



**DesignNews**

## Developing IoT Applications with Nordic nRF Modules

**Day 4:**

**A Custom BLE Design Using the Raytac MDBT50Q Module**

Sponsored by

**DigiKey**



## Webinar Logistics

- Turn on your system sound to hear the streaming presentation.
- If you have technical problems, click “Help” or submit a question asking for assistance.
- Participate in ‘Attendee Chat’ by maximizing the chat widget in your dock.

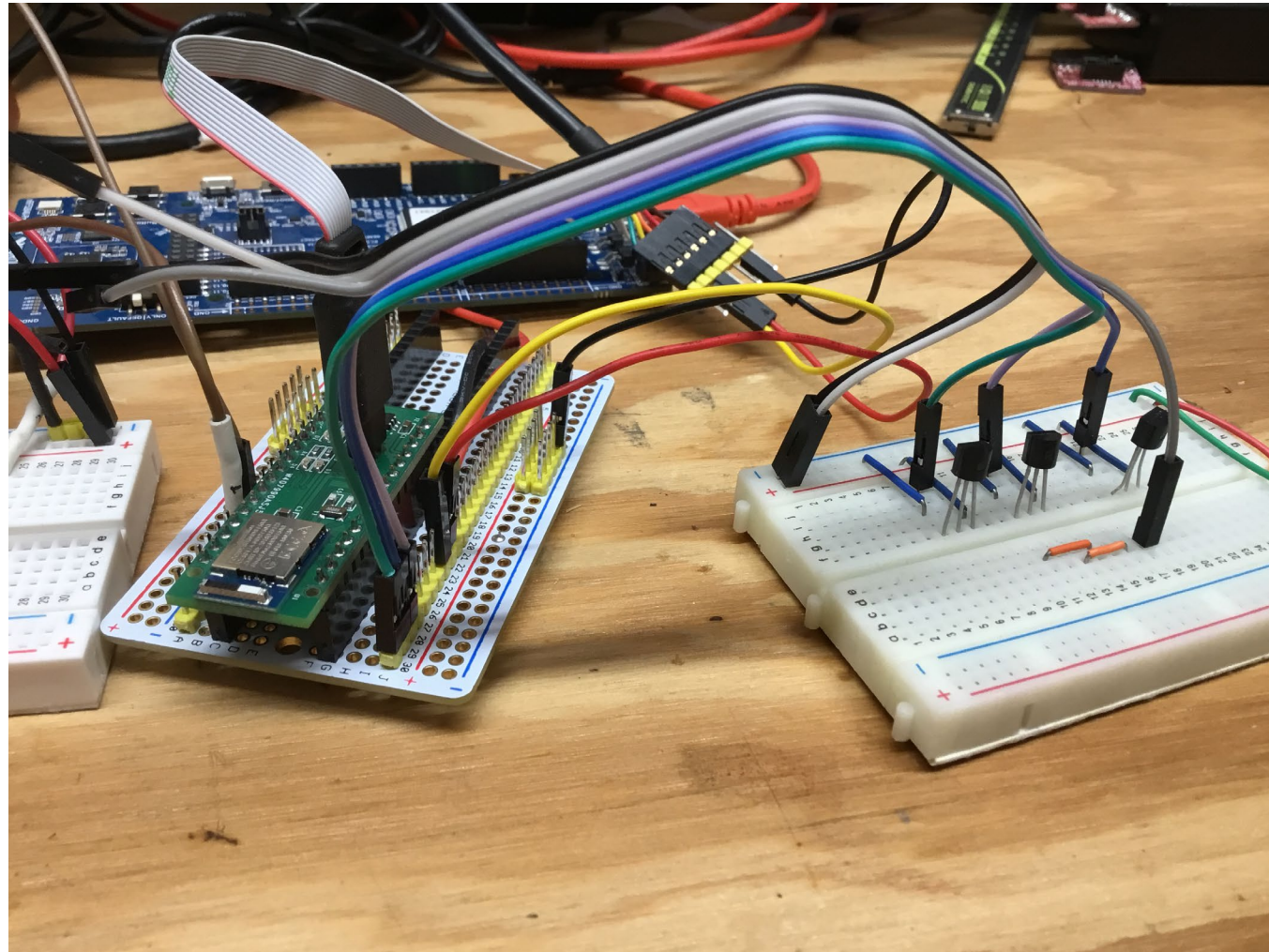


## Fred Eady

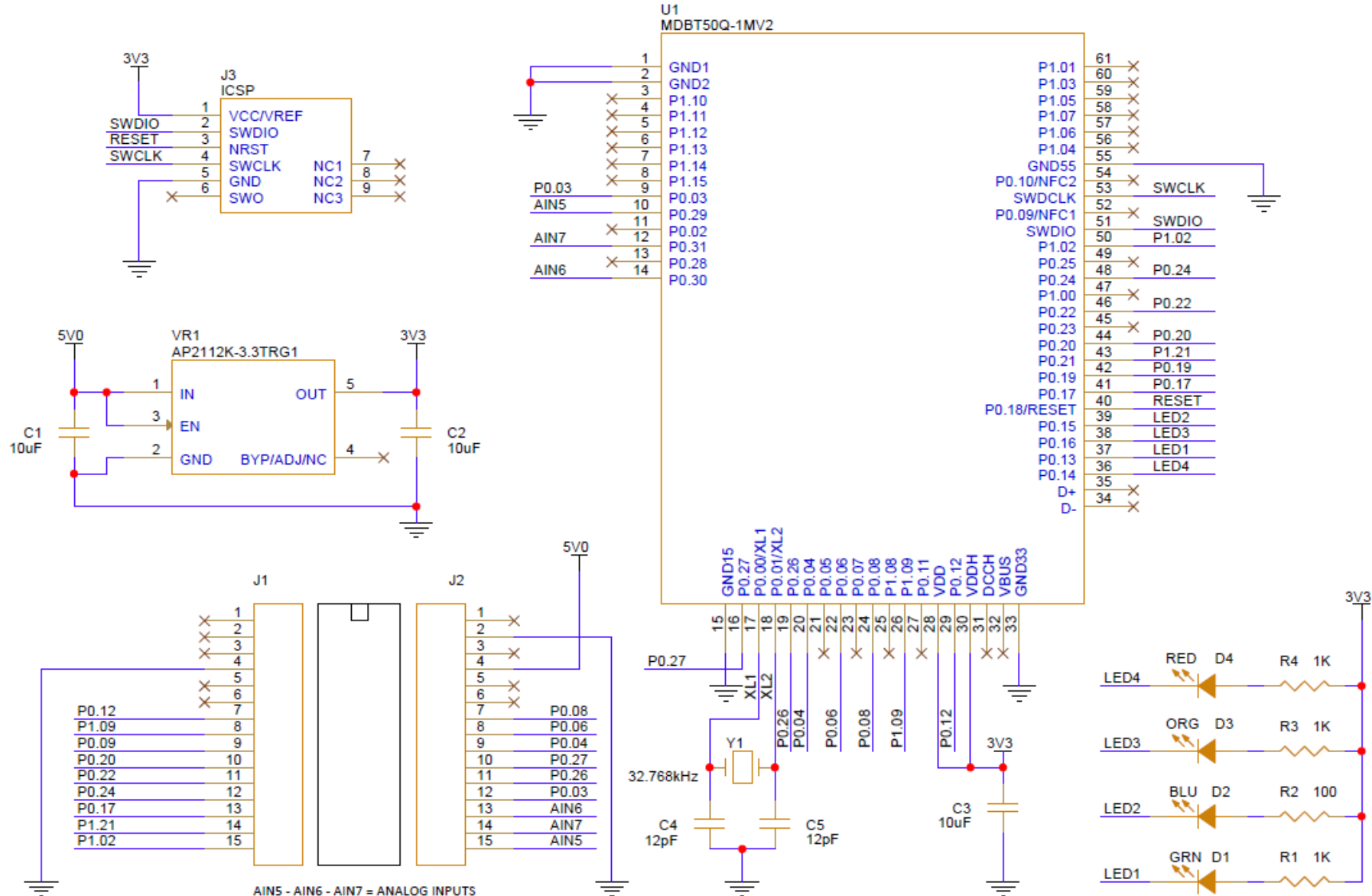
Visit 'Lecturer Profile' in your console for more details.

