

Porting RZK/ZTP Applications to the eZ80L925148MODG Module

AN036602-1013

Abstract

This document describes procedures for porting ZTP 2.4.x applications to the eZ80L925148MODG Module (Zilog Module Label 99C1400-001G) and the eZ80Acclaim! Development Platform (99C0858-001). The new eZ80L925148MODG Module features the same physical form factor as the eZ80L925048MODG Module (99C0857-001) it replaces; however, the new eZ80L925148MODG Module uses a different external Flash module. This new eZ80L925148MODG Module (99C1400-001G) is affixed to the eZ80Acclaim! Development Platform, as indicated in Figure 1.

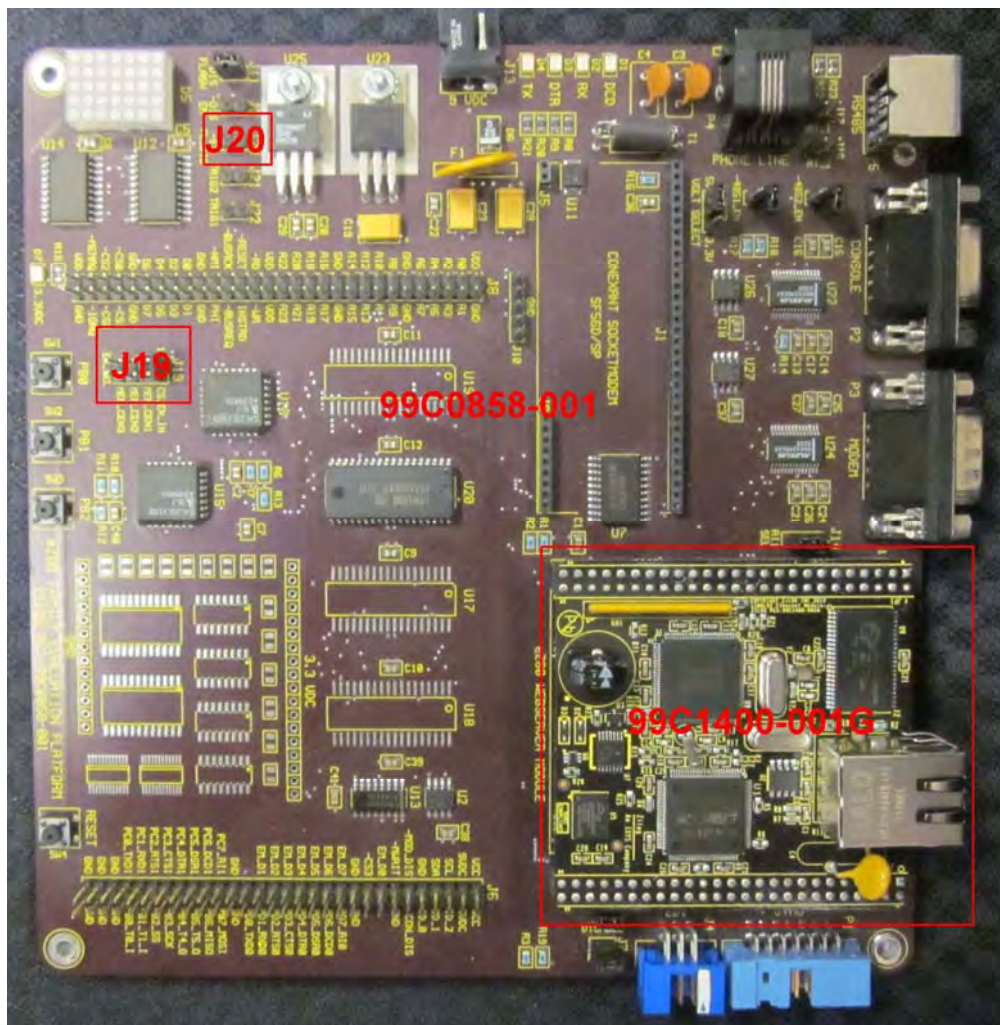


Figure 1. eZ80L92 Module Affixed to the eZ80Acclaim! Development Platform



► **Note:** The source code file associated with this application note, [AN0366-SC01](#), is available free for download from the Zilog website. This source code has been tested with ZDSII – eZ80Acclaim! version 5.2.1 and ZTP version 2.4.1. Subsequent releases of ZDSII may require you to modify the code supplied with this application note.

