Introducing SPI Storm

Serial Protocol Host Adapter on USB

SPI Storm is Byte Paradigm's USB Serial Protocol host adapter. SPI Storm supports SPI (Serial Peripheral Interface), 3-wires SPI, dual-SPI, quad-SPI protocols as a master. It also allows the control of custom serial protocol interfaces up to 100 MHz from PC.

This white paper introduces the unique features of SPI Storm and explains why this device can be conveniently used for ASIC, FPGA, SoC and embedded system development, testing and debugging.

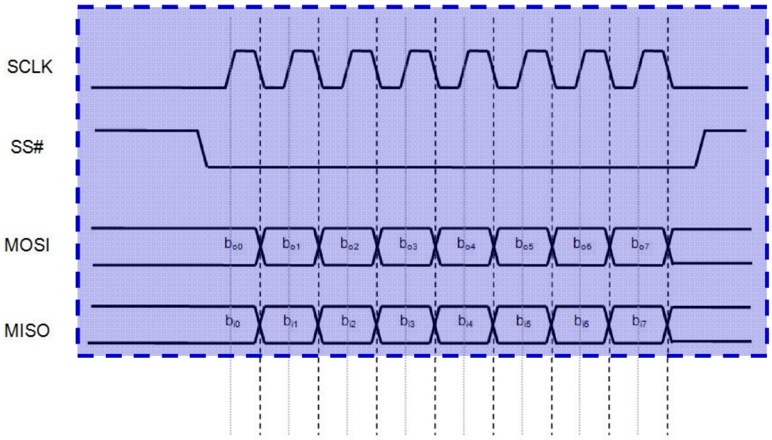
SPI, Dual-SPI and Quad-SPI protocols

In the world of protocols used for communications between integrated circuits, SPI (Serial Peripheral Interface) has been an all-around protocol because of its simplicity, especially when there is a need to stream large data quantities with low pin counts.

It is based on 4 signal lines (see Figure 1)

- One clock signal (SCLK) sent from the bus master to the slave(s); this clock is active when data is exchanged between the master and the slave;
- One slave select (SS) line for each slave used to select the slave the master communicates with;
- One data line from the master to the slave ('Master-Out-Slave-In' - MOSI);
- One data line from the slave to the master ('Master-In-Slave-Out' – MISO).

Figure 1: Serial Peripheral Interface (SPI-4 protocol)



SPI is what about what any electronic engineer would think of if he/she had to define a simple interface between 2 integrated circuits. As 'de facto standard', the SPI protocol does not define much more – no specific addressing scheme, no specific higher level formatting; data length was initially 8 bits (1 byte) but is more or less implicitly 'any length'.

There are many advantages to using SPI protocol:

- Clock rate is not restricted: within the physical limits of the ICs, increasing the clock rate would increase the data rate;
- The protocol is full-duplex: data can be received as data is sent;
- It is equally suitable for various types of applications: from register access with a fixed addressing / data scheme to the streaming of large data quantities;
- SPI does not care about the I/O voltage.

This protocol also presents some drawbacks. Among others:

- For advanced uses, it requires building full protocol stacks from scratch (as SPI does not define anything);
- SPI lacks a built-in acknowledgement mechanism.

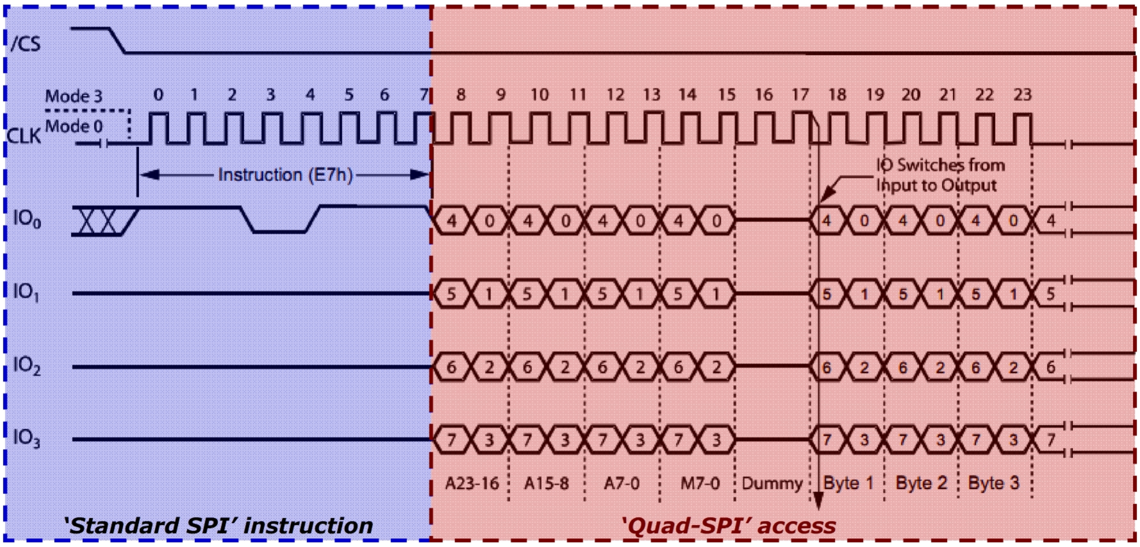
With time, SPI has known variants and extensions – a few examples:

- The clock signal is sometimes generated as continuously toggling signal, as the SS line is enough to detect phases when data is sent and phases when data is not sent;
- Some variants merge MOSI and MISO data lines, implementing a half duplex SPI protocol with a bidirectional data line (which is often referred to '3-wires SPI');
- Especially for flash memories, dual-SPI and quad-SPI protocol were created to increase the available bandwidth. Because they use multiple data lines in parallel, dual-SPI and quad-SPI protocols cannot strictly be considered as 'serial' protocols any more, though.

Dual-SPI and Quad-SPI protocols are more or less 'standardized' extensions of the SPI protocol.
They are based on the following principles:

- Standard SPI protocol is used by default by dual-SPI and quad-SPI protocol slaves;
- Specific SPI instructions are used to set the slave in 'dual-SPI or quad-SPI mode'
- Once the slave is configured in 'dual- or quad-SPI' mode, the data transfers (read or write) use some additional control lines as data lines, resulting in a data bus extended from 1 data signal line to 2 (dual-SPI) or 4 (quad-SPI) data lines. Of course, the purpose is to increase the available bandwidth (see Figure 2).

Figure 2: Example of transition from simple SPI to quad-SPI protocol



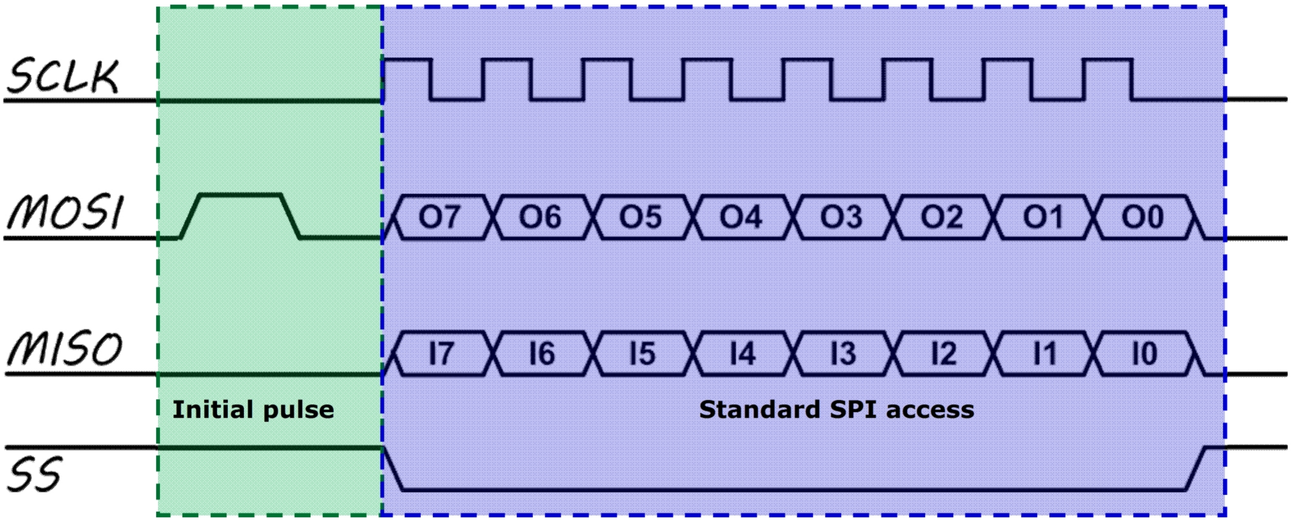
Custom serial protocols

Designing custom IC such as ASIC, SoC or FPGA allows choosing between standard or custom protocols for IC-to-IC communications.

