

SPI Storm Studio

User's Guide



Table of Contents

1About this guide	4
2Installing SPI Storm Studio	4
2.1System requirements	4
2.2Installation Wizard	4
2.3Installing the USB driver	6
2.4Installing the License File	7
3Your first SPI Storm Studio project – example	8
3.1Starting the application	8
3.2Connecting and configuring your SPI Storm device	9
3.2.1SPI Storm Device at a glance	9
3.2.2Establish a connection between SPI Storm Studio and SPI Storm USB device	
3.3Step-by-step: how to define a protocol with SPI Storm Studio?	
3.3.1The basics: defining a simple access with the standard SPI protocol:	
4Defining Standard SPI protocol	20
4.1Standard protocols in SPI Storm Studio	20
4.2Data formats	22
4.3Using standard protocols in SPI Storm Studio	23
4.3.10verview	23
4.3.2Defining a standard protocol 'device'	24
4.3.3Defining a standard protocol 'macro'	
5Defining custom serial protocol	27
5.1Defining a custom segment	
5.2Rules for custom segment definition	29
5.3Custom segment example – detailed	
5.4Defining a custom macro	
6GPO patterns sequence	
6.1How to define GPO patterns	
6.1.1Defining GPO segments	
6.1.2Defining GPO macros	
7Defining a program	
7.1Program tab overview	
7.2Power supply and clock selection	
7.3SPI trigger	
7.4GPO trigger	
7.5SPI Program	
7.5.10verview	
7.5.2Building up a program from SPI Storm Studio GUI	43
7.6GPO Program	
7.7File formats	
7.7.1Standard and custom macro file format	46
7.7.2GPO segment file format	
7.7.3Output file format	
8Running a program	
9SPI Storm API	
9.10verview	
9.2Detailed Functions Description	
9.3Files Needed to Use the API	
9.4Programming example	51



References

[]

History

Version	Date	Description
0.90	29 July 2011	Preliminary revision – to be completed.
1.00	August 2012	Completed missing sections
1.01	August, 22 nd , 2012	Added file formats description
1.02	Sept., 20 th , 2012	Added bit ordering for data representation option description + minor changes
1.03	Sept. 30 th , 2013	Added description of custom segments using SS2 and SS3 to apply constant values. Completed usage description of WE line.



About this guide 1

This user's guide describes SPI Storm Studio software, used to control Byte Paradigm's SPI Storm device.

Installing SPI Storm Studio 2

2.1 System requirements

- PC installed with Microsoft Windows XP, Windows VISTA or Windows 7 32-bit or 64-bit versions. _
- 20 MB of free space. _
- One free USB 2.0 port.
- Microsoft .NET Framework 4 Client Profile runtime installed.

2.2 Installation Wizard

- _ Download SPI Storm Studio from http://www.byteparadigm.com/download-16.html
- Double-click on archive to start the installation wizard. _
- At the wizard welcome screen, click on Next> _



The next screen lets you choose the 'Start Menu' folder where the SPI Storm Studio shortcuts will be installed. The folder 'Byte Paradigm' is chosen by default.

Select the destination folder and click on Next>



15	Setup - SPI Storm Studio
	Select Start Menu Folder Where should Setup place the program's shortcuts?
	Setup will create the program's shortcuts in the following Start Menu folder.
	To continue, click Next. If you would like to select a different folder, click Browse.
1	Byte Paradigm Browse
⊢	
	< Back Next > Cancel

- The next screen lets you choose the destination directory on your PC hard drive. Default is: 'c:\Program Files\Byte Paradigm'.

Select the destination directory and click on Next>

弱 Setup - SPI Storm Studio	
Select Destination Location Where should SPI Storm Studio be installed?	
Setup will install SPI Storm Studio into the following folder.	
To continue, click Next. If you would like to select a different folder, clic	ck Browse.
C:\Program Files\Byte Paradigm	Browse
At least 12,2 MB of free disk space is required.	
< <u>B</u> ack Next >	Cancel

- Finally, click in 'Install' at the 'Ready to install' screen.



📴 Setup - SPI Storm Studio	
Ready to Install Setup is now ready to begin installing SPI Storm Studio on your computer.	
Click Install to continue with the installation, or click Back if you want to review change any settings.	or
Destination location: C:\Program Files\Byte Paradigm Start Menu folder: Byte Paradigm	*
4	F
< Back	Cancel

- Once setup is complete, the final screen offers to launch SPI Storm Studio. Select the appropriate option and click on 'Finish' to finish the software installation.



2.3 Installing the USB driver

- Connect your SPI Storm device to one free USB port of your PC with the provided cable.
- When prompted, locate and install the USB driver:
 - 32-bit operating system, the driver is located in: <Installation root>\SPIStormStudio\Drivers\x86
 - 64-bit operating system, the driver is located in: <Installation root>\SPIStormStudio\Drivers\x64



			Ja 1	Byte Paradigm		
2.4	Installing the L	icense File		BPI Control I	Parrel	
				a BH Eaprorn I	Contraction (Contraction)	
-	Start SPI Storm Studi	o \rightarrow	_	👗 BPI Test Inte	finia	
-	In the SPI Storm Stu	dio main window, click on the 'install licens	se'	SPI Storm St	udio	
	button from the tool	bar or select Help > Install license		doc		
		·		CDI Storm	Studio 1 0 0	
	SPI Storm Studio			🔂 Uninstall	1360001.0.0	
	<u>File T</u> ools <u>H</u> elp					
	🗄 🔁 🖑 💕 🖬 🖪 🖉	P 1				
	Processes 💌 🕂 🗙	Priect Description Initial St		$\langle \rangle$		
	1. Project Description	Project Description		/		
	2. Initial State	rojeet Description		Start SPI St	orm Studio fron	n the
	3. Standard SPI	Use the space below to enter you		Winde	ows Start Menu	
	F. CDO	Description				
	5. GPO		rm Stu	udio		
	6. Program	Install License button on the toolbar				
	7. Run	00	IS H	lelp		
			2	🔎 Install	License	
		s t [St:	Des 🕻	🕽 Online	Help	ripti ;cri
		аго		i) About		, belo
		n	SPI			
			Ins	tall License	from the 'Help'	menu

The 'License Manager' window opens.

🖉 License Manager		
Installed licenses		
Serial Number	License Key	
		Install License Done

- Click on 'Install License' button. A browser window opens.
- If you have not received your license file, please go to <u>http://www.byteparadigm.com/download-16.html</u> and follow the instructions about how to receive your license file.
- Select the received license file and click on 'Open'.
 The License Manager now lists the installed devices and the corresponding license strings.

🖉 License Manager	
Installed licenses	
Serial Number	License Key
04011003006	5E6C7A728EC6D84A049257FB459F6B9D3B0FF573D4C696A2B31DDF3D2B7DABDFE13119E659470F893BA3E75D750FF4E1D550BD613E4F40B359BA7A0119647FAF
	Install License Done



Please note:

- Each installed device is designated with its serial number, in the 'Serial Number' column of the License Manager.
- You can find the device serial number written on a label at the back of your SPI Storm device.
- You can install more than one device.
- When upgrading your version of SPI Storm Studio software, you need not to install the license file again.

3 Your first SPI Storm Studio project – example

3.1 Starting the application

- Locate and click on the 'SPI Storm Studio' icon from your start menu program.



- At startup, SPI Storm Studio main window opens:

SPI Storm Studio		x
<u>F</u> ile <u>T</u> ools <u>H</u> elp		
🗄 🔁 🐸 📂 🖬 🖪 🗷	P 🛈	
Processes 💌 🕂 🗙	Project Description Initial State Standard SPI Custom SPI GPO Program Run	⇒×
1. Project Description 2. Initial State	Project Description	
3. Standard SPI 4. Custom SPI	Use the space below to enter your project description. This field is optional.	
5. GPO	Description	
6. Program		
7. Run		
Output		•
SPI Storm Studio v1.0.0		
Device connected : 🔇 Pro	iject defined : 🔇 Project saved : 🔇 Project : -	



3.2 Connecting and configuring your SPI Storm device



3.2.1 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 (SSO 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. Each I/O line has its specifics and limitations. Please refer to sections below.
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	ско скі	CKO mirrors an internally generated clock signal; CKI can be used as an input pin for providing an external reference clock signal.



3.2.2 Establish a connection between SPI Storm Studio and SPI Storm USB device

- Connect your SPI Storm device to a USB port of your PC with the provided USB cable.
- SPI Storm device is powered through the USB port; once connected, a blue and a red LED are on.
- In SPI Storm Studio main window, click on 'Select Device' button or select Tools > Select Device from the window menu.
- Alternatively, you can click on the 'Device connected area' in the status bar at the bottom of the main window:

SPI Storm Studio		SPI Storm Studio	
File Iools Help Image: Processes Image: Processes Image: Processes Project Description Image: Project Description Image: Image: Image: Project Description Image: Project Description Image: Image: Image: Image: Project Description Image: Project Description Image: Image: Image: Image: Image: Image: Image: Project Description Image: Im	Project Descript Project Descri Use the spase bela Description	File Tools Help Select Device Processes T X 1. Project Description 2. Initial State	Project De Project De
Device connected : 🚫 Pro	oject defined : 🔇 Proje	ct saved : 🚫 Project : -	

- The 'Device Selection window' opens. The SPI Storm devices physically connected to the PC are listed. They are designated with their serial number.

