P-SRAM

Evaluation Board

User Guide for Arduino UNO R3

P-SRAM-ADRKIT01

Revision History

Revision No.	Date	History
1.0	08/14/2018	Initial Release
1.01	02/11/2019	Cosmetic Change
1.02	06/26/2019	Updated Part #
1.03	10/25/2019	Fixed header numbering Renamed section 1 to Overview Corrected part numbers
1.04	01/14/2020	Updated Figures 1 and Figure 2 Added Figures 3 and 4 for SOIC-8 Updated Schematics and BOM Updated part numbers Removed SPnvSRAM

Contents

1. Overview

The Avalanche Persistent SRAM (P-SRAM) evaluation board contains a P-SRAM device utilizing Spin-Transfer Torque Magneto-Resistive Random-Access Memory (STT-MRAM) technology enabling users to develop interactive hardware solutions using the Avalanche P-SRAM evaluation board and an Arduino UNO Rev. 3 host board connected via standard SPI interface. The Arduino UNO host board communicates with a computer using a USB 2.0 cable type A/B and terminal emulator software. Avalanche software bundled APIs and test program provide basic access to functionality of the Avalanche P-SRAM device. The Arduino UNO SPI can operate up to 8MHz.

The P-SRAM evaluation kit includes:

- One Arduino UNO REV 3 host board
- One Avalanche P-SRAM evaluation board
    - One populated P-SRAM device (see Table 1 below)
- One USB 2.0 cable type A/B

2. Avalanche P-SRAM Product Support

The Avalanche P-SRAM evaluation board can be populated with one of the following Serial P-SRAM devices:

Table 1: Avalanche Product Support

Part Number	Density	Voltage (Vcc/Vccq)	Frequency	Interface	Package
AS3016101-0010XxxWAx	16Mb	2.7V to 3.6V	Up to 10Mhz	SPI	8-Pin WSON
AS3016101-0010XxxSAx	16Mb	2.7V to 3.6V	Up to 10Mhz	SPI	8-Pin SOIC
AS1016101-0010XxxWAx	16Mb	1.71V to 2.0V	Up to 10Mhz	SPI	8-Pin WSON
AS1016101-0010XxxSAx	16Mb	1.71V to 2.0V	Up to 10Mhz	SPI	8-Pin SOIC
AS3016204-0054XxxWAx	16Mb	2.7V to 3.6V	Up to 54Mhz	SPI, QSPI	8-Pin WSON
AS3016204-0054XxxSAx	16Mb	2.7V to 3.6V	Up to 54Mhz	SPI, QSPI	8-Pin SOIC
AS1016204-0054XxxWAx	16Mb	1.71V to 2.0V	Up to 54Mhz	SPI, QSPI	8-Pin WSON
AS1016204-0054XxxSAx	16Mb	1.71V to 2.0V	Up to 54Mhz	SPI, QSPI	8-Pin SOIC

Part Number	Density	Voltage (Vcc/Vccq)	Frequency	Interface	Package
AS3016204-0108XxxWAx	16Mb	2.7V to 3.6V	Up to 108Mhz	SPI, QSPI	8-Pin WSON
AS3016204-0108XxxSAx	16Mb	2.7V to 3.6V	Up to 108Mhz	SPI, QSPI	8-Pin SOIC
AS1016204-0108XxxWAx	16Mb	1.71V to 2.0V	Up to 108Mhz	SPI, QSPI	8-Pin WSON
AS1016204-0108XxxSAx	16Mb	1.71V to 2.0V	Up to 108Mhz	SPI, QSPI	8-Pin SOIC

3. Host Board Support

The Avalanche P-SRAM evaluation board is compatible with the Arduino UNO REV3 host platform. For details of the Arduino UNO REV3 host board go to Arduino UNO REV3.

