

Application Notes

Programming AURIX TC3xx UCBs with UDE® Memtool

Overview

The members of Infineon AURIX™ TC3xx controller family provide a mechanism to store configuration information on-chip within special data FLASH sectors. These sectors are called **User Configuration Blocks (UCBs)**. Information stored in these UCBs e.g. configure read- and write protection of on-chip FLASHs, affect the startup behavior or lock the debug interface of the controller. For details on which controller type provides which UCBs and the content of the UCBs please refer to the User's Manual of the respective controller type.

GENERAL WARNING:

Programming invalid information to an UCB or completely erasing an UCB FLASH sector may result in a state where the controller can't execute its on-chip startup software anymore. In this case it won't be possible to connect the controller with UDE® Universal Debug Engine anymore to alter the UCB configuration. So make sure to always use proper UCB programming data!

UDE® Universal Debug Engine provides two ways to program UCBs:

1. **UCB Configuration Function** ... provides configuration dialogs to set-up UCB content for programming manually. It is meant for the developer how tries to prepare the right configuration for his use case. It also allows changing or removing an existing UCB configuration as long as the respective UCB allows this. The function also includes some measures to avoid invalid UCB configurations. Finally a successfully tested UCB configuration may be exported to a hex file for production usage.
UCB Configuration Function is also supported by UDE® Memtool object model so it can be used by COM client software and scripting clients. (For general information on UDE® Memtool scripting please refer to application note **UDE® Memtool Automation Basics**)
2. **Direct UCB Programming** ... allows to load and program HEX files as usual with other memory devices (e.g. PFLASH).
This method is meant for production usage. Prepared HEX files containing valid UCB configuration information can be loaded and programmed to one or more UCBs.
Memtool command line interface MemtoolCli.exe supports this method.
PLEASE NOTE:
UDE® Memtool's Memory Device **DF_UCBS** meant for direct UCB programming is disabled by default to avoid misconfiguration because of invalid an untested UCB sections in hex files. It needs to be enabled by user before direct UCB programming is possible.
UDE® Memtool can only program UCBs completely. A hex file that cover only parts of an UCB will be rejected.
There are also additional checks implemented to prevent programming UCBs with invalid data (e.g. invalid confirmation codes).

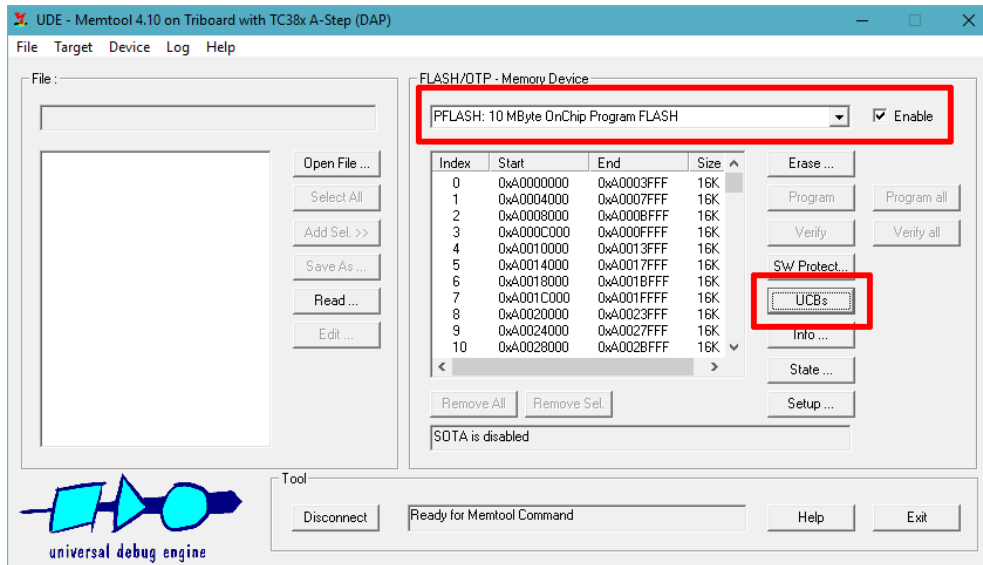
GENERAL HINT:

A changed UCB configuration only affects the controller's behavior after a reset. When using UDE® Memtool this usually means disconnecting and re-connecting the target before the result of a new configuration can be displayed.