Typical differences between these protocol and the standard SPI:

- Higher levels of the protocol stacks are implemented in the communication protocol itself, requiring a more strict use of the clock and control lines;
- Specific data pulse sequences, with or without clock are used to set the slave components into specific modes, or to 'wake them up';
- Open-drain I/Os are used;
- Like for Quad-SPI protocol, additional data lines are used;
- Other implementations tend to save on I/O counts and require bi-directional data lines;
- Edges used for data is generation and sampling vary from what is specified for SPI.

Figure 3: Example of custom protocol composed of a pulse on MOSI, and a standard SPI access.



SPI Storm: Beyond Serial Peripheral Interface (SPI)

SPI Storm offers unprecedented possibilities to the electronic engineer, who wishes to develop, test and debug embedded systems. These unique capabilities include:

- The ability to access chips / embedded systems with **standard and extended standard protocols.**
  SPI Storm is notably the world first dual-SPI and quad-SPI host adapter on USB.

- The ability to go beyond standard protocols and to **support user-defined types of accesses** through a graphical user interface.
  SPI Storm is based on the principle that a **chip-to-chip communication protocol can be defined as the assembly of simple 'segments'** characterized by 'properties' in terms of clocking, signal lines, control lines behavior, data sampling and data generation.

- A dramatic upgrade in terms of **hardware capabilities in a USB form factor PC instrument.** SPI Storm is able to clock serial interfaces up to **100 MHz**, a reasonable higher limit for single-ended signal lines that still guarantees adequate signal integrity. Its **32 MByte** memory buffer guarantees real-time signal generation and data sampling over a wide range of applications.

In brief, SPI Storm is a unique and fast-to-deploy host adapter solution for interfacing the standard and non-standard serial interfaces commonly used in embedded systems for chip-to-chip communications.

SPI Storm is ideally used for:

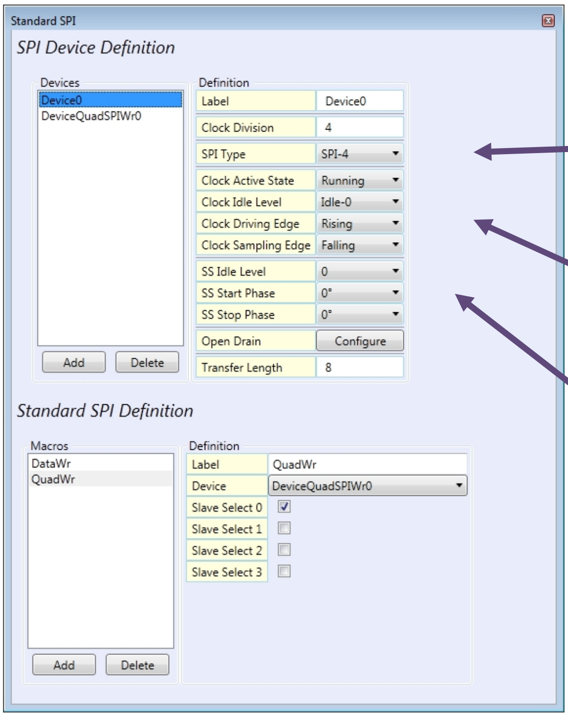
- Troubleshooting a digital design;
- Accessing registers;
- Streaming data inputs through serial ports;
- Demonstrating ICs and embedded systems that use serial protocols;
- Program SPI, dual-SPI and quad-SPI flash memories;
- Accessing ADC, DAC and other components that use standard SPI protocols;
- Accessing custom chips (FPGA, ASIC) using non-standard protocols for IP testing and design validation on prototype.

Unique features and performance level

The following table compares SPI Storm features with SPI Xpress and their nearest competitor.