# AGENDA

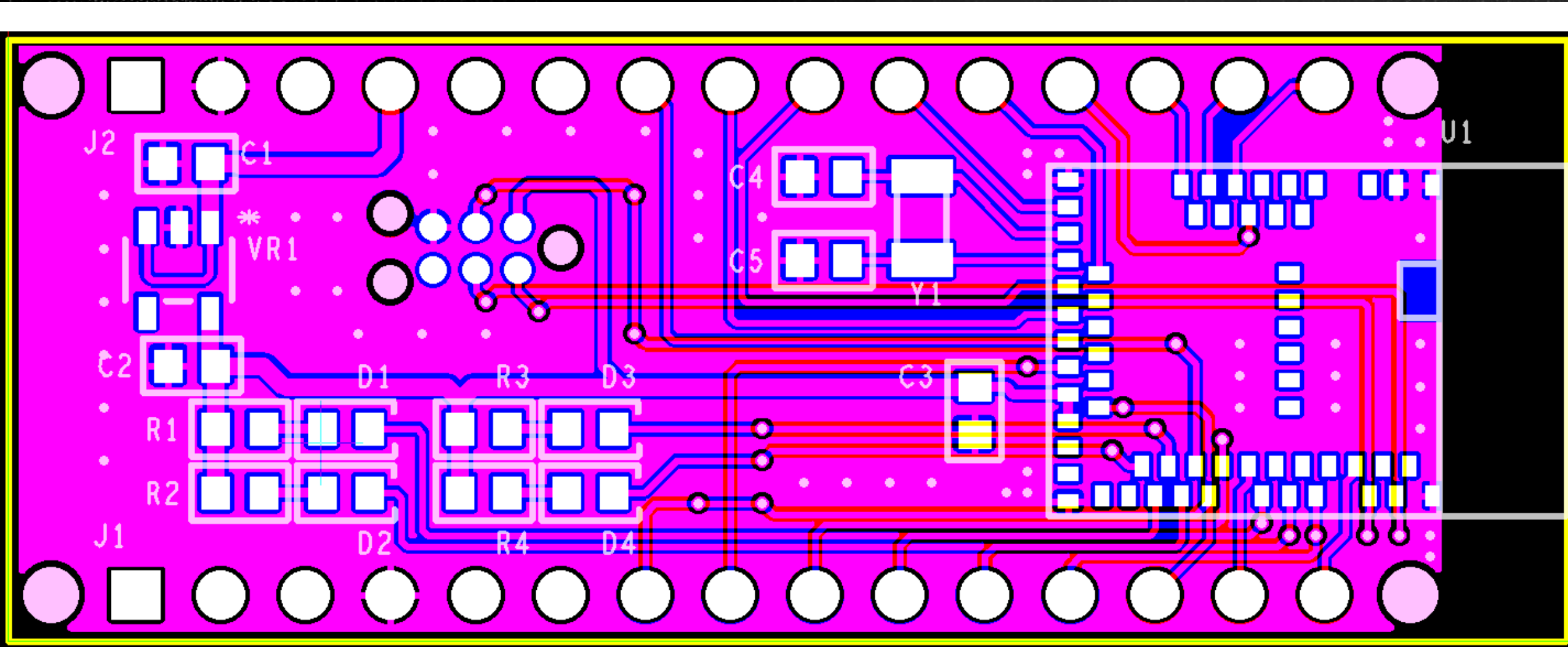
- **MDBT50Q Hardware Design**
- **BLE Firmware Customization**