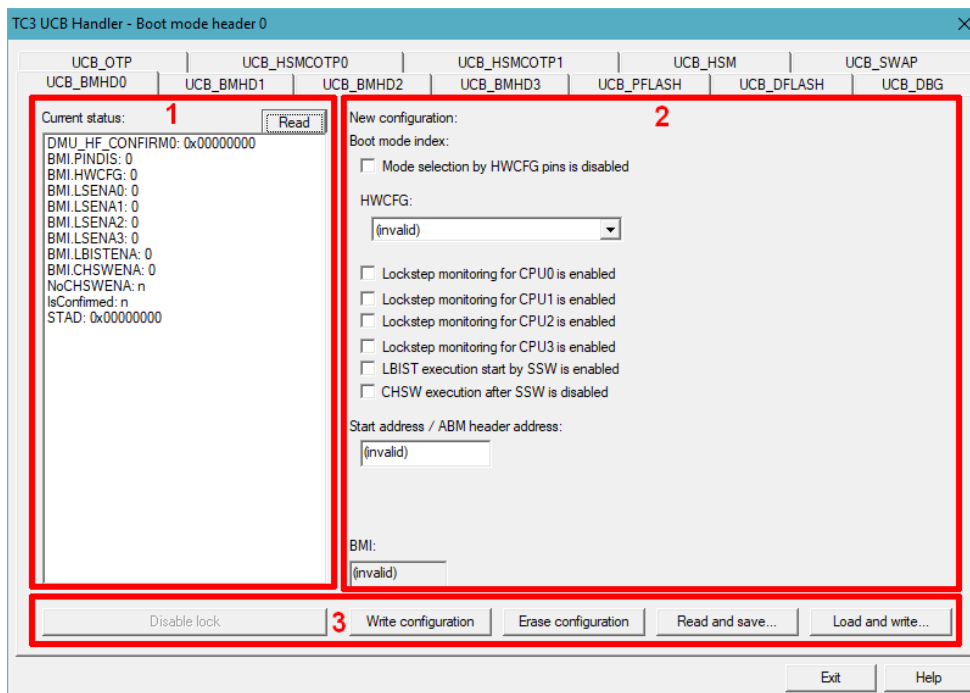
Application Notes

Using UCB Configuration Function

UCBs programming function is always associated with UDE's PFLASH Memory Device. So PFLASH must be selected and enabled for UCB handling:



Button UCBs opens the TC3xx (AURIX TC3xx) UCB Handler dialog:



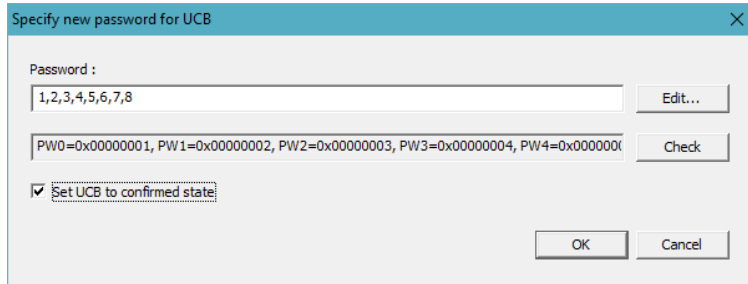
Each programmable UCB is represented by a separate dialog tab. All tabs look similar but the functions available to the user depend on the features of the respective UCB and its current configuration:

1. At the left side the current status of registers and bit-fields are shown which are affected by the respective UCB.
2. The main part of the dialog tab allows preparing the new UCB configuration.
3. The buttons at bottom line allow executing the actual UCB handling functions.

