SAML10 CPU Support Package Guide

Version: 4.1

Contents

SAML10 Support Package

This guide describes the following features of the SAML10 CPU support package:

- How to create SAML10 projects
- How to open SAML10 sample projects
- SAML10 specific project properties
- SAML10 specific project templates
- Supported SAML10 devices

Creating SAML10 Projects

Creating an SAML10 C/C++ executable project

To create a new minimal C/C++ executable project:

Select the **File** > **New** > **New Project** menu item.

Select the **A C/C++ executable for Microchip SAML10** project template.

Set the required project name and location directory.

Click **Next**.

If required, change any of the default project settings.

Click **Finish** to create the project.

Creating an SAML10 library project

To create a new library project:

Select the **File** > **New** > **New Project** menu item.

Select the **A library for Microchip SAML10** project template.

Set the required project name and location directory.

Click **Next**.

If required, change any of the default project settings.

Click **Finish** to create the project.

Creating an SAML10 externally built executable project

To create a new project that will allow you to debug an existing externally built executable file:

Select the **File** > **New** > **New Project** menu item.

Select the **An externally built executable for Microchip SAML10** project template.

Set the required project name and location directory.

Click **Next**.

Set the **Load File** project property to point to the executable file you want to download and debug.

If required, change any of the other default project settings.

Click **Finish** to create the project.

Creating an SAML10 CrossWorks Tasking Library executable project

To create a new C/C++ executable project configured to use the CrossWorks Tasking Library:

Select the **File** > **New** > **New Project** menu item.

Select the **A CrossWorks Tasking Library executable for Microchip SAML10** project template.

Set the required project name and location directory.

Click **Next**.

If required, change any of the other default project settings.

Click **Finish** to create the project.

Creating an SAML10 assembly code only executable project

Please note, this template does not add any C/C++ startup code or libraries and is therefore not suitable for creating projects that include C/C++ code.

To create a new assembly code only executable project without:

Select the **File** > **New** > **New Project** menu item.

Select the **An assembly code only executable for Microchip SAML10** project template.

Set the required project name and location directory.

Click **Next**.

If required, change any of the other default project settings.

Click **Finish** to create the project.

Opening SAML10 Sample Solutions

SAML10 Samples Solution

This solution contains general sample projects that run on SAML10 devices. To open the SAML10 Samples Solution:

Select the **Tools** > **Show Installed Packages** menu item.
Select the **Microchip SAML10 CPU Support Package** link.
Select the **Samples Solutions** > **SAML10 Samples Solution** link.

SAML10 CMSIS-DSP Samples Solution

This solution contains sample projects that use the CMSIS-DSP library running on SAML10 devices. To open the SAML10 CMSIS-DSP Samples Solution:

Select the **Tools** > **Show Installed Packages** menu item.
Select the **Microchip SAML10 CPU Support Package** link.
Select the **Sample Solutions** > **SAML10 CMSIS-DSP Samples Solution** link.

SAML10 Project Properties

Projects creating using the project templates in this support package have the following device specific project properties:

Heap Size

The heap size is set to be 256 bytes when a project is created. The heap size can be modified using the **Heap Size** project property.

Section Placement

You can select the memory configuration you require using the **Section Placement** project property.

For SAML10 projects, the set of placements are:

Flash - The application runs in internal Flash memory *(default)*.
Flash Vectors In RAM - The application runs in internal Flash memory and exception vectors are copied to RAM memory.
Flash Copy To RAM - The application starts in internal flash and copies itself to run from internal RAM memory.
RAM - The application runs from internal RAM memory only.

Stack Sizes

The main stack size is set to be 256 bytes when a project is created.

The process stack size is set to be 0 bytes when a project is created.

The main and process stack sizes can be modified using the **Main Stack Size** and **Process Stack Size** project properties.

To change the location of the stacks, edit the section placement file and place the *.stack* and *.stack_process* sections as required.

Startup From Reset

By default, the application will only startup from power-on/reset in *Release* configuration. This acts as a safety net in case you accidently download a program in FLASH during development that crashes and prevents the debugger from taking control of the target over the debug interface thus rendering the device unusable.

For SAML10 projects, the **Startup From Reset** project property can be set to one of the following:

No - The application will not startup from reset.

