AT91-AN01: Using the Two-wire interface (TWI) in Master Mode on AT91SAM Microcontrollers

AT91 ARM Thumb-based Microcontrollers

Application Note

1. Scope

The purpose of this document is to help the hardware and software developer in the design of a system using the Two-wire Interface (TWI) embedded in the AT91SAM product family of ARM® Thumb®-based microcontrollers with Atmel's Two-wire Interface slave devices and I²C-Bus compatible slave devices. It describes the connection with a digital temperature sensor, a Real Time Clock (RTC), a Graphic LCD Module (GLCD) and a 1Mbit Two-wire Serial EEPROM. In this TWI/I²C-compatible network, only one Master is present, the AT91SAM device, and all other devices are slave devices. This application note also provides a demonstration software project which displays static and animated images, temperature, time and date.

The board used in this application note is the AT91SAM7S256-EK.

- I²C Temperature Sensor: LM75 (National Semiconductor®).
- I²C Real Time Clock: DS1337 (Maxim-Dallas).
- TWI Serial EEPROM: AT24C1