

SPI Storm Studio

Getting Started Guide



Table of Contents

1Introduction.....	4
2Starting up with an existing SPI Storm Studio project.....	4
2.1What do you need to get started?.....	4
2.2Connecting and configuring your SPI Storm device.....	4
2.3Loading an existing SPI Storm Studio project.....	6
3Using SPI Storm for the first time.....	7
3.1SPI Storm Device at a glance.....	7
3.2What is a “Project” in SPI Storm Studio ?.....	8
3.3Step-by-Step: how to define a protocol with SPI Storm Studio?.....	8

References

[]

History

Version	Date	Description
v1.00	05 April 2011	Initial revision
v1.02	September 2012	Corrected missing pictures

1 Introduction

2 Starting up with an existing SPI Storm Studio project

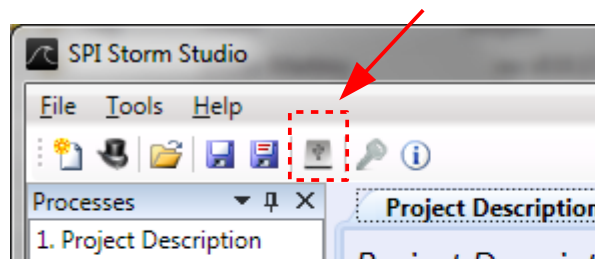
2.1 What do you need to get started?

To start using SPI Storm, you need:

- A SPI Storm Device;
- SPI Storm Studio Software;
- A PC installed with Microsoft Windows XP or Microsoft Windows 7 (32 bit or 64 bit version);
- .NET 4.0 or .NET 4.0 client profile framework installed (or a more recent version) – download link: <http://www.microsoft.com/downloads/en/details.aspx?FamilyID=e5ad0459-cbcc-4b4f-97b6-fb17111cf544>
- One free USB port.

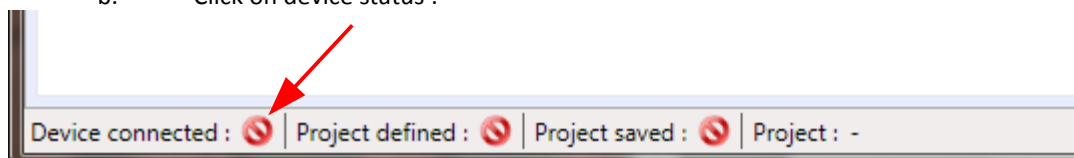
2.2 Connecting and configuring your SPI Storm device

- 1) Install SPI Storm Studio software
- 2) Copy your license file into the SPI Storm Studio installation directory
- 3) Connect your SPI Storm device to the PC USB port with the provided USB cable
- 4) When prompted, locate and install the USB driver:
 - a. 32-bit operating system - the driver is located in: \drivers\x86
 - b. 64-bit operating system – the driver is located in: \drivers\x64
- 5) Start SPI Storm Studio Software
- 6) Open the device selection window:
 - a. Locate and click on the 'Select Device' icon in the main toolbar:



- or:

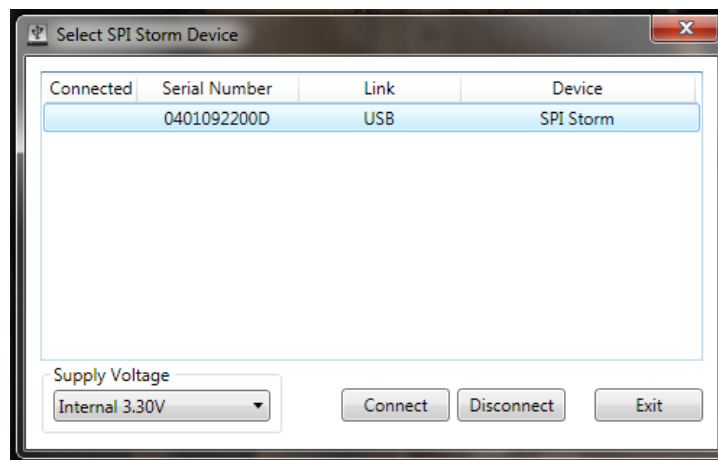
- b. Click on device status :



- or:

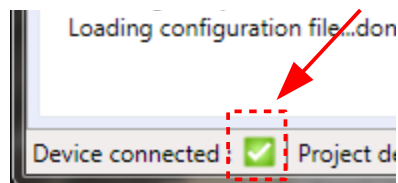
- c. Go to **Tools > Select Device**

- 7) Click on your device in the device selection window and click on button “Connect”.



If your device is not physically connected to your PC or if the USB driver is not properly installed, the device won't be recognized and the list in the 'Select Device' Window will be empty.

After a few seconds, the device is properly configured and ready to be used. The 'device selection window disappears and the device status turns green:



Repeat steps 5) to 7) above each time you restart a SPI Storm Studio session.

REMARK: SPI Storm device is now properly configured to run with a 3.3V LVCMOS signaling standard.

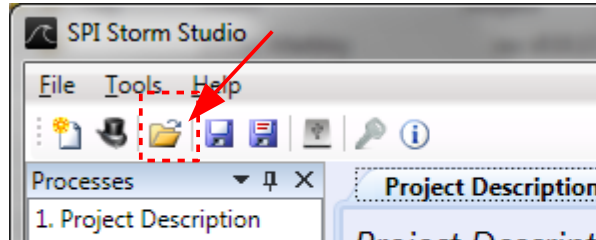
To use a different voltage for the I/Os, you must:

- 1) Select the right range in the 'Select SPI Storm Device' window**
- 2) Supply the proper corresponding voltage through the I/O external voltage connector.**

! If you don't supply the external voltage, the device will automatically switch back to 3.3V voltage for the I/Os!

