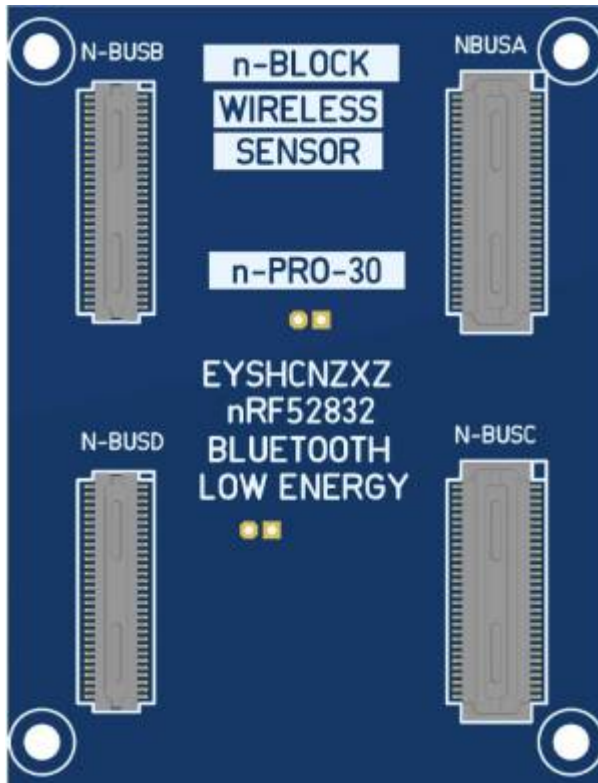
n-PRO-30 Features

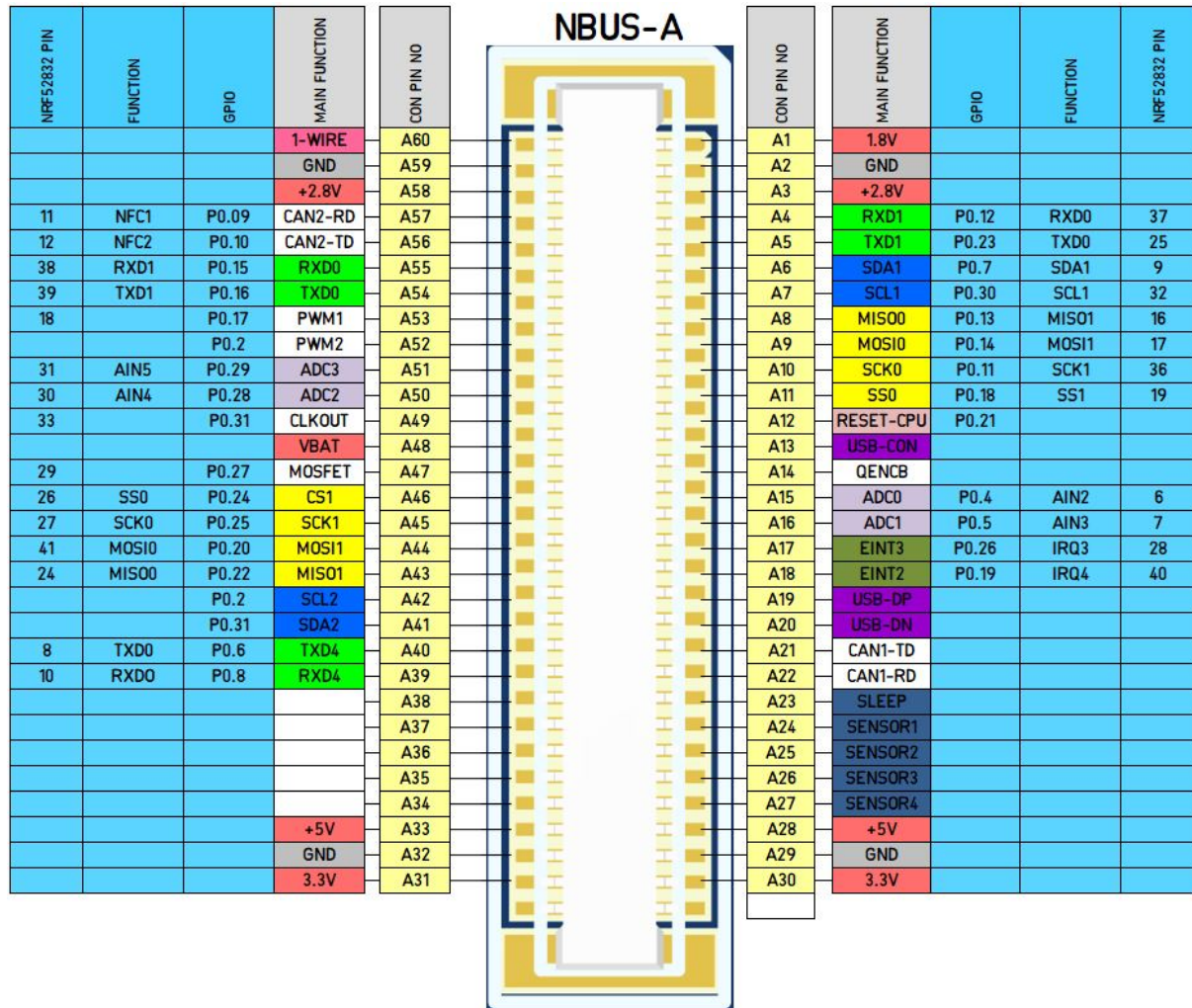
- Tayio module integrates nRF52832 and antenna
- Simple and Low Power
- Additional “industry standard” [RF module socket](#)
- Miniature Li-Ion battery connector (optional)