Differences Between the Modules

The hardware differences between the older eZ80L925048MOD Module and the new eZ80L925148MODG Module are summarized in Table 1. Click the links in this table to learn more about how these changes affect ZTP applications.

Table 1. Hardware Differences Between eZ80L925x48MOD Modules

Component/ Feature	99C0857 Hardware Implementation	99C1400 Hardware Implementation	Affected Project Settings	Application Modification
External Flash on CS0	1 MB Micron MT28F008B3	8 MB Spansion S29GL064N eZ80Acclaim! Development Platform Jumper Changes	CS0 Configuration New ZTGT Files and Address Spaces eZ80L925148MODG Preprocessor Definition	Optional ZFS Configuration
IrDA	Yes	No	N/A	Remove IRDA code

The hardware changes listed in Table 1 require modifications to your ZTP project before the application will function with the eZ80L925148MODG Module. Although this document describes the changes required specifically when using the eZ80Acclaim! Development Platform, similar project file changes may also be required when using hardware platforms with the eZ80L925148MODG Module.

To learn more about the eZ80L925148MODG Module, refer to the [eZ80L92 Module Product Specification \(PS0318\)](#). For a demonstration program that includes the changes required to support the eZ80L925148MODG Module, refer to the ZTPDemo_eZ80L925148MODG.zdsproj file in the AN0366-SC01_<version>.exe installation package. The remaining sections of this document describe the project and/or application changes required to support these hardware changes.

CS0 Configuration

1 MB of external Flash memory is mapped to the CS0 on the eZ80L925048MOD Module. Applications that use this older module typically configured the CS0 as follows for RAM build configurations:

CS0 Lower Bound	0x100000
CS0 Upper Bound	0x1FFFFFF

CS0 Control	0x88
CS0 Bus Mode	0x82

The new eZ80L925148MODG Module is equipped with 8MB of Flash mapped to the CS0. To avoid conflicts with SRAM memory that is mapped to the first 512KB to 1MB of the eZ80L92 address space in RAM build configurations, the CS0 should continue to be configured with a lower bound of 0x100000. This configuration would imply a CS0 upper bound of 0x8FFFFFF on the eZ80L925148MODG Module.

However, be advised that when using the eZ80Acclaim! Development Platform (99C0858) with the eZ80L925148MODG Module, the GPIO and LED matrix are mapped into the CS2 memory at 0x800000. Also, the gate array logic on the eZ80Acclaim! Development Platform limits the upper bound of the CS0 mapping to 0x7FFFFFF. Consequently, Zilog recommends the following CS0 settings for use with the eZ80L925148MODG Module on the eZ80Acclaim! Development Platform for RAM builds:

CS0 Lower Bound	0x100000
CS0 Upper Bound	0x7FFFFFF
CS0 Control	0x88
CS0 Bus Mode	0x01

When using a Flash (or Copy to RAM) build configuration, the lower bound on the CS0 is typically set to 0x00 and the chip selects for SRAM are configured such that the static RAM is located above the highest memory address occupied by the external Flash.

eZ80Acclaim! Development Platform Jumper Changes

When using the eZ80L925148MODG Module with the eZ80Acclaim! Development Platform, it is typically necessary to install Jumper J20. By default, the external chip select signal on the eZ80Acclaim! Development Platform, nCS_EX, features a jumper in the 1–2 position of J19, which routes memory requests in the CS0 range of 0x400000 to 0x7FFFFFF to the eZ80Acclaim! Development Platform. As a result, access is prevented to the upper 4MB of the external 8MB Flash space on the eZ80L925148MODG Module. To maximize the amount of external Flash memory, the eZ80L92 MCU can access on the eZ80L925148MODG Module. It is necessary to install a shunt on J20.

New ZTGT Files and Address Spaces

The AN0366–SC01_<version>.exe installation and later ZDSII – eZ80Acclaim! releases include two new Zilog target (ZTGT) files for the eZ80L925148MODG Module. These files are named eZ80L925148MODG_RAM.ztgt and eZ80L925148MODG_Flash.ztgt.

RAM Build Configuration

For applications using the RAM build configuration, select the eZ80L925148MODG_RAM target by observing the following brief procedure.

