TERIDIAN SEMICONDUCTOR CORP.
A Maxim Integrated Products Brand

71M653X
Energy Meter IC

APPLICATION NOTE

AN_653X_005 JANUARY 2009

Operating the 71M653X with External Serial SRAM

Introduction

This Application Note demonstrates methods for extending the internal 4KB XRAM of the 71M653X Energy Metering ICs to a larger size. An external static memory IC may be connected to a 71M653X using its micro-wire interface. The external SRAM can be treated as an extra storage space outside the meter IC. Thus, internal bytes or blocks of bytes can be copied onto anywhere in the extended memory and read back.

This application scheme can be utilized to store demand logs or tables and other applications that require memory space larger than the space provided by the internal MPU RAM.

An even more interesting application is to use SRAM components with shadow RAM capabilities, such as the RAMTRON FM25H20. This type of IC is a commercially available memory that automatically transfers the RAM contents to a non-volatile memory portion on-chip as soon as the power supply is lost. Using such ICs can be beneficial since they require no MPU involvement for the process of saving billing and other data upon loss of the mains voltage.

The entire project with documents, firmware, and hardware data sheets can be found in the file *sram.zip that can be obtained from Teridian*.

System Overview

TSC Energy Meter 71M653X family products support the micro-wire interface. When equipped with an external static memory IC, that supports micro-wire, data transfer between internal memory and external memory is feasible.

A general system configuration for this project is shown in Figure 1:

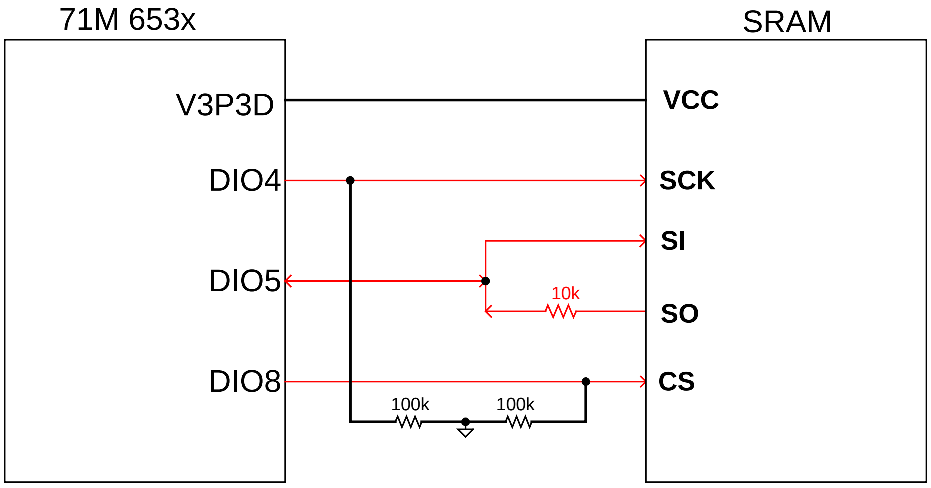

Figure 1 - 653X & External SRAM Micro-Wire Connection

The VCC pin of the SRAM memory chip is connected to V3P3D of the 71M6533 which supplies operating power.

The SCK pin of the SRAM is connected to the DIO4 pin of the 71M6533, where a clock signal is generated upon data transfer. SCK is also connected to ground through a 100kΩ resistor to protect the memory from unexpected power glitches.

The SI and SO pins are sharing the same data port (DIO5), since the micro-wire interface implements only a half-duplex data transfer. There is a 10kΩ resistor between DIO5 and the SO pin. This resistor is used to prevent interference between the SI and SO signals coming in and out the SRAM.

CS is chip select pin that is connected to DIOn (any available DIO on the 71M6533). DIO8 was selected for this function in this firmware demo. It is also connected to ground through a 100kΩ resistor to prevent activation of the SRAM during power transitions.