Release Only - The application will only startup from reset when built in *Release* configuration *(default)*.
Yes - The application will always startup from reset.

Target Processor

Once a project has been created you can target different devices by modifying the **Target Processor** project property. See the **SAML10 Devices** section for details on the files, preprocessor definitions and macro definitions used when a device is selected.

SAML10 Project Templates

The project template system simplifies the creation of new projects with the IDE, it also system makes it easy to create new projects with a text editor or script. All that needs to be specified is the project name, the support packages that the project is dependent on, the target processor and the source files you want to add to the project. For example, create a file called *example.hzp* with the following contents:

```
<!DOCTYPE CrossStudio_Project_File>
<solution Name="Example Solution">
  <project Name="Example Project" template_name="SAML10_EXE">
    <configuration Name="Common" package_dependencies="SAML10" Target="ATSAML10E16A" />
    <folder Name="Source Files">
      <file file_name="file1.c" />
      <file file_name="file2.c" />
    </folder>
  </project>
</solution>
```

You can also add any other property settings that the project requires such as preprocessor definitions or include paths using the property save name, for example:

```
<!DOCTYPE CrossStudio_Project_File>
<solution Name="Example Solution">
  <project Name="Example Project" template_name="SAML10_EXE">
    <configuration Name="Common" package_dependencies="SAML10" Target="ATSAML10E16A"
 c_preprocessor_definitions="MYDEF1=1;MYDEF2=TWO" c_user_include_directories="$(ProjectDir)/
include1;$(ProjectDir)/include2" />
    <folder Name="Source Files">
      <file file_name="file1.c" />
      <file file_name="file2.c" />
    </folder>
  </project>
</solution>
```

Available SAML10 project templates

Template Name	Template Description
SAML10_ASM_EXE	SAML10 Assembly Code Only Executable
SAML10_CTL_EXE	SAML10 CTL Executable
SAML10_EXE	SAML10 C/C++ Executable
SAML10_EXT_EXE	SAML10 Externally Built Executable
SAML10_LIB	SAML10 Library

SAML10 Devices

This package supports the following SAML10 devices:

- ATSAML10D15A
- ATSAML10E14A
- ATSAML10D14A
- ATSAML10D16A
- ATSAML10E15A
- ATSAML10E16A

ATSAML10D15A

Device Details	
CMSIS Header File	$(TargetsDir)/SAML10/CMSIS/include/sam.h
CMSIS Include Path	$(TargetsDir)/SAML10/CMSIS/include
CMSIS System File	$(TargetsDir)/SAML10/CMSIS/gcc/system_saml10d15a.c
Family	SAML10
Loader File	$(TargetsDir)/SAML10/Loader/SAML10_Loader.elf
Memory Map File	$(TargetsDir)/SAML10/XML/ATSAML10D15A_MemoryMap.xml
Register Definition File	$(TargetsDir)/SAML10/XML/ATSAML10D15A_Registers.xml
Vectors File	$(TargetsDir)/SAML10/Source/ATSAML10D15A_Vectors.s

Preprocessor Definitions
ARM_MATH_ARMV8MBL
__SAML10D15A__
__SAML10_FAMILY

Memory Segments	
FLASH	0x00000000 - 0x00007FFF
RAM	0x20000000 - 0x20001FFF
FLASH2	0x00400000 - 0x004007FF

Project Macros
DeviceIncludePath=$(TargetsDir)/SAML10/CMSIS/include
DeviceHeaderFile=$(TargetsDir)/SAML10/CMSIS/include/sam.h
DeviceLoaderFile=$(TargetsDir)/SAML10/Loader/SAML10_Loader.elf
DeviceRegisterDefinitionFile=$(TargetsDir)/SAML10/XML/ATSAML10D15A_Registers.xml
DeviceSystemFile=$(TargetsDir)/SAML10/CMSIS/gcc/system_saml10d15a.c
DeviceVectorsFile=$(TargetsDir)/SAML10/Source/ATSAML10D15A_Vectors.s
DeviceFamily=SAML10