1. From the **Project** menu in ZDSII, choose **Settings...** to open the Project Settings dialog.
2. In the left pane, click **Debugger**. In the Target pane, select the **eZ80L925148MODG_RAM** target.
3. Click the **Setup** button. The Configure Target dialog will appear, as shown in Figure 2, which presents the default settings for the eZ80L925148MODG_RAM ZTGT file. If you have not previously modified the contents of the eZ80L925148MODG_RAM file, then you should see the values in Figure 2 after clicking the Setup button. Verify that your settings match the values shown in this figure.

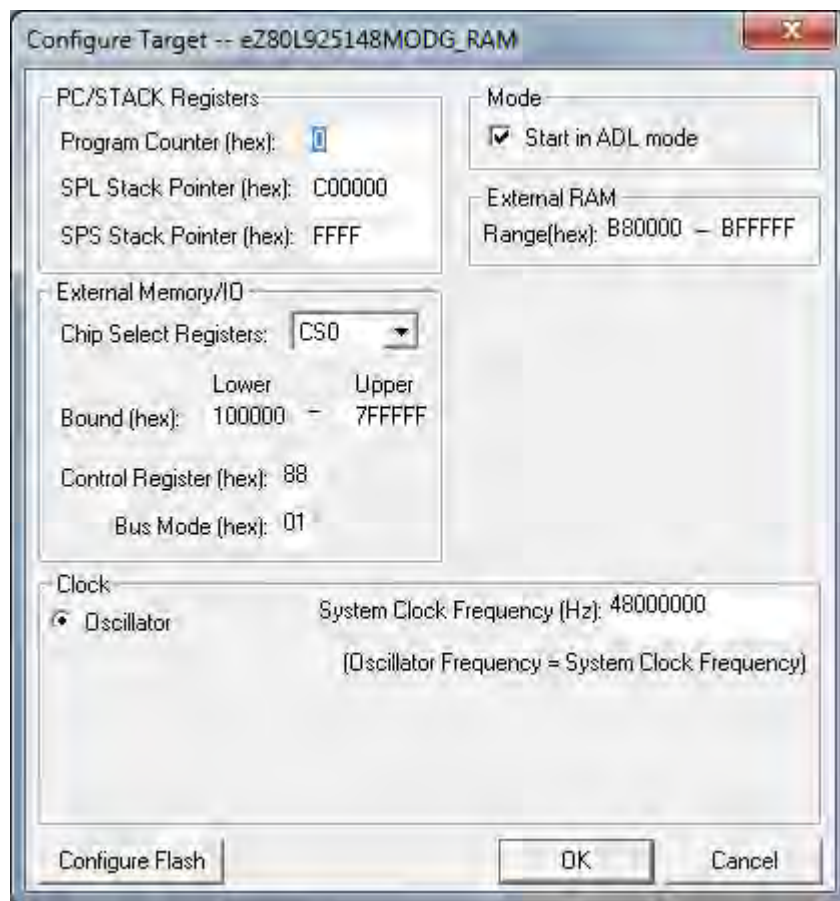


Figure 2. The Configure Target Dialog for the RAM Build

-
4. Examine the settings of the remaining chip selects in the **Chip Select Registers:** drop-down menu, and verify that the settings for CS0 to CS3 correspond to the values shown in Table 3.

Table 2. Chip Select Settings for the RAM Build

Chip Select	Memory Range	CS Control	CS Bus Mode
CS0	0x100000–0x7FFFFFFF	0x88	0x01
CS1	0x000000–0x07FFFF	0x08	0x01
CS2	0x800000–0xBFFFFFFF	0x28	0x01
CS3	0x030000–0x030000	0x18	0x84

5. When configuring RAM build configurations using the eZ80L925148MODG Module with the eZ80Acclaim! Development Platform, Zilog recommends using the settings in the Address Spaces selection of the Project Settings dialog, which is shown in Figure 3. Ensure that the project’s settings match the values displayed in this figure.