2.3 Loading an existing SPI Storm Studio project

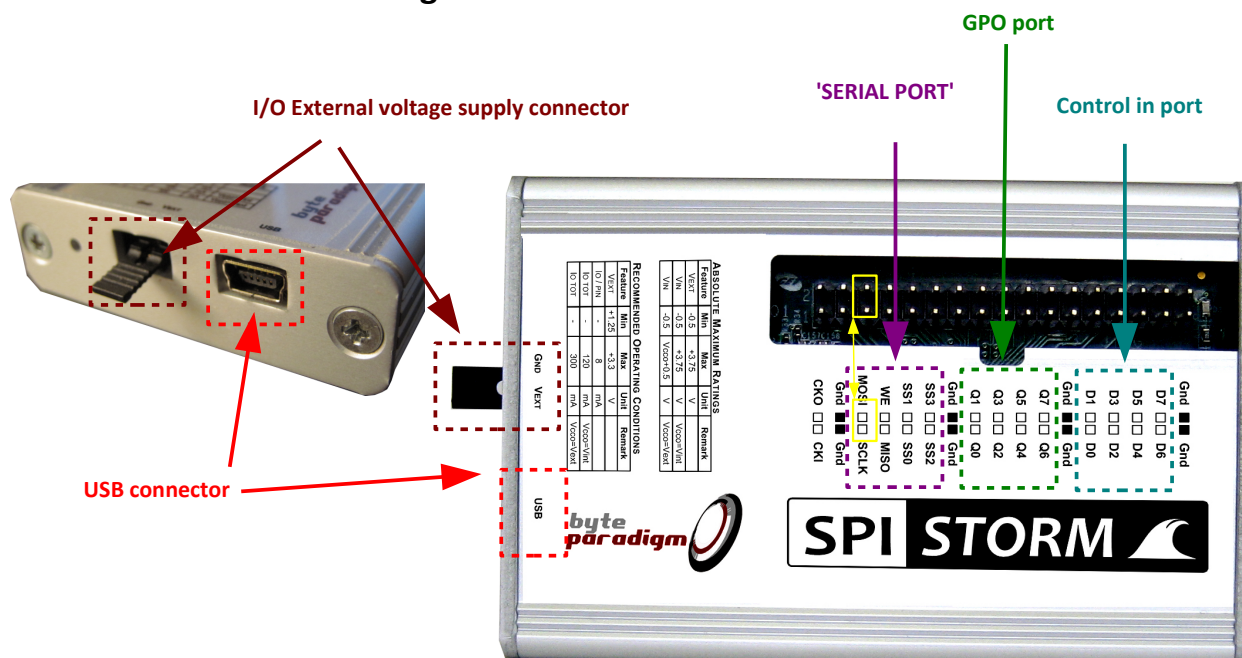
- 1) Click on “Open” icon or **File > Open**



- 2) Locate the desired .ssp (SPI Storm Project) file – in this document, we use 'ExampleFile1.ssp'.
- 3) Click on **OK**.

3 Using SPI Storm for the first time

3.1 SPI Storm Device at a glance



There are 3 main parts to the SPI Storm device connector:

Port name	Number of I/O pins	I/O names	Purpose
SERIAL PORT	8, with 4 multi-purpose I/Os.	SCLK MOSI (DQ0) MISO (DQ1) WE SS0 SS1 SS2 (DQ2) SS3 (DQ3)	Flexible port for the generation of digital protocols using a clock signal (SCLK), up to 4 slave select signals (SS0 ... SS3), and up to 4 data lines (MOSI, MISO / DQ0... DQ3). The direction of the data lines can be defined as output, input or hi-Z. Open drain I/O is also supported.
GPO PORT	8	Q0 ... Q7	General-purpose outputs: set of outputs where arbitrary digital pattern can be generated.
Control in PORT	8	D0 ... D7	Set of input pins used for triggering.
GND pins	8	GND	Ground pins – SPI Storm device and slave devices must share the same ground signals.
Special clock I/Os	2	CKO CKI	CKO mirrors an internally generated clock signal; CKI can be used as an input pin for providing an external reference clock signal.

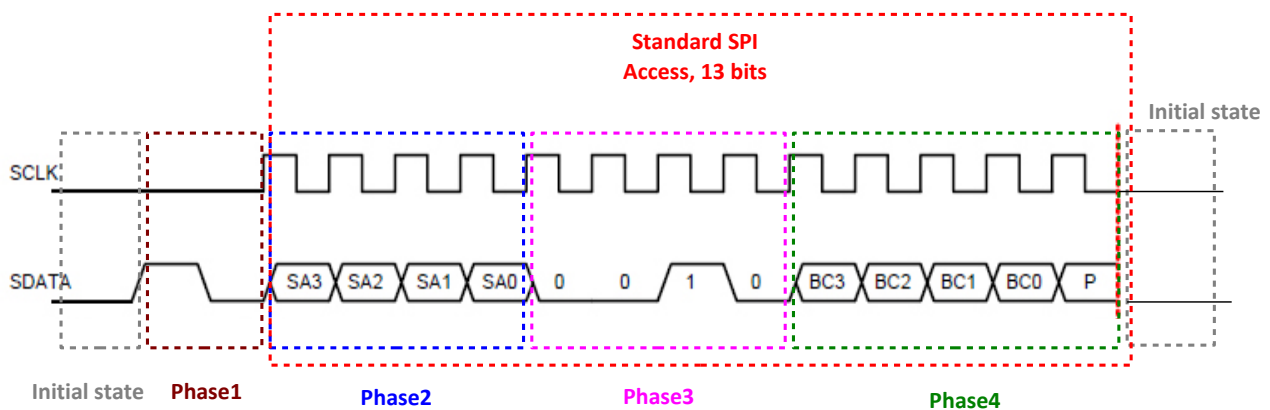
3.2 What is a “Project” in SPI Storm Studio ?

A 'Project' in SPI Storm Studio is a set of settings and definitions that can be saved as a .ssp file and later recalled. Each .ssp file encloses:

- The definition of a “Initial State” for the I/O connector of SPI Storm device
- The definition of 'Devices' and 'Macros' for 'Standard SPI' protocols
- The definition of 'Segments' and 'Macros' for 'Custom SPI' protocols (or custom serial protocols)
- The definition of 'Segments' and 'Macros' for the GPO port
- The definition of programs.
- Additional information about project name, source and destination files for data and logging.

3.3 Step-by-Step: how to define a protocol with SPI Storm Studio?

Let us imagine we would like to communicate over a serial connection by using the following protocol:



This is a simple access with a protocol that uses a single clock line and a single DATA line. In this protocol, the 2 signals are called SCLK and SDATA respectively; they'll be mapped onto SCLK and MOSI (DQ0) on the SPI Storm device. When toggling, the clock frequency is 25 MHz.

This example is pre-configured in file 'ExampleFile1.ssp'.