ATSAML10E14A

Device Details	
CMSIS Header File	$(TargetsDir)/SAML10/CMSIS/include/sam.h
CMSIS Include Path	$(TargetsDir)/SAML10/CMSIS/include
CMSIS System File	$(TargetsDir)/SAML10/CMSIS/gcc/system_saml10e14a.c
Family	SAML10
Loader File	$(TargetsDir)/SAML10/Loader/SAML10_Loader.elf
Memory Map File	$(TargetsDir)/SAML10/XML/ATSAML10E14A_MemoryMap.xml
Register Definition File	$(TargetsDir)/SAML10/XML/ATSAML10E14A_Registers.xml
Vectors File	$(TargetsDir)/SAML10/Source/ATSAML10E14A_Vectors.s

Preprocessor Definitions
ARM_MATH_ARMV8MBL
__SAML10E14A__
__SAML10_FAMILY

Memory Segments	
FLASH	0x00000000 - 0x00003FFF
RAM	0x20000000 - 0x20000FFF
FLASH2	0x00400000 - 0x004007FF

Project Macros
DeviceIncludePath=$(TargetsDir)/SAML10/CMSIS/include
DeviceHeaderFile=$(TargetsDir)/SAML10/CMSIS/include/sam.h
DeviceLoaderFile=$(TargetsDir)/SAML10/Loader/SAML10_Loader.elf
DeviceRegisterDefinitionFile=$(TargetsDir)/SAML10/XML/ATSAML10E14A_Registers.xml
DeviceSystemFile=$(TargetsDir)/SAML10/CMSIS/gcc/system_saml10e14a.c
DeviceVectorsFile=$(TargetsDir)/SAML10/Source/ATSAML10E14A_Vectors.s
DeviceFamily=SAML10

ATSAML10D14A

Device Details	
CMSIS Header File	$(TargetsDir)/SAML10/CMSIS/include/sam.h
CMSIS Include Path	$(TargetsDir)/SAML10/CMSIS/include
CMSIS System File	$(TargetsDir)/SAML10/CMSIS/gcc/system_saml10d14a.c
Family	SAML10
Loader File	$(TargetsDir)/SAML10/Loader/SAML10_Loader.elf
Memory Map File	$(TargetsDir)/SAML10/XML/ATSAML10D14A_MemoryMap.xml
Register Definition File	$(TargetsDir)/SAML10/XML/ATSAML10D14A_Registers.xml
Vectors File	$(TargetsDir)/SAML10/Source/ATSAML10D14A_Vectors.s

Preprocessor Definitions
ARM_MATH_ARMV8MBL
__SAML10D14A__
__SAML10_FAMILY

Memory Segments	
FLASH	0x00000000 - 0x00003FFF
RAM	0x20000000 - 0x20000FFF
FLASH2	0x00400000 - 0x004007FF

Project Macros
DeviceIncludePath=$(TargetsDir)/SAML10/CMSIS/include
DeviceHeaderFile=$(TargetsDir)/SAML10/CMSIS/include/sam.h
DeviceLoaderFile=$(TargetsDir)/SAML10/Loader/SAML10_Loader.elf
DeviceRegisterDefinitionFile=$(TargetsDir)/SAML10/XML/ATSAML10D14A_Registers.xml
DeviceSystemFile=$(TargetsDir)/SAML10/CMSIS/gcc/system_saml10d14a.c
DeviceVectorsFile=$(TargetsDir)/SAML10/Source/ATSAML10D14A_Vectors.s
DeviceFamily=SAML10

ATSAML10D16A

Device Details	
CMSIS Header File	$(TargetsDir)/SAML10/CMSIS/include/sam.h
CMSIS Include Path	$(TargetsDir)/SAML10/CMSIS/include
CMSIS System File	$(TargetsDir)/SAML10/CMSIS/gcc/system_saml10d16a.c
Family	SAML10
Loader File	$(TargetsDir)/SAML10/Loader/SAML10_Loader.elf
Memory Map File	$(TargetsDir)/SAML10/XML/ATSAML10D16A_MemoryMap.xml
Register Definition File	$(TargetsDir)/SAML10/XML/ATSAML10D16A_Registers.xml
Vectors File	$(TargetsDir)/SAML10/Source/ATSAML10D16A_Vectors.s

Preprocessor Definitions
ARM_MATH_ARMV8MBL
__SAML10D16A__
__SAML10_FAMILY

Memory Segments	
FLASH	0x00000000 - 0x0000FFFF
RAM	0x20000000 - 0x20003FFF
FLASH2	0x00400000 - 0x004007FF