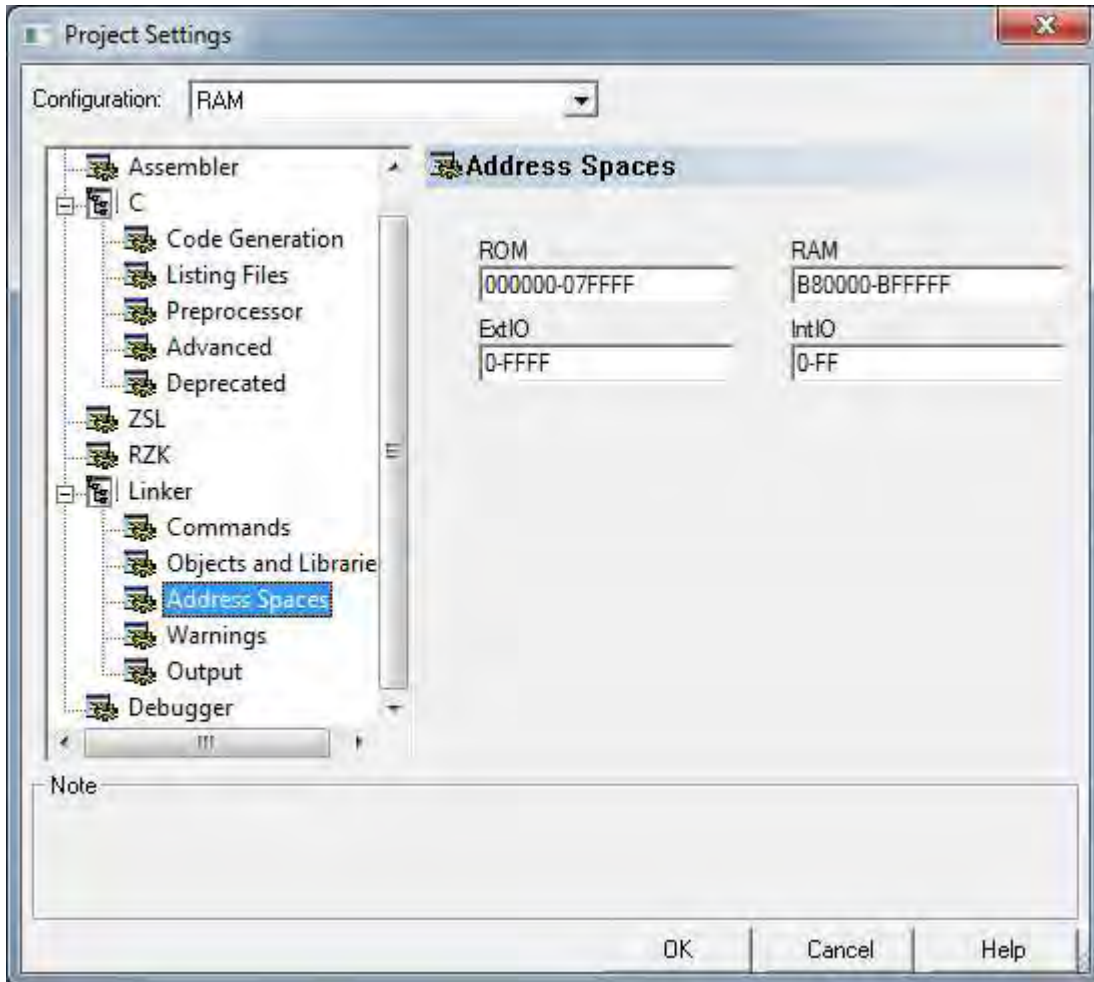


Figure 3. The Configure Target Dialog

Flash and Copy to RAM Build Configurations

Applications using the FLASH or COPY TO RAM build configurations should select the eZ80L925148MODG_Flash target by observing the following brief procedure:

1. From the **Project** menu in ZDSII, choose **Settings...** to open the Project Settings dialog.
2. In the left pane, click **Debugger**. In the Target pane, select the **eZ80L925148MODG_Flash** target.
3. Click the **Setup** button. The Configure target dialog will appear, as shown in Figure 4, which presents the default settings for the eZ80L925148MODG_Flash ZTGT file. If you have not modified the contents of the eZ80L925148MODG_Flash.ztgt file, then you should see the values in Figure 4 after clicking the Setup button. Verify that your settings match the values shown in this figure.