Connected	Serial Number	Link	Device	
	0401092200D	USB	SPI Storm	
Course by Mark				
Supply volt	age			

- To establish a connection between one device and this session of SPI Storm Studio, select the desired device in the list and click on '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:



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

In this section, we'll show how to define a simple protocol with SPI Storm Studio. We'll start from a standard SPI protocol, and then show how to customize it to form a more specific serial protocol for communication.

3.3.1 The basics: defining a simple access with the standard SPI protocol:

Let's get started. Here is the SPI protocol that should be used with one SPI slave that we'll call 'DeviceO'. This protocol is a standard SPI protocol having the following characteristics:

- Initially, all signals lines are at low level, except SS lines, at high level.
- Access length : 8 bits (1 byte).
- Slave select signal's polarity : active low.
- SCLK IDLE level is 'low'.
- Data on MOSI / MISO are generated on SCLK rising edge.
- SCLK frequency is 25 MHz.







3.3.1.1 Setting up the initial state

- Select the 'Initial State' tab from the main window:

SPI Storm Studio		x
<u>File T</u> ools <u>H</u> elp		
i 🔁 👶 💕 🛃 🛃 🖉		
Processes 💌 🖡 🗙	Project Description Initial State Standard SPI Custom SPI GPO Program Run	₹×
1. Project Description 2. Initial State	Initial State Definition	
3. Standard SPI 4. Custom SPI	Initials	
5. GPO	SCLK Idle-0	
6. Program	MOSI (DQ0) 0	
7. Run	MISO (DQ1) High-Z	
	WE 0	
	SSO 1	
	SS1 1 •	
	SS2 (DQ2) 1	
	SS3 (DQ3)	
	Open Drain Configure	
	Q7Q0 0000000	
Output		чх
SPI Storm Studio v1.0.0		
Device connected : 🔇 Pro	oject defined : 🥎 Project saved : 🥎 Project : -	

A drop-down box allows setting the initial value of each signal or group of signals. For this example, click on the drop-down boxes and select the following values:

Signal / Group of signals / Control	Possible values	Selected value	Comments
SCLK	Idle-0, Idle-1, Running	Idle-0	The clock is non-running and at Idle-0 level at initial state
MOSI (DQ0)	High-Z, 0, 1	0	MOSI is the data line from the master to the slave(s). It is set at '0' initially.
MISO (DQ1)	High-Z, 0, 1	High-Z	MISO line is the data line from the slave to the master. It should not be driven in the initial state.
WE	0,1	Don't care	WE line is not used in this example. Please refer to section 5.1 below for more information about this signal.
SSO	0,1	1	We'll use SSO as slave select line. As active-low signal, it is set to logic '1' during the initial state.
SS1	0,1	Don't care	



Signal / Group of signals / Control	Possible values	Selected value	Comments
SS2 (DQ2)	High-Z, 0, 1	Don't care	The other SS lines are not used in this example. Please refer to ### for
SS3 (DQ3)	High-Z, 0, 1	Don't care	more information about SS lines
Open Drain	Clicking on this button	opens the 'Op	pen Drain' controls for the I/Os during the initial phase.
	Open Drain	A tick bo	is available for each signal.
	Select the open drain configurat	tion: Che	cking the box sets the corresponding I/O in 'open-drain' mode.
			S voltage levels).
	MOSI (DO0)		
	MISO (DQ1)		
	WE		
	SSO 🔲		
	SS1		
	SS2 (DQ2)		
	SS3 (DQ3)		
	Cancel O	k	
Q7 Q0	0 or 1 for each bit of the vector	Don't care	This controls the initial values of the GPO port. Please refer to ### for more details about the GPO port.

_



3.3.1.2 Setting up the standard SPI access

- Select the 'Standard SPI' tab from the main window.
 - There 2 main areas in this tab: 'Devices Definition' and 'Macros Definition'.
 - The 'Devices Definition' area defines the characteristics of the SPI protocol used for each device;
 - The 'Macros Definition' area associates each device to one of the four physical slave select lines of the device.

SPI SPI Storm Studio					٢.
<u>F</u> ile <u>T</u> ools <u>H</u> elp					
🗄 😫 💕 😼 🖪	1 🔎 🗯 🚺				
Processes 🔻 🖡 🗙	Project Descri	ption Initial	State Standard SPI Custom SPI GPO Program Ru	un 🗸	×
1. Project Description 2. Initial State	SPI Device	Definition			
3. Standard SPI					
4. Custom SPI	Devices		Definition		
5. GPO					
6. Program			Clock Erequency		
7. Run	1 F 1				
			Bit Order	Standard protocol	
			Byte Order		
	18		Clock Artive State	Devices Definition area	
			Clock Idle Level		
			Clock Driving Edge		
			Clock Sampling Edge		
			SS Idle Level		
	Add	Delete	Open Drain Configure		
	Standard Sl	PI Definitio	n		
	Massas		Definition		
	Wacros		Label		
			Device		
	1		Slave Select 0		
			Slave Select 1	Standard protocol	
			Slave Select 2	'Macros Definition' area	
			Slave Select 3		
	Add	Delete			
	1				
Output				* #	×
SPI Storm Studio v1.1.14					
Loading project file:					
Checking file syntaxdo	one.				
Loading configuration f	iledone.				
Device connected : 🔇 Pr	roject defined : 🔇	Project saved :	🔇 Project : -		

SPI Storm Studio User's Guide



To define a device:

- 1) In the SPI Device Definition, click on 'Add' button.
- 2) In the window that opens, specify a name for the device let's use 'Device0'. Click on OK.

Add SPI Device	x
Please enter a label for the new device:	
Device0	
Cancel	Ok

3) Now your device is created and listed. Select Device0 in the list. You are now able to define the parameters of the access related to 'Device0'. To do so, modify the parameters contained in the text boxes and drop-down lists located to the right of the Devices list.

F	Project Descripti	on Initial Sta	ate Standard SPI	Custom SPI G	PO Program Run
SP	l Device De	efinition			
	Devices		Definition		
	Device0		Label	Device0	
			Clock Division	4	
			Clock Frequency	25 MHz	
			SPI Type	SPI-4	•
			Bit Order	LSBit First	•
			Byte Order	LSByte First	•
			Clock Active State	Running	•
			Clock Idle Level	Idle-0	-
			Clock Driving Edge	Rising	-
			Clock Sampling Edge	Falling	-
			SS Idle Level	0	•
			Open Drain	Configur	e
	Add	Delete	Transfer Length	13	bits

	Possible values	Selected value	Comments
Label	Any – this is a text label used to designate the access being defined	Device0	The text box can be used to change the name of the Device.
Clock Division	Any positive integer value from 1 to 1024	4	This defines the SCLK frequency as a dividing factor from a reference 100 MHz clock, according to the formula: FSCLK = 100 / (Clock Division). Defining a 'clock division' of 4 will result in a frequency equal to 100 / 4 = 25 MHz for SCLK
SPI Туре	SPI-4, SPI-3, SPI-Dual, SPI-Quad	SPI-4	This drop down box lets you specify the type of standard protocol you would like to use. In this case, we are using a standard 'SPI' protocol – SPI-4
Bit Order	LSBit First, MSBit First	LSBit First	Defines the bit ordering within each byte of data.



	Possible values	Selected value	Comments
Byte Order	LSByte First, MSByte First	LSByte First	Defines the byte ordering
Clock Active State	Running, Idle-0, Idle-1	Running	This defines the behavior of SCLK while data is sent or received. In our example, SCLK toggles while data is sent on MOSI and data is received on MISO.
Clock Idle level	Running, Idle-0, Idle-1	Idle-0	Defines the level of the SCLK signal when the clock is not toggling.
Clock Driving edge	Rising, Falling	Rising	Defines on which edge of SCLK the data is generated.
Clock Sampling Edge	Rising, Falling	Falling	Defines on which edge of SCLK data is sampled.
SS Idle level	0,1	1	Defines the level of SS when it is not active.
Open Drain	Clicking on this button opens the from the 'Standard SPI tab'.	'Open drain' controls A tick box is available fo ☑ Checking the box s □ Leaving the box mode' (firm LVCMOS vo	of the I/Os when executing accesses defined or each signal. ets the corresponding I/O in 'open-drain' mode. unchecked sets the corresponding I/O in 'normal oltage levels).
Transfer length	Any positive integer value > 1	8	Length of the transfer counted in bits.