Application Notes

Disable Lock

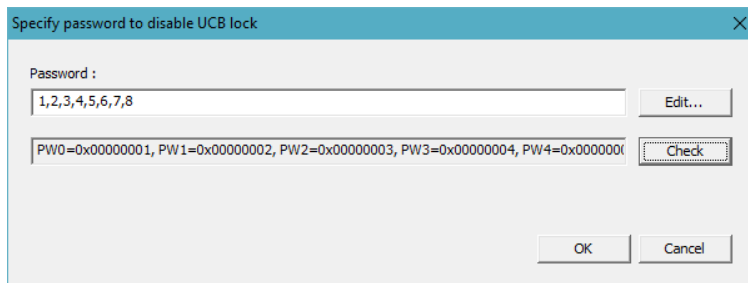
Some UCBs provide a password protection mechanism. The user has to specify a 256bit (8x32bit) password to unlock the UCB. Otherwise its content can't be accessed nor changed.



The dialog box titled "Specify new password for UCB" contains a "Password:" label, a text input field with the value "1,2,3,4,5,6,7,8", an "Edit..." button, a text area with the value "PW0=0x00000001, PW1=0x00000002, PW2=0x00000003, PW3=0x00000004, PW4=0x00000005", a "Check" button, a checked checkbox labeled "Set UCB to confirmed state", and "OK" and "Cancel" buttons at the bottom.

Write configuration

This function writes the new configuration to the respective UCB. When the UCB provides a password protection mechanism the function asks the user to specify a new 256bit (8x32bit) password.



The dialog box titled "Specify password to disable UCB lock" contains a "Password:" label, a text input field with the value "1,2,3,4,5,6,7,8", an "Edit..." button, a text area with the value "PW0=0x00000001, PW1=0x00000002, PW2=0x00000003, PW3=0x00000004, PW4=0x00000005", a "Check" button, and "OK" and "Cancel" buttons at the bottom.

Erase configuration

This doesn't actually erase the UCB content. Instead of this the UCB is overwritten with an empty configuration. Please note: The OTP UCBs can't be erased at all. So this function is not available for these UCBs.

Read and save ...

This function reads the current content of an UCB and asks the user to store it to a HEX file.

Load and write ...

This function allows loading a HEX file and programming its content to the UCB.

Application Notes

Automate UCB Configuration Function

UCB configuration can be automated by using UDE® Memtool object model functions (API).

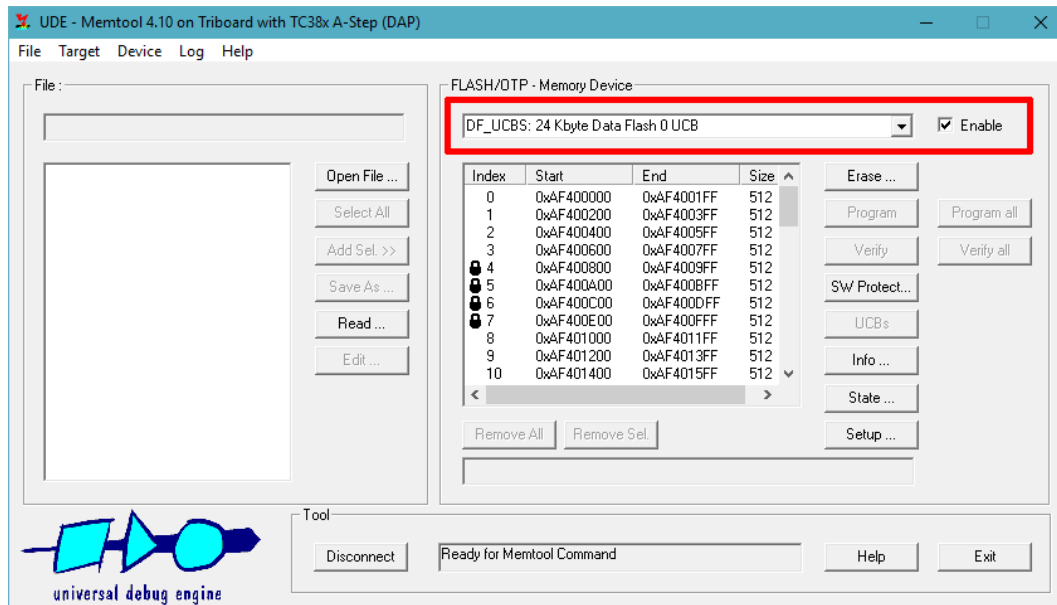
Here is a small example on how to do this with a C# application:

```
// access FlashModExt object
UDELib.DIUDEFashMod FlashMod = m_Memtool.GetFlashMod("PFLASH0");
UDELib.DIUDEFashModExt FlashModExt = FlashMod.GetExtension("UCB_DFLASH");
// read current status from target
if(!FlashModExt.ReadStatus())
{
    System.Console.WriteLine("FlashModExt.ReadStatus failed !");
    return -1;
}
// check if UCB is locked
if(FlashModExt.Locked)
{
    // try to unlock UCB with password
    if(!FlashModExt.DisableLock("1,2,3,4,5,6,7,8"))
    {
        System.Console.WriteLine("FlashModExt.DisableLock failed !");
        return -1;
    }
}
// prepare new configuration based on current status
FlashModExt.PrepareNewConfig();
// access new configuration and change some options
var Config=FlashModExt.Config;
Config.SetItem("PROCONRAM.RAMINSEL1",1);
Config.SetItem("PROCONRAM.RAMINSEL2",1);
// write new configuration to target
if(!FlashModExt.WriteConfig())
{
    System.Console.WriteLine("FlashModExt.WriteConfig failed !");
    return -1;
}
```