It is composed of the following segments:

- 1) Initial state : SCLK and SDATA are at IDLE0
- 2) Phase 1: SCLK remains inactive and a pulse is generated onto the SDATA signal.
- 3) Phase 2: a 4 bits data (address) is sent on SDATA while the SCLK signal toggles. Data changes on the rising edge of the clock signal.
- 4) Phase 3: a 4 bits constant pattern “0010” is sent on the data line, clock toggling
- 5) Phase 4: 4 bits data + a parity bit are sent on the data line, clock toggling.
- 6) All lines return to the initial state

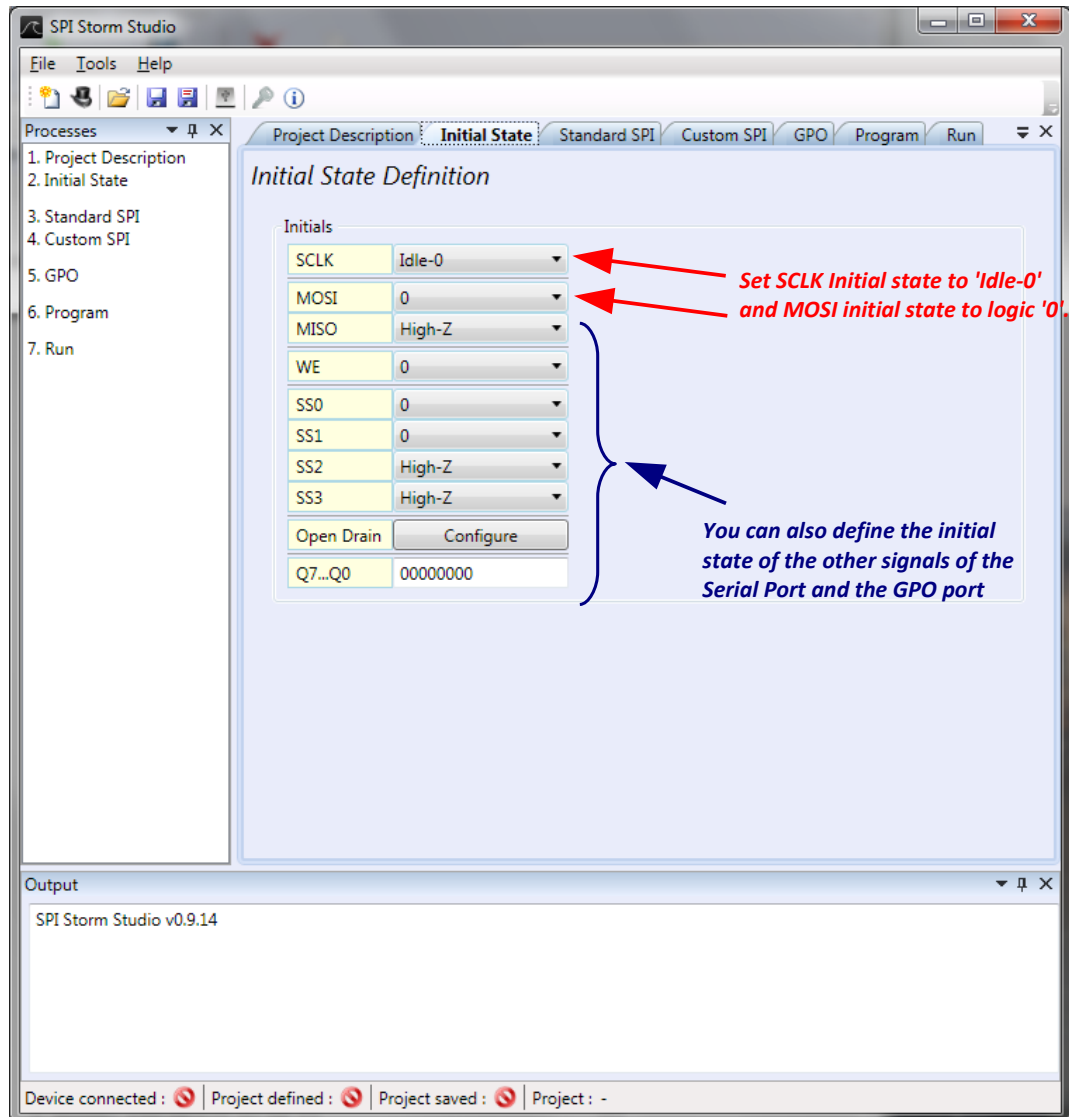
Alternatively, the same protocol can be seen as:

- 1) Initial state
- 2) Phase 1
- 3) Standard SPI access (Serial Peripheral Interface) – 13 bits

This alternative is provided in the ExampleFile1.ssp file as well. Please check 'Standard SPI' tab.