To define a macro in the 'Standard SPI' tab:

- 1. Click on 'Add' in the 'macros definition area'.
- 2. Specify a name for the macro being defined: example: 'AccessDevice0'
- 3. Select which 'device' you wish to associate with this macro click on the drop-down list next to 'Device'. This list contains all the defined devices.
- 4. Select 'Device0'.
- 5. Associate a slave select line to this access by selecting the corresponding tick box.

Standard SPI Definition

	Definition		
AccessDevice0	Label	Access_Device0	3 4.
	Device	Device0 5.	-
	Slave Select 0		
	Slave Select 1		
	Slave Select 2		
	Slave Select 3		
Add Dalata			
Add Delete	J		



3.3.1.3 Setting up a simple program that uses the configured macros

Switch to 'Program Tab'.

SPI Storm Studio	
<u>F</u> ile <u>T</u> ools <u>H</u> elp	
🗄 🔁 💕 🖬 🗐 🖉	
Processes 💌 🖡 🗙	Project Description / Initial State / Standard SPI / Custom SPI / GPO / Program / Run / T
Processes ◆ ₽ × 1. Project Description 2. Initial State 3. Standard SPI 4. Custom SPI 5. GPO 6. Program 7. Run	Project Description Initial State Standard SPI Cuttom SPI Program Run × ×
Output	- 1 X
SPI Storm Studio v1.1.5 Loading project file: Checking file syntaxdor Loading configuration fil	ie. edone. siert defined : 🔽 Project caved : 🔊 Project : CAllsers\frederic\Documents\EvampleFile.ssp

We'll now edit the 'SPI Program' to run a simple access with the SPI port.

> Right-click on the 'End clause' in the 'SPI Program' area

Select 'Insert Before' from the contextual menu that pops up. A window opens.

In the 'Add macro' window, use the drop-down lists to select: 'Standard SPI' in the 'Macro type' and 'AccessDevice0' in the Macro Label.

Then click on **'OK'**.



Be	gin		
En	1	Insert Before	
	4	Insert After	
	/	Edit	
	×	Delete	
		Move Up	
	▼	Move Down	

Macro Type	SPI -	Macro Label		•
			Cancel	Ok
Spi transfer :	: "AccessDe	vice0"		
Spi transfer : Define SPI dat	. "AccessDe	vice0"		
Spi transfer : Define SPI dat Data source	: "AccessDe ta Data	vice0"		

The window that opens allows defining the data sent on MOSI for this access. Type a value in hexadecimal (without '0x' prefix) here: A3. Click on 'OK'.

The Program tab now contains a program defined for the SPI port.

C SPI Storm Studio				
<u>File T</u> ools <u>H</u> elp				
: 🎦 🍕 💕 🛃 🗷	۵			
Processes 💌 म 🗙	Project Description Initial State Standard SPI GPO Program Run = X			
1. Project Description 2. Initial State	Power Supply and Clock Selection			
3. Standard SPI 4. Custom SPI 5. GPO 6. Program	Status Not selected External Clock Not present Selected Clock Internal *			
7. Run	SPI Trigger Definition Enable Condition D7 D6 D5 D4 D3 D2 D1 D0 Source : Buffer Definition D7 D6 D5 D4 D3 D2 D1 D0 SpiStandard : "AccessDevice0" Data source : Buffer Definition Begin End Definition End End Definition End Definition Definition			
Output	▲ 廿 ×			
SPI Storm Studio v1.1.5 Loading project file: Checking file syntaxdon Loading configuration file	ie. edone.			
Device connected : 📎 Project defined : 🔄 Project 3 ved : 💟 Project : C.\Users\treateric\Documents\DocExamplehile.ssp				



3.3.1.4 Running the program

Switch to the 'Run' tab.

Connect your device (see section 3,2,2) and click on 'RUN'.

SPI Storm Studio	Concession in the local division of the loca					-		X
<u>File T</u> ools <u>H</u> elp								
: 🔁 🤻 💕 📓 🖉 🛛	P 😫 🛈							-
Processes 🔻 🖡 🗙	Project Description Initial Sta	ate Standard SP	Custom SPI GPO Progr	ram Run				⇒ ×
1. Project Description 2. Initial State	Output File							
3. Standard SPI 4. Custom SPI	C:\Users\frederic\Documents\	DocExampleFile.ou	t					
5. GPO								
6. Program	Progress							
7. Run	Run Initialising	Preloading	SPI Triggering	SPI R	unning	Receiving Data	Done	
			GPO Triggering			GPO Running		
	Result							
	Nr Spi Type	Label		Mosi			Miso	
	1 SPI-4 Acce	essDevice0	A3			00		_
	٠) F
Output								▼ ‡ ×
SPI Storm Studio v1.1.5								*
Loading project file:								
Checking file syntaxdone.								
Loading configuration file	aone.							
								*
Device connected : 🔽 Project defined : 💟 Project saved : 💟 Project : C:\Users\frederic\Documents\DocExampleFile.ssp								



4 Defining Standard SPI protocol

4.1 Standard protocols in SPI Storm Studio

The 'standard protocols' in SPI Storm Studio are summarized in the table below.

Description			
This is the standard SPI protocol on 4 wires. It uses MOSI, MISO, at least 1 SS line and a SCLK signal.			
<u>MOSI (07 \ 06 \ 05 \ 04 \ 03 \ 02 \ 01 \ 00 \</u>			
$\underline{MISO} \qquad \qquad \boxed{17 \ 16 \ 15 \ 14 \ 13 \ 12 \ 11 \ 10 \ }$			
55			
SPI is a master-slave protocol simultaneously sends data onto the MOSI data line and samples data from the MISO line. 4 modes exist, according to the clock IDLE state and the clock phase relative to the data. SPI defines one SS (slave select) line per slave. The master also sends its own clock signal SCLK. Usually, SS is active-low but another convention can be used too. Usually, SCLK is held IDLE between transfers but a continuously toggling SCLK signal can be generated by SPI Storm too.			
This is a variant of the SPI protocol that uses 3 wires only. It uses MOSI, at least 1 SS line and a SCLK signal.			
<u>saukan nan nan nan nan nan nan nan nan nan </u>			
<u>MOSI / MISO 07 X 06 X 05 X 04 X 03 X 02 X 01 X 00 X 13 X 12 X 11 X 10 </u>			
55 MOSI Phase Bus direction Swap MISO Phase SPI-3 uses one single data line for writing (data out) or reading data (data in). This allows saving on slave I/Os count. SPI-3 protocol is composed of 3 phases: - 'MOSI phase' (Master-Out-Slave-In), during which the DQ0 data line of SPI Storm is used as an output of the device. 'Bus direction Swap' phase – an arbitrary number of clock cycles during which the data line is held at HI-Z (high impedance state). This phase is necessary for switching bus direction and avoid shortcuts. - 'MISO phase' (Master-In-Slave-Out), during which the DQ0 data line of SPI Storm is used as an input. During this phase, the slave answers to the master and the data that it sends is sampled by SPI Storm.			



Protocol	Description			
Dual-SPI	This is the dual-SPI protocol. It is a mixed protocol optionally initiated with a standard SPI (SPI-4) access followed by read and/or write commands that use 2 signals in parallel for the data.			
	55	۱	_	
	SCLK	mmm	mmmm	uuuuu
	M051 (DQO)	<u>(</u>	<mark>(06)(04)(02)(00)</mark>	
	MI50 (DQ1)		<u>(07) (05) (03) (01</u>	
		Command Write (SP14)	Dummy Dual-SPI Dumm Cycles 1 Write Cycles	y Dual-SPI 2 Read
	The Dual-SPI protoc the phases describe 0.	ol command defined in SPI Stor d in the picture above must not	m Studio is generic. According to be sent. To skip one of the phas	the context and the slave, some of es above, its length has to be set to
Quad-SPI	This is the quad-SPI read and/or write c	protocol. It is a mixed protoco ommands that use 4 signals in p	l optionally initiated with a star arallel for the data.	dard SPI (SPI-3) access followed by
	SCLK	mmm	mmm	unnnn
	MOSI (DQO)	<u>(</u>	<mark>(04)(00)(04)(00)</mark>	
	MISO (DQI)		<u>(05)(01)(05)(01)</u>	
	552 (DQ2)		(<u>06)(02)(06)(02</u>)	
	553 (DQ3)		(<u>07)(03)(07)(03</u>)	(17)(13)(17)(13)(17)(13)
		Command Write (SP14)	Dummy Quad-SPI Dumm Cycles 1 Write Cycles	y Quad-SPI 2 Read



4.2 Data formats

Hexadecimal string without prefix representing the bits sent on / sampled from the data lines

Default format: most significant bit first / MS byte first.

Complete bytes must always be entered. (from SPI Storm Studio version 1.1.14)

Example: if 12 bits 0x123 must be used, it must be padded by one character.

