



SAML11 CPU Support Package Guide

Version: 4.1



Contents

SAML11 Support Package	5
Creating SAML11 Projects	6
Opening SAML11 Sample Solutions	8
SAML11 Project Properties	9
SAML11 Project Templates	11
SAML11 Devices	12
ATSAML11D16A	13
ATSAML11E16A	14
ATSAML11E14A	15
ATSAML11E15A	16
ATSAML11D15A	17
ATSAML11D14A	18



SAML11 Support Package

This guide describes the following features of the SAML11 CPU support package:

- [How to create SAML11 projects](#)
- [How to open SAML11 sample projects](#)
- [SAML11 specific project properties](#)
- [SAML11 specific project templates](#)
- [Supported SAML11 devices](#)

Creating SAML11 Projects

Creating an SAML11 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 SAML11** 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 SAML11 library project

To create a new library project:

- Select the **File > New > New Project** menu item.
- Select the **A library for Microchip SAML11** 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 SAML11 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 SAML11** 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 SAML11 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 SAML11** 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 SAML11 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 SAML11** 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 SAML11 Sample Solutions

SAML11 Samples Solution

This solution contains general sample projects that run on SAML11 devices. To open the SAML11 Samples Solution:

- Select the **Tools > Show Installed Packages** menu item.
- Select the **Microchip SAML11 CPU Support Package** link.
- Select the **Samples Solutions > SAML11 Samples Solution** link.

SAML11 CMSIS-DSP Samples Solution

This solution contains sample projects that use the CMSIS-DSP library running on SAML11 devices. To open the SAML11 CMSIS-DSP Samples Solution:

- Select the **Tools > Show Installed Packages** menu item.
- Select the **Microchip SAML11 CPU Support Package** link.
- Select the **Sample Solutions > SAML11 CMSIS-DSP Samples Solution** link.

SAML11 Rescue Solution

This solution contains a project that can be used to carry out a chip erase if you can no longer connect to the device using a regular project (this can happen if the UROW or BOCOR CRCs are invalid for example). It does not contain a program that can be run, it is simply a project that has been set up to allow you to connect when the device is in this state. To open the SAML11 Rescue Solution:

- Select the **Tools > Show Installed Packages** menu item.
- Select the **Microchip SAML11 CPU Support Package** link.
- Select the **Sample Solutions > SAML11 Rescue Solution** link.

To carry out a chip erase:

- Click **View > Targets** to open the targets window.
- Right click on the target interface in the targets window and select **Connect**.
- Right click on the target interface in the targets window and select **Chip Erase**.

SAML11 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 SAML11 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 accidentally 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 SAML11 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 [SAML11 Devices](#) section for details on the files, preprocessor definitions and macro definitions used when a device is selected.

SAML11 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="SAML11_EXE">
    <configuration Name="Common" package_dependencies="SAML11" Target="ATSAML11E16A" />
    <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="SAML11_EXE">
    <configuration Name="Common" package_dependencies="SAML11" Target="ATSAML11E16A"
      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 SAML11 project templates

Template Name	Template Description
SAML11_ASM_EXE	SAML11 Assembly Code Only Executable
SAML11_CTL_EXE	SAML11 CTL Executable
SAML11_EXE	SAML11 C/C++ Executable
SAML11_EXT_EXE	SAML11 Externally Built Executable
SAML11_LIB	SAML11 Library

SAML11 Devices

This package supports the following SAML11 devices:

[ATSAML11D16A](#)

[ATSAML11E16A](#)

[ATSAML11E14A](#)

[ATSAML11E15A](#)

[ATSAML11D15A](#)

[ATSAML11D14A](#)

ATSAML11D16A

Device Details

CMSIS Header File	\$(TargetsDir)/SAML11/CMSIS/include/sam.h
CMSIS Include Path	\$(TargetsDir)/SAML11/CMSIS/include
CMSIS System File	\$(TargetsDir)/SAML11/CMSIS/gcc/system_saml11d16a.c
Family	SAML11
Loader File	\$(TargetsDir)/SAML11/Loader/SAML11_Loader.elf
Memory Map File	\$(TargetsDir)/SAML11/XML/ATSAML11D16A_MemoryMap.xml
Register Definition File	\$(TargetsDir)/SAML11/XML/ATSAML11D16A_Registers.xml
Vectors File	\$(TargetsDir)/SAML11/Source/ATSAML11D16A_Vectors.s