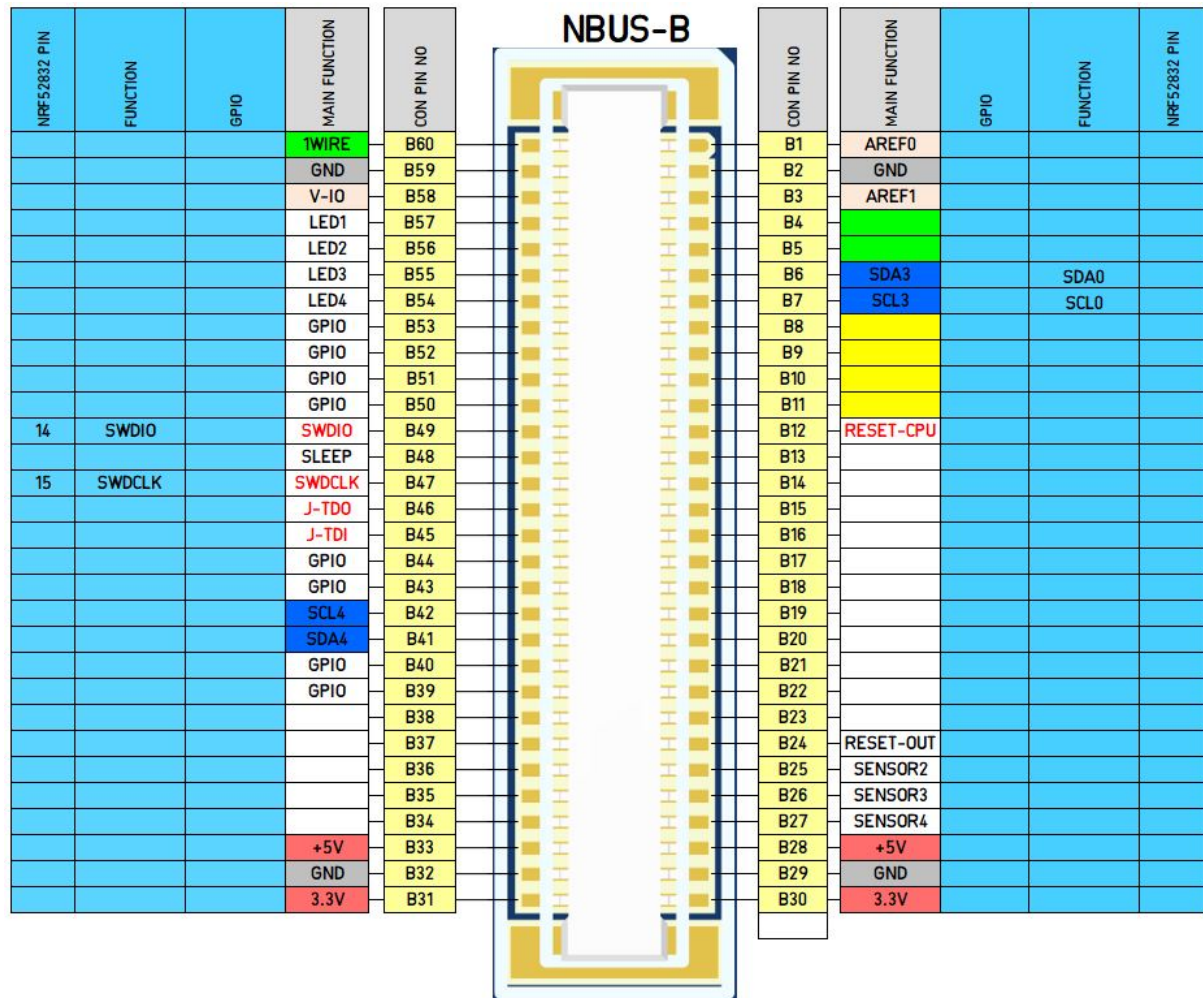
- [mbed](#) compatibility supports BLE and easy development (use [Nordic nRF52-DK](#) platform)
- [nRF5 SDK for Mesh](#)
- [IoT SDK for applications using IPv6 over Bluetooth low energy](#)