Examples: <u>Convention : MS Byte first / MS bit first</u> Length: 12 bit – data : 5A30

Chronology	1	2
Byte level	0x5 A	0x 3 0
Bit level	0-1-0-1-1-0-1-0	0-0-1-1

Length: 17 bit – data : 305AF0

Chronology	1	2	3
Byte level	0x30	0x5A	0xF0
Bit level	0-0-1-1-0-0-0-0	0-1-0-1-1-0-1-0	1 (1-1-1-0-0-0-0 is not sent)

Examples: Convention: LS Byte first / Is bit first Length: 12 bit – data : 05A3

Chronology	1	2
Byte level	0x A3	0x05
Bit level	1-1-0-0-0-1-0-1	1-0-1-0

Length: 17 bit – data : 0305A3

Chronology	1	2	3
Byte level	0xA3	0x05	0x03
Bit level	1-1-0-0-0-1-0-1	1-0-1-0-0-0-0	1

Other more advanced combinations can be used, however less intuitive.

Protocols using more than one data line (example: dual-SPI, quad-SPI) use an interlaced convention. Please refer to figures at section 4.1.

4.3 Using standard protocols in SPI Storm Studio

4.3.1 Overview

The 'Standard SPI tab' is used for setting up SPI Storm device to run with standard protocols.

SPI SPI Storm Studio				
<u>F</u> ile <u>T</u> ools <u>H</u> elp				
🗄 🔁 😂 🚰 🛃 🗷	P 🛱 🛈			
Processes 💌 म 🗙	Project Description Initial	State Standard SPI	Custom SPI GPO Program	Run = X
1. Project Description 2. Initial State	SPI Device Definition	Device defi	nition area	<u> </u>
3. Standard SPI	Devices	Definition		
4. Custom SPI	Device0	Label	Device0	
5. GPO		Clock Division	4	
6. Program		Clock Frequency	25 MHz	
7. Run		SPI Type	SPI-4 🔻	
		Bit Order	LSBit First 🔹	
		Byte Order	LSByte First 🔹	
		Clock Active State	Running	
		Clock Idle Level	Idle-0 🔻	
		Clock Driving Edge	Rising •	
		Clock Sampling Edge	Falling 🔻	
		SS Idle Level	0 -	
		Open Drain	Configure	E
	Add Delete	Transfer Length	13 bits	
	Standard SPI Definitio	on		
	Macros	Definition		
	DataWr	Label DataWr		
		Device Device0	•	
		Slave Select 0		
		Slave Select 1		
		Slave Select 2		
		Slave Select 3		
		Macros def	inition area	
	Add Delete			•
Output				- ↓ ×
Loading configuration file	edone.			4
Loading project files				
Loading project file: Checking file syntaxdone.				
Loading configuration filedone.				
	· · · · · · · · · · · · · · · · · · ·			
Device connected : 🔇 Pro	oject defined : 🔽 Project saved :	View Project : C:\Program	n Files (x86)\Byte Paradigm\SPISto	ormStudio\Examples\Exa

To set up a standard protocol:

- One or multiple 'devices' must be defined. A 'device' specifies the characteristics of the protocol.
- One or multiple 'macros' must be defined. A 'standard protocol macro' associates a device to a specific slave select I/O.

4.3.2 Defining a standard protocol 'device'

To create a device, click on 'Add' in the 'Device definition area'. A window pops up, prompting for the device name.

SPI Add SPI Device
Please enter a label for the new device:
Device0
Cancel Ok

The table below summarizes the parameters to be specified for each SPI Storm Studio standard protocol.

Parameter	Parameter Applies to Valid values		Description	
Label	ALL	String composed of letters and numerical characters	Device name	
Clock Division	ALL	Integer value from 1 to 1024	Clock dividing factor. Defines the clock rate used by the device. The resulting clock rate is the RefClock frequency / Clock Division. If the internal clock is used, the RefClock frequency is 100 MHz.	
Clock Frequency	ALL	Non editable	Displays the clock rate resulting from the specified division factor.	
SPI Туре	ALL	SPI-4, SPI-3, SPI-Dual, SPI-Quad	Allows choosing a standard protocol for the device	
Bit Order	ALL	LSBit First, MSBit First	Defines the bit ordering within each byte of data.	
Byte Order	ALL	LSByte First, MSByte First	Defines the byte ordering	
Clock Active State	ALL	Running, Idle-0, Idle-1	Defines the state of the SCLK signal while sending / receiving data with the given protocol. Running = the clock toggles. Idle-0 = the clock is held at constant low logic level; Idle-1 = the clock is held at constant high logic level.	
Clock Idle Level	ALL	Running, Idle-0, Idle-1	Defines the state of the SCLK signal while not sending / receiving data with the given protocol. Running = the clock toggles. Idle-0 = the clock is held at constant low logic level; Idle-1 = the clock is held at constant high logic level.	
Clock Driving Edge	ALL	Rising, Falling	Defines the phase of the SCLK clock signal relative to the generated data. Rising = data is sent out with the SCLK signal rising edge Falling = data is sent out with the SCLK signal falling edge	
Clock Sampling Edge	ALL	Rising; Falling	Defines the edge of the clock used for sampling	

Parameter	Applies to	Valid values	Description
			data in. Rising = data is sampled with the clock rising edge Falling = data I sampled with the clock falling edge
SS Idle level	ALL	0, 1	Defines the IDLE level of the SS line used for this device.
			0 = SS is at low logic level when IDLE 1 = SS is at high logic level when IDLE
Open Drain	ALL	Click on the 'Configure button' to set I/Os open-drain controls	SelectOpenDrainSpi Select the open drain configuration: ScLK MOSI (DQ0) MISO (DQ1) WE SS0 SS1 SS2 (DQ2) SS3 (DQ3) Cancel Ok
Transfer Length	SPI-4	Integer value from 0 to 1.000.000	Defines the length in bits (clock cycles) of the SPI transfer.
Write Length	SPI-3	Integer value from 0 to 1.000.000	Defines the length in bits (clock cycles) of the write phase (MOSI phase) of the SPI-3 transfer.
Read Length	SPI-3	Integer value from 0 to 1.000.000	Defines the length in bits (clock cycles) of the read phase (MISO phase) of the SPI-3 transfer.
Swap latency	SPI-3	Integer value from 0 to 1.000.000	Defines the length in bits (clock cycles) of the swap latency phase of the SPI-3 transfer.
Swap Clock	SPI-3	Disabled, Enabled	Specifies whether the clock signal on SCLK should be active or IDLE during the swap latency phase of the SPI-3 transfer
Swap Level	SPI-3	High-Z, 0, 1	Defines the level of the MOSI (DQ0) data line during the swap latency phase of the SPI-3 transfer.
Command Write Length	SPI-Dual, SPI-Quad	Integer value from 0 to 1.000.000	Defines the length in bits of the 'Command write phase' of Dual- and Quad-SPI protocols transfers.
Dual Dummy 1 Length	SPI-Dual	Integer value from 0 to 1.000.000	Defines the length clock cycles of the 'Dual Dummy 1 phase' of Dual-SPI protocol transfers.
Dual Write Length	SPI-Dual	Integer value from 0 to 1.000.000 and multiple of 2	Defines the length in bits of the 'Dual Write' of Dual-SPI protocol transfers. This is the total number of bit interlaced onto the 2 data lines. This number must be a multiple of 2.
Dual Dummy 2 Length	SPI-Dual	Integer value from 0 to 1.000.000	Defines the length in clock cycles of the 'Dual

Parameter	Applies to	Valid values	Description	
			Dummy 2 phase' of Dual-SPI protocol transfers.	
Dual Read Length	SPI-Dual	Integer value from 0 to 1.000.000 and multiple of 2	Defines the length in bits of the 'Dual Read' of Dual-SPI protocol transfers. This is the total number of bit interlaced onto the 2 data lines. This number must be a multiple of 2.	
Quad Dummy 1 Length	SPI-Quad	Integer value from 0 to 1.000.000	Defines the length clock cycles of the 'Quad Dummy 1 phase' of Quad-SPI protocol transfers.	
Quad Write Length	SPI-Dual	Integer value from 0 to 1.000.000 and multiple of 4	Defines the length in bits of the 'Quad Write' of Quad-SPI protocol transfers. This is the total number of bit interlaced onto the 4data lines. This number must be a multiple of 4.	
Quad Dummy 2 Length	SPI-Dual	Integer value from 0 to 1.000.000	Defines the length clock cycles of the 'Quad Dummy 2 phase' of Quad-SPI protocol transfers.	
Quad Read Length	SPI-Dual	Integer value from 0 to 1.000.000 and multiple of 4	Defines the length in bits of the 'Quad Read' of Quad-SPI protocol transfers. This is the total number of bit interlaced onto the 4data lines. This number must be a multiple of 4.	

4.3.3 Defining a standard protocol 'macro'

'Macros' are what is executed by SPI Storm Studio's program.

To create a new macro, click on 'Add button' and enter its name in the prompt that pops up.