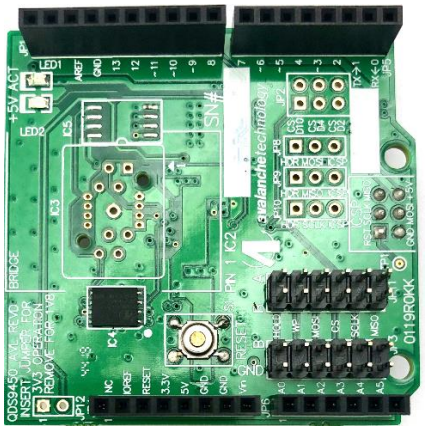


Figure 1: Avalanche P-SRAM Evaluation Board populated with an WSON-8 MRAM device

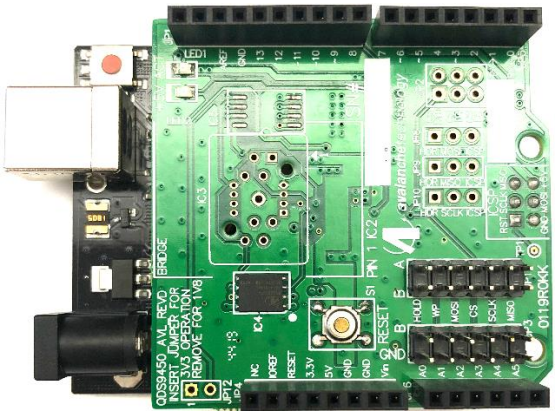


Figure 2: Arduino Uno Host with Avalanche P-SRAM Evaluation Board on Top with WSON-8 MRAM device

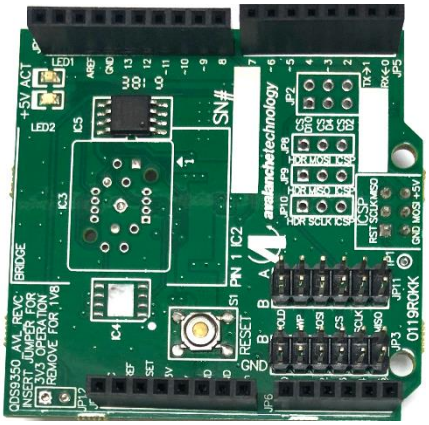


Figure 3: Avalanche P-SRAM Evaluation Board populated with an SOIC-8 MRAM device

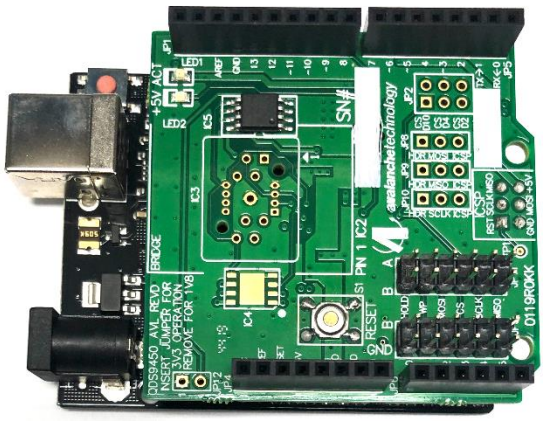


Figure 4: Arduino Uno Host with Avalanche P-SRAM Evaluation Board on Top with SOIC-8 MRAM device

4.1 Requirements

- A PC system with one available USB 2.0/3.0 port
- Windows 7/8/10 with 32/64-bit Operation System
- An Arduino UNO host board R3
- A USB 2.0 cable Type A/B

4.2 Software Installations

1. Downloading the Arduino software IDE (Integrated Development Environment). This software is required to program the Arduino UNO host board.
    - Download the latest version of Windows installer from (https://www.arduino.cc/en/Main)

Figure 5: USB Driver Installation

2. Installing Arduino application software and drivers
    - Click on “Arduino-x.x.x-windows.exe” file to install Arduino application software and USB drivers

4.3 Connecting the Arduino UNO Host Board

Follow the step-by-step instructions in the following order below to configure and connect the Arduino UNO host board to your computer:

1. Attach the Avalanche P-SRAM evaluation board on top of the Arduino UNO host board via the UNO R3 headers (refer to Figures 1 and 2 or Figures 3 and 4 on page 5).