Hardware Requirements

1. Energy Meter IC 71M6531, 71M6533, or 71M6534 by Teridian Semiconductor Corp (TSC)
2. 256KB Low-power Serial SRAM by AMI Semiconductor (N256S0818HDA/N256S0830HDA) *or* 2Mb Serial 3V F-RAM Memory by RAMTRON (FM25H20)
3. ADM51 ICE for loading code to the meter, by Signum Systems (www.signum.com)

Software Requirements

1. Keil C Compiler Cx51 7.0, Keil Corp
2. Microvision 3, Keil IDE, Keil Corp
3. Wemu51, Debugging Software , Signum Systems

Firmware Location

A demo C code firmware is provided in the zip file, *sram.zip*. The project is created using Microvision IDE 3. It has the following five components:

1. io653x.c: I/O functions and interfaces to TSC power meter chip
2. uwr_lite.c: small and light-weight micro-wire library functions
3. irq.c: interrupt enable/disable/init functions for TSC power meter chip
4. sram.c: special functions and interfaces for external SRAM
5. sramtest.c: main read/write test to external SRAM through micro-wire interface

Note: Each specific memory chip has its own dedicated project folder with the above five components. Please find the right project according to the integrating chip (i.e. N256S0818HDA or FM25H20).

Supporting Firmware

Programmers can use the functions provided in *sram.c* to fully access the external RAM, as listed below:

- *mem_config():* Configures and initializes the necessary components for the external memory to be enabled or disabled. This function is re-written as macros sram_enable() and sram_disable() functions for easy access and simplicity of the code.
- *memr_stat_reg():* Reads the status register byte of the external memory by sending the RDSR byte out.
- *memw_stat_reg():* Writes the status register byte of the external memory by sending the WRSR byte out.
- *memcpy_srx():* Copies a number of bytes from XDATA to SRAM.
- *memcpy_xsr():* Reads a number of bytes from SRAM to XDATA.

There are two tests used to verify the access of external memory:

1) The first test is to write an array with increasing numbers up to BUFFER_SIZE value to the memory and then verify by reading the data back. The test begins with creating and clearing two arrays of numbers: *bf1* stores the original values that are assigned to each array member; *bf2* stores the values that are read back after writing succeeds. Next, it enables the external memory in word mode and assigns values to *bf1*. The entire array is written by using the *memcpy_srx()* function. Then, the program disables the SRAM connection and re-enables it to make sure that there is no interference between writing and reading processes. *memcpy_xsr()* performs the reading process to retrieve all the written bytes back. At the end of the test, the program compares the newly read array *bf2* with the original array *bf1*. If there is any difference, the software will hang.

2) The second test in this demo is similar to the first one; however, the array to be written *bf1* is randomly generated, and reading or writing bytes is set in burst mode instead of word mode.

Using the AMI N256S0818HDA - 256Kb Low-Power Serial SRAM

Project location: *"AMI_N256S0818HDA"* folder

Interface

In this project, a 71M6533 demo board was used in conjunction with a 256KB SRAM (N256S0818HDA). The hardware is configured as shown in Figure 2:

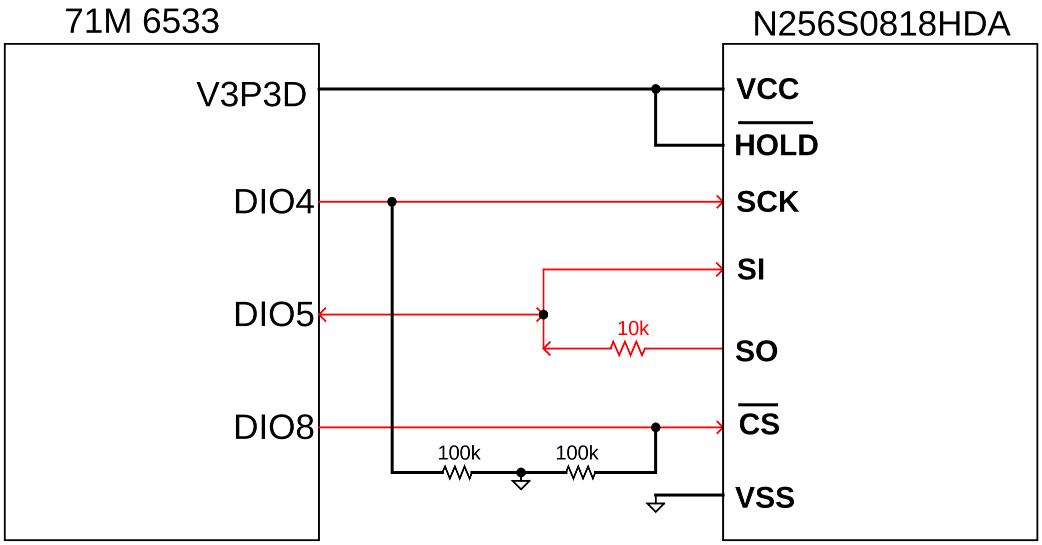

Figure 2: Micro-Wire Connection between 71M6533 and N256S0818HDA

Table 1 shows the description for each pin.