SPI Add Standard SPI Macro	Standard SPI Definition
Please enter a label for the new macro: AccessSS0 Cancel Ok	Macros Definition DataWr Label AccessSS0 QuadSPI Access12 AccessSS0 Slave Select 0 Slave Select 1 Slave Select 2 Slave Select 3 Slave Select 3

- Then, select a device from the 'Device drop-down list'. This associates a device to the macro being defined.
- Finally, select one or multiple slave select signals that have to be activated when executing this macro. Given slave select lines are multi-purpose I/O on SPI Storm, not all choices are available for all protocols. For instance, SS2 cannot be selected if the associated device uses SPI-Quad protocol: this line is automatically used as data line. If you need more control lines with such protocols, please use GPO lines. An example or error message is shown below.

5 Defining custom serial protocol

'Custom protocols' are defined from the 'Custom SPI tab'.

They are based on the definition of 'segments' and 'macros'. **'Segments'** are the **building blocks of macros**. One or multiple segments are assembled to form a macro.

SPI SPI Storm Studio	A R OTHER DRIVE	1		byte	
<u>File Tools H</u> elp					
: 🞦 🤻 💕 🔙 🗐 🖻	P 🗘 🛈				3
Processes 💌 म 🗙	Project Description Initial S	tate	PO Program Run		₹×
1. Project Description 2. Initial State	SPI Segment Definition	1			
3. Standard SPI 4. Custom SPI	Segments Segment0clkIdle	Definition Label Segment0clkIdle			
5. GPO	Segment1clkIdle	length 1	; Seg	gment definition area	
6. Program	SegmentData4 SegmentData5	SCLK Idle-0 •		_	
7. Run	Segment1clkRunning	MOSI (DQ0) 0			
		MISO (DQ1) High-Z			
		WE 0 V			
		SS0 0 •			
		SS2 (DO2) High-7 •		Macros definition a	rea
	Add Delete	SS3 (DQ3) High-Z			
	Custom SPI Definition				
	Macros	Definition		Segments	
	Phase1	Label Phase1		Segment0clkIdle	
	Phase2 Phase3	Clocks & Phases Configure		Segment1clkIdle SegmentData4	
	Phase4	Open Drain Configure		SegmentData5	
	Return I oInitial	Bit Order LSBit First	•	SegmentUclkRunning Segment1clkRunning	
		Byte Order LSByte First	•		
		Kemove Me	ove Up		
		Segments SegmentIcikIdle Segment0cikIdle			
	Add Delete			Add To Macro	
Output					÷ † ×
Loading configuration fi	ledone.				^
Loading project file: Checking file syntaxdo	ne.				
Loading configuration fi	ledone.				=
					*
Device connected : 🔇 Pro	oject defined : 🌄 Project saved : 🤅	🔇 Project : C:\Program Files (x86)\Byte	Paradigm\SPIStormStudio\	\Examples\ExampleFile1.ssp	

5.1 Defining a custom segment

To define a new segment, click on 'Add' button in the 'SPI Segment Definition' area. A window pops up, prompting for the segment name.

SPI Segment Definition	Definition	
Segment0clkIdle	Label	Segment1
Segment1clkIdle SegmentData4	Length	1
SegmentData5 SegmentOclkRupping	SCLK	Idle-0 🔹
Segment1clkRunning	MOSI (DQ0)	Unused 🔹
Segment1	MISO (DQ1)	Unused 🔻
	WE	0 🗸
	SS0	0 🔹
	SS1	0 🔹
	SS2 (DQ2)	Unused 🔻
Add Delete	SS3 (DQ3)	Unused 🔻

Add SPI Segment	×
Please enter a label for the ne	w segment:
Segment1	
	Cancel Ok

When the segment's name is highlighted in the 'SPI Segment Definition' list, its properties can be read and edited in the 'Definition area'. The following properties can be defined:

	Possible values	Description
Label	Any – this is a text label used to designate the segment's name	The text box can be used to change the name of the segment.
Length	1 to 500	This is the length of the segment in clock cycles. The clock period / frequency is defined in the 'macros area'. So, the same segment can be used at different clock rates.
SCLK	Running, Idle-0, Idle-1	Defines the behavior of the SCLK signal while the segment is executed. 'Running' means that SCLK will be toggling during segment execution; Idle-0 and Idle-1 will hold SCLK at low and high logic levels respectively.
MOSI (DQ0)	Unused, High-Z, 0, 1, Data, Sample	Defines the behavior of the corresponding signal,
ISO (DQ1)	Unused, High-Z, 0, 1, Data, Sample	according to the list of possible values: Unused = unused
SS0	0, 1	High-Z = held high-impedance during segment
SS1	0, 1	execution. 0 = held at low logic level during segment execution
SS2 (DQ2)	Unused, High-Z, 0, 1, Data, Sample	1 = held at high logic level during segment execution.
SS3 (DQ3)	Unused, High-Z, 0, 1, Data, Sample	Data = used generate data out. Sample = used to sample data in.
		IMPORTANT: Not all arbitrary combinations are possible. See section '4.2 Rules for segment definition'
WE	0, 1	0 = active-low; 1 = active-high. WE will be made active at the above level if a data 'write' operation is executed on bidir data bus. Otherwise, remains at the defined default value.

5.2 Rules for custom segment definition

Not all combinations of values on the 'data lines' of the device are possible. The 'data lines of the device' are MOSI (DQ0), MISO (DQ1), SS2(DQ2) and SS3 (DQ3).

When an illegal combination of the data lines is attempted, the following window pops up, enabling the selection of a valid combination:

SPI DQx Cont	iguration				
Description The selected DQx lines combination is not supported. Please use this wizard to configure the DQx lines according to your needs.					
Configurat	ion				
Send Dat	a Send Constant Value Sample Data				
Possible (Configurations 💌				
Result					
MOSI	Send Constant Value (possible values: High-Z, 0 and 1)				
MISO	Unused				
SS2	Unused				
SS3	Unused				
	Done				

The table below shows the list of possible combinations.

Combination name	Description		
Unused	MOSI	Unused	
	MISO	Unused	
	SS2	Unused	
	SS3	Unused	
SPI-Wr			
	MOSI	Data	
	MISO	Unused	
	SS2	Unused	
	SS3	Unused	
SPI-Wr-SS23			
	MOSI	Data	
	MISO	Unused	
	SS2	Send Constant Value (possible values: High-Z, 0 and 1)	
	SS3	Send Constant Value (possible values: High-Z, 0 and 1)	
SPI-WrRd	MOSI	Data	
	MISO	Sample	
	SS2	Unused	
	SS3	Unused	
		-	

Combination name	Description		
SPI-WrRd-SS23	MOSI	Data	
	MISO	Sample	
	SS2	Send Constant Value (possible values: High-Z, 0 and 1)	
	SS3	Send Constant Value (possible values: High-Z, 0 and 1)	
		·	
SPI-Rd3	MOSI	Sample	
	MISO	Unused	
	SS2	Unused	
	SS3	Unused	
SPI-Rd3-SS23	MOSI	Sample	
	MISO	Unused	
	SS2	Send Constant Value (possible values: High-Z, 0 and 1)	
	SS3	Send Constant Value (possible values: High-Z, 0 and 1)	
		·	
SPI-Rd4	MOSI	Unused	
	MISO	Sample	
	SS2	Unused	
	SS3	Unused	
		*	
SPI-Rd4-SS23	MOSI	Unused	
	MISO	Sample	
	SS2	Send Constant Value (possible values: High-Z, 0 and 1)	
	SS3	Send Constant Value (possible values: High-Z, 0 and 1)	
SPI-WrD	MOSI	Data	
	MISO	Data	
	SS2	Unused	
	SS3	Unused	
SPI-WrD-SS23	MOSI	Data	
	MISO	Data	
	SS2	Send Constant Value (possible values: High-Z, 0 and 1)	
	SS3	Send Constant Value (possible values: High-Z, 0 and 1)	
SPI-RdD	MOSI	Sample	
	MISO	Sample	
	SS2	Unused	
	SS3	Unused	

Combination name	Description		
SPI-RdD-SS23	MOSI	Sample	
	MISO	Sample	
	SS2	Send Constant Value (possible values: High-Z, 0 and 1)	
	SS3	Send Constant Value (possible values: High-Z, 0 and 1)	
SPI-Wr-RdD	MOSI	Data	
	MISO	Data	
	SS2	Sample	
	SS3	Sample	
SPI-WrQ	MOSI	Data	
	MISO	Data	
	SS2	Data	
	SS3	Data	
SPI-RdQ	MOSI	Sample	
	MISO	Sample	
	SS2	Sample	
	SS3	Sample	
Apply-4	MOSI	Send Constant Value (possible values: High-7, 0 and 1)	
	MISO	Send Constant Value (possible values: High-Z, 0 and 1)	
	\$\$2	Send Constant Value (possible values: High-7, 0 and 1)	
	\$52	Send Constant Value (possible values: High-Z, 0 and 1)	
	555	Sena constant value (possible values, high-2, 0 and 1)	

5.3 Custom segment example – detailed

Here is an example of segment definition:

Segments	Definition	
Segment0clkIdle	Label	Segment1
Segment1clkIdle SegmentData4	Length	12
SegmentData5 SegmentOclkRunning	SCLK	Idle-0 🔻
Segment1clkRunning	MOSI (DQ0)	Unused 🔻
Segment1	MISO (DQ1)	Sample 🔻
	WE	0 •
	SS0	0 🔹
	SS1	1 •
	SS2 (DQ2)	Unused 🔻
Add Delete	SS3 (DQ3)	Unused 🔻

Label name : Segment1 Length : 12 clock cycles SCLK is held at low level during the segment MISO is sampled (12 bit) WE : held at low level SSO : held at low level SS1 : held at high level Other bits are left unused.