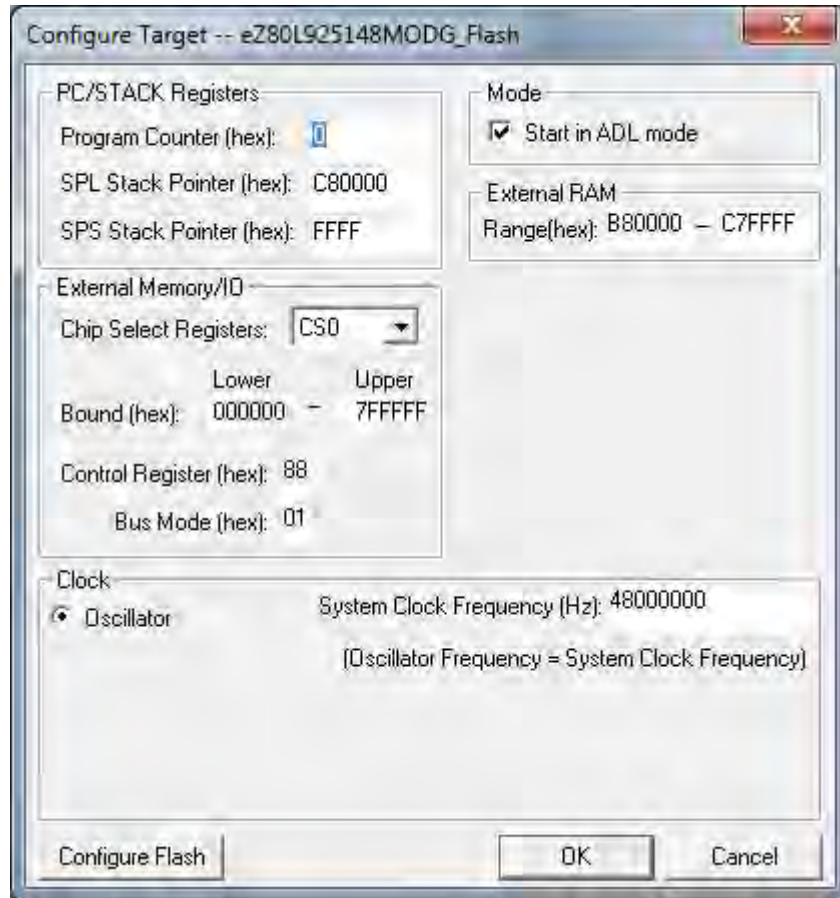


Figure 4. The Configure Target Dialog for the Flash/Copy to RAM Build

- Examine the settings of the remaining chip selects in the **Chip Select Registers:** drop-down menu, and verify that the settings for CS0 to CS3 correspond to the values shown in Figure 3.

Table 3. Chip Select Settings for the Flash/Copy to RAM Build

Chip Select	Memory Range	CS Control	CS Bus Mode
CS0	0x000000–0x7FFFFFFF	0x88	0x01
CS1	0xC00000–0xC7FFFF	0x08	0x01
CS2	0x800000–0xBFFFFFFF	0x28	0x01
CS3	0x030000–0x030000	0x18	0x84

- When using the eZ80L925148MODG Module with the eZ80Acclaim! Development Platform, Zilog recommends the settings for the Address Spaces tab in the Project Set-

tings menu for Flash build configurations which are shown in Figure 5. Verify that the settings match the values shown in this figure.