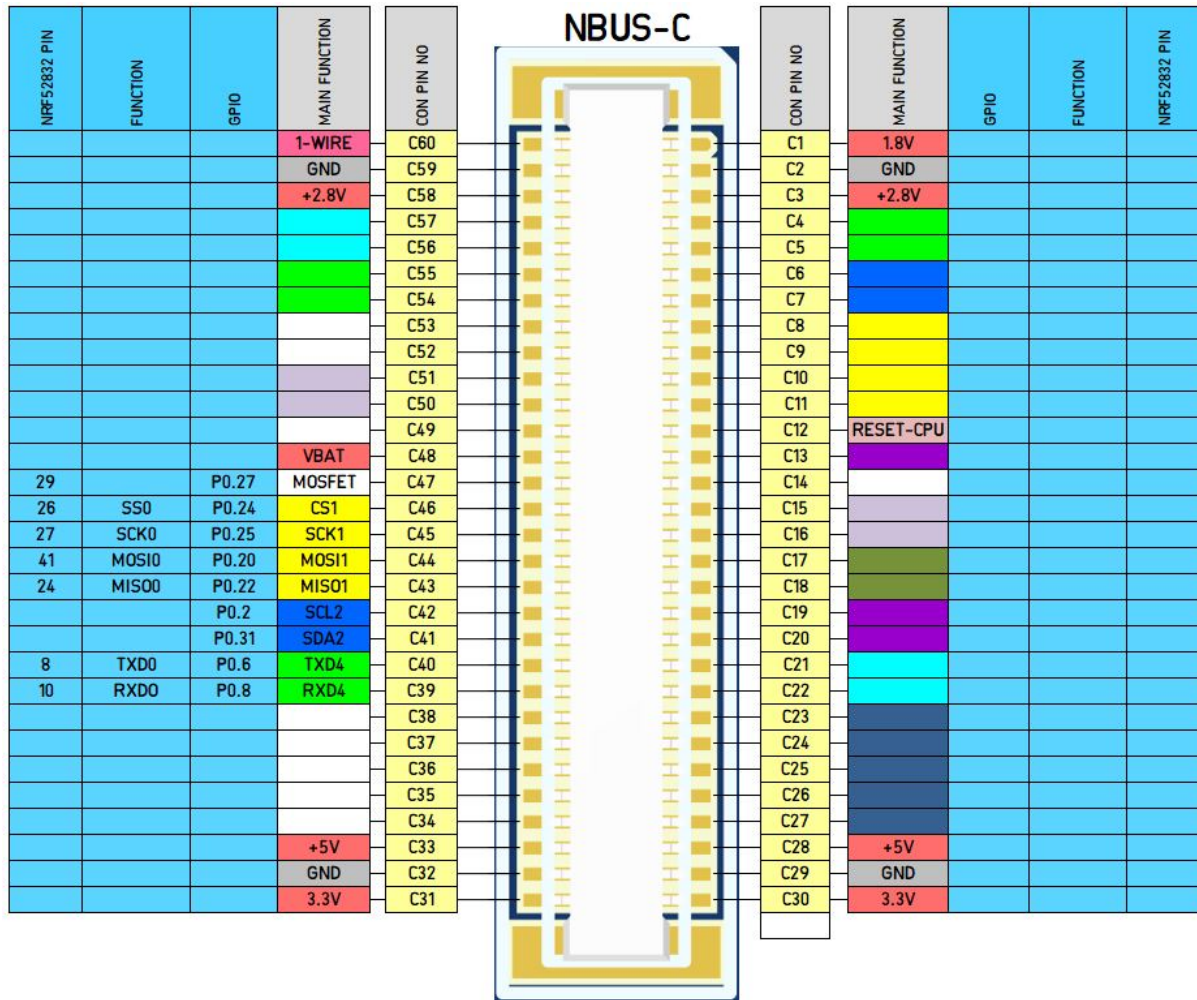
Board Pinout

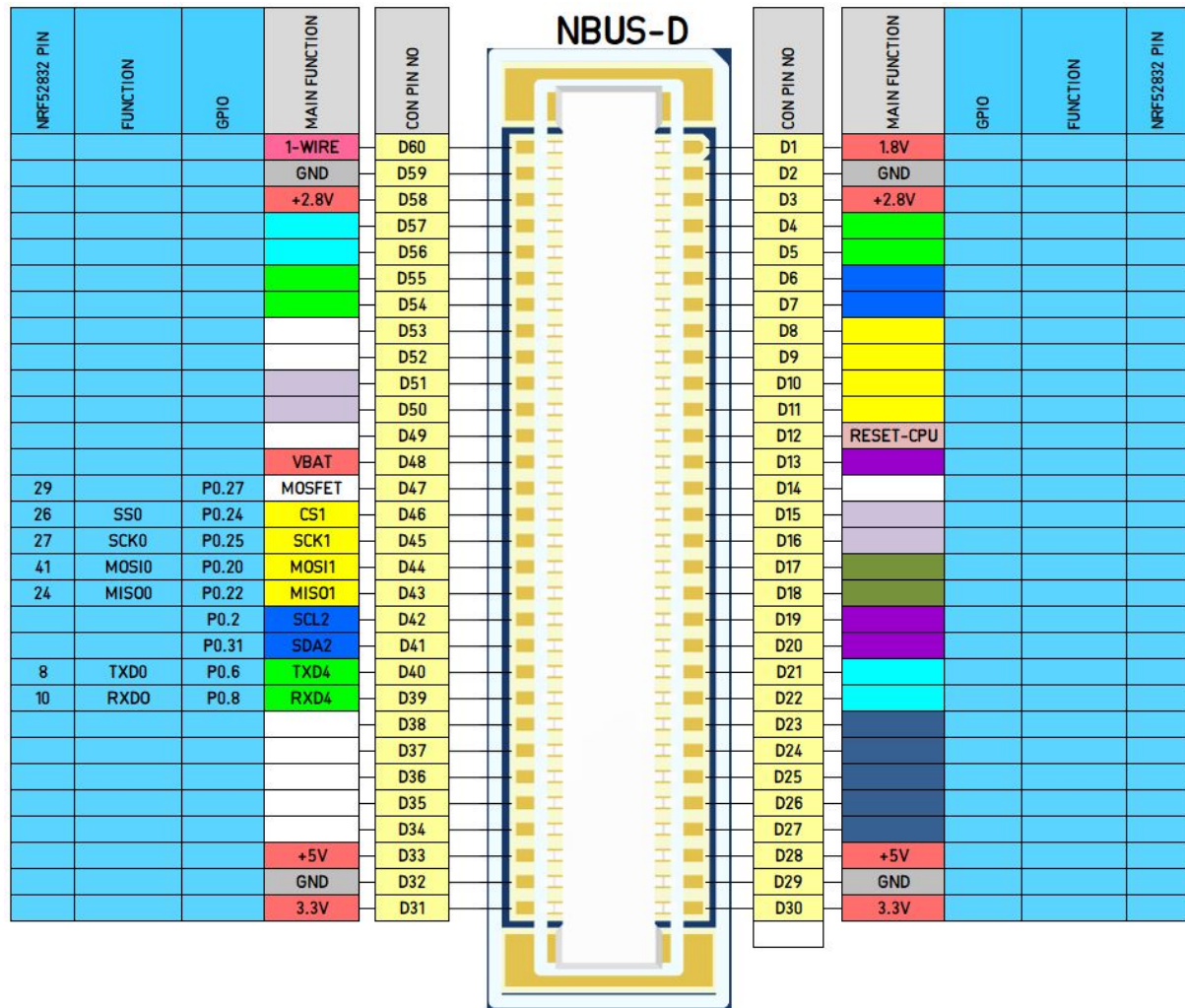
n-PRO-30 is a **HOST** board with four Hirose DF30-series 60-pin low profile connectors at bottom side, following the [n-Blocks PRO form factor](#).