STEP 1: Define initial state

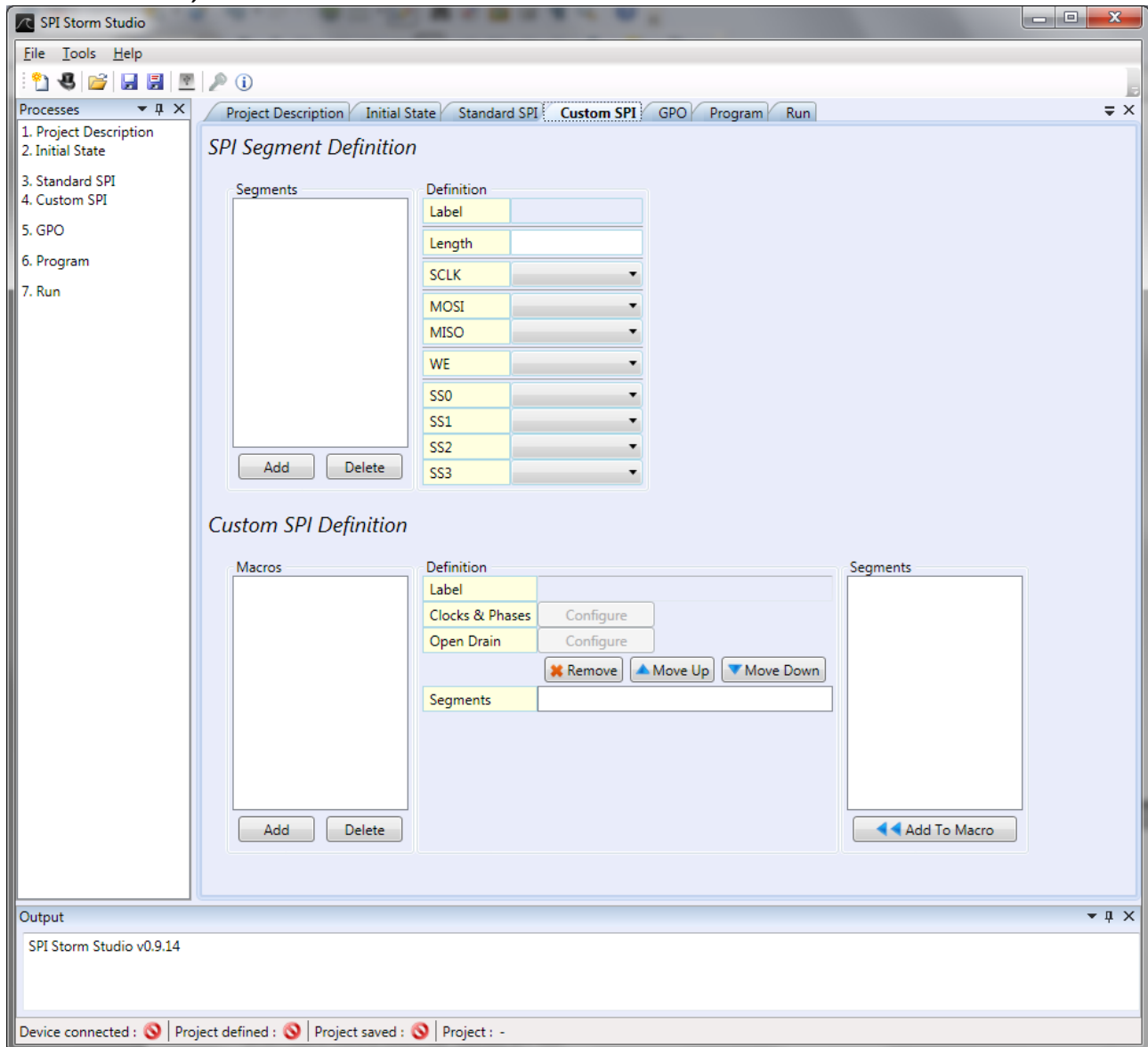
In SPI Storm Studio, select the 'Initial State' tab:



STEP 2: Define custom SPI segments and macros

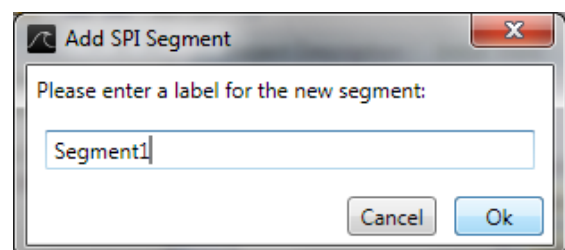
Schematically, we'll define 'macros' corresponding to each phase of the protocol. A macro is made out of one or more successive segments.

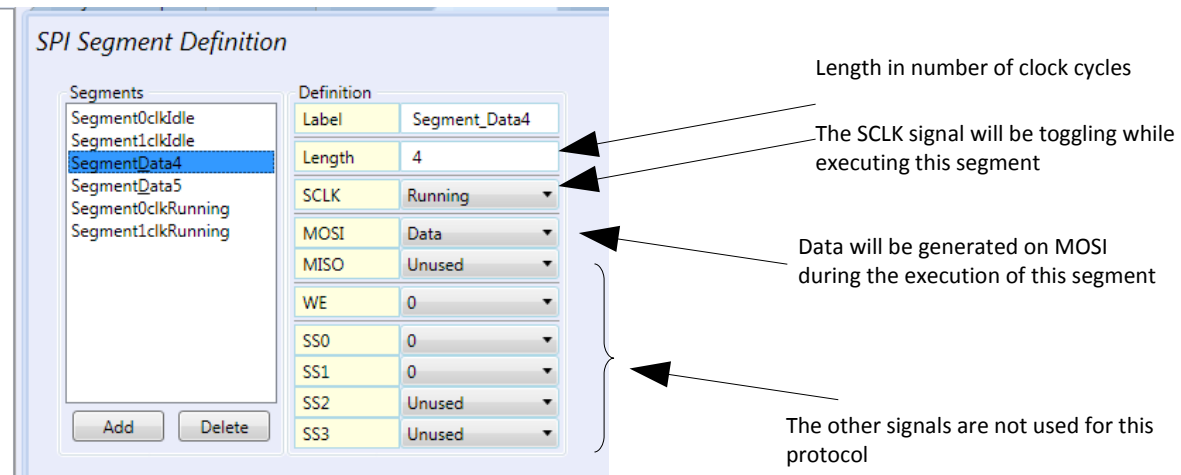
In SPI Storm Studio, select the 'Custom SPI' tab:



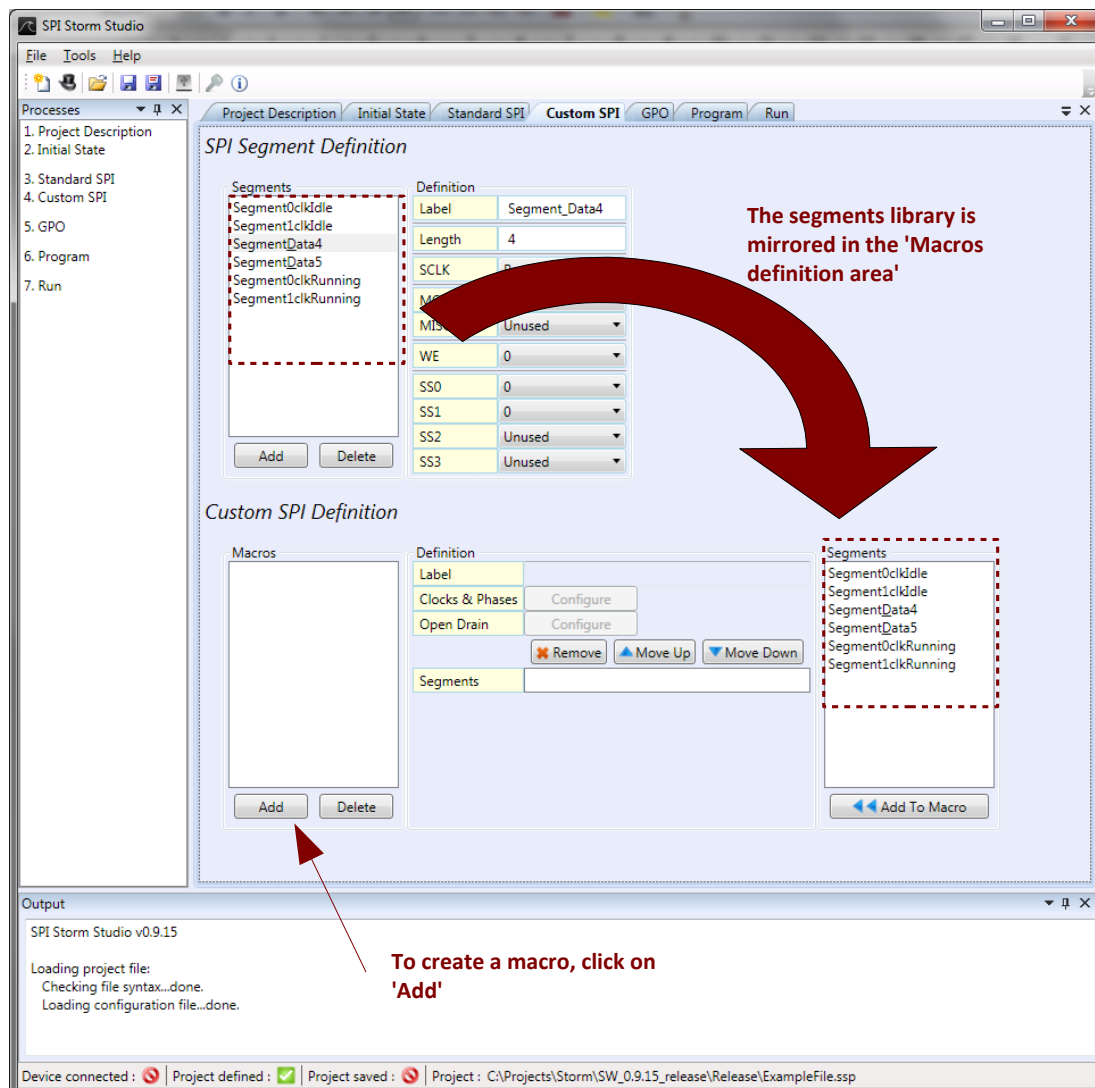
To add a new segment:

- Click on 'Add' button in the segment definition area and enter a name for it in the window that popped; click on 'OK':
- Setup this segment by using the controls in the 'definition area' for the segment: select the segment name in the segments list
- Repeat this procedure until you have defined define all the segments that you need of the protocol (load ExampleFile1.ssp to check the details of each segment).

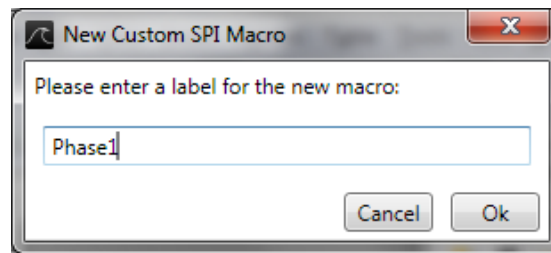




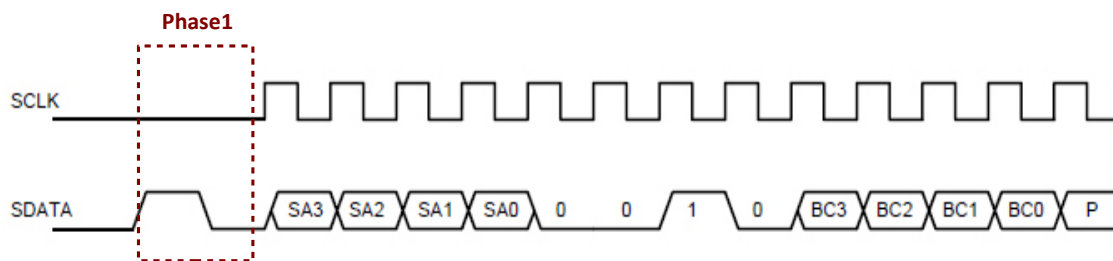
The 'segments' defined in this example are the basic building blocks for implementing the complete protocol. Here is how the 'Custom SPI' tab looks like after segments definition:



To create a macro, click on 'Add' in the macro definition area. Specify a name in the pop-up window:



We'll match each macro that we define with one phase of the protocol.



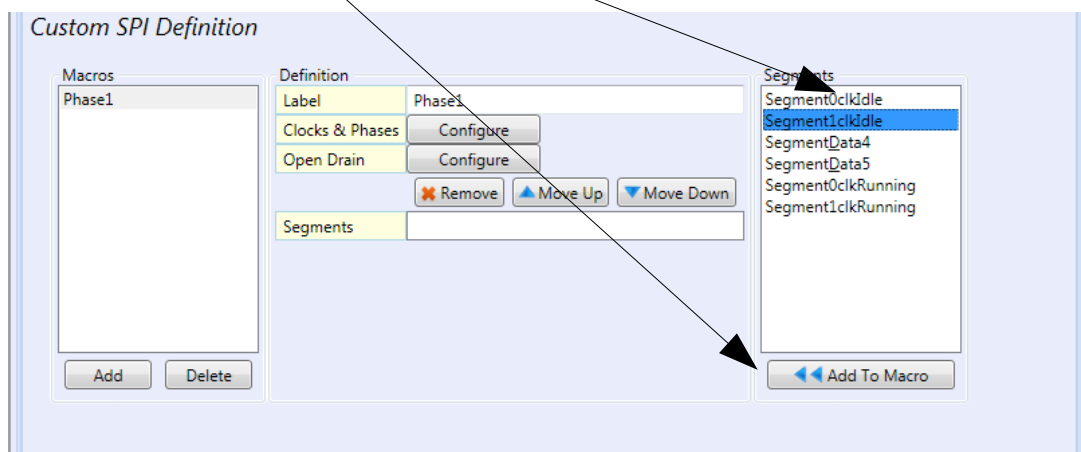
For instance, we'll detail the definition of 'Phase1':

This macro is composed of the following segments:

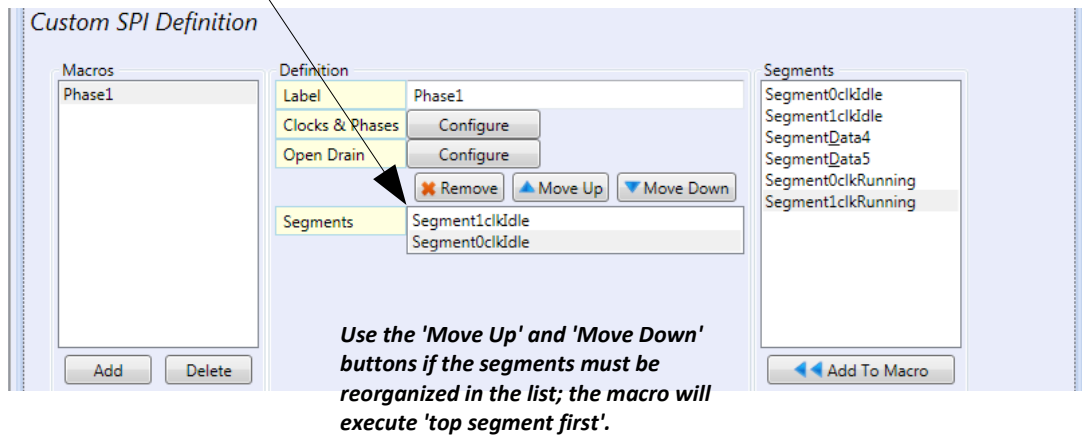
- Segment 'Segment1clkIdle', which sets MOSI to '1' while the SCLK is held low;
- Segment 'Segment0clkIdle', which sets MODI to '0' while the SCLK is held low.

Proceed as follow:

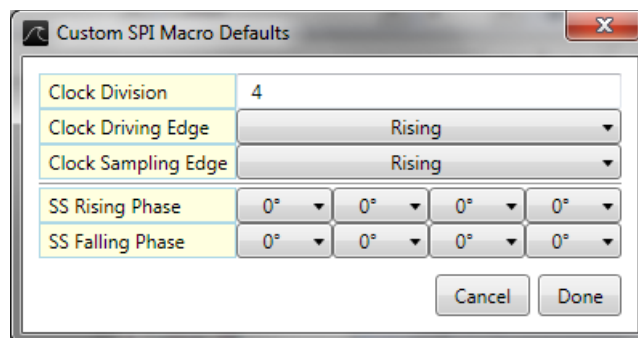
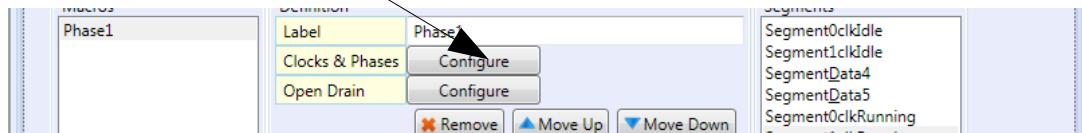
- 1) Select one segment in the library
- 2) Click on 'Add to macro' button.
- 3) Do so for all segments you wish to add to the macro.



The segments are added to the macro:



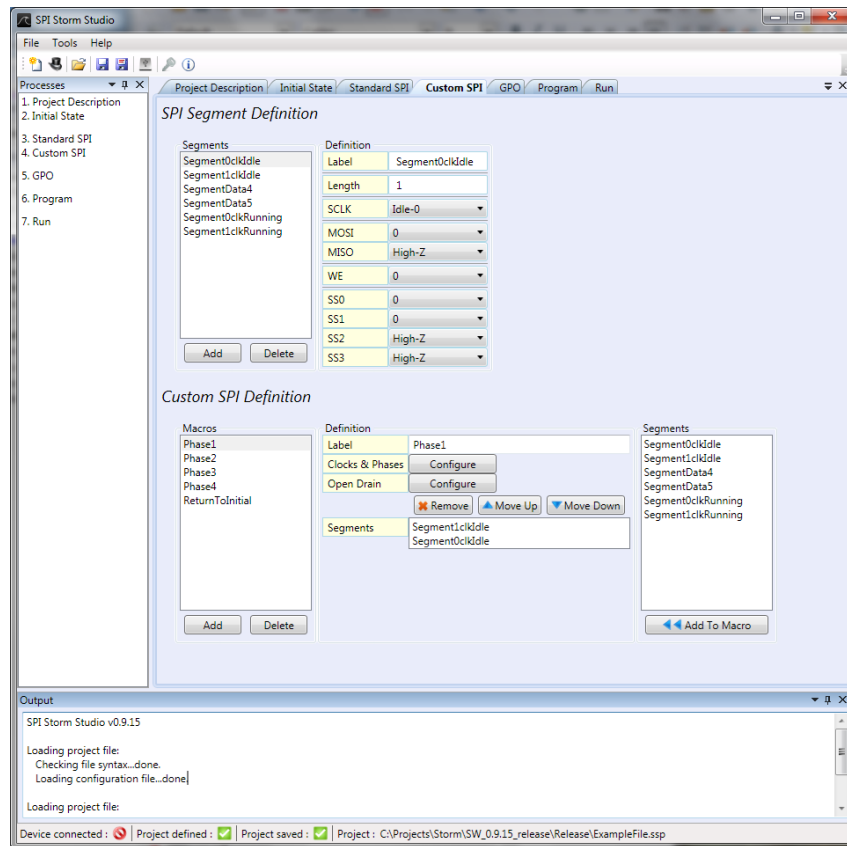
Now, configure the clock for the current macro (click):



In the 'Custom SPI Macro Defaults' window, you can specify the clock frequency and other settings related to clock. Clock frequency is actually defined by 'Clock division', which is a dividing factor for the clock relative to a 100 MHz reference clock.