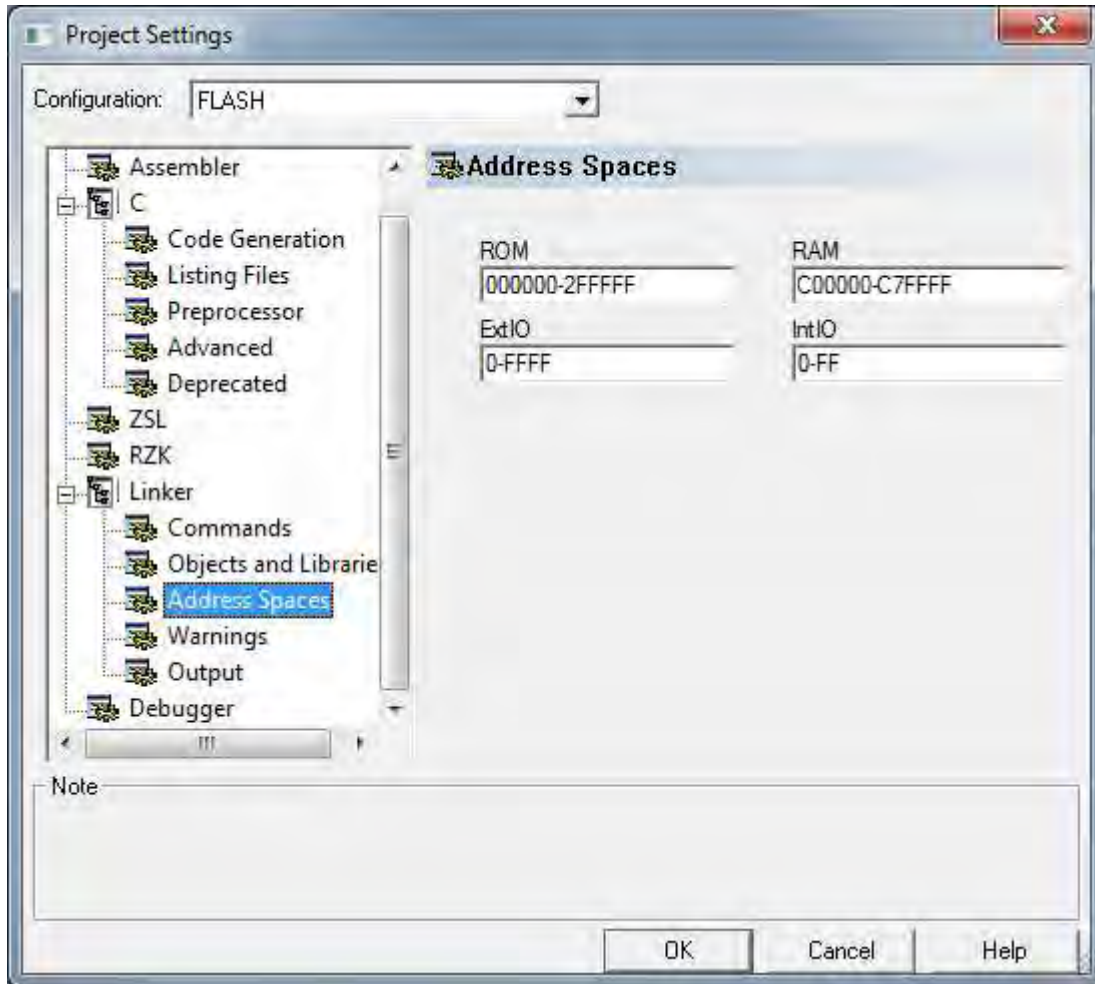


Figure 5. The Address Spaces Dialog for the Flash Build

► **Note:** The ROM address space shown in Figure 5 only includes the first 3 MB of external Flash memory. If the project requires additional ROM memory (i.e., code space), the upper range of the ROM address space can be increased up to 0x7FFFFFFF.

6. When configuring Copy to RAM build configurations using the eZ80L925148MODG Module with the eZ80Acclaim! Development Platform, Zilog recommends using the settings in the Address Spaces selection of the Project Settings dialog which are shown in Figure 6. Ensure that your project's settings match the values displayed in the following figure.