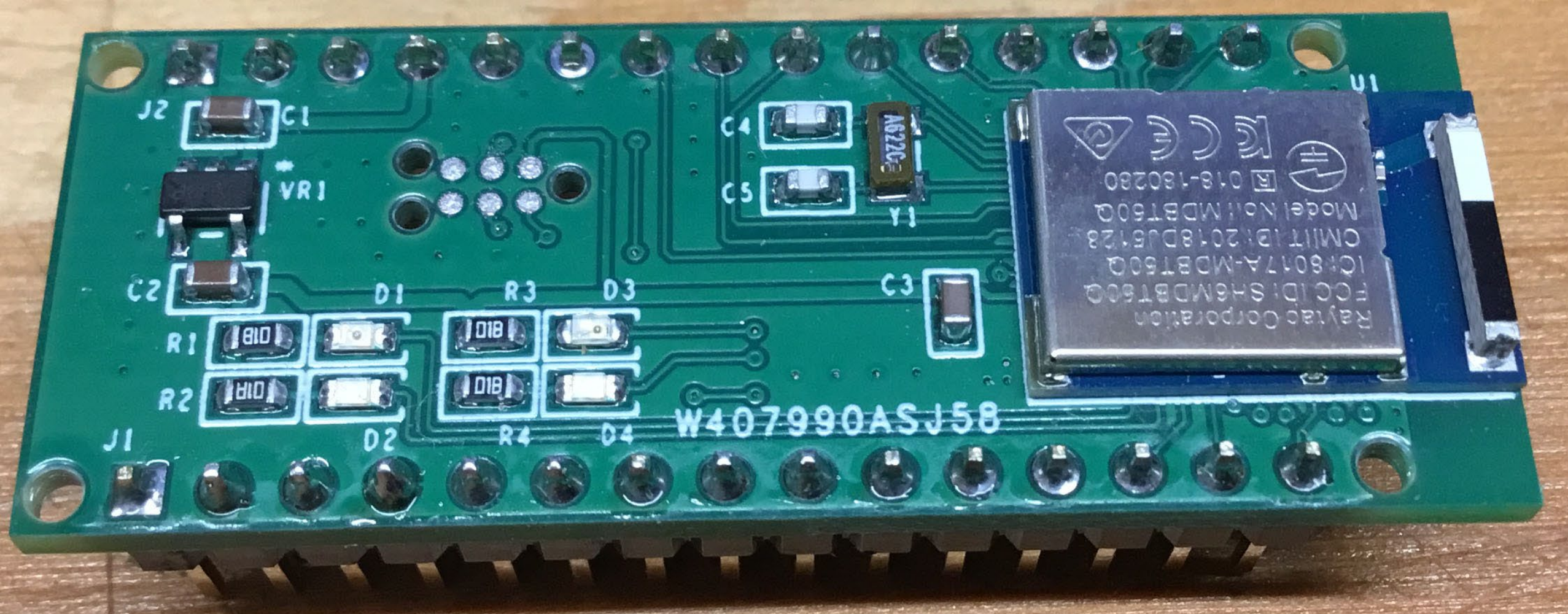
## MDBT50Q Custom Board Schematic

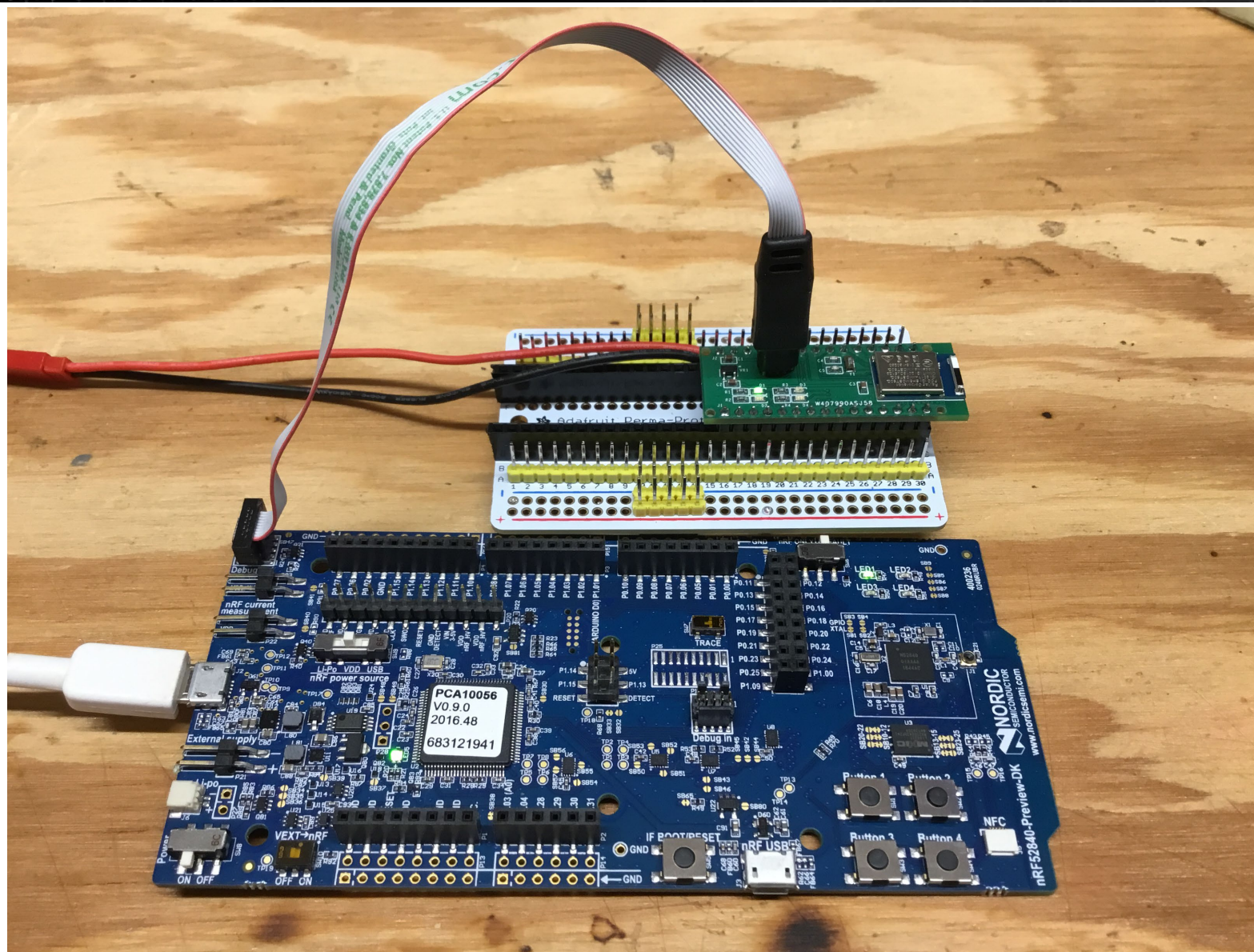


# MDBT50Q Custom Board Gerber View



# MDBT50Q Custom Board In the Flesh



**MDBT50Q Custom Hardware Development**



## Create a New Custom Board

<p>NRF CONNECT ...</p> <p>WELCOME</p> <p>Manage toolchains v2.6.0 (newest)</p>		<p>Create New Board (1/5)</p> <p>cec_mdbt50q</p> <p>Press 'Enter' to confirm your input or 'Escape' to cancel</p>
<p>NRF CONNECT ...</p> <p>WELCOME</p> <p>Manage toolchains v2.6.0 (newest)</p> <p>Manage SDKs v2.6.0 (newest)</p>		<p>Create New Board (2/5)</p> <p>cec_mdbt50q</p> <p>Enter a machine-readable board ID, to be referenced by the west build command. (Press 'Enter' to confirm or 'Escape' to cancel)</p>
<p>NRF CONNECT ...</p> <p>WELCOME</p> <p>Manage toolchains v2.6.0 (newest)</p> <p>Manage SDKs v2.6.0 (newest)</p>		<p>Create New Board (3/5)</p> <p>Select the SoC device family and the device variant...</p> <p>nrf52840 QIAA</p> <p>nrf52840 QFAA</p>
<p>NRF CONNECT ...</p> <p>WELCOME</p> <p>Manage toolchains v2.6.0 (newest)</p> <p>Manage SDKs v2.6.0 (newest)</p>		<p>Create New Board (4/5)</p> <p>/home/fred/nrf_day4</p> <p>Set board root directory (Press 'Enter' to confirm or 'Escape' to cancel)</p>
<p>NRF CONNECT ...</p> <p>WELCOME</p> <p>Manage toolchains v2.6.0 (newest)</p> <p>Manage SDKs v2.6.0 (newest)</p> <p>Open an existing application</p> <p>Create a new application</p> <p>Create a new board</p> <p>Browse samples</p>		<p>Create New Board (5/5)</p> <p>edtp</p> <p>Press 'Enter' to confirm your input or 'Escape' to cancel</p>

## Create a New Custom Board

EXPLORER ... **cec\_mdbt50q\_defconfig** X

boards > arm > cec\_mdbt50q > **cec\_mdbt50q\_defconfig**

```

1 # Copyright (c) 2024 Nordic Semiconductor ASA
2 # SPDX-License-Identifier: Apache-2.0
3
4 CONFIG_SOC_SERIES_NRF52X=y
5 CONFIG_SOC_NRF52840_QIAA=y
6 CONFIG_BOARD_CEC_MDBT50Q=y
7
8 # Enable MPU
9 CONFIG_ARM_MPU=y
10
11 # Enable hardware stack protection
12 CONFIG_HW_STACK_PROTECTION=y

```

OPEN EDITORS

- cec\_mdbt50q\_defconfig boards...
- NRF\_DAY4
  - boards/arm/cec\_mdbt50q
    - board.cmake
    - cec\_mdbt50q\_defconfig**
    - cec\_mdbt50q.dts
    - cec\_mdbt50q.yaml
    - Kconfig.board
    - Kconfig.defconfig

EXPLORER ... **cec\_mdbt50q.dts** X

boards > arm > cec\_mdbt50q > **cec\_mdbt50q.dts** > ...

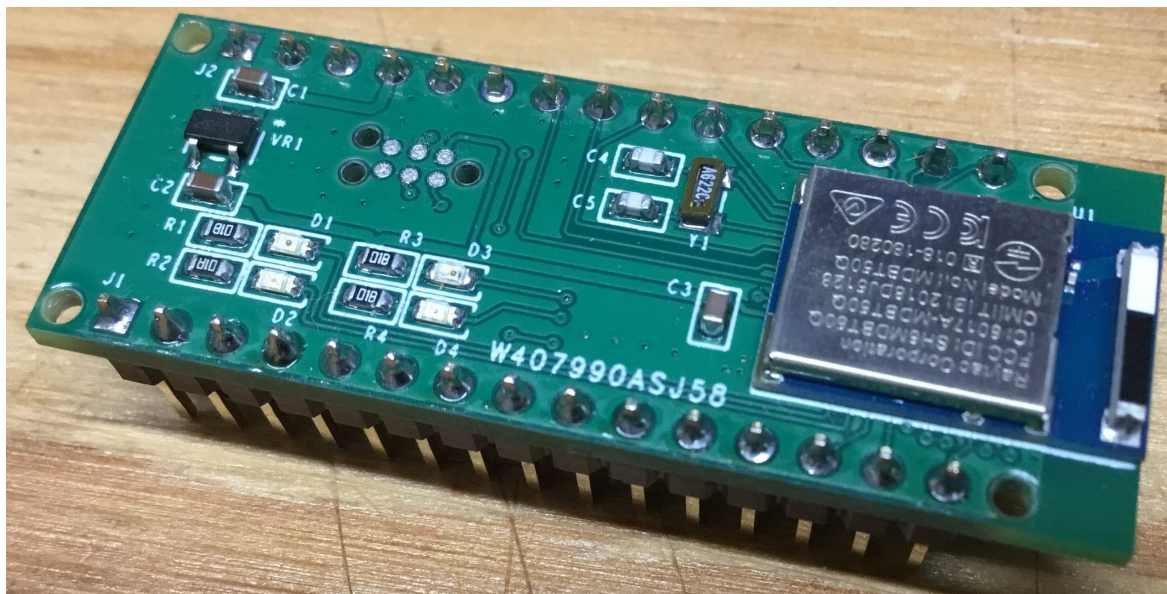
```

1 // Copyright (c) 2024 Nordic Semiconductor ASA
2 // SPDX-License-Identifier: Apache-2.0
3
4 /dts-v1/;
5 #include <nordic/nrf52840_qiaa.dtsi>
6
7 / {
8     model = "cec_mdbt50q";
9     compatible = "edtp,cec-mdbt50q";
10
11     chosen {
12         zephyr,sram = &sram0;
13         zephyr,flash = &flash0;
14         zephyr,code-partition = &slot0_partition;
15     };
16 };
17
18 &flash0 {
19     partitions {
20         compatible = "fixed-partitions";
21         #address-cells = <1>;
22         #size-cells = <1>;
23
24         boot_partition: partition@0 {
25             label = "mcuboot";
26             reg = <0x0 0xc000>;
27         };
28         slot0_partition: partition@c000 {
29             label = "image-0";
30             reg = <0xc000 0x72000>;
31         };
32         slot1_partition: partition@7e000 {
33             label = "image-1";
34             reg = <0x7e000 0x72000>;
35         };
36         scratch_partition: partition@f0000 {
37             label = "image-scratch";
38             reg = <0xf0000 0xa000>;
39         };
40         storage_partition: partition@fa000 {
41             label = "storage";
42             reg = <0xfa000 0x6000>;
43         };
44     };
45 };

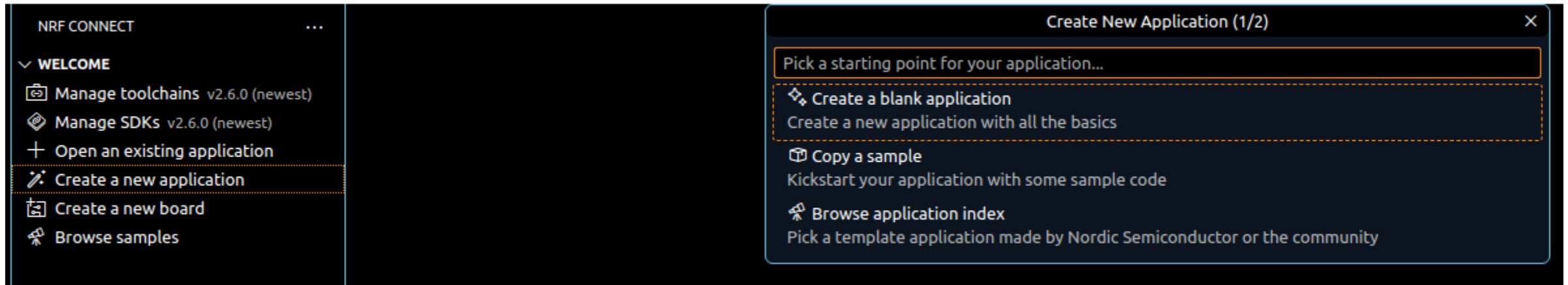
```