Table 1 - N256S0818HDA Pin Descriptions

Pins	Description
VCC	Connected to the supply voltage (3.3V)
HOLD	Active-low hold function. It's currently not part of this demo code. Please see N256S0818HDA data sheet for more detail
SCK	Feeding clock from the host, connected directly to DIO4
SI	Serial data input, connected directly to DIO5
SO	Serial data output, connected through a 10kΩ resistor
CS	Active-low chip select, controlled by DIO8
VSS	Ground

Data Package Format

Any access to the external SRAM must be in a complete form that includes a command byte, two-byte address and data byte(s). The data package is sent based on MSB to LSB fashion.

The first byte of the package is constructed using 1-byte command op-code. Table 2 provides command op-codes used in this demo code.

Table 2 - N256S0818HDA Command Op-codes

Instruction	Command op-code	Description
READ	`0000 0011b`	Read data from memory starting at selected address
WRITE	`0000 0010b`	Write data to memory starting at selected address
RDSR	`0000 0101b`	Read status register
WRSR	`0000 0001b`	Write status register

Following the first command byte are the two-byte address and the data.

The amount of data is determined based on the transmitting mode selected. The table below shows three different data packages for each mode.

Table 3 – Data Package Format in Different Modes

Mode	Data Package Format (HEX)	Description
Single	MM AA AA DD	Single byte data transfer.
Page	MM AA AA DD …. DD (limited)	Specified size page data transfer
Burst	MM AA AA DD …. DD (infinite)	Infinite data transfer as long as providing clock signal

Note: M – command, A – address, D – data

Using the RAMTRON FM25H20 – 2Mb Serial 3V F-RAM

Project location: *“RAMTRON_FM25H20”* folder

Interface

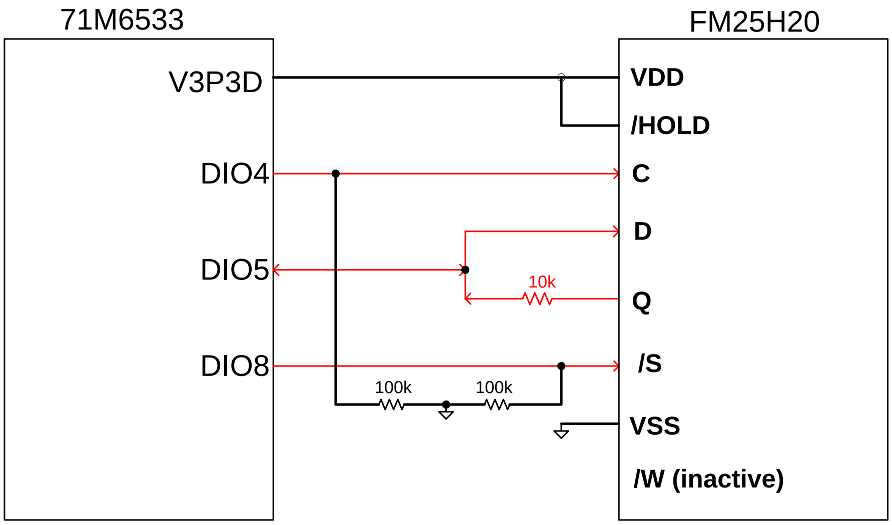

Figure 3- Micro-Wire Connection between 71M6533 and N256FM25H20

Table 4 shows the description for each pin:

Table 4 - FM25H20 Pin Description

Pins	Description
VDD	Connected to the supply voltage (3.3V)
/HOLD	Active-low hold function. It’s currently not part of this demo code. Please see N256FM25H20 data sheet for more detail
C	Feeding clock from the host, connected directly to DIO4
D	Serial data input, connected directly to DIO5
Q	Serial data output, connected through a 10kΩ resistor
/S	Active-low chip select, controlled by DIO8
VSS	Ground
/W	Active-low write protect function. It’s in inactive mode (not connected) in this demo code. Please see N256FM25H20 data sheet for more detail.