	SPI Storm	***SPI Xpress***	***Nearest Competitor***
Device type	**Master**	**Master / Analyzer**	**Master**
Host Adapter / Master characteristics			
Clock max. frequency	**100 MHz**	**50 MHz**	**40 MHz**
All SPI modes supported	●	●	●
Number of slaves	**4**	**5**	**3**
Selectable SS polarity	●	●	●
GUI	●	●	●
Free API	●	●	●
Continuous clock generation	●	●	
Selectable SS edge position step	●	●	
Bi-directional (3-wires) protocols	**Full flexibility**	**Basic, fixed protocol (Read after Write)**	
Dual-SPI	●		
Quad-SPI	●		
Custom protocol	●		
Open-drain I/O	●		
GPO / Digital Pattern Generator	**Side 8 bits GPO / Digital Pattern Generator**		
Streaming mode	●		
Total memory buffer	**32 MB**	**16 kB**	**Information not available**
Control Software	**SPI Storm Studio**	**8PI Control Panel**	●

SPI Storm Studio protocol definition engine



Standard SPI

SPI Device Definition

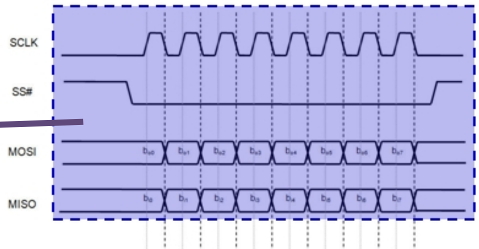
Devices
- Device0
- DeviceQuadSPIWr0

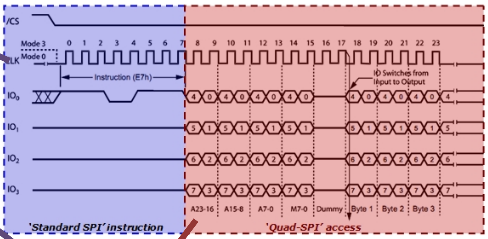
Definition	
Label	Device0
Clock Division	4
SPI Type	SPI-4
Clock Active State	Running
Clock Idle Level	Idle-0
Clock Driving Edge	Rising
Clock Sampling Edge	Falling
SS Idle Level	0
SS Start Phase	0°
SS Stop Phase	0°
Open Drain	Configure
Transfer Length	8

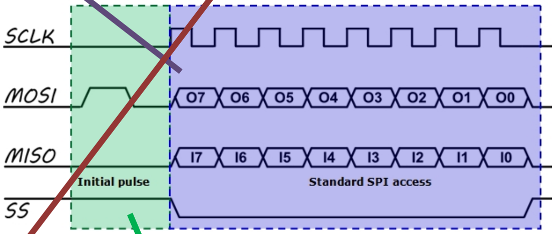
Standard SPI Definition

Macros
- DataWr
- QuadWr

Definition	
Label	QuadWr
Device	DeviceQuadSPIWr0
Slave Select 0	☑
Slave Select 1	□
Slave Select 2	□
Slave Select 3	□

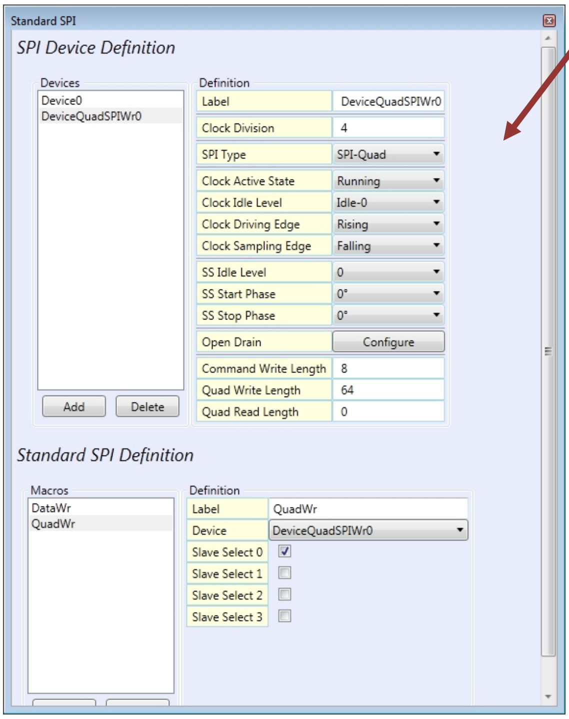






Standard SPI and standard Quad-SPI 'macros' are defined from the 'Standard SPI' tab of SPI Storm Studio.

Each protocol characteristic is built-in the software and selected with simple controls.