2. Connect the Arduino UNO host board to your computer's USB port using the USB 2.0 cable Type A/B. The green power LEDs on both the Arduino UNO host board and the Avalanche P-SRAM evaluation board should go on.

3. For first time installation, Windows should initiate the USB driver installation process. You can check to ensure the USB drivers have been properly installed by opening Windows Device Manager, and looking under "Ports (COM & LPT)". Windows should assign COM port # to the Arduino UNO host board. If no COM port has been assigned to the Arduino UNO host board, then remove and re-insert USB connector from/into PC's USB port for Windows to re-enumerate the USB port. If that still doesn't work, then you may need to re-install the USB drivers.

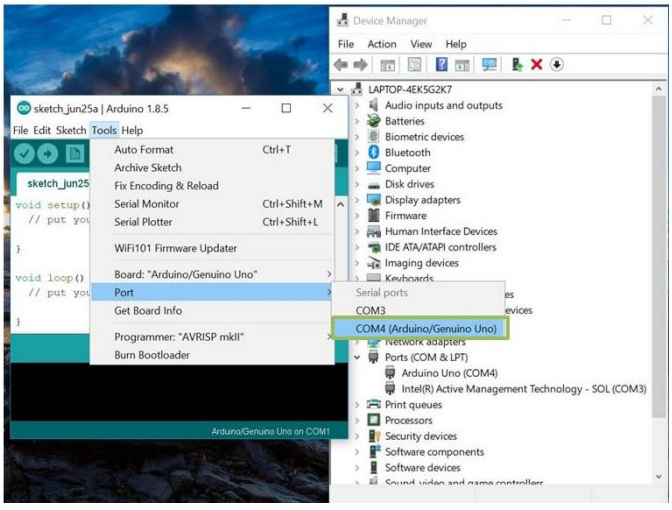


Figure 6: Selecting COM Port for Arduino UNO

4.3.1 Configuring PuTTY

Follow the step-by-step instructions below to configure the PuTTY UART terminal:

1. Double click on Putty icon to open PuTTY Configuration Window.
2. Under “SSH”, select “Serial”. Under “Flow control” pull-down menu, select “None” (refer to Figure 7).
3. Under Category, select “Terminal”, and check “Implicit CR in every LF” checkbox (refer to Figure 8).
4. Select “Session” under “Connection Type”, select “Serial” (refer to Figure 9).
5. In the “Serial line” box, type “COMx” where x is the COM port # that Windows has assigned to Arduino UNO board (refer to Figure 9).
6. In the “Speed” box, type “115200” to set the baud rate (refer to Figure 9).
7. In the “Saved Sessions” box, type “COMx” where x is the COM port # that Windows has assigned to Arduino UNO board (refer to Figure 9).
8. Click “Save” to save the COMx configuration file (refer to Figure 9).
9. Click “Open” to launch PuTTY (refer to Figure 9).
10. The Arduino UNO host board and the Avalanche P-SRAM evaluation board are now up-and-running. The Terminal Monitor window will display a menu of read/write/compare tests for the Avalanche P-SRAM device (refer to Figure 10).