Preprocessor Definitions

ARM_MATH_ARMV8MBL

__SAML11D16A__

__SAML11_FAMILY

Memory Segments

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

Project Macros

DeviceIncludePath=\$(TargetsDir)/SAML11/CMSIS/include

DeviceHeaderFile=\$(TargetsDir)/SAML11/CMSIS/include/sam.h

DeviceLoaderFile=\$(TargetsDir)/SAML11/Loader/SAML11_Loader.elf

DeviceRegisterDefinitionFile=\$(TargetsDir)/SAML11/XML/ATSAML11D16A_Registers.xml

DeviceSystemFile=\$(TargetsDir)/SAML11/CMSIS/gcc/system_saml11d16a.c

DeviceVectorsFile=\$(TargetsDir)/SAML11/Source/ATSAML11D16A_Vectors.s

DeviceFamily=SAML11

ATSAML11E16A

Device Details	
CMSIS Header File	\$(TargetsDir)/SAML11/CMSIS/include/sam.h
CMSIS Include Path	\$(TargetsDir)/SAML11/CMSIS/include
CMSIS System File	\$(TargetsDir)/SAML11/CMSIS/gcc/system_saml11e16a.c
Family	SAML11
Loader File	\$(TargetsDir)/SAML11/Loader/SAML11_Loader.elf
Memory Map File	\$(TargetsDir)/SAML11/XML/ATSAML11E16A_MemoryMap.xml
Register Definition File	\$(TargetsDir)/SAML11/XML/ATSAML11E16A_Registers.xml
Vectors File	\$(TargetsDir)/SAML11/Source/ATSAML11E16A_Vectors.s

Preprocessor Definitions	
ARM_MATH_ARMV8MBL	
__SAML11E16A__	
__SAML11_FAMILY	

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

Project Macros	
DeviceIncludePath	\$(TargetsDir)/SAML11/CMSIS/include
DeviceHeaderFile	\$(TargetsDir)/SAML11/CMSIS/include/sam.h
DeviceLoaderFile	\$(TargetsDir)/SAML11/Loader/SAML11_Loader.elf
DeviceRegisterDefinitionFile	\$(TargetsDir)/SAML11/XML/ATSAML11E16A_Registers.xml
DeviceSystemFile	\$(TargetsDir)/SAML11/CMSIS/gcc/system_saml11e16a.c
DeviceVectorsFile	\$(TargetsDir)/SAML11/Source/ATSAML11E16A_Vectors.s
DeviceFamily	SAML11

ATSAML11E14A

Device Details

CMSIS Header File	\$(TargetsDir)/SAML11/CMSIS/include/sam.h
CMSIS Include Path	\$(TargetsDir)/SAML11/CMSIS/include
CMSIS System File	\$(TargetsDir)/SAML11/CMSIS/gcc/system_saml11e14a.c
Family	SAML11
Loader File	\$(TargetsDir)/SAML11/Loader/SAML11_Loader.elf
Memory Map File	\$(TargetsDir)/SAML11/XML/ATSAML11E14A_MemoryMap.xml
Register Definition File	\$(TargetsDir)/SAML11/XML/ATSAML11E14A_Registers.xml
Vectors File	\$(TargetsDir)/SAML11/Source/ATSAML11E14A_Vectors.s

Preprocessor Definitions

ARM_MATH_ARMV8MBL

__SAML11E14A__

__SAML11_FAMILY

Memory Segments

FLASH	0x00000000 - 0x00003FFF
RAM	0x20000000 - 0x20001FFF
FLASH2	0x00400000 - 0x004007FF

Project Macros

DeviceIncludePath=\$(TargetsDir)/SAML11/CMSIS/include

DeviceHeaderFile=\$(TargetsDir)/SAML11/CMSIS/include/sam.h

DeviceLoaderFile=\$(TargetsDir)/SAML11/Loader/SAML11_Loader.elf

DeviceRegisterDefinitionFile=\$(TargetsDir)/SAML11/XML/ATSAML11E14A_Registers.xml

DeviceSystemFile=\$(TargetsDir)/SAML11/CMSIS/gcc/system_saml11e14a.c

DeviceVectorsFile=\$(TargetsDir)/SAML11/Source/ATSAML11E14A_Vectors.s

DeviceFamily=SAML11

ATSAML11E15A

Device Details

CMSIS Header File	\$(TargetsDir)/SAML11/CMSIS/include/sam.h
CMSIS Include Path	\$(TargetsDir)/SAML11/CMSIS/include
CMSIS System File	\$(TargetsDir)/SAML11/CMSIS/gcc/system_saml11e15a.c
Family	SAML11
Loader File	\$(TargetsDir)/SAML11/Loader/SAML11_Loader.elf
Memory Map File	\$(TargetsDir)/SAML11/XML/ATSAML11E15A_MemoryMap.xml
Register Definition File	\$(TargetsDir)/SAML11/XML/ATSAML11E15A_Registers.xml
Vectors File	\$(TargetsDir)/SAML11/Source/ATSAML11E15A_Vectors.s