OPEN EDITORS

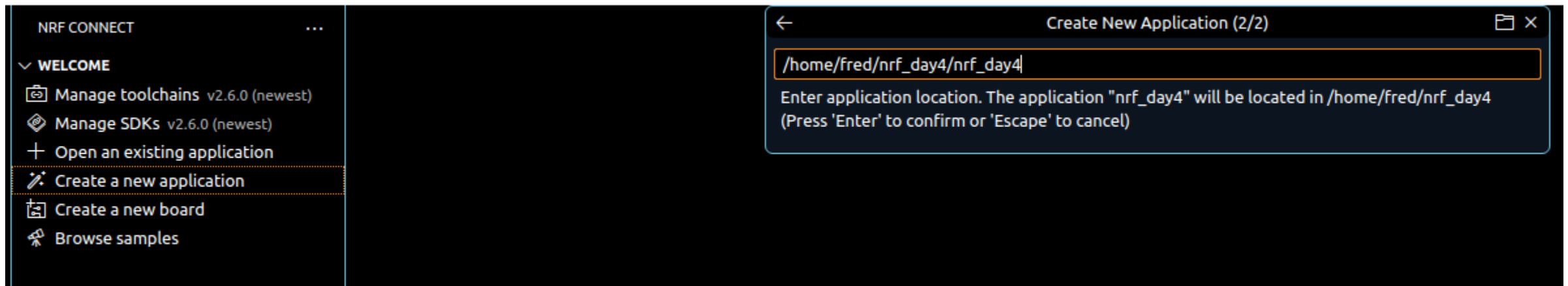
- cec\_mdbt50q.dts boards/arm/ce...
- NRF\_DAY4
  - boards/arm/cec\_mdbt50q
    - board.cmake
    - cec\_mdbt50q\_defconfig
    - cec\_mdbt50q.dts**
    - cec\_mdbt50q.yaml
    - Kconfig.board
    - Kconfig.defconfig



## Create a New Application



The screenshot shows the NRF CONNECT application interface. On the left, a sidebar menu is visible with the following options: Manage toolchains v2.6.0 (newest), Manage SDKs v2.6.0 (newest), Open an existing application, Create a new application (highlighted with a dashed border), Create a new board, and Browse samples. The main area displays a dialog box titled "Create New Application (1/2)". The dialog contains a search bar with the placeholder text "Pick a starting point for your application...". Below the search bar, three options are listed: "Create a blank application" (with a diamond icon), "Copy a sample" (with a folder icon), and "Browse application index" (with a magnifying glass icon). The "Create a blank application" option is selected and highlighted with a dashed border. The text below this option reads: "Create a new application with all the basics".



The screenshot shows the NRF CONNECT application interface. On the left, the sidebar menu is the same as in the previous screenshot, with "Create a new application" highlighted. The main area displays a dialog box titled "Create New Application (2/2)". The dialog contains a text input field with the path `/home/fred/nrf_day4/nrf_day4` entered. Below the input field, the text reads: "Enter application location. The application 'nrf\_day4' will be located in /home/fred/nrf\_day4 (Press 'Enter' to confirm or 'Escape' to cancel)".

## Create a New Application

The screenshot shows the Visual Studio Code interface with the 'Add Build Configuration' dialog open for a project named 'nrf\_day4'. The dialog is titled 'Add Build Configuration' and contains the following fields and options:

- Board:** A dropdown menu set to 'cec\_mdbt50q'. To its right is a 'Revision' field with a question mark icon.
- Configuration:** A dropdown menu set to 'Use build system default'.
- Kconfig fragments:** A section with an 'Add fragment' button.
- Devicetree overlays:** A section with an 'Add overlay' button.
- Snippets:** A section with an 'Add snippet' button.
- Extra CMake arguments:** A section with an 'Add argument' button.
- Build directory name:** A text input field containing 'build'.
- Optimization level:** A dropdown menu set to 'Use project default'.
- Build after generating configuration:** A checked checkbox.
- Use sysbuild:** An unchecked checkbox.

At the bottom right of the dialog is a 'Build Configuration' button. The background shows the Visual Studio Code interface with the 'nrf\_day4' project selected in the left sidebar.

## Create a New Application

The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left. The Explorer sidebar is expanded to show the 'nrf\_day4' project structure, including 'Source files', 'Config files', and 'Output files'. The 'ACTIONS' section is expanded to show 'Build', 'Debug', 'Flash', 'Devicetree Board File', 'nRF Kconfig GUI', and 'Memory report'. The terminal window at the bottom shows the output of a build command, including the path to the nRF Connect SDK and the successful completion of the build.

```
[12:53:20] Update available https://nsscprodmedia.blob.core.windows.net/prod/software-and-other-downloads/desktop-software/nrf-command-line-tools/sw/versions-10-x-x/10-24-1/nrf-command-line-tools\_10.24.1\_amd64.deb
[12:59:57] west build --build-dir /home/fred/nrf_day4/nrf_day4/build /home/fred/nrf_day4/nrf_day4 --pristine --board cec_mdbt50q --no-sysbuild -- -DNCS_TOOLCHAIN_VERSION=NONE -DBOARD_ROOT=/home/fred/nrf_day4
[13:00:07] ===== Toolchain validation report for nRF Connect SDK Toolchain v2.6.0 =====
🍌 Using the nRF Connect SDK v2.6.0 at /home/fred/ncs/v2.6.0
🕒 dtc is outdated. Found 1.5.0, but 1.5.1-1 is required by the SDK.
[13:00:12] Success.
```