Standard SPI

SPI Device Definition

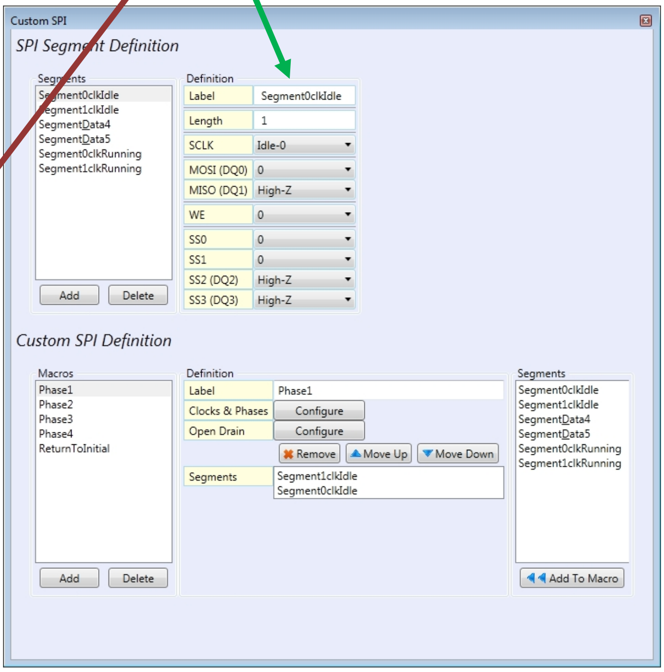
Devices
- Device0
- DeviceQuadSPIWr0

Definition	
Label	DeviceQuadSPIWr0
Clock Division	4
SPI Type	SPI-Quad
Clock Active State	Running
Clock Idle Level	Idle-0
Clock Driving Edge	Rising
Clock Sampling Edge	Falling
SS Idle Level	0
SS Start Phase	0°
SS Stop Phase	0°
Open Drain	Configure
Command Write Length	8
Quad Write Length	64
Quad Read Length	0

Standard SPI Definition

Macros
- DataWr
- QuadWr

Definition	
Label	QuadWr
Device	DeviceQuadSPIWr0
Slave Select 0	☑
Slave Select 1	□
Slave Select 2	□
Slave Select 3	□



Custom SPI

SPI Segment Definition

Segments
- Segment0clkIdle
- Segment1clkIdle
- SegmentData4
- SegmentData5
- Segment0clkRunning
- Segment1clkRunning

Definition	
Label	Segment0clkIdle
Length	1
SCLK	Idle-0
MOSI (DQ0)	0
MISO (DQ1)	High-Z
WE	0
SS0	0
SS1	0
SS2 (DQ2)	High-Z
SS3 (DQ3)	High-Z

Custom SPI Definition

Macros
- Phase1
- Phase2
- Phase3
- Phase4
- ReturnToInitial

Definition	
Label	Phase1
Clocks & Phases	Configure
Open Drain	Configure
Segments	Segment1clkIdle Segment0clkIdle