Application Notes

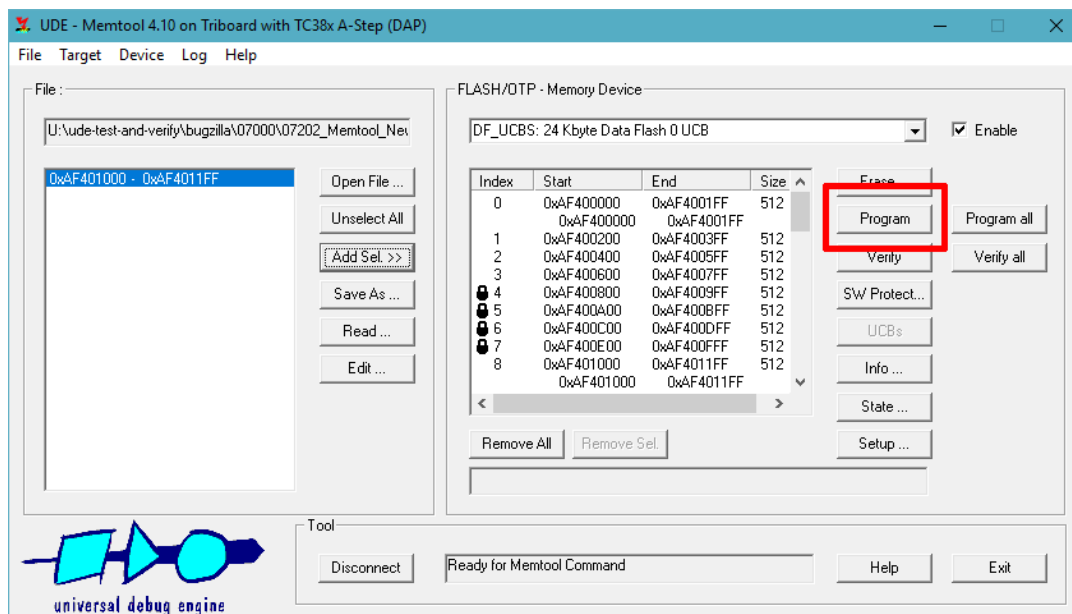
Using Direct UCB Programming

A separate Memory Device **DF_UCBS** is provided for direct UCB programming. This needs to be selected and enabled:



The several UCBs are shown as separate FLASH sectors. Some UCBs can't be programmed by the user. This is shown by the black lock symbol.

Now the user can load one or more HEX files and program the UCBs:



Please note: There is no **Erase** function available. However the user may load a program an HEX file that contains empty UCB configuration.