5.4 Defining a custom macro

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

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 'SegmentOclkIdle', 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 would like to add to the macro.

Macros	Definition		Segments
Phase1 Phase2 Phase3 Phase4 ReturnToInitial	Label Clocks & Phases Open Drain Bit Order	Phase1 Configure Configure LSBit First	SegmentOclkIdle Segment1clkIdle SegmentData4 SegmentData5 SegmentOclkRunning
	Byte Order	LSByte First Remove Move Up Move Down	Segmenticikkunning
Add Delete	Segments	Segment1clkIdle Segment0clkIdle	Add To Macro

The segments are added to the macro:

Macros	Definition		Segments
Phase1	Label	Phase1	Segment0clkIdle
Phase2	Clocks & Phases	Configure	Segment1clkIdle
Phase3 Phase4	Open Drain	Configure	SegmentData4
ReturnToInitial	Bit Order	LSBit First 🔹	SegmentOclkRunning
	Byte Order	LSByte First 🔹	Segment1clkRunning
		Remove Move Up	
	Segments	Segment1clkIdle	
		Segment0clkIdle	
	Use the 'N	Nove Up' and 'Move Down'	
Add Delete	buttons if	the segments must be ed in the list; the macro will	Add To Macro
	execute 't	on seament first'	
		op segment just .	

Use the 'Bit Order' and 'Byte Order' drop down boxes to configure bit and byte ordering (see section 4.2 for more information).

Use the 'Configure' buttons to set up the clock and the I/O open-drain features;

Macros	Definition			Segments
Phase1	Label	Phase1		Segment0clkIdle
Phase2	Clocks & Phases	/ Configure		Segment1clkIdle
Phase4	Open Drain	Configure		SegmentData5
Thuse	;			ocginentoatao
		SPI	SelectOpenDrainSpi	
SPI Custom SPI Macro De	efaults		Select the open	drain configuration:
Clock Division	4		SCLK	
Clock Frequency	25 MHz		MOSI (DQ0)	
Clock Driving Edge	Rising 🔹		MISO (DQ1)	
Clock Sampling Edge	Falling 🔻		WE	
			SS0	
0	Cancel Done		SS1	
			SS2 (DQ2)	
			SS3 (DQ3)	
				Cancel Ok

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.

In the 'SelectOpenDrainSpi' window, checking a tick box enables the 'open drain' configuration of the I/O.

6 GPO patterns sequence

6.1 How to define GPO patterns

GPO (General Purpose Output) sequences are sequences of arbitrary digital patterns generated onto the 'GPO port' of SPI Storm. Unlike the 'Serial Port', SPI Storm's GPO port is output-only.

GPO sequences are defined from the 'GPO' tab in the main window. Clicking on 'Add' in the 'GPO Segment Definition' prompts for the name of a new GPO segment. As for custom serial protocols, GPO patterns are defined by 'macros'. Each macro is the assembly of one or multiple GPO segments.

6.1.1 Defining GPO segments

	SPI Add GPO Segment
PI/ GPO	Please enter a label for the new segment:
	GPOSegment1
	Cancel Ok

Once it is created, the segment name appears in the 'segments list'.

Segments	Definition	
GPOSegment1	Label	GPOSegment1
	Repeat	5
	Segments	Data

The table below shows the parameters used to define a GPO segment.

Parameter	Valid values	Description
Label	Any string composed of letters and figures	GPO segment name
Repeat	1 to 2147483647	Sets the number of times the data defined in 'Segments' has to be repeated
Segments	Series of 8-bit values representing the logic levels to be generated onto the GPO port	The following options are available for entering data: - 'Data' (selected from the drop down list): in this case, the logic patterns must be entered in the dialog window, one pattern for each line, according to the following format: b7b6b5b4b3b2b1b0 with b# = 0 or 1 value.

Parameter	Valid values	Description
		Example: 00110111 - 'File' is selected from the drop down list: in this case, a text file must be specified. This text file contains the list of patterns to be sent according to the same format – that is, a list of 8-bit patterns; 1 pattern per line in the file.
		Example:
		11001101
		00011101
		11111111

6.1.2 Defining GPO macros

GPO macros are composed of one or multiple GPO segments. GPO Macros are defined from the 'GPO macros definition area' in the GPO tab.

To define a new GPO macro:

- 1) Click on 'Add' button below the Macros list.
- 2) A window pops up, prompting for macro name.
- 3) Once the macro is defined, set up its properties as follows:
- To set up a GPO macro:

Macros	1 Definition	: Select a GPO seg	ment fro	m the list	Segments
GROMacro1	Definition	CDOM and			Segments
GPOMacrol	Label	GPOMacrol			GPOSegment2
	Clock Division	4			or obeginente
	Clock Frequency	25 MHz			
	Open Drain	Configure		2:	Click on 'Add To Macro'
	Segments	🔀 Remove 🔺 Move	e Up 🔻	Move Down	
3: Editing	the 'Clock				
division' a	llows to change				
the freque	ency of the GPO				
patterns.					
The frequ	ency is compute	d			
Add from the	nternal 100 MH	2			Add To Macro
clock – Clo	ock frequency =				· · · · · · · · · · · · · · · · · · ·
100 MHz / Macros	Clock division				Segments
GPOMacro1	Label	GP@Macro1			GPOSeament1
	Clock Division	4			GPOSegment2
	Clock Division	4			
	Clock Frequency	25 MHz			When added, the segm
	Open Drain	Configure			appears in the macro
	Segments	😫 Remove 🛋 Mov	e Up 🛛 🔍	Move Down	definition list.
		GPOSegment1			Repeat this to assemble
		GPOSegment2			multiple segments as a
		GPOSegment2			macro.
		GPOSegment1			iviacros execute segme
			-		from top to down.
Add Delete			SPI Selec	tOpenDrainGp	
				C - I + +	ducia configurations
	4. Einally	each CBO output		select the oper	drain configuration:
	4. Filidily,	t as open drain		GPO0	
		click on		GPO1	
	'Configure	e' and select the		GPO2	
	correspor	nding tick box. A		GPO3	
	selected t	ick box means		GPO4	
	that the L	/O is configured		GPO5	
	as oppen-	drain.		GPO6	
				GPO7	
					Cancel Ok

The following table summarizes the parameters of a GPO macro.

Parameter	Possible value	Description
Label	Any string composed of letters and numerical characters	GPO macro name
Clock division	Integer value from 1 to 1024.	Clock division factor. Defines the clock rate at which the GPO macro will be executed. The output rate of the GPO macro is: 100 MHz / (Clock division). The generated clock frequency for the GPO macro is shown in the 'Clock Frequency' field.
Open drain	(Click on button 'Configure' to change open drain configuration of GPO)	The 'Configure button' opens up a windows that allows configuring the GPO outputs as open-drain.
Segments	(list)	List of segments executed when running the selected GPO macro.

SPI Storm Studio User's Guide

7 Defining a program

In SPI Storm Studio, 'programs' are defined through the 'Program' tab.

7.1 Program tab overview

SPI SPI Storm Studio		
<u>F</u> ile <u>T</u> ools <u>H</u> elp		
1 🕄 💐 💕 🔛 🗷	P 🛱 🛈	a
Processes V I X	Project Description Initial State Standard SPI Custom SPI GPO	Program Run 🗢 X
2. Initial State 3. Standard SPI 4. Custom SPI 5. GPO	Power Supply and Clock Selection Status External Power Supply Not selected External Clock Not present - refer to se	ly and clock selection ection 7.2
6. Program	Selected Clock Internal	
7 Run	· · · · · · · · · · · · · · · · · · ·	1
	SPI Trigger SPI trigger –	GPO Trigger GPO trigger – refer to section 7.4
	Pefinition refer to section 7.3	Enable
	D7 D6 D5 D4 D3 D2 D1 D0	Slave to SPI
		Condition D7 D6 D5 D4 D3 D2 D1 D0
	SPI Program	GPO Program
	Definition	Definition
	Begin	Begin
	SPI Program – refer to section 7.5 SpiCustom : "Phase1"	GpioMacro : "GPOMacro1"
	Data source : No data	End
	For-loop begin Iterations : 10	
	Spifustom - "Phase3"	GPO Program – refer to section 7.6
	Data source : No data	
	SpiCustom : "Phase4"	
	Data source : Buffer	
	For-loop end	
	SpiStandard : "DataWr"	
	Data source : Buffer MOSI Data : ffff	
	SpiCustom : "ReturnToInitial" Data source : No data	
	End	
		(<u></u>)
Output		- ₫ ×
		*
Checking devicesdone		
Checking datadone		
Saving project filedone.		
		*
Device connected : 🔇 Pro	nject defined : 🔽 Project saved : 🔕 Project : C:\Program Files (x86)\Byte Par	adigm\SPIStormStudio\Examples\ExampleFile1.ssp

7.2 Power supply and clock selection

This area shows the status of external power supply and external clock presence. It also allows selecting between clock sources.