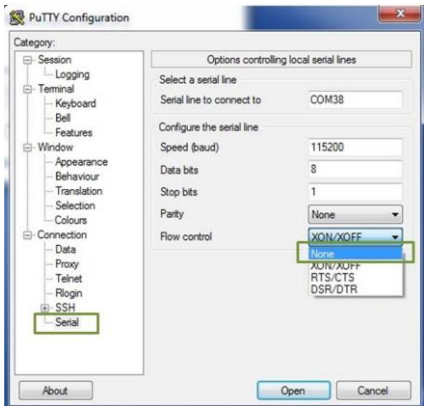


Figure 7: PuTTY Configuration - 1

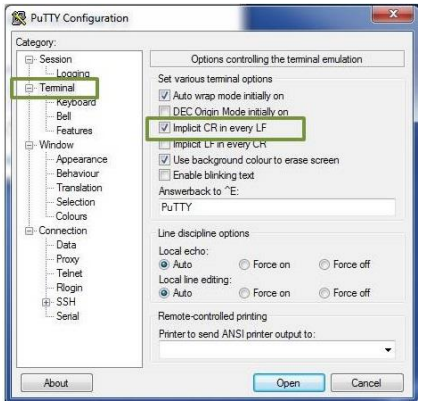


Figure 8: PuTTY Configuration - 2

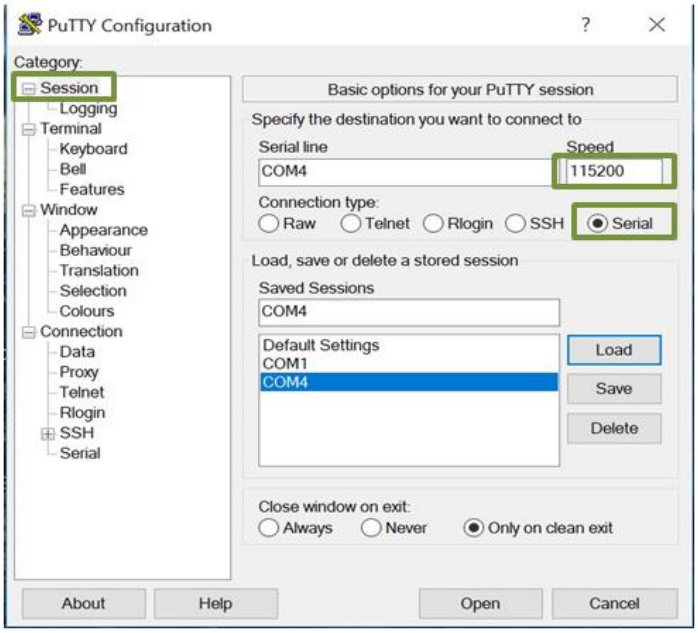


Figure 9: PuTTY Configuration - 3

```
Avalanche Technology STT-MRAM Test Program V4.6.0
Device ID: E6 01 02 02
Avalanche HP, 3V, 85C, 4Mb, 54MHz


Test Menu
a. Read Device ID
b. Read Status Register
c. Write Status Register
d. Read Configuration Registers
e. Write Configuration Registers
f. Read Serial Number Register
g. Write Serial Number Register
h. Read Unique ID Register
i. SDR Read Memory Device
j. SDR Write Memory Device
k. SDR Write-Read-Compare Memory Device
l. SDR Read 256-Byte Special Sector
m. SDR Write-Read-Compare 256-Byte Special Sector
n. Reset Status/Configuration Registers to Factory Default Values
o. Exit Menu

Selection ?
```

Figure 10: Avalanche Test Menu on P-SRAM - PuTTY

4.4 APIs

4.4.1 SPI Write

Purpose: API to write to P-SRAM

Arguments: P-SRAM address (2-byte address; 3-byte address when THREEBYTEADDRESS is enabled)

P-SRAM data: 2-byte data

Parameter: unint32_t (address); unit16_t (write data)

Return: None

Usage: SPI_Write (0x0000, 0x55); Write 2-byte data 0x55 to P-SRAM starting address 0x0000

4.4.2 SPI Read

Purpose: API to read from P-SRAM device

Arguments: P-SRAM address (2-byte address; 3-byte address when THREEBYTEADDRESS is enabled)

P-SRAM data: 2-byte data

Parameter: unint32_t (address)