Getting started

Blinky using Nordic SDK

- Follow the first part of this [tutorial](#)
- Download and install [GNU Arm Embedded Toolchain](#)
- Update the PATH variable(s) to include the GNU tools, something like:

```
C:\Program Files (x86)\GNU Tools ARM Embedded\7 2017-q4-major\bin
```

- Test the path by typing

```
arm-none-eabi-gcc --version
```

- Download and UNZIP in a convenient directory [nRF5 SDK Software Development Kit for the nRF51 Series and nRF52](#)
- Download and UNZIP in a convenient directory the build tools from [here](#)
- Update the PATH to include Build Tools directory, or copy the .exe files from .bin directory to blinky project directory
- Set the toolchain path in **makefile.windows**, located in

```
<SDK>/components/toolchain/gcc
```

- It should be something like:

```
GNU_INSTALL_ROOT := $(PROGFILES)/GNU Tools ARM Embedded/4.9 2015q3
// Toolchain path
GNU_VERSION := 4.9.3
GNU_PREFIX := arm-none-eabi
```

- Copy the .exe files from .bin directory to blinky project directory
- Edit main.c, changing the GPIO line that is connected to the LED:

```
#include <stdbool.h>
#include <stdint.h>
#include "nrf_delay.h"
#include "boards.h"

int main(void)
{
    NRF_GPIO->DIRSET = (1 << 17); // GPIO17 output, n-DAP LED

    /* Toggle LED */
    while (true)
    {
        NRF_GPIO->OUTSET = (1 << 17); // GPIO17 HIGH
        nrf_delay_ms(100);
        NRF_GPIO->OUTCLR = (1 << 17); // GPIO17 LOW
        nrf_delay_ms(100);
    }
}
```

- Open command-line and change directory to

```
<SDK>\examples\peripheral\blinky\pca10040\blank\armgcc
```

- Run

make

- Program the Flash Memory n-BLE with the created `.bin` file:

```
F:\prj_soft\nRF5_SDK_12.3.0_d7731ad\examples\peripheral\blinky\pca10040\blank\armgcc\_build\nrf52832_xxaa.bin
```