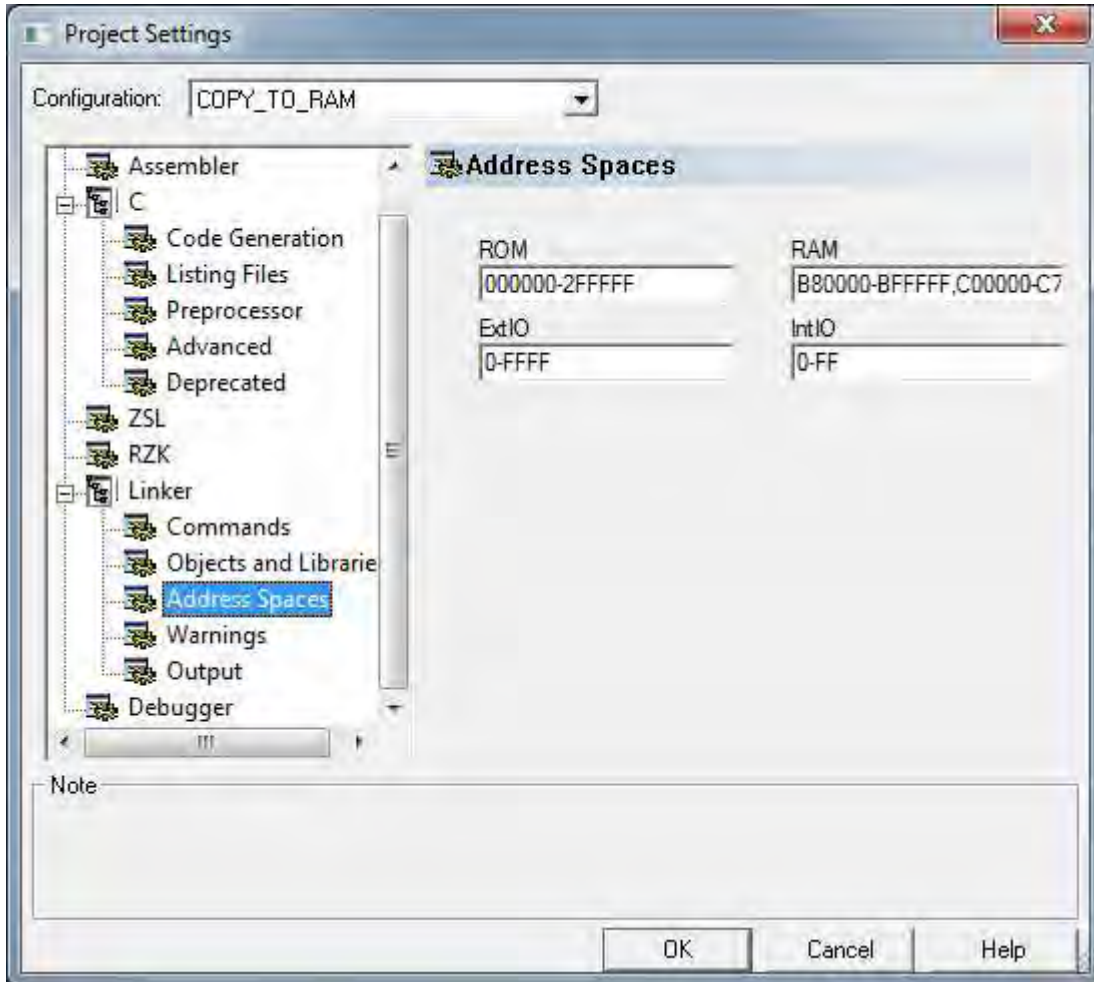


Figure 6. The Address Spaces Dialog for the Copy to RAM Build

eZ80L925148MODG Preprocessor Definition

The eZ80L925148MODG preprocessor definition is used to simplify and unify project settings between the various build configurations (RAM, Flash and Copy to RAM). This new preprocessor definition replaces the following preprocessor definitions used with the eZ80L925048MOD Module: RAM_BUILD and ZFS_COPY_TO_RAM. RZK/ZTP projects that target the eZ80L925148MODG Module should include the eZ80L925148MODG preprocessor definition regardless of which build configuration is used and should not use the RAM_BUILD or ZFS_COPY_TO_RAM preprocessor definitions.

The eZ80L925148MODG preprocessor definition is used to select a memory configuration compatible with the optional Zilog File System (ZFS) and the optional data persistence module.

If the eZ80L925148MODG preprocessor definition is included in the project settings the compiler uses the settings in the `ZFS_Conf_eZ80L925148MODG.c` file to configure ZFS, as shown in the following code fragment from `ZFS_Conf.c`.

```
#ifndef EZ80F910300KITG
    #include "ZFS_Conf_eZ80F910300KITG.c"
#else
    #ifndef EZ80L925148MODG
        #include "ZFS_Conf_eZ80L925148MODG.c"
    #else
        #include "ZFS_Conf_old.c"
    #endif
#endif
```

The eZ80L925148MODG preprocessor definition also determines the location of the block of Flash memory used to hold the Data Persistence variables as shown in the following code fragment in the `DataPer_Conf.c` file:

```
#ifndef _EZ80L92
    #ifndef EZ80L925148MODG
        DATA_PER_CFG_t g_data_per_cfg =
        {
            ( void * ) 0x7F0000,          /* starting address */
            ( UINT32 ) 0x002000UL,      /* erasable block size */
            FS_AM29LV160_Init,
            FS_AM29LV160_Read,
            FS_AM29LV160_Write,
            FS_AM29LV160_Erase,
            FS_AM29LV160_Close
        };
    #else
        DATA_PER_CFG_t g_data_per_cfg =
        {
            ( void * ) 0x004000,          /* starting address */
            ( UINT32 ) 0x002000UL,      /* erasable block size */
            MT28F008_Init,
            MT28F008_Read,
            MT28F008_Write,
            MT28F008_Erase,
            MT28F008_Close
        };
    #endif
#else
    DATA_PER_CFG_t g_data_per_cfg =
    {
        ( void * ) 0x000000,          /* starting address */
        ( UINT32 ) 0x000100UL,      /* erasable block size */
        IntFlash_IP_Init,
        IntFlash_IP_Read,
    }
```

```
    IntFlash_IP_Write,  
    IntFlash_IP_Erase,  
    IntFlash_IP_Close  
};  
#endif
```