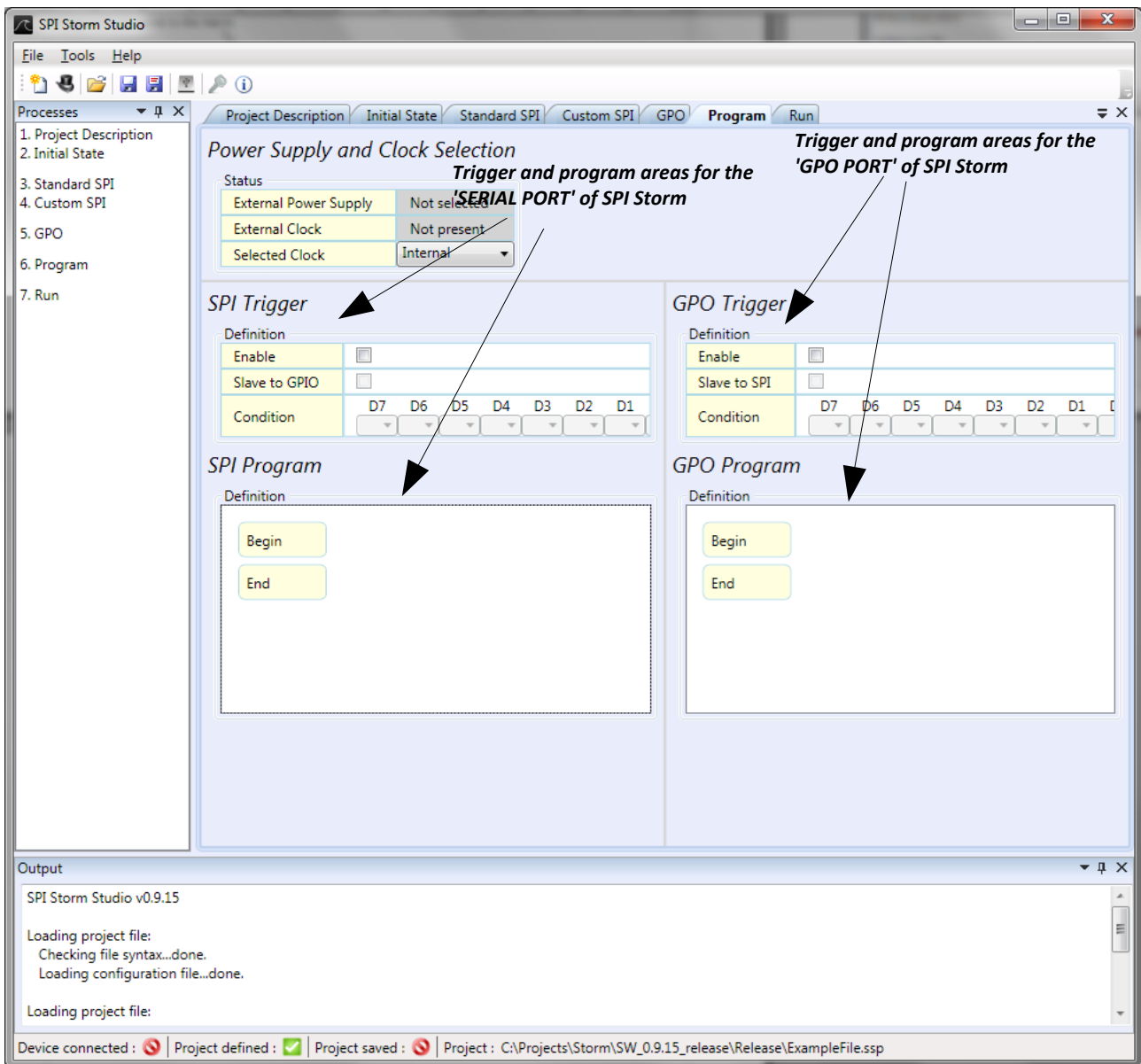
For instance, typing '4' for clock division will result in $100 / 4 = 25$ MHz clock for SCLK.

Please do so for the definition of all Phases / Macros for the protocol.



STEP 3: Define a program

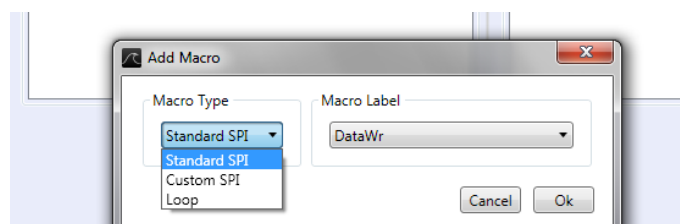
Switch to 'Program tab'.



The program tab is used to organize how the defined macros are executed.

It works like a sequential program that starts with 'Begin' and finishes with 'End'.

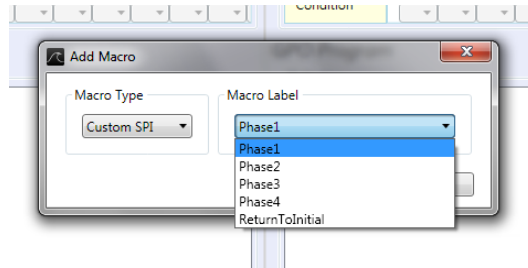
To add a command to this program, right-click where you wish to insert it in the program. Select the 'Insert after' or 'Insert before' to insert a new macro. The following window opens:



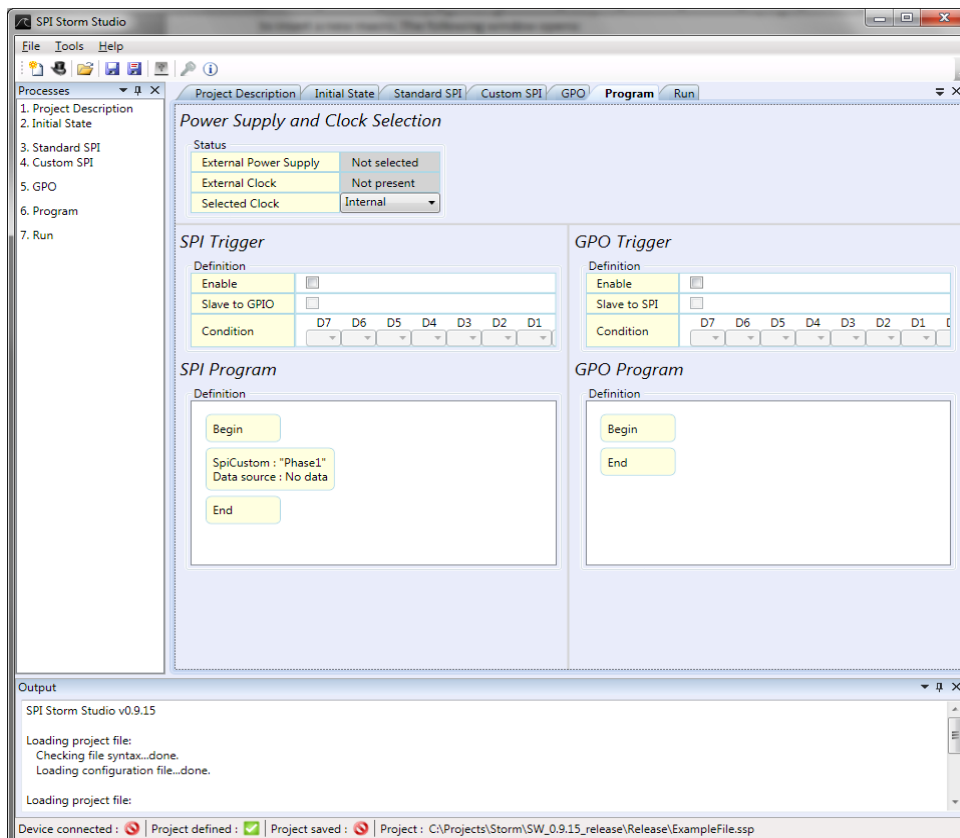
In the drop-down list under 'Macro Type', select the category of macro / command that you wish to insert.