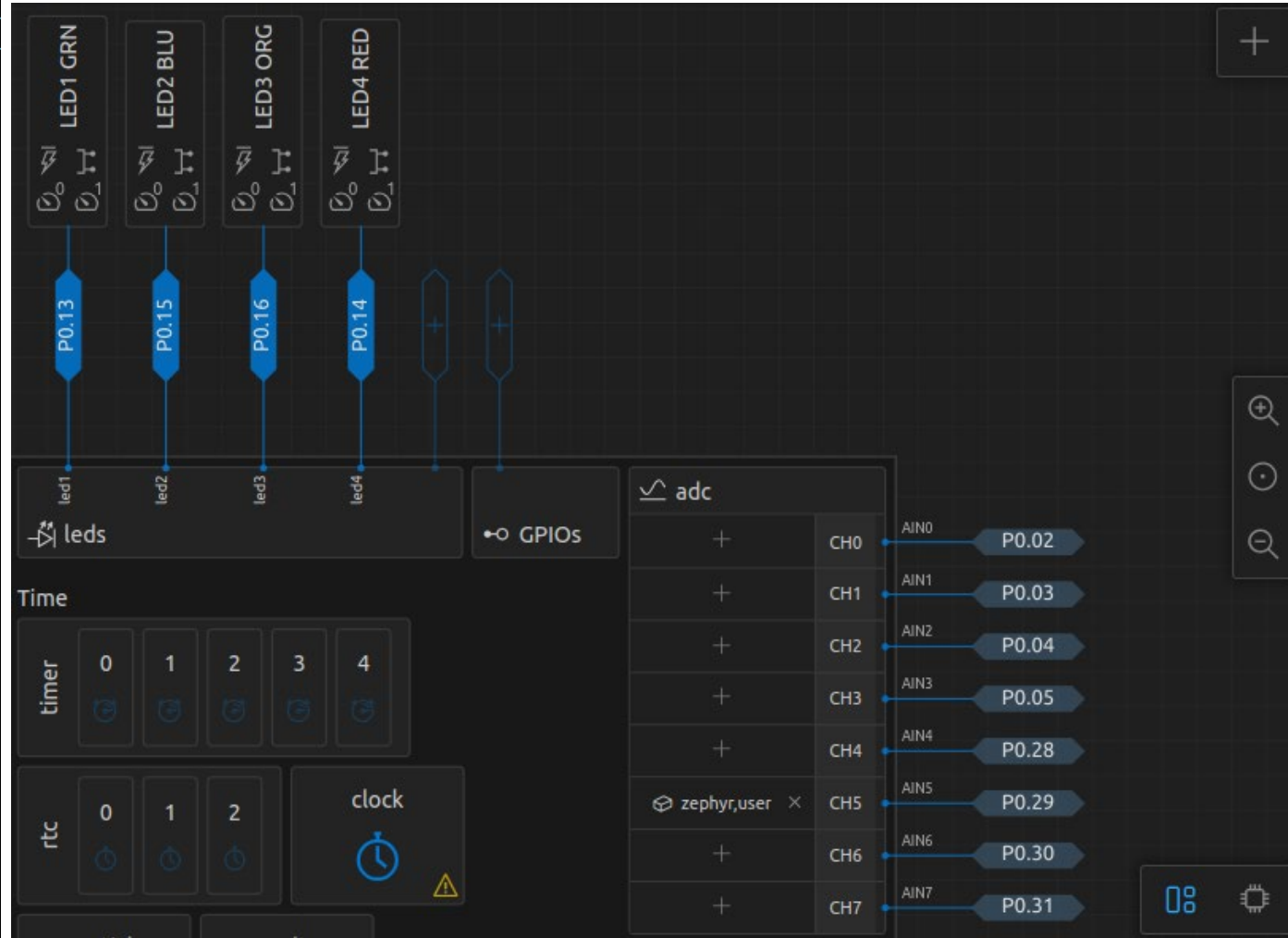
## Customize the Devicetree

```

EXPLORER
...
boards > arm > cec_mdbt50q > cec_mdbt50q.dts > / > aliases

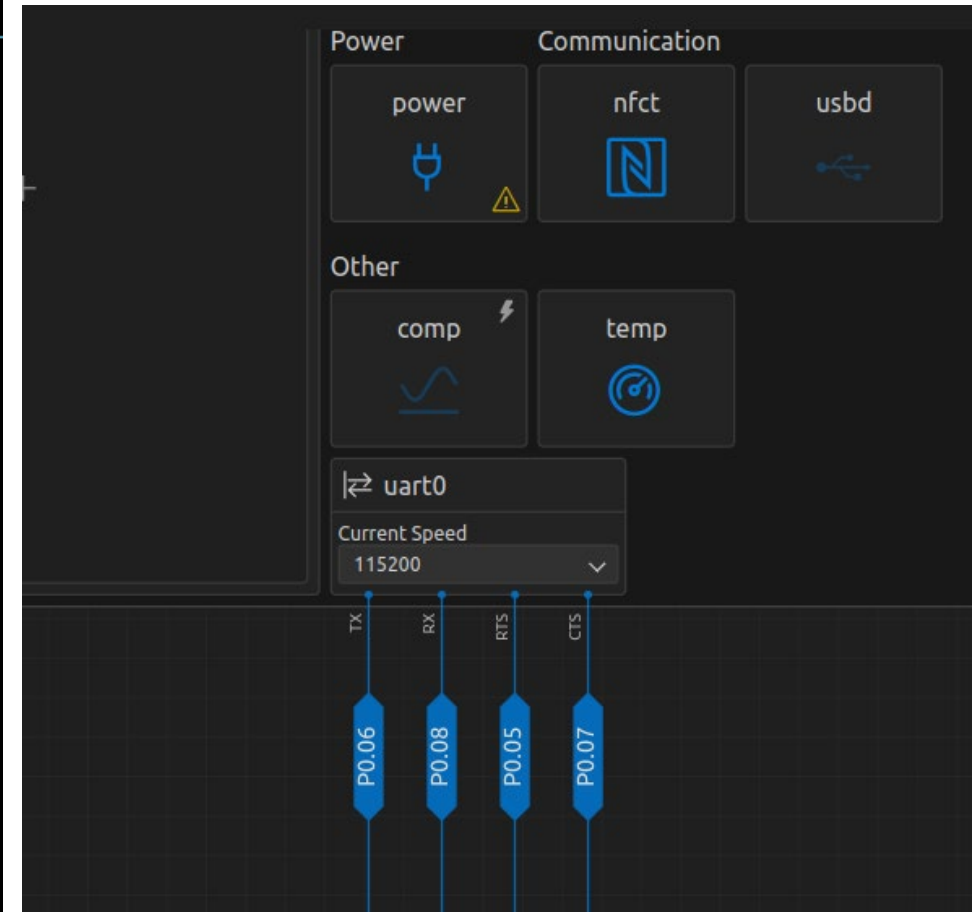
OPEN EDITORS
x cec_mdbt50q.dts boards/... 3
NRF_DAY4
> .vscode
> boards/arm/cec_mdbt50q
  board.cmake
  cec_mdbt50q_defconfig
  cec_mdbt50q.dts 3
  cec_mdbt50q.yaml
  Kconfig.board
  Kconfig.defconfig
> nrf_day4
  > build
  > src
    C main.c
    .gitignore
    CMakeLists.txt
    prj.conf

boards > arm > cec_mdbt50q > cec_mdbt50q.dts > / > aliases
4 /dts-v1/;
5 #include <nordic/nrf52840_qiaa.dtsi>
6 #include "cec_mdbt50q-pinctrl.dtsi"
7
8 / {
9     model = "cec_mdbt50q";
10    compatible = "edtp,cec-mdbt50q";
11
12    chosen {
13        zephyr,shell-uart = &uart0;
14        zephyr,console = &uart0;
15        zephyr,uart-mcumgr = &uart0;
16        zephyr,bt-mon-uart = &uart0;
17        zephyr,bt-c2h-uart = &uart0;
18        zephyr,sram = &sram0;
19        zephyr,flash = &flash0;
20        zephyr,code-partition = &slot0_partition;
21    };
22
23    leds {
24        compatible = "gpio-leds";
25        led1: led1 {
26            gpios = <&gpio0 13 GPIO_ACTIVE_LOW>;
27            label = "LED1 GRN";
28        };
29        led2: led2 {
30            gpios = <&gpio0 15 GPIO_ACTIVE_LOW>;
31            label = "LED2 BLU";
32        };
33        led3: led3 {
34            gpios = <&gpio0 16 GPIO_ACTIVE_LOW>;
35            label = "LED3 ORG";
36        };
37        led4: led4 {
38            gpios = <&gpio0 14 GPIO_ACTIVE_LOW>;
39            label = "LED4 RED";
40        };
41    };
42
  