Blinky with mbed online compiler

- Select target `Nordic nRF52-DK`
- Create a blinky program like below

```
#include "mbed.h"

DigitalOut led1(P0_17);

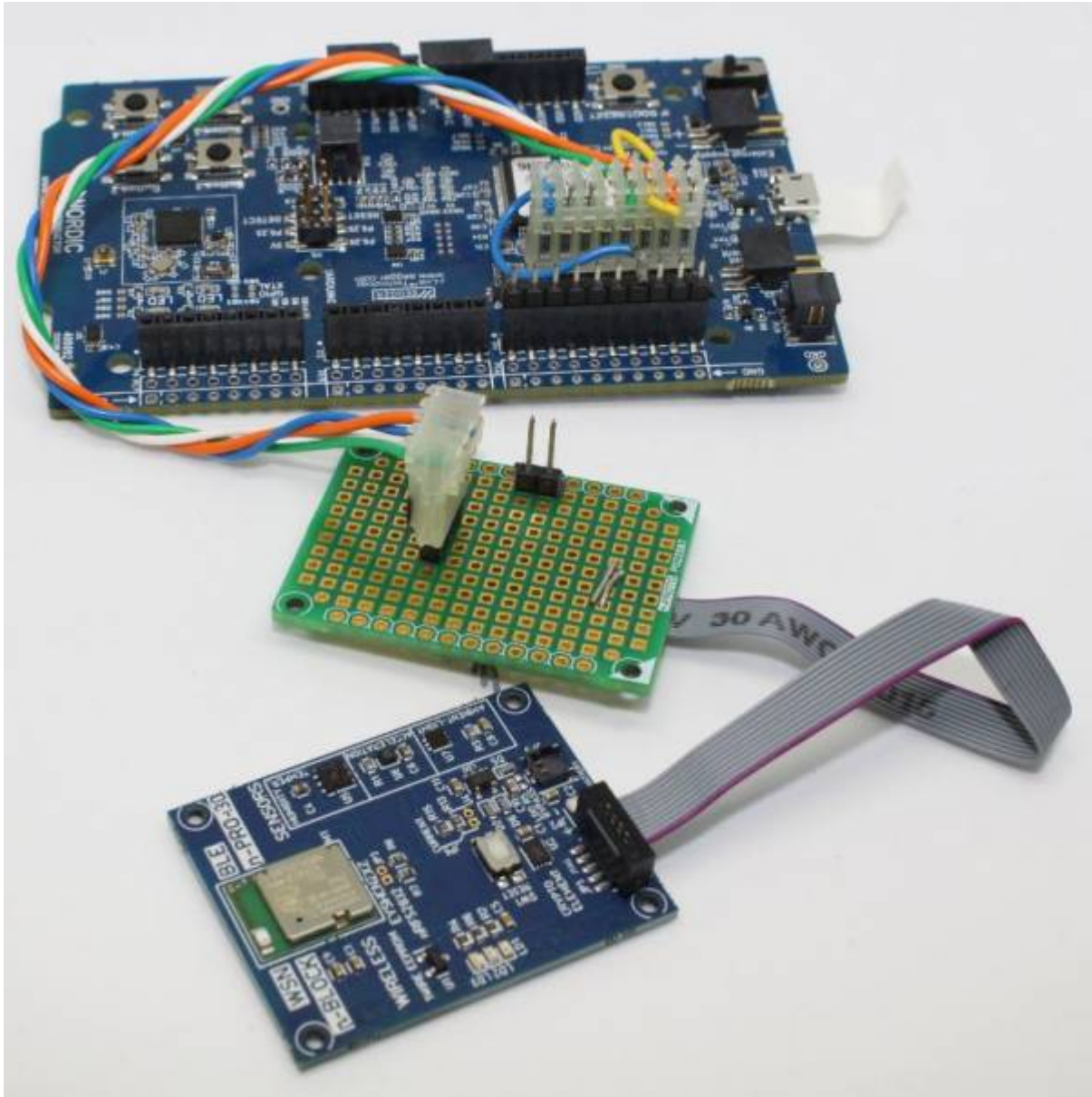
// main() runs in its own thread in the OS
int main() {
    while (true) {
        led1 = !led1;
        wait(0.05);
    }
}
```

- Program the Flash memory with the created `.hex` file

Flash Memory Programming

Programming with Nordic nRF52-DK board

- `nRF52-DK` can be used as SWD programmer.
- A DIY adaptation cable is needed.



- nRF52-DK appears as a USB DISK. Just drag and drop the file created from compiler to the disk.
- nRF52-DK can use both `.hex` and `.bin` files.
- mbed on-line compiler produces `.hex`, while GCC produces `.bin` and `.hex`

USE-CASE examples

- BLE Beacon Humidity & Temperature Sensor
- BLE Beacon Environmental Sensor
- LoRa Sensor with BLE Commissioning
- BLE simple Beacon for Localisation

Related articles in this Wiki

- [n-pro-30](#)

[RF](#), [CPU](#), [nblock](#), [BLE](#), [nsensorRF](#)

IMPORTANT NOTICE - PLEASE READ CAREFULLY

Nimbus Centre reserve the right to make changes, corrections, enhancements, modifications, and improvements to Nimbus Centre products and/or to this document at any time without notice.

All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.



Address: Cork Institute of Technology
Campus, Bishopstown, Cork

Phone: (021) 433 5560

© 2019 Nimbus Centre - All rights reserved