Select 'Custom SPI'.

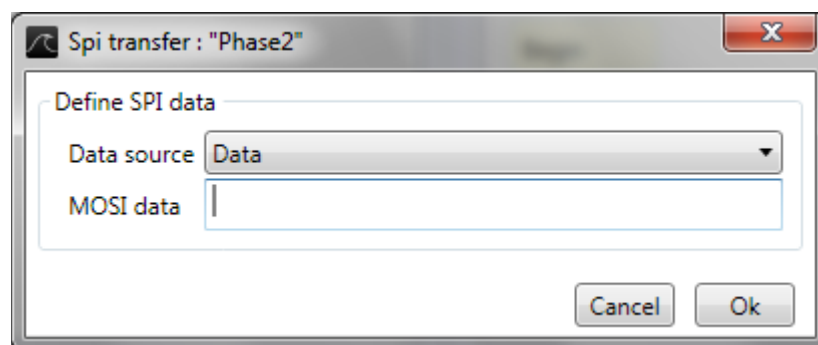
The list under 'Macro label' changes to the list of macros that you defined:



To insert one macro, select it and click on 'Ok'.



When a macro requires a variable data field, the following window pops up:

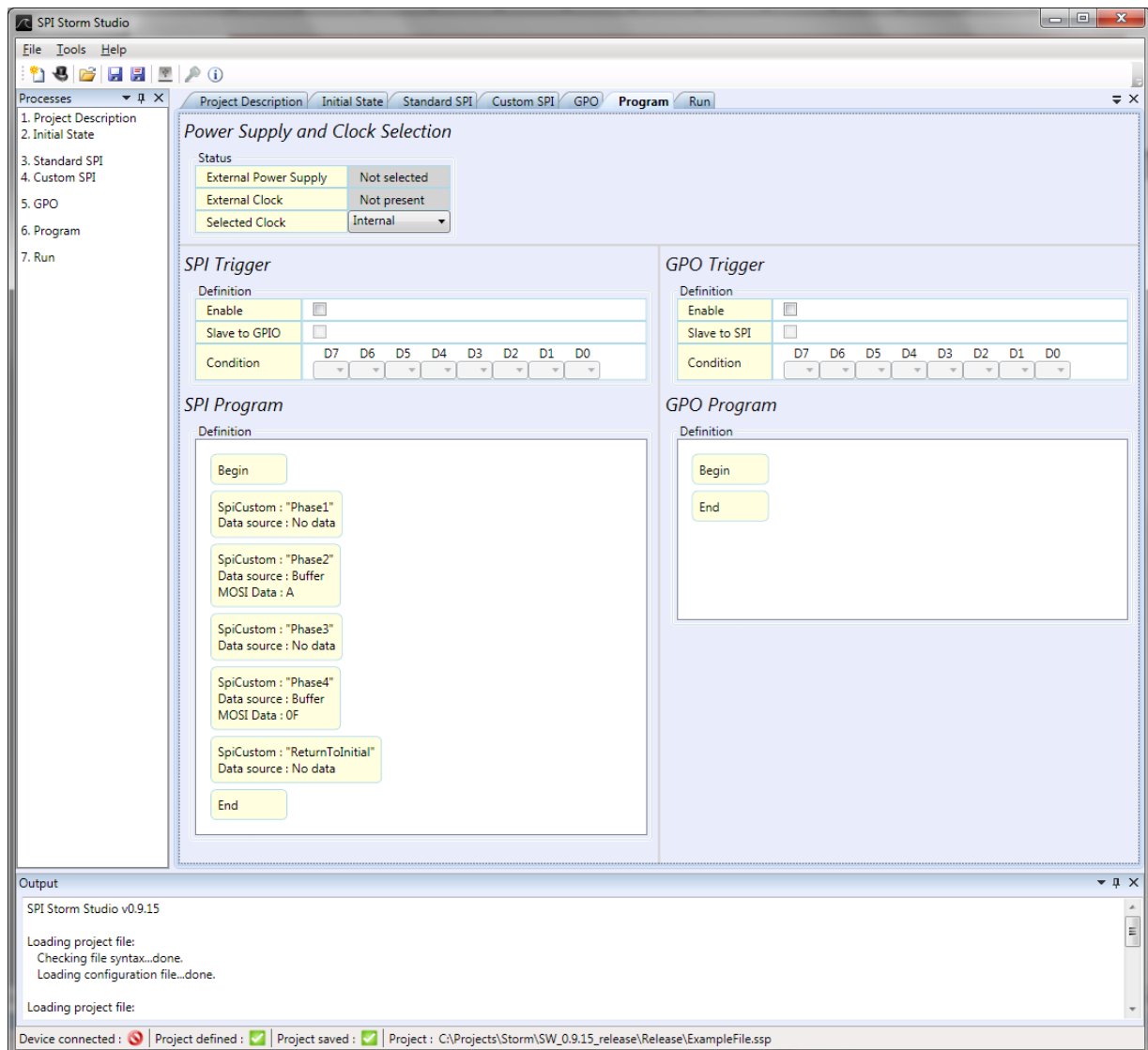


Data can be specified as constant data or with a file. Data is entered as hexadecimal without prefix:

Oxa will be entered as 'a'

Data is sent least significant bit first.

The resulting program window looks like:



STEP 4: Run a program

Switch to 'Run tab'. Click on 'RUN'.

You are able to define an output file. Result of the RUN appears in the 'Result' window.

