

[https://infocenter.nordicsemi.com/topic/com.nordic.infocenter.sdk5.v15.0.0/twi\\_master\\_with\\_twis\\_slave\\_example.html](https://infocenter.nordicsemi.com/topic/com.nordic.infocenter.sdk5.v15.0.0/twi_master_with_twis_slave_example.html)

[Copy URL](#)

[Download  
offline documentation](#)

## TWIS Slave and TWI Master Mode Drivers Example

This example demonstrates how to use the following two drivers:

- [TWIS driver - legacy layer](#) (in asynchronous mode)
- [TWI driver - legacy layer](#) (in synchronous mode)

### TWI Slave

On the slave, EEPROM memory is simulated. The size of the simulated EEPROM is configurable in the `config.h` file. Default memory value of the device is 320 bytes. It is simulated using internal RAM. This RAM area is accessed only by the simulated EEPROM so the rest of the application can access it only by using TWI commands through hardware configured pins.

The selected memory chip has a 7-bit address constant. During a write operation, a word is selected to access the memory. The first byte sent is used as the current address pointer.

During a single access, the entire memory can be read and a maximum of an 8-byte page can be written.

The following are the differences between the actual chip and the simulated chip:

1. Simulated chip has practically 0 ms write time. This example does not poll the memory for readiness.
2. During sequential read, when memory end is reached, there is no support for roll-over. It is recommended for master to assure that it does not try to read more than the page limits. If the master is not tracking this and trying to read after the page ends, then the slave will start to NACK for trying to over-read the memory. The master should end the transaction when slave starts NACKing, which could mean that the master has read the end of the page.
3. It is possible to write a maximum of `EEPROM_SIM_SEQ_WRITE_MAX_BYTES` bytes in a single sequential write. However, in simulated EEPROM the whole address pointer is incremented. In a real device, writing would roll-over in memory page.

### TWI Master

The master communicates with the EEPROM memory and additionally has read and write access. Simple UART commands can be used to check the memory.

Pins to short (see `config.h` file example directory):

- `TWI_SCL_M` - `EEPROM_SIM_SCL_S`
- `TWI_SDA_M` - `EEPROM_SIM_SDA_S`

In order to list all commands, press the Tab button.

You can find the source code and the project file of the example in the following folder:

`<InstallFolder>\examples\peripheral\twi_master_with_twis_slave`

### Testing

Test the TWIS slave and TWI master mode drivers example application by performing the following steps:

1. The pins should be shorted as described above.
2. Compile and program the application.
3. Start a terminal emulator like PuTTY (recommended) and establish a serial connection with the following [Terminal settings](#).
4. Read the welcome message.
5. Press Tab to view all available commands.

6. Send the eeprom read command to read data from the simulated EEPROM.
7. Send the eeprom write command with "any string".
8. Send the eeprom read command again and check whether your string is at the beginning of the memory.
9. Send the eeprom clear command to clear EEPROM.
10. Send the eeprom read command again and see if all values are 0xff.
11. Send the eeprom error command to check transmission error.

---

[Documentation feedback](#) | [Developer Zone <https://devzone.nordicsemi.com/questions/>](https://devzone.nordicsemi.com/questions/) | [Subscribe](#) | Updated 2018-04-03