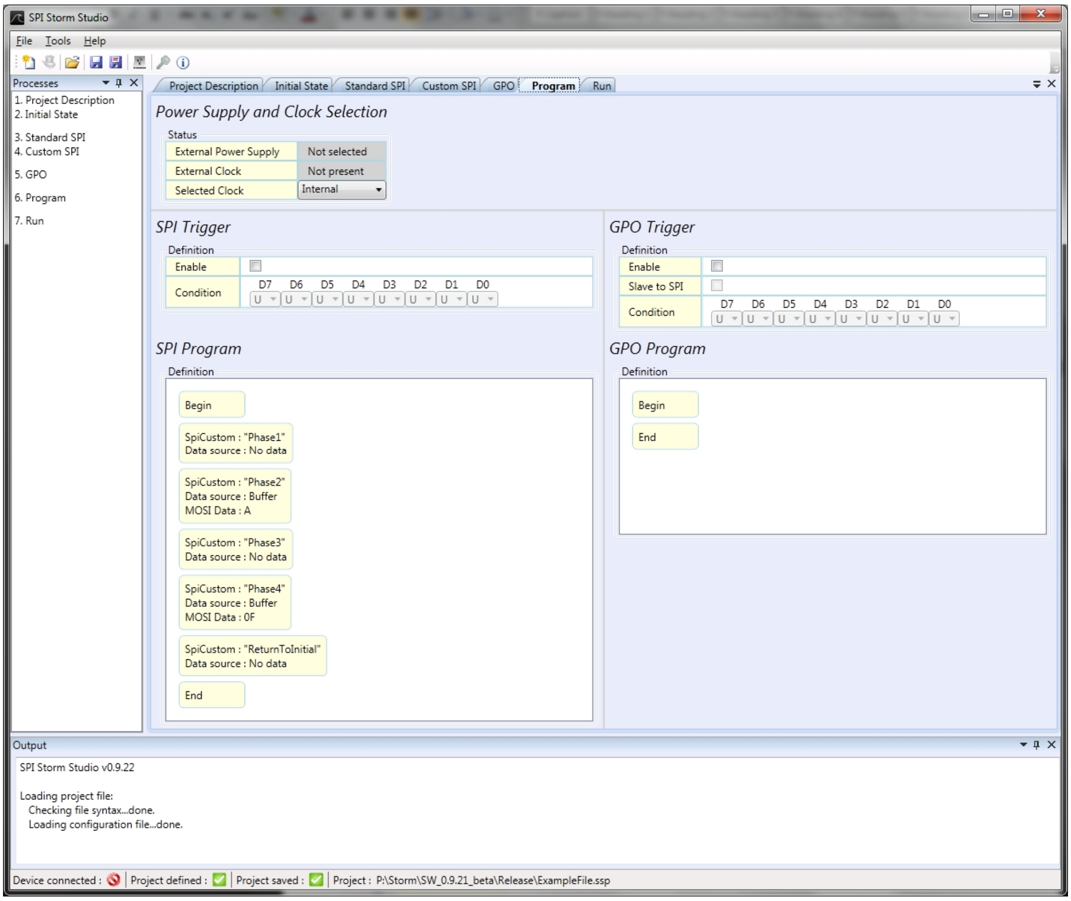
Segments
- Segment0clkIdle
- Segment1clkIdle
- SegmentData4
- SegmentData5
- Segment0clkRunning
- Segment1clkRunning

Custom protocols macros and segments are defined from the 'Custom SPI tab' of SPI Storm Studio.

Segments are the elementary building blocks of the protocols and macros are composed of one or more segments.

Multiple programming interfaces

Once the protocols are defined in SPI Storm Studio, the user can choose to setup and execute real access sequences with the graphical user interface or with C function calls from the provided API.



Power Supply and Clock Selection

Status

External Power Supply	Not selected
External Clock	Not present
Selected Clock	Internal

SPI Trigger

Definition

Enable	□
Condition	D7 U, D6 U, D5 U, D4 U, D3 U, D2 U, D1 U, D0 U

GPO Trigger

Definition

Enable	□
Slave to SPI	□
Condition	D7 U, D6 U, D5 U, D4 U, D3 U, D2 U, D1 U, D0 U

SPI Program

Definition

Begin

SpiCustom : "Phase1"
Data source : No data

SpiCustom : "Phase2"
Data source : Buffer
MOSI Data : A

SpiCustom : "Phase3"
Data source : No data

SpiCustom : "Phase4"
Data source : Buffer
MOSI Data : 0F

SpiCustom : "ReturnToInitial"
Data source : No data

End

GPO Program

Definition

Begin

End

Output

```
SPI Storm Studio v0.9.22

Loading project file:
   Checking file syntax...done.
   Loading configuration file...done.
```

Device connected : 🚫 | Project defined : ✓ | Project saved : ✓ | Project : P:\Storm\SW_0.9.21_beta\Release\ExampleFile.ssp

Figure 4: SPI Storm Studio Program Page

This enables tasks automation and building up complete custom application interfaces from environment able to call standard C functions.

Conclusion

Standard and custom serial protocols used for chip-to-chip communications are everywhere. SPI Storm is a convenient and powerful tool that allows the design engineer access them, for FPGA, ASIC, SoC and full embedded system development, testing and debugging. It offers a dramatic improvement over traditional host adapter solutions, as the user is now able to define his/her own serial protocol and fine-tune the characteristics of standard protocols such as SPI. SPI Storm is also the first USB Host adapter that supports dual-SPI and quad-SPI protocols.

For more advanced information: http://www.byteparadigm.com/product-spi-storm-39.html.