Return: None

Usage: SPI_Read (0x0000); Read 2-byte data from P-SRAM starting address 0x0000, and assigns to variable 'value'

4.4.3 SPI Burst Write

Purpose: API to burst-write to P-SRAM device

Arguments: P-SRAM address (2-byte address; 3-byte address when THREEBYTEADDRESS is enabled)
SPnvSRAM_wr_data_ptr (data buffer which needs to be written into P-SRAM)
Total_count (total number of words to write)

Parameter: unint32_t (address); unit16_t (write data); unint32_t (total count)

Return: None

Usage: SPI_BurstWrite (0x0000, wr_buf, 16); Write 16 bytes of data from wr_buf to

P-SRAM starting address 0x0000

4.4.4 SPI Burst Read

Purpose: API to burst-read to P-SRAM device

Arguments: P-SRAM address (2-byte address; 3-byte address when THREEBYTEADDRESS is enabled)
SPnvSRAM_rd_data_ptr (data buffer to hold the data being read from P-SRAM)
Total_count (total number of words to write)

Parameter: unint32_t (address); unit8_t (read_data_ptr); unint32_t (total_count)

Return: None

Usage: SPI_BurstWrite (0x0000, rd_buf, 16); Read 16 bytes of data from P-SRAM starting address 0x0000