Optional ZFS Configuration

RZK/ZTP applications that use the Zilog File System (ZFS) can optionally modify the ZFS configuration file to allocate more, or less, Flash memory to ZFS. The default settings of the ZFS_Conf_eZ80L925148MODG.c file are listed below.

```
#define ZFS_TOTAL_NUM_BLOCKS ( 32 )  
#define ZFS_TOTAL_NUM_SECTORS( 0x200000/ZFS_SEC_SIZE )  
#define ZFS_TOTAL_NUM_VOLUMES( 1 )  
#define ZFS_MAX_FILE_OPEN_COUNT( 20 )  
#define ZFS_MAX_DIRS_SUPPORTED( 50 )  
#define ERASE_FLASH( 0 )  
ZFS_CONFIG_t g_zfs_cfg[ ZFS_TOTAL_NUM_VOLUMES ] =  
{  
    "EXTF", // vol name  
    ZFS_EXT_FLASH_DEV_TYPE, // vol type  
    (UINT8*)0x300000, // vol_start_addr  
    0x200000, // vol_size  
    ZFS_TOTAL_NUM_BLOCKS, // vol_blocks  
    ZFS_TOTAL_NUM_SECTORS, // number of sectors  
    FS_AM29LV160_Init,  
    FS_AM29LV160_Read,  
    FS_AM29LV160_Write,  
    FS_AM29LV160_Erase,  
    FS_AM29LV160_Close  
};
```

These settings will configure the ZFS to use 2MB of CS0 Flash from addresses 0x300000 to 0x4FFFFFF for the ZFS. In total, 32 Flash erase blocks (each 64KB in size) are allocated to the ZFS giving a volume size of 0x200000 bytes. Source code to the Flash driver used to read and write data to this volume is contained in the AM29LV160_Driver.c file.

-
- **Note:** The AM29LV160_Driver.c file (and other ZFS Flash driver files) is modified in the ZTP 2.4.2 release to explicitly disable interrupts while modifying the contents of Flash. This modification is necessary to ensure that an interrupt does not cause a higher-priority thread to start running from external Flash or attempt to read data from external Flash while it is being programmed or erased.
-

Customer Support

To share comments, get your technical questions answered, or report issues you may be experiencing with our products, please visit Zilog's Technical Support page at <http://support.zilog.com>.

To learn more about this product, find additional documentation, or to discover other facts about Zilog product offerings, please visit the Zilog Knowledge Base at <http://zilog.com/kb> or consider participating in the Zilog Forum at <http://zilog.com/forum>.

This publication is subject to replacement by a later edition. To determine whether a later edition exists, please visit the Zilog website at <http://www.zilog.com>.



Warning: DO NOT USE THIS PRODUCT IN LIFE SUPPORT SYSTEMS.

LIFE SUPPORT POLICY

ZILOG'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS PRIOR WRITTEN APPROVAL OF THE PRESIDENT AND GENERAL COUNSEL OF ZILOG CORPORATION.

As used herein

Life support devices or systems are devices which (a) are intended for surgical implant into the body, or (b) support or sustain life and whose failure to perform when properly used in accordance with instructions for use provided in the labeling can be reasonably expected to result in a significant injury to the user. A critical component is any component in a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system or to affect its safety or effectiveness.

Document Disclaimer

©2013 Zilog, Inc. All rights reserved. Information in this publication concerning the devices, applications, or technology described is intended to suggest possible uses and may be superseded. ZILOG, INC. DOES NOT ASSUME LIABILITY FOR OR PROVIDE A REPRESENTATION OF ACCURACY OF THE INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED IN THIS DOCUMENT. ZILOG ALSO DOES NOT ASSUME LIABILITY FOR INTELLECTUAL PROPERTY INFRINGEMENT RELATED IN ANY MANNER TO USE OF INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED HEREIN OR OTHERWISE. The information contained within this document has been verified according to the general principles of electrical and mechanical engineering.

eZ80Acclaim!, ZTP and ZDSII are trademarks or registered trademarks of Zilog, Inc. All other product or service names are the property of their respective owners.