Power Supply and Clock Selection

Status		_	 External power supply status
External Power Supply	Not selected		
External Clock	Not present	-	 External clock presence status
Selected Clock	Internal	•	Clock source selection control

Field	Possible values	Description
External Power Supply	Not Selected / Selected	Returns the presence status of an external power supply source on the I/O External voltage supply connector.
External Clock	Not present / present	Returns the status of a clock signal presence detection on the Cki input pin of SPI Storm connector. A valid continuous clock signal must be present on this pin if an external clock reference signal is to be used.
Selected Clock	Internal / External	Controls the clock source to be used for executing a program in SPI Storm studio. If 'Internal' is selected, SPI Storm will use its own internal 100 MHz reference clock
		If 'external is selected', an external reference clock signal must be supplied onto the CKI pin.

Remark:

If an **'external clock source'** is selected and the external clock signal is not present at the CKI input pin, SPI Storm Studio cannot run its program. At program run (see section 7), the following message will appear if no valid reference is applied on the external clock input pin:

7.3 SPI trigger

This area is used to configure the trigger conditions used to start execution on the SPI port.

A 'trigger condition' is a logic expression built on the inputs signals of the 'control port'. Upon occurrence of this condition, SPI Storm will start executing the 'SPI Program' defined from the Program tab (see section 7.5).

SPI Trigger	
Definition	
Enable 🔲	
D7 D6 D5 D4 D3 D2 D1 D0 U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U U	

To enable SPI trigger, check the 'Enable tick box' of the SPI trigger. Leaving this box unchecked disables the trigger: SPI Storm program will be executed as soon as the user clicks on the 'RUN' button in the 'Run' tab.

When a trigger is defined, clicking on the 'RUN' button in the 'Run' tab will just 'arm' the trigger: SPI Storm will wait until it detects the defined trigger condition and then start executing the program.

Once SPI trigger is **enabled**, the 'Condition' in SPI Trigger area become active and the trigger condition can be defined on the D7...D0 input pins.

SPI Trigger		
	Definition	
	Enable	
	Condition	D7 D6 D5 D4 D3 D2 D1 D0 U • U • U • R • U • 0 • 0 • 1 •
Trigger definition controls. The trigger		

condition is defined as a AND equation.

D7 D0 possible values	Description
U	Undefined – don't care: this input pin is not used for the definition of the trigger condition.
0 Low logic level : in this case, a low logic level must be detected.	
1 High logic level : in this case, a high logic level must be detected.	
R	Rising edge : in this case, a transition from low logic level to high logic level must be detected.
F	Falling edge : in this case, a transition from high logic level to low logic level must be detected.

Example:

If the value 'UUUU1001' is defined in the D7...D0 bits of the SPI trigger, the SPI Program will be executed upon the detection of the binary value "1001" on the control port input pins D3 down to D0.

7.4 GPO trigger

GPO Trigger		
Definition		
Enable		
Slave to SPI		
Condition	D7 D6 D5 D4 D3 D2 D1 D0 U + U + U + U + U + U + U + U +	

The GPO trigger works on the same fashion as SPI trigger; it is also defined on the control port input pins.

When it is enabled, the option 'Slave to SPI' allows using the SPI trigger as trigger condition for the GPO program execution. This allows synchronizing GPO program execution with the execution of SPI Program.

If the 'Slave to SPI condition' tick box is left unchecked, GPO trigger is defined independently .

7.5 SPI Program

7.5.1 Overview

This area allows defining 'programs' for the SPI port – that is, the sequence of SPI port macros to be executed.

l Program		
efinition		
Begin		
SpiCustom : "Phase1" Data source : No data		
For-loop begin Iterations : 10		
SpiCustom : "Phase3" Data source : No data		
SpiCustom : "Phase4" Data source : Buffer MOSI Data : 0F		
For-loop end		
SpiStandard : "DataWr" Data source : Buffer MOSI Data : ffff		
SpiCustom : "ReturnToIniti Data source : No data	r	
End		

The table below gives the list of items that can be inserted in a 'program'.

Item type	Parameters	Description
Standard SPI macro	Optional: data out field ('MOSI data'), depending on macro.	Macros defined as standard accesses from the SPI tab
Custom SPI macro	Optional: data out field ('MOSI data'), depending on macro.	Macros defined as custom accesses from the Custom SPI tab
Loop	Number of iterations	For loop statement: allows executing a subsequence a predefined number of times.

(Explain the format of simple, dual and quad data). (Explain where the data in is read back).

7.5.2 Building up a program from SPI Storm Studio GUI

7.5.2.1 Overview

When 'Insert Before' or 'Insert After' is selected, the following window pops up:

SPI Add Macro	X
Macro Type Standard SPI -	Macro Label DataWr Cancel Ok
n drop-down menu.	
	Select macro to insert f

drop down. The list is filled in with the macros of each types that were defined with 'Standard SPI' and 'Custom SPI' tabs of SPI Storm Studio

Select Macro T

- Valid choices a
- Standard SPI
- Custom SPI
- Loop

7.5.2.2 Inserting a Standard SPI or Custom SPI macro

Once the macro that you wish to insert is selected, click on 'OK'.

If the selected macro requires a data parameter, the following window appears. According to the selected output data source, the window format changes.

Macro name		
	Define SPI data	
	Data source Data	Macro output
	MOSI data	data source 'Data'
	Cancel Ok	
		1
	Spi transfer: Datawr	
Data out source file	Define SPI data Data source Text File (.txt)	Macro output
path and name	MOSI file	'Text File (.txt)'
Data in destination file path and name	MISO file	
	Cancel Ok	
	SPI Spi transfer : "DataWr"	
Data out source file	Define SPI data	
path and name	Data source Binary File (.bin)	Macro output
	MOSI file	'Binary File
Data in destination file path and name	Reset file index	(.bin)'
	Cancel Ok	

Macro parameter	Valid values	Description
Data source	Data Text File (.txt) Binary File (.bin)	Specifies the source for the data sent out by the macros. 'Data' specifies a constant value inserted in the program. 'Text file' specifies an external text-formatted file containing the data to be sent. 'Binary file' specifies an external binary-formatted file containing the data to be sent.
MOSI data	String representing a serial data as hexadecimal string.	 Only valid if 'Data Source' is set to 'Data'. If the corresponding macro expects a constant data to be sent out onto the data lines. The number of used data lines and the total length depend of the defined macro. Example: if a 8 bit data string is defined, 'A9' (without the quotes) is an acceptable data format. Current default bit ordering is 'LS bit', LS byte first. Input data sampled by the called macros can be found from the log file (see 'Running a program'). Optional bit ordering will be available in future SPI Storm Studio versions. Please contact support@byteparadigm.com for availability.
MOSI file	Path and file name containing the data to be sent out.	 Only valid if 'Data Source' is set to 'Text File (.txt)' or 'Binary File (.bin)'. In this case, the source data is supplied by means of an external source file. If text file is selected, each line bears 1 data vector formatted as an hexadecimal string without prefix. 'Reset file index' allows re-starting at the beginning of the file at each loop iteration (if loops are used in the program). Otherwise, each successive call to the file will increment the index in the file and fetch the 'next data'.
MISO file	Path and file name used as destination for the data being read back by the called macros.	Only valid if 'Data Source' is set to 'Text File (.txt)' or 'Binary File (.bin)'. If the called macros sample data, this file is used as destination to store the sampled data.

7.5.2.3 Inserting a For Loop

In this case, the type 'Loop' is selected from the 'Macro type' drop-down menu.

SPI Add For-loop			
Please enter the n	umber of iterations:		
10			
	Cancel Ok		

A dialog box pops up, requesting the for loop count. Nested loops (loops in loops) are not allowed.

7.6 GPO Program

The method for inserting GPO macros and loops in the GPO Program area is identical to this of SPI Program. Please refer to section 6.5 for more details.

As opposed to SPI Program, a GPO Program will only make use of GPIO Macros defined from the GPO tab. For loops are also allowed, as shown in the 'Add Macro window' below.

SPI Add Macro					
Macro Type GPIO Macro GPIO Macro Loop	Macro Label	Cancel Ok			

7.7 File formats

This section provides a detailed description of the file formats used in SPI Storm Studio.

7.7.1 Standard and custom macro file format

Context: Program tab, when inserting a macro in the SPI program.

When applicable, the data can be provided for macros as follows:

- Entering the data in the GUI (Data option see below);
- Providing a text file (.txt) (Text File (.txt) option see below);
- Providing a binary file (.bin) (Binary File (.bin) option see below).