Project Macros
DeviceIncludePath=$(TargetsDir)/SAML10/CMSIS/include
DeviceHeaderFile=$(TargetsDir)/SAML10/CMSIS/include/sam.h
DeviceLoaderFile=$(TargetsDir)/SAML10/Loader/SAML10_Loader.elf
DeviceRegisterDefinitionFile=$(TargetsDir)/SAML10/XML/ATSAML10D16A_Registers.xml
DeviceSystemFile=$(TargetsDir)/SAML10/CMSIS/gcc/system_saml10d16a.c
DeviceVectorsFile=$(TargetsDir)/SAML10/Source/ATSAML10D16A_Vectors.s
DeviceFamily=SAML10

ATSAML10E15A

Device Details	
CMSIS Header File	$(TargetsDir)/SAML10/CMSIS/include/sam.h
CMSIS Include Path	$(TargetsDir)/SAML10/CMSIS/include
CMSIS System File	$(TargetsDir)/SAML10/CMSIS/gcc/system_saml10e15a.c
Family	SAML10
Loader File	$(TargetsDir)/SAML10/Loader/SAML10_Loader.elf
Memory Map File	$(TargetsDir)/SAML10/XML/ATSAML10E15A_MemoryMap.xml
Register Definition File	$(TargetsDir)/SAML10/XML/ATSAML10E15A_Registers.xml
Vectors File	$(TargetsDir)/SAML10/Source/ATSAML10E15A_Vectors.s

Preprocessor Definitions
ARM_MATH_ARMV8MBL
__SAML10E15A__
__SAML10_FAMILY

Memory Segments	
FLASH	0x00000000 - 0x00007FFF
RAM	0x20000000 - 0x20001FFF
FLASH2	0x00400000 - 0x004007FF

Project Macros
DeviceIncludePath=$(TargetsDir)/SAML10/CMSIS/include
DeviceHeaderFile=$(TargetsDir)/SAML10/CMSIS/include/sam.h
DeviceLoaderFile=$(TargetsDir)/SAML10/Loader/SAML10_Loader.elf
DeviceRegisterDefinitionFile=$(TargetsDir)/SAML10/XML/ATSAML10E15A_Registers.xml
DeviceSystemFile=$(TargetsDir)/SAML10/CMSIS/gcc/system_saml10e15a.c
DeviceVectorsFile=$(TargetsDir)/SAML10/Source/ATSAML10E15A_Vectors.s
DeviceFamily=SAML10

ATSAML10E16A

Device Details	
CMSIS Header File	$(TargetsDir)/SAML10/CMSIS/include/sam.h
CMSIS Include Path	$(TargetsDir)/SAML10/CMSIS/include
CMSIS System File	$(TargetsDir)/SAML10/CMSIS/gcc/system_saml10e16a.c
Family	SAML10
Loader File	$(TargetsDir)/SAML10/Loader/SAML10_Loader.elf
Memory Map File	$(TargetsDir)/SAML10/XML/ATSAML10E16A_MemoryMap.xml
Register Definition File	$(TargetsDir)/SAML10/XML/ATSAML10E16A_Registers.xml
Vectors File	$(TargetsDir)/SAML10/Source/ATSAML10E16A_Vectors.s

Preprocessor Definitions
ARM_MATH_ARMV8MBL
__SAML10E16A__
__SAML10_FAMILY

Memory Segments	
FLASH	0x00000000 - 0x0000FFFF
RAM	0x20000000 - 0x20003FFF
FLASH2	0x00400000 - 0x004007FF

Project Macros
DeviceIncludePath=$(TargetsDir)/SAML10/CMSIS/include
DeviceHeaderFile=$(TargetsDir)/SAML10/CMSIS/include/sam.h
DeviceLoaderFile=$(TargetsDir)/SAML10/Loader/SAML10_Loader.elf
DeviceRegisterDefinitionFile=$(TargetsDir)/SAML10/XML/ATSAML10E16A_Registers.xml
DeviceSystemFile=$(TargetsDir)/SAML10/CMSIS/gcc/system_saml10e16a.c
DeviceVectorsFile=$(TargetsDir)/SAML10/Source/ATSAML10E16A_Vectors.s
DeviceFamily=SAML10