Data Package Format

Any access to the external SRAM must be in a complete form that includes a command byte, three-byte address and data byte(s). The data package is sent based on MSB to LSB fashion.

The first byte of data package is constructed using 1-byte command op-code. Table 5 provides command op-codes used in this demo code.

Table 5 - FM25H20 Command Op-codes

Instruction	Command op-code	Description
WREN	`0000 0110b`	Set write enable latch for write access
WRDI	`0000 0100b`	Remove write enable latch to block write access
READ	`0000 0011b`	Read data from memory starting at selected address
WRITE	`0000 0010b`	Write data to memory starting at selected address
RDSR	`0000 0101b`	Read status register
WRSR	`0000 0001b`	Write status register

This FRAM has a large memory space (2Mb) compare to some other memories. Thus, the address section contains three bytes instead of two like others. However, only lower 18 bits are used, and the first higher 6 bits are ignored.

The amount of data in this particular memory chip is not limited per data transfer. Any number of bytes can be transferred each time as long as the host can provide the clock signal.

The overall data package must be similar to the form below (in binary format):

```
mmmm mmmm   xxxx xxaa aaaa aaaa aaaa aaaa   dddd dddd … dddd dddd
<-------->  <------------------------------> <-------------------->
8-bit command        18-bit address              n data bytes
                   first 6 bits ignored     must be multiples of 8-bit
```

Any write access can only be done to this memory by setting the write enable latch (WREN). The status register should report whether or not this bit is set. The WREN must be sent separately right before any WRITE command.

Advanced Techniques for Data Transfer

It can be desirable to make data transfers transparent to the user. This is possible by assigning special memory types to data residing in the external SRAM.

Requirement: Keil C compiler Cx51 7.0 or higher with an extended linker Lx51

Far memory refers to extended address space that can be used for two Cx51 memory types, far and const far, access variables when adding an extended RAM space. Please refer to *Far Memory* section in *Cx51 Compiler User's Guide* for more information.

Caching is a memory technique that speeds up the access of variables located on far memory by mapping some external memory pages onto internal memory.

APPENDIX A – N256S0818HDA Read/Write Time Measurements

6533 CPU speed: 4.9MHz with MPU_DIV = 0

Time to write 1 single byte:

t_{byte} = 40.25 us

Time delay between each byte when writing multiple bytes:

t_{byte_delay} = 19.83 us

Time to send 1 single command (4 bytes: 1 command byte, 2 address bytes, 1 data byte):

t_{cmd} = 4 x (19.83 us + 40.25 us) = 240.32 us

Time delay between each command:

t_{cmd_delay} = 145.2 us

Writing SRAM in WORD mode: every byte sent needs a command byte and two address bytes

To write (n) bytes:

t_W = n x (t_{cmd} + t_{cmd_delay}) = n x (240.32 + 145.2) = 385.52n (us)

Writing SRAM in BURST mode: 1st byte sent with 1-byte command and 2-byte address, all other bytes sent continuously after without any command or address bytes

To write (n) bytes:

t_B = (240.32 + 145.2) + (n-1) x (19.83 + 40.25) = 19.83n + 345.27 (us)

Comparison of WORD and BURST modes:

n = 1: $\frac{t_W}{t_B} = \frac{385.52us}{361.1\ us} = 1.07$

n = 256: $\frac{t_W}{t_B} = \frac{98{,}693.12us}{5{,}421.75\ us} = 18.20$

n = 1024: $\frac{t_W}{t_B} = \frac{394{,}772.48us}{20{,}651.19\ us} = 19.17$

n = ∞: $\lim_{n\to\infty}(\frac{t_W}{t_B}) = \lim_{n\to\infty}(\frac{385.52\text{n}}{19.83\text{n} + 345.27}) = \frac{385.52}{19.83} = 19.44$

Revision History

Revision	Date	Description
Rev. 1.0	09/11/2008	First publication.
Rev. 2.0	01/28/2009	Added description of operation with RAMTRON FM25H20.

Maxim cannot assume responsibility for use of any circuitry other than circuitry entirely embodied in a Maxim product. No circuit patent licenses are implied. Maxim reserves the right to change the circuitry and specifications without notice at any time.

Maxim Integrated Products, 120 San Gabriel Drive, Sunnyvale, CA 94086 408-737-7600