9.6.2.1 Erase protection

Erase protection can be used to prevent a device from being erased.

A device ERASEALL operation can be initiated either by the non-volatile memory controller, or through the control access port. The following table describes the protection of the CTRL-AP ERASEALL operation. For more information, see Access port protection on page 817

Access port protection state		ERASEALL operation
UICR.ERASEPROTECT	TAMPC.PROTECT.ERASEPROTECT	ERASEALL
Unprotected	Unprotected	Allowed
Protected	-	Disabled
-	Protected	Disabled

Table 75: Erase all protection

The debugger can read the erase protection status in the register ERASEPROTECT.STATUS on page 828.

When erase protection is enabled, both the debugger and on-board firmware are required to disable it. The same non-zero 32-bit KEY value must be written to the debugger register ERASEPROTECT.DISABLE and CPU register ERASEPROTECT.DISABLE to disable erase protection. When both registers have been written with the same non-zero 32-bit KEY value, the device is automatically erased as described in Erase all on page 823. The access ports will be re-enabled on the next reset once the secure erase sequence has completed.

Set the write-once register ERASEPROTECT.LOCK on page 834 to Locked as early as possible in the start-up sequence, preferably as soon as the on-chip firmware has determined it does not need to communicate with a debugger over the CTRL-AP mailbox interface. Once written, it will not be possible to remove the erase protection until the next pin reset, power-on reset, brownout reset, or watchdog timer reset, and therefore ERASEPROTECT.DISABLE is also disabled.

9.6.3 Mailbox interface

CTRL-AP implements a mailbox interface which enables the CPU to communicate with a debugger over the SWD interface.

The mailbox interface consists of a transmit register MAILBOX.TXDATA on page 829 with its corresponding status register MAILBOX.TXSTATUS on page 829, and a receive register MAILBOX.RXDATA on page 829 with its corresponding status register MAILBOX.RXSTATUS on page 829. Status bits in registers TXSTATUS/RXSTATUS will be set and cleared automatically when registers TXDATA/RXDATA are written to and read from, independently of the direction.