4.4.5 SPI Status Register Write

Purpose: API to write status register byte

Arguments: 1-byte status register data

Return: None

Usage: SPI_Status_Reg_Write (0x00); Write one byte 0x00 to status register

4.4.6 SPI Status Register Read

Purpose: API to read status register byte

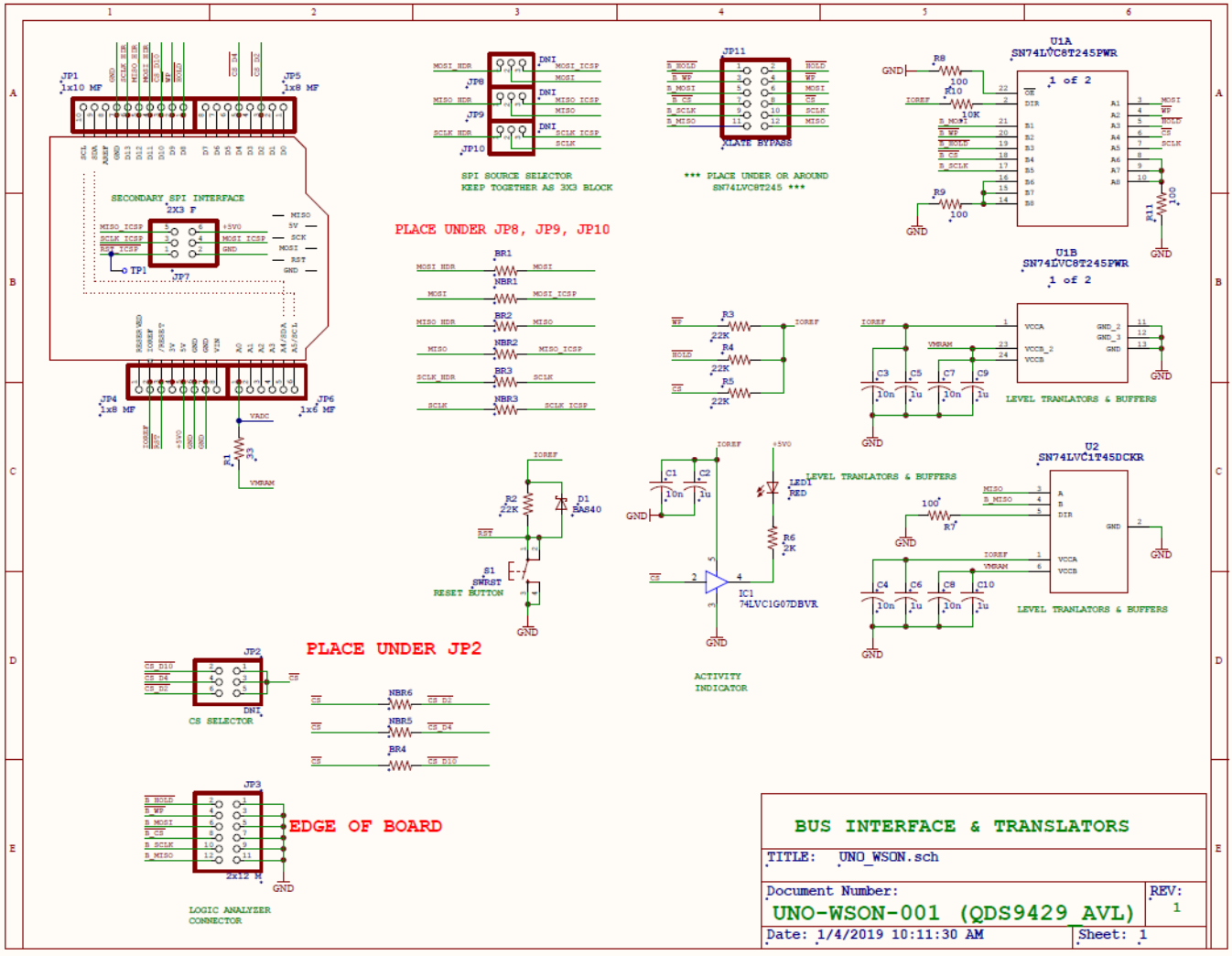
Arguments: 1-byte status register data

Return: None

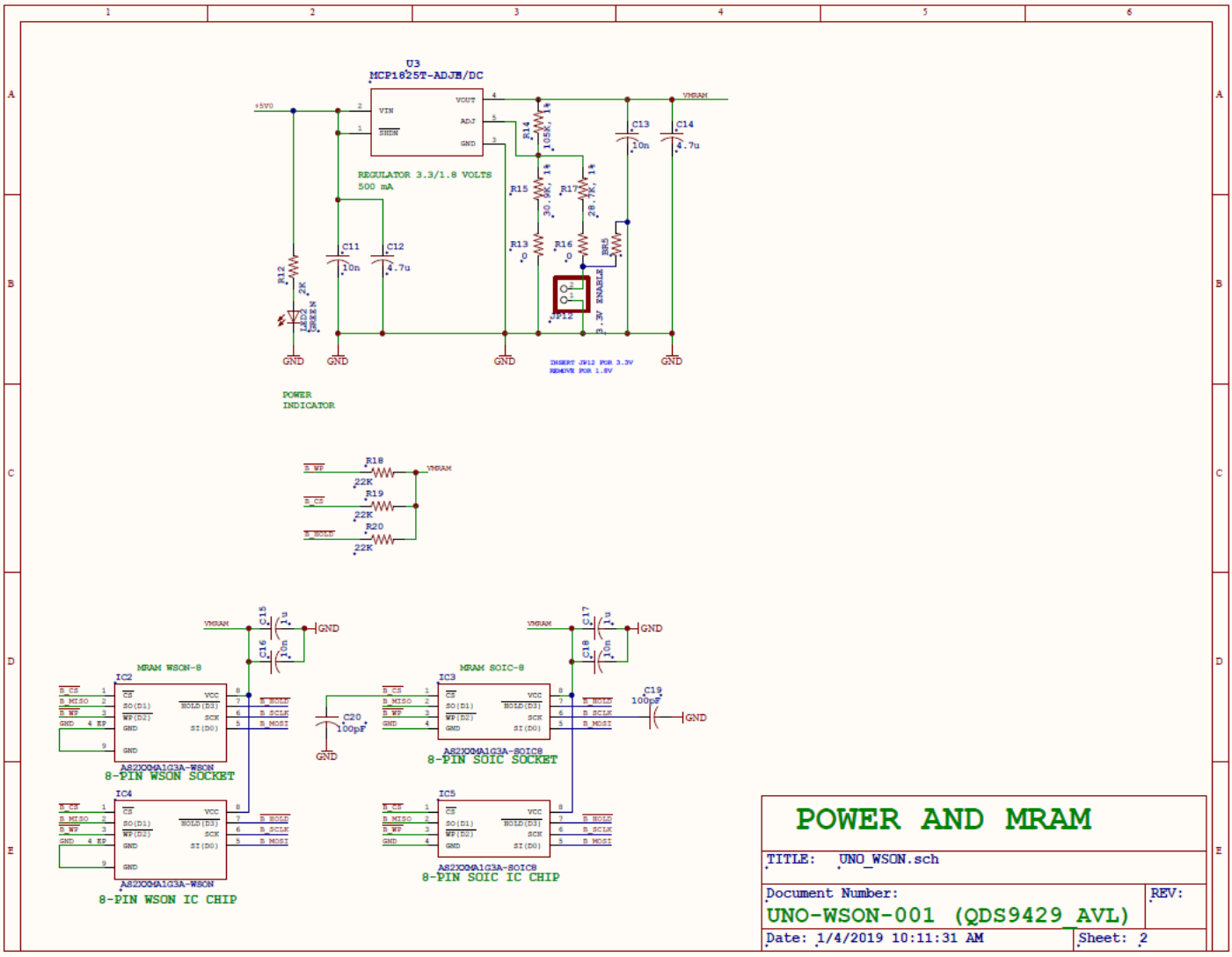
Usage: SPI_Status_Reg_Read (); Read one byte data from status register, and assigns to variable ‘status’

