

The screenshot displays the nRF Connect IDE interface. On the left, the project tree shows the 'chat' application under 'APPLICATIONS', with a 'build' folder containing 'nRF52833 DK NRF52833'. The 'CHAT' folder is expanded, showing 'src' with files 'chat_cli.c', 'main.c', and 'model_handler.c'. The 'NRF52833DK NRF52833' folder is also expanded, showing hardware details like GPIO, Flash (512 kB), Interrupts & nvic, Buses, and ADCs & adc. The 'ACTIONS' section includes Build, Kconfig, Debug, Flash, and Memory report. The 'CONNECTED DEVICES' section shows a device with ID 685586543, model NRF52833_xxAA_REV1, and interfaces VCOM0 COM5 and RTT.

The main editor shows the source code for 'chat_cli.c'. The code includes copyright information for Nordic Semiconductor ASA, SPDX license identifier, and various include statements for Zephyr Bluetooth mesh, chat_cli.h, mesh/net.h, mesh/transport.h, and string.h. It also includes logging headers and declares the LOG_MODULE_DECLARE(chat). The code defines BUILD_ASSERT macros for message length constraints and implements the encode_presence function, which takes a net_buf_simple pointer and a bt_mesh_chat_cli_presence enum value.

The bottom terminal window shows the flashing process. It starts with 'runners.nrfjprog: mass erase requested', followed by 'runners.nrfjprog: Flashing file: C:\nordic\myapps\chat\build\zephyr\merged.hex'. The log shows the file being erased (0.257s), the program being programmed (3.507s), and the file being verified (3.474s). The process concludes with 'Enabling pin reset. Applying pin reset.' and 'runners.nrfjprog: Board with serial number 685586543 flashed successfully.' A Korean message at the bottom states: '* 터미널이 작업에서 다시 사용됩니다. 닫으려면 아무 키나 누르세요.'

We tested the nrf / sample / mesh chat with NRF52833 demo board as the above.

And after completing provisioning, the application key can not be added.