Preprocessor Definitions

ARM_MATH_ARMV8MBL

__SAML11E15A__

__SAML11_FAMILY

Memory Segments

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

Project Macros

DeviceIncludePath=\$(TargetsDir)/SAML11/CMSIS/include

DeviceHeaderFile=\$(TargetsDir)/SAML11/CMSIS/include/sam.h

DeviceLoaderFile=\$(TargetsDir)/SAML11/Loader/SAML11_Loader.elf

DeviceRegisterDefinitionFile=\$(TargetsDir)/SAML11/XML/ATSAML11E15A_Registers.xml

DeviceSystemFile=\$(TargetsDir)/SAML11/CMSIS/gcc/system_saml11e15a.c

DeviceVectorsFile=\$(TargetsDir)/SAML11/Source/ATSAML11E15A_Vectors.s

DeviceFamily=SAML11

ATSAML11D15A

Device Details

CMSIS Header File	\$(TargetsDir)/SAML11/CMSIS/include/sam.h
CMSIS Include Path	\$(TargetsDir)/SAML11/CMSIS/include
CMSIS System File	\$(TargetsDir)/SAML11/CMSIS/gcc/system_saml11d15a.c
Family	SAML11
Loader File	\$(TargetsDir)/SAML11/Loader/SAML11_Loader.elf
Memory Map File	\$(TargetsDir)/SAML11/XML/ATSAML11D15A_MemoryMap.xml
Register Definition File	\$(TargetsDir)/SAML11/XML/ATSAML11D15A_Registers.xml
Vectors File	\$(TargetsDir)/SAML11/Source/ATSAML11D15A_Vectors.s

Preprocessor Definitions

ARM_MATH_ARMV8MBL
 __SAML11D15A__
 __SAML11_FAMILY

Memory Segments

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

Project Macros

DeviceIncludePath=\$(TargetsDir)/SAML11/CMSIS/include
 DeviceHeaderFile=\$(TargetsDir)/SAML11/CMSIS/include/sam.h
 DeviceLoaderFile=\$(TargetsDir)/SAML11/Loader/SAML11_Loader.elf
 DeviceRegisterDefinitionFile=\$(TargetsDir)/SAML11/XML/ATSAML11D15A_Registers.xml
 DeviceSystemFile=\$(TargetsDir)/SAML11/CMSIS/gcc/system_saml11d15a.c
 DeviceVectorsFile=\$(TargetsDir)/SAML11/Source/ATSAML11D15A_Vectors.s
 DeviceFamily=SAML11

ATSAML11D14A

Device Details	
CMSIS Header File	\$(TargetsDir)/SAML11/CMSIS/include/sam.h
CMSIS Include Path	\$(TargetsDir)/SAML11/CMSIS/include
CMSIS System File	\$(TargetsDir)/SAML11/CMSIS/gcc/system_saml11d14a.c
Family	SAML11
Loader File	\$(TargetsDir)/SAML11/Loader/SAML11_Loader.elf
Memory Map File	\$(TargetsDir)/SAML11/XML/ATSAML11D14A_MemoryMap.xml
Register Definition File	\$(TargetsDir)/SAML11/XML/ATSAML11D14A_Registers.xml
Vectors File	\$(TargetsDir)/SAML11/Source/ATSAML11D14A_Vectors.s
Preprocessor Definitions	
ARM_MATH_ARMV8MBL	
__SAML11D14A__	
__SAML11_FAMILY	
Memory Segments	
FLASH	0x00000000 - 0x00003FFF
RAM	0x20000000 - 0x20001FFF
FLASH2	0x00400000 - 0x004007FF
Project Macros	
DeviceIncludePath	\$(TargetsDir)/SAML11/CMSIS/include
DeviceHeaderFile	\$(TargetsDir)/SAML11/CMSIS/include/sam.h
DeviceLoaderFile	\$(TargetsDir)/SAML11/Loader/SAML11_Loader.elf
DeviceRegisterDefinitionFile	\$(TargetsDir)/SAML11/XML/ATSAML11D14A_Registers.xml
DeviceSystemFile	\$(TargetsDir)/SAML11/CMSIS/gcc/system_saml11d14a.c
DeviceVectorsFile	\$(TargetsDir)/SAML11/Source/ATSAML11D14A_Vectors.s
DeviceFamily	SAML11