Appendix A

Figure 11: P-SRAM-ARDKIT01 – Schematic (sheet 1)



P-SRAM-ARDKIT01 – Schematic (sheet 2)



Appendix B

Table 2: P-SRAM-ARDKIT01 – BOM

Qty	Value	Part
9	10nF	C1, C3, C4, C7, C8, C11, C13, C16, C18
7	1uF	C2, C5, C6, C9, C10, C15, C17
2	4.7uF	C12, C14
2	100pF	C19, C20
1	BAS40	D1
1	SN74LVC1G07DBVR	IC1
1	ASx016101-0010XxxWAx ASx016204-0054XxxWAx ASx016204-0108XxxWAx	IC2
1	ASx016101-0010XxxSAx ASx016204-0054XxxSAx ASx016204-0108XxxSAx	IC3
1	ASx016101-0010XxxWAx ASx016204-0054XxxWAx ASx016204-0108XxxWAx	IC4
1	ASx016101-0010XxxSAx ASx016204-0054XxxSAx ASx016204-0108XxxSAx	IC5
1	1x10 MF	JP1
1	DNI	JP2
1	2x12 M	JP3
2	1x8 MF	JP4, JP5
1	1x6 MF	JP6

Qty	Value	Part
1	2x3 F	JP7
3	DNI	JP8, JP9, JP10
1	2x6 (XLATE BYPASS)	JP11
1	1x2 (3.3V ENABLE)	JP12
1	RED	LED1
1	GREEN	LED2
1	33	R1
7	22K	R2, R3, R4, R5, R18, R19, R20
2	2K	R6, R12
4	100	R7, R8, R9, R11
1	10K	R10
2	0	R13,R16
1	105K (1%)	R14
1	30.9K (1%)	R15
1	28.7K (1%)	R17
1	SWRST	S1
1	TP	TP1
1	SN74LVC8T245PWR	U1
1	SN74LVC1T45DCKR	U2
1	MCP1825T-ADJE/DC	U3

Qty	Value	Part
4	Bridge	BR1, BR2, BR3, BR4
5	Not Bridge	NBR1, NBR2, NBR3, NBR5, NBR6