SPI Spi transfer : "DataWr"			
Define SPI data			
Data source	Data 🔹		
MOSI data	Data		
	Text File (.txt)		
	Binary File (.bin)		
	Cancel Ok		

If one of the **file formats** is selected, one of the following dialog boxes appears:

SPI Spi transfer : "DataWr"	SPI Spi transfer : "DataWr"
Define SPI data	Define SPI data
Data source Text File (.txt)	Data source Binary File (.bin)
MOSI file	MOSI file
MISO file	MISO file
Reset file index 🛛	Reset file index 🛛
Cancel Ok	Cancel Ok

MOSI file: path to the file for the data that are sent as output by the corresponding macro. This is the location from where the macro reads the data that it has got to send out.

MISO file: path to the file for the data that is read back by the corresponding macro. This is where the macro will write the read back data. No content for this file must be specified, as this file will be written after the execution of the macro.

Revision 1.03 - 01-Oct-2013

File format:	Text file (.txt)						
File structure:	Each line in the file contains a single output data as required by the corresponding macro.						
	Successive calls to the same macro increment the index in the file, allowing placing all the data related to a macro in the same file. Please refer to section 7.5.2.2 for additional options, such as 'reset file index'.						
	Each line must contain the information equivalent to 1 data for the corresponding macro. The size of this data depends on the defined macro.						
	For example, if the macro requires a total of 13 bit out, each line will have to provide 13 bit of data.						
Data format in the file:	Hexadecimal number without prefix.						
	The default hexadecimal format is MSBit first / MSByte first – this convention was adopted as a default from SPI Storm Studio version 1.1.14. Each macro bit / byte ordering is defined in the .ssp file through SPI Storm Studio interface.						
	Hereafter is an example with convention "most significant byte first, most significant bit first" = MSBit, MSByte first.						
	For instance: data specified on 16 bits as: 1A45 will be sent: 1A - 45 msbit first:						
	0-0-0-1-1-0-0-0-0-0-0-1-0-1						
	Below is an example with convention "least significant byte first, least significant bit first = LSBit, LSByte first.						
	For instance: data specified on 16 bits as: 1A45 will be sent: 45 - 1A , lsbit first:						
	1-0-1-0-0-1-0 0-1-0-1-1-0-0-0						
Example	Let's assume that we specify a file for macro 'DataWr', a standard SPI access with 13 bit length.						
	Each data will have to be 13 bit long, which is formatted onto 4 hexadecimal characters.						
	1A45						
	0444						
	19A4						
	0017						
	1002						

File format:	Binary file (.bin)		
File structure:	This file contains all the output data required by the corresponding macro.		
	Successive calls to the same macro increment the position in the file. The macro will fetch the data required for its execution and then increment its position in the file to fetch the next data. Please refer to section 7.5.2.2 for additional options, such as 'reset file index'.		
Data format in the file:	Byte-padded data. E.g.: 9 bits of data is mapped onto 2 bytes of data. There is no carriage return i binary files. Data is sent least significant byte first, least significant bit first. Example:		

If the binary file contains 2 bytes of data:
A1 56
If serialized (depends on the format of the macro), data will be sent:
Convention MSBit first:, MSByte first
0-1-0-1-1-0-1-0-0-0-0-1-
Convention: LSBit first:, LSByte first
1-0-0-0-1-0-1 0-1-1-0-1-0

7.7.2 GPO segment file format

Data for GPO are specified at the 'segment' level. When a GPO segment is created, the data source can be entered in the GUI or specified as a file. See picture below.

<mark>д х</mark> п		(i) Project Description / Initial S	State Standar	d SPI Custom SPI GPO Program Run
	GF	Segments GPOSegment Add Delete	ON Definition Label Repeat Segments	GPOSegment 1 Data • File Data

File format:	Text file				
File structure:	Each line in the file contains the data of a new 8-bit vector to be applied successively onto the GPO lines. Each line must contain the information for 8 bit (1 byte).				
GPO vector format in the file:	Hexadecimal number without prefix, 2 characters, representing a byte of information.				
	Convention: most significant bit first.				
Example of file:	A5				
	33				
	55				
	12				
	66				
	70				
	01				
	10				

7.7.3 Output file format

The output file lists the accesses executed by the SPI Storm Studio program. It is notably used to collect the data that is read back, when no input file is specified.

[Xfer]	[Xfer] : marks the beginning of a tranfer.
Nr = 1	Nr = index of the transfer
Spi4 = 131072	<macro identifier="" type=""> : keyword showing whether a standard SPI or a custom SPI macro was</macro>
Mosi = 1A45	executed. The value that follows is an internal reference number for the macro.
Miso = 0000	Mosi : if applicable and variable, shows the data sent by the macro. Nothing is written here if not used.
	Miso : if applicable, shows the data read back by the macro. Nothing is written here if not used.

Example:

[Xfer] Nr = 1Spi4 = 131072 Mosi = 1A45Miso = 0000 [Xfer] Nr = 2 Spi4 = 131072 Mosi = 0444Miso = 0000 [Xfer] Nr = 3 SpiC = 262145 Mosi = 8 Miso = [Xfer] Nr = 4 SpiC = 262144 Mosi = Miso =

8 Running a program

Programs are run from the 'Run tab' of SPI Storm Studio. To run a program, a SPI Storm device must be connected to the PC. Please refer to section 3.2 to know how to connect your SPI Storm device.

SPI SPI Storm Studio				Competition in Street,				- • ×
<u>F</u> ile <u>T</u> ools <u>H</u> elp								
Processes • I × Project Description Initial State Standard SPI Custom SPI GPO Program Run = ×								
1. Project Description								
2. Initial State	Jal State Output File							
3. Standard SPI 4. Custom SPI	.\Exampl	AcxampleFile.out Cutput file path and name						
5. GPO						Pro	ogram progress	har
6. Program	Progress							·····,
7. Run	Pur	Initialising	Declaration	SPI Triggering	SPI Ru	nning	Receiving Data	Done
	Kui	Initialising	Freioduling	GPO Trigger	ing	GP	O Running	Done
			°i					
Click on 'Run' bu	tton to r	un program						
	Nr	Spi Type	Lab	el	1	Mosi		<u>^</u>
	1	SPI-4 Acc	ess12	5A3			000	
	2	SPI-CUSTOM Pha	ise1	-			-	
	3	SPI-4 Dat	aWr	00AA			0000	
	4	SPI-CUSTOM Pha	ise3	-			-	=
	5	SPI-CUSTOM Pha	ise4	OF			000	
	6	SPI-4 Dat	aWr	00AA	Program e	vecution lo	0000	
	7	SPI-CUSTOM Pha	ise3	-			· ·	
	8	SPI-CUSTOM Pha	ise4	OF			000	
	9	SPI-4 Dat	aWr	00AA			0000	
	10	SPI-CUSTOM Pha	ise3	-			-	
	11	SPI-CUSTOM Pha	ise4	OF			000	
	12	SPI-4 Dat	aWr	00AA			0000	
	13	SPI-CUSTOM Pha	ise3	-			-	
	14	SPI-CUSTOM Pha	ise4	OF			000	
	15	SPI-4 Dat	aWr	00AA			0000	
	16	SPI-CUSTOM Pha	ise3	-			-	
	17	SPI-CUSTOM Pha	ise4	OF			000	Ψ
								· ·
Output								~ ↓ ×
								A.
Checking devicesdone								*
Device connected : 🗹 Project defined : 💟 Project saved : 💟 Project : C:\Program Files (x86)\Byte Paradigm\SPIStormStudio\Examples\Examples\ExampleFile1.ssp								

SPI Storm Studio GUI allows running 'programs'.

For individual calls to macros, please refer to section 9: SPI Storm API.

9 SPI Storm API

9.1 Overview

SPI Storm Studio GUI must not be used to control SPI Storm device. SPI Storm Studio C API can be used to build up your own application / environment.

There is no difference between what can be achieved with the DLL or SPI Storm Studio GUI.

Basically, controlling SPI Storm device always consists in defining a SPI Storm Studio project file which contains the settings of the SPI Storm device and the library of macros that you need. Using the API from within your own environment can be more flexible, according to your application.

For instance, SPI Storm Studio API allows executing macros or programs directly. Calling macros directly can provide more flexibility to organize Standard SPI, Custom SPI and/or GPO accesses.

9.2 Detailed Functions Description

Please refer to document: SPI Storm Studio – C library user's guide: ug_SPIStormStudio_CLib.pdf, available from Byte Paradigm web site: <u>http://www.byteparadigm.com/files/documents/ug_SPIStormStudio_CLib.pdf</u>

9.3 Files Needed to Use the API

Please copy the following file into your application directory:

- SpiStorm.dll
- xspi*.bin
- Your SPI Storm Studio project file (.ssp)

9.4 Programming example

Programming examples can be found from Byte Paradigm's website <u>http://www.byteparadigm.com/support/software-downloads/</u>