```



## Create and Customize pinctrl

```
EXPLORER
...
ceci_mdbt50q-pinctrl.dtsi x
boards > arm > ceci_mdbt50q > ceci_mdbt50q-pinctrl.dtsi > ...
4  */
5
6  &pinctrl {
7      uart0_default: uart0_default {
8          group1 {
9              psels = <NRF_PSEL(UART_TX, 0, 6)>,
10             <NRF_PSEL(UART_RTS, 0, 5)>;
11          };
12          group2 {
13              psels = <NRF_PSEL(UART_RX, 0, 8)>,
14              <NRF_PSEL(UART_CTS, 0, 7)>;
15              bias-pull-up;
16          };
17      };
18
19      uart0_sleep: uart0_sleep {
20          group1 {
21              psels = <NRF_PSEL(UART_TX, 0, 6)>,
22              <NRF_PSEL(UART_RX, 0, 8)>,
23              <NRF_PSEL(UART_RTS, 0, 5)>,
24              <NRF_PSEL(UART_CTS, 0, 7)>;
25              low-power-enable;
26          };
27      };
28
```



## Customize prj.conf

```
1  #*****
2  #* A Custom BLE Design Using the Raytac MDBT50Q Module
3  #* REV 1.0.0
4  #* LAST UPDATE 04-05-2024
5  #* NOTES:
6  #*****
7
8  # CONFIG FOR LOGGING
9  # These settings can alternately be entered in cec_mdbt50q_defconfig
10 CONFIG_CONSOLE=y
11 CONFIG_SERIAL=y
12 CONFIG_UART_CONSOLE=y
13
14 #CONFIG ADC and GPIO
15 CONFIG_GPIO=y
16 CONFIG_ADC=y
17
18 # Bluetooth LE
19 CONFIG_BT=y
20 CONFIG_BT_PERIPHERAL=y
21 CONFIG_BT_DEVICE_NAME="CEC_BLE_DAY4"
22
23 # CONFIG LOGGER                25  #include <zephyr/logging/log.h>
24 CONFIG_LOG=y                    26
25 CONFIG_USE_SEGGER_RTT=y        27  LOG_MODULE_DECLARE(bleday4, LOG_LEVEL_DBG);
26 CONFIG_LOG_BACKEND_RTT=y
27 CONFIG_LOG_BACKEND_UART=y
28 CONFIG_LOG_PRINTK=y
```

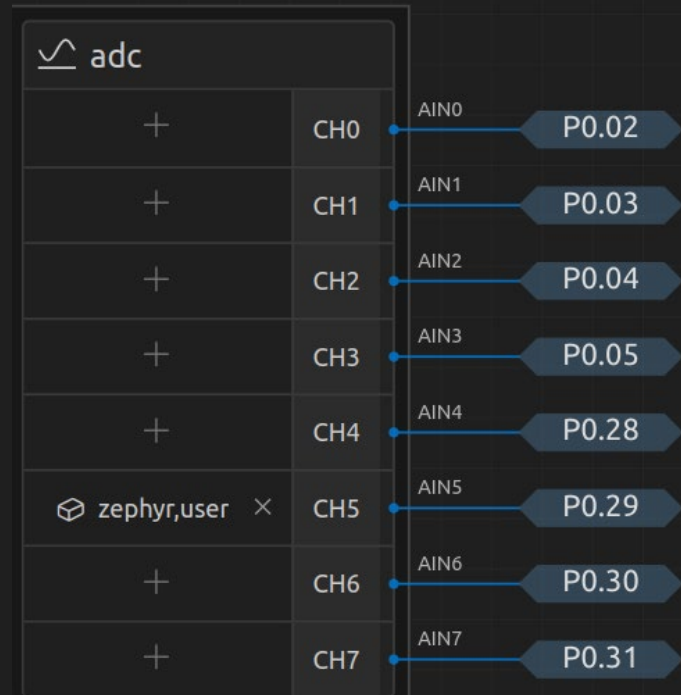


Create the `cec_mdbt50q.overlay`

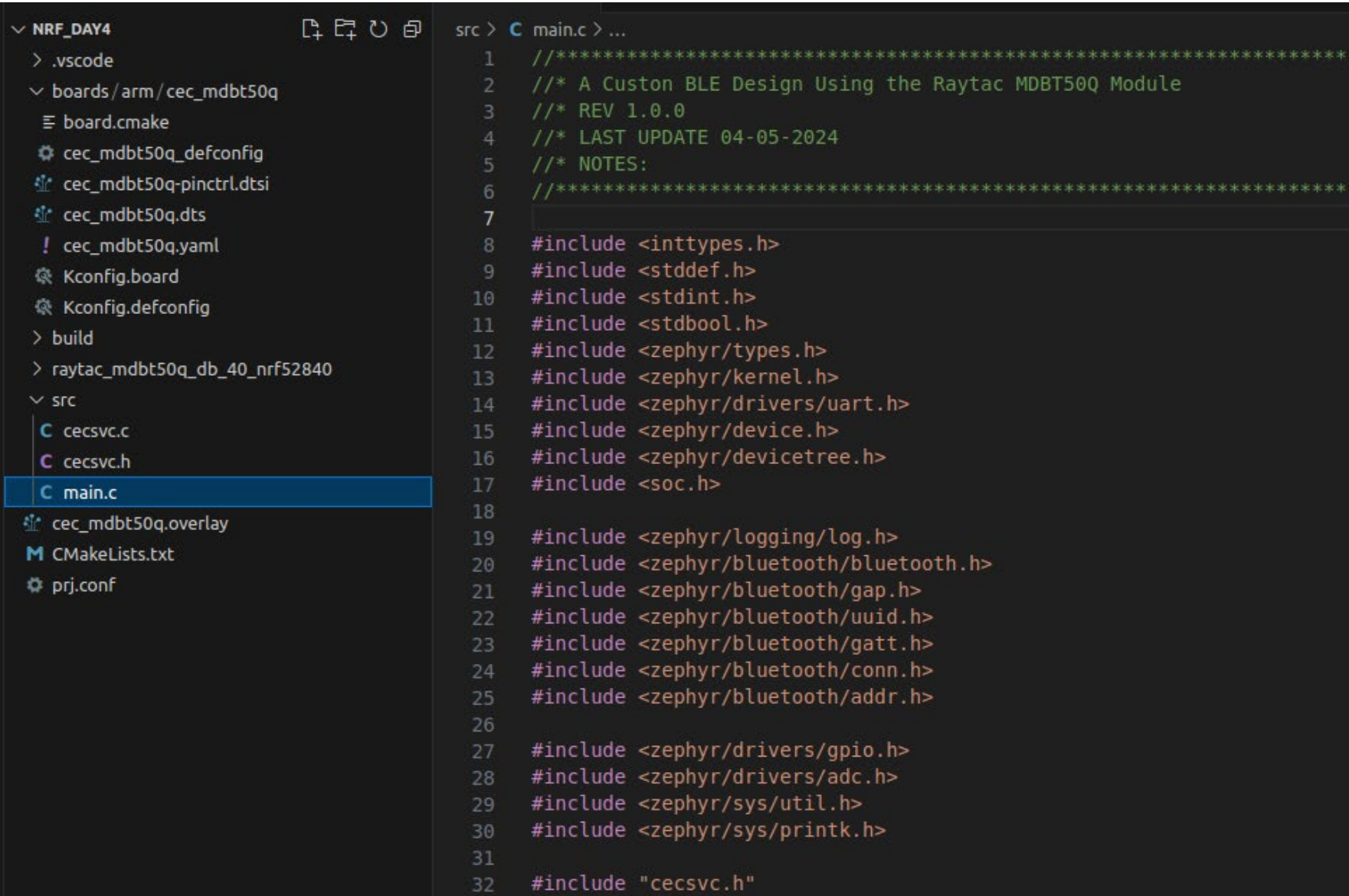
```

1 // LAST UPDATE: 04-05-2024
2 // To get started, press Ctrl+Space to bring up the completion menu and view the available nodes.
3
4 // You can also use the buttons in the sidebar to perform actions on nodes.
5 // Actions currently available include:
6
7 // * Enabling / disabling the node
8 // * Adding the bus to a bus
9 // * Removing the node
10 // * Connecting ADC channels
11
12 // For more help, browse the DeviceTree documentation at https://docs.zephyrproject.org/latest/guides/dts/index.html
13 // You can also visit the nRF DeviceTree extension documentation at https://nrfconnect.github.io/vscode-nrf-connect/devicetree/nrfdevicetree.html
14
15 /{
16     zephyr,user {
17         io-channels = <&adc 5>;
18     };
19 };
20
21 &adc {
22     compatible = "nordic,nrf-saadc";
23     status = "okay";
24     #address-cells = <1>;
25     #size-cells = <0>;
26
27     channel@5 {
28         reg = <5>;
29         zephyr,gain = "ADC_GAIN_1_6";
30         zephyr,reference = "ADC_REF_INTERNAL";
31         zephyr,acquisition-time = <ADC_ACQ_TIME_DEFAULT>;
32         zephyr,input-positive = <NRF_SAADC_AIN5>; // P0.29
33         zephyr,resolution = <12>;
34         zephyr,oversampling = <8>;
35     };
36 };
37

```



## Copy nrf\_day3 Source Files



```
src > C main.c > ...
1  //*****
2  /* A Custom BLE Design Using the Raytac MDBT50Q Module
3  /* REV 1.0.0
4  /* LAST UPDATE 04-05-2024
5  /* NOTES:
6  //*****
7
8  #include <inttypes.h>
9  #include <stddef.h>
10 #include <stdint.h>
11 #include <stdbool.h>
12 #include <zephyr/types.h>
13 #include <zephyr/kernel.h>
14 #include <zephyr/drivers/uart.h>
15 #include <zephyr/device.h>
16 #include <zephyr/devicetree.h>
17 #include <soc.h>
18
19 #include <zephyr/logging/log.h>
20 #include <zephyr/bluetooth/bluetooth.h>
21 #include <zephyr/bluetooth/gap.h>
22 #include <zephyr/bluetooth/uuid.h>
23 #include <zephyr/bluetooth/gatt.h>
24 #include <zephyr/bluetooth/conn.h>
25 #include <zephyr/bluetooth/addr.h>
26
27 #include <zephyr/drivers/gpio.h>
28 #include <zephyr/drivers/adc.h>
29 #include <zephyr/sys/util.h>
30 #include <zephyr/sys/printk.h>
31
32 #include "cecsvc.h"
```

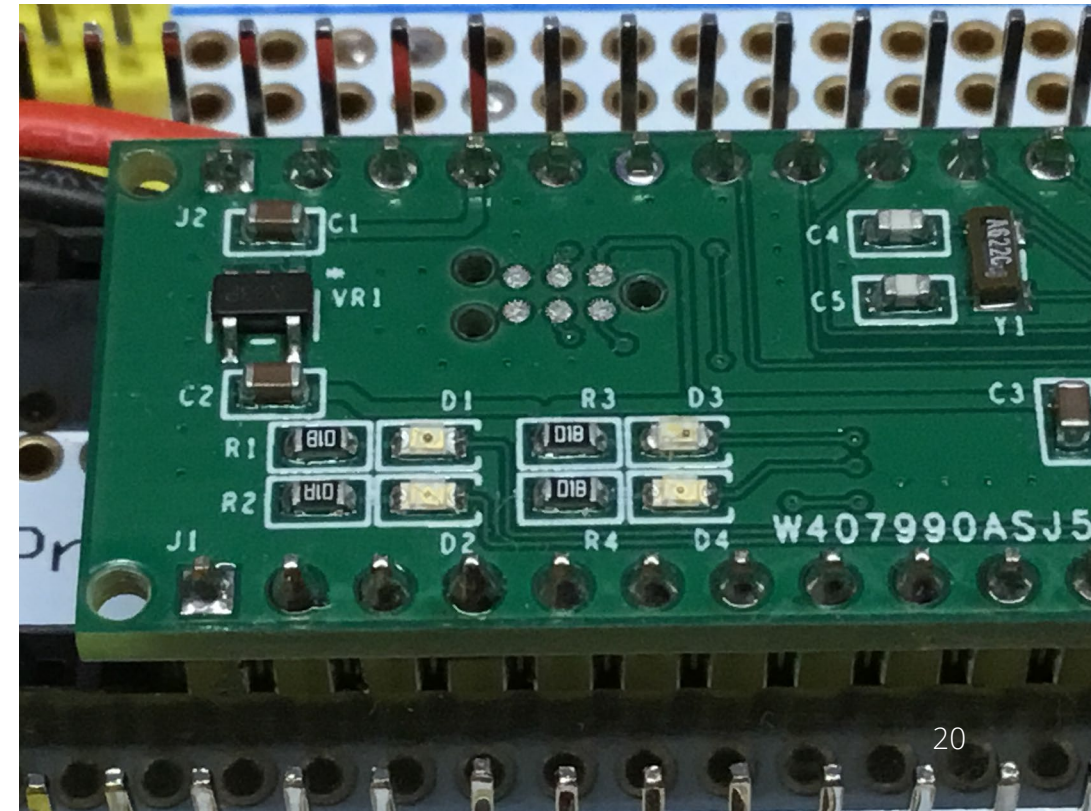
## Add BLE Notify Capability – Sensor Service

```
32 static uint16_t adc_val;
33 static bool status_notify;
34
35 static void status_cccd_changed(const struct bt_gatt_attr *attr, uint16_t value)
36 {
37     LOG_DBG("status_cccd_changed called");
38     status_notify = value == BT_GATT_CCC_NOTIFY;
39 }
40
41 //*****
42 /* Read Temp Sensor Callback - No Changes
43 //*****
44 //*****
45 /* Create and add the SENSOR service to the Bluetooth LE stack
46 //*****
47 BT_GATT_SERVICE_DEFINE(sensor_svc,
48 BT_GATT_PRIMARY_SERVICE(BT_UUID_SENSORSVC),
49 // Create and add the sensor characteristic
50 BT_GATT_CHARACTERISTIC(BT_UUID_SENSORCHR,
51     BT_GATT_CHRC_READ | BT_GATT_CHRC_NOTIFY,
52     BT_GATT_PERM_READ, read_temp_sensor, NULL,
53     &adc_val),
54 BT_GATT_CCC(status_cccd_changed,
55     BT_GATT_PERM_READ | BT_GATT_PERM_WRITE),
56 );
57
58
59 void status_nfy(struct bt_conn *conn, uint16_t adcValue)
60 {
61     uint8_t buf[2];
62     buf[0] = (adcValue & 0xFF00) >> 8;
63     buf[1] = (uint8_t) adcValue & 0x00FF;
64
65     bt_gatt_notify(conn, &sensor_svc.attrs[1], buf, sizeof(buf));
66 }
```

## Replace DK Library

```
68 #define RUN_STATUS_LED DT_ALIAS(led1)
69 #define CON_STATUS_LED DT_ALIAS(led2)
70 #define LED_USER DT_ALIAS(led3)
71 #define LED_CNTL DT_ALIAS(led4)
72
73 static const struct gpio_dt_spec runstatus = GPIO_DT_SPEC_GET(RUN_STATUS_LED, gpios);
74 static const struct gpio_dt_spec constatus = GPIO_DT_SPEC_GET(CON_STATUS_LED, gpios);
75 static const struct gpio_dt_spec leduser = GPIO_DT_SPEC_GET(LED_USER, gpios);
76 static const struct gpio_dt_spec ledcntl = GPIO_DT_SPEC_GET(LED_CNTL, gpios);
```

```
205 //*****
206 /* Configure GPIO Function
207 //*****
208 static void configure_gpio(void)
209 {
210     gpio_pin_configure_dt(&runstatus, GPIO_OUTPUT_INACTIVE);
211     gpio_pin_configure_dt(&constatus, GPIO_OUTPUT_INACTIVE);
212     gpio_pin_configure_dt(&leduser, GPIO_OUTPUT_INACTIVE);
213     gpio_pin_configure_dt(&ledcntl, GPIO_OUTPUT_INACTIVE);
214 }
```



## Add Notify to the Main Loop

```
278     for (;;)
279     {
280         gpio_pin_set_dt(&runstatus, (++blink_status) % 2);
281         k_sleep(K_MSEC(RUN_LED_BLINK_INTERVAL));
282
283         if(CHK_FLAG(fconnected))
284         {
285             adc0_val = sensor_cb();
286             LOG_INF("adc0 = %04X",adc0_val);
287             LOG_INF("adc0 %d mV",adc0_val);
288
289             status_nfy(NULL,adc0_val);
290         }
291     }
```

3:53 92%  
Devices DISCONNECT  
BONDED ADVERTISER CEC\_BLE\_DAY4 DE:88:83:B1:AB:15  
CONNECTED CLIENT SERVER  
Generic Attribute  
UUID: 0x1801  
PRIMARY SERVICE  
Generic Access  
UUID: 0x1800  
PRIMARY SERVICE  
LED SERVICE  
UUID: 5328b08a-0c4a-4bda-996c-41064568e694  
PRIMARY SERVICE  
SENSOR SERVICE  
UUID: 97252dff-72ca-44ea-910b-82864cae65f3  
PRIMARY SERVICE  
SENSOR CHRC  
UUID: a64a76ef-d522-4783-9f46-bfdf95f4510d  
Properties: NOTIFY, READ  
Value: (0x) 01-12  
Descriptors:  
Client Characteristic Configuration  
UUID: 0x2902  
Value: Notifications enabled



# Test Drive

```

Monitor Mode Serial View Mode Text Port /dev/ttyUSB0 - FTDI Baud rate 115200 Line ending CRLF Stop Monitoring
*** Booting nRF Connect SDK d96769faceca ***
[00:00:00.000,274] <inf> bleday4: A Custom BLE Design Using the Raytac MDBT50Q BLE Module

[00:00:00.000,488] <inf> bt sdc hci driver: SoftDevice Controller build revision:
                        36 f0 e5 0e 87 68 48 fb 02 fd 9f 82 cc 32 e5 7b |6....hH. ....2.{
                        91 b1 5c ed |..\

[00:00:00.002,624] <inf> bt hci core: HW Platform: Nordic Semiconductor (0x0002)
[00:00:00.002,655] <inf> bt hci core: HW Variant: nRF52x (0x0002)
[00:00:00.002,685] <inf> bt hci core: Firmware: Standard Bluetooth controller (0x00) Version 54.58864 Build 1214809870
[00:00:00.003,479] <inf> bt hci core: Identity: DE:88:83:B1:AB:15 (random)
[00:00:00.003,509] <inf> bt hci core: HCI: version 5.4 (0x0d) revision 0x118f, manufacturer 0x0059
[00:00:00.003,540] <inf> bt hci core: LMP: version 5.4 (0x0d) subver 0x118f
[00:00:00.003,570] <inf> bleday4: Bluetooth initialized

[00:00:00.004,455] <inf> bleday4: Advertising successfully started

[00:00:29.276,763] <inf> bleday4: Connected

[00:00:30.009,185] <inf> bleday4: adc0 = 00FF
[00:00:30.009,185] <inf> bleday4: adc0 255 mV
[00:00:31.012,237] <inf> bleday4: adc0 = 0102
[00:00:31.012,268] <inf> bleday4: adc0 258 mV
[00:00:32.015,258] <inf> bleday4: adc0 = 00FF
[00:00:32.015,289] <inf> bleday4: adc0 255 mV
[00:00:33.018,280] <inf> bleday4: adc0 = 0101
[00:00:33.018,310] <inf> bleday4: adc0 257 mV
[00:00:34.021,301] <inf> bleday4: adc0 = 00FE
[00:00:34.021,331] <inf> bleday4: adc0 254 mV
[00:00:35.024,353] <inf> bleday4: adc0 = 00FE
[00:00:35.024,383] <inf> bleday4: adc0 254 mV
[00:00:36.027,374] <inf> bleday4: adc0 = 0100
[00:00:36.027,404] <inf> bleday4: adc0 256 mV
[00:00:37.030,395] <inf> bleday4: adc0 = 0102
[00:00:37.030,426] <inf> bleday4: adc0 258 mV
[00:00:38.033,416] <inf> bleday4: adc0 = 0100
[00:00:38.033,447] <inf> bleday4: adc0 256 mV
[00:00:39.036,437] <inf> bleday4: adc0 = 0101
[00:00:39.036,468] <inf> bleday4: adc0 257 mV
[00:00:39.198,028] <inf> bleday4: Disconnected (reason 19)
  
```

The screenshot shows an Android application interface for managing BLE devices. The top bar shows the time as 1:48 and battery at 67%. The main screen displays a list of devices, with 'CEC\_BLE\_DAY4' (DE:88:83:B1:AB:15) selected. The device status is 'CONNECTED' and 'CLIENT'. Below the device name, there are sections for 'Generic Attribute', 'Generic Access', 'LED SERVICE', 'SENSOR SERVICE', and 'SENSOR CHRC'. The 'SENSOR CHRC' section is expanded, showing details for a Client Characteristic Configuration (UUID: 0x2902) with the value 'Notifications enabled'. A yellow arrow points from the 'adc0 = 0102' log entry in the terminal to the 'SENSOR CHRC' section in the app.

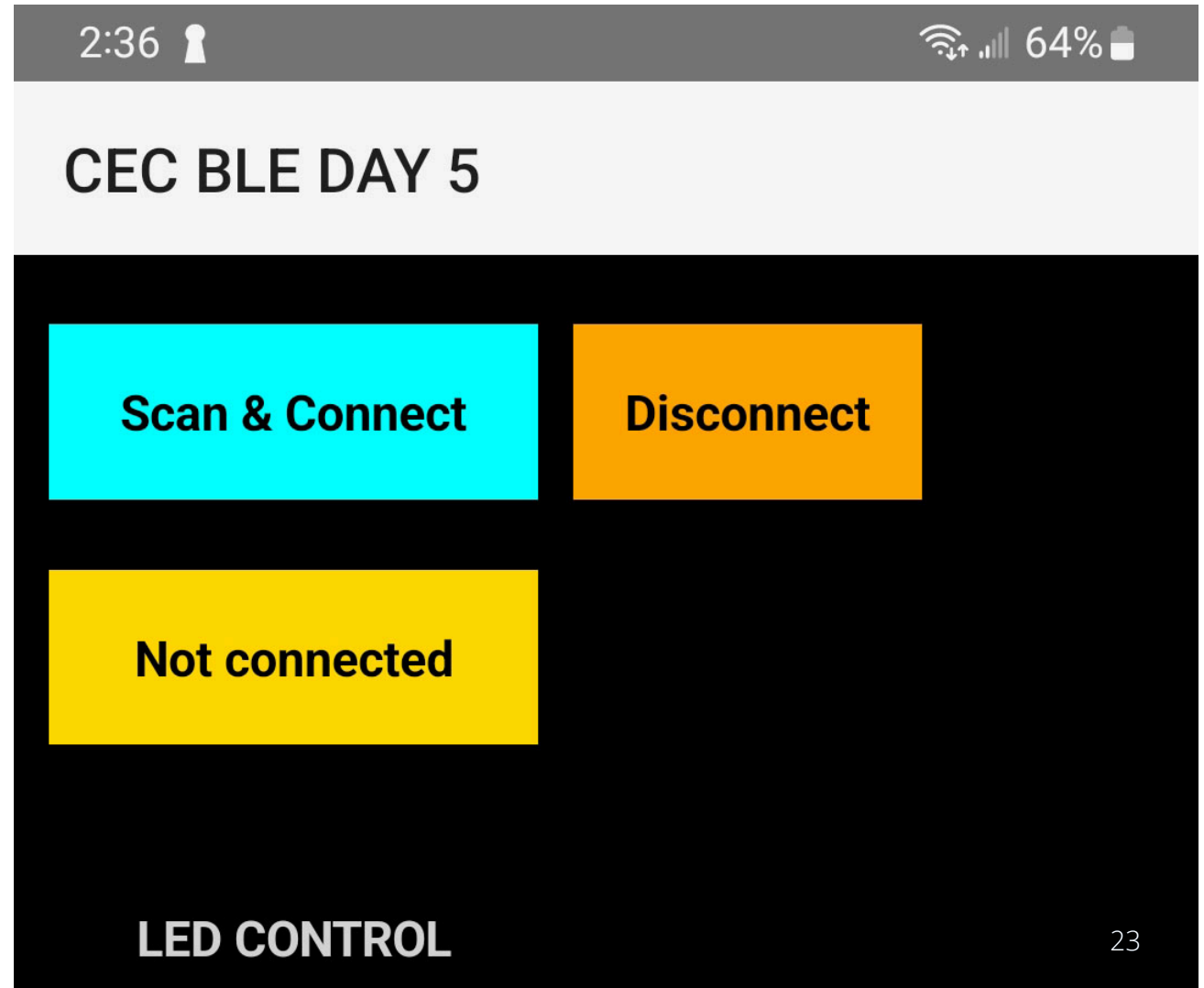
Next Time...

MORE TO COME..

# Thank you for attending!!!

Please consider the resources below:

- [Today's Download Package](#)
- [nordicsemi.com](http://nordicsemi.com)
- [nRF52840 User Guide](#)
- [raytac.com](http://raytac.com)





**DesignNews**

Thank You

Sponsored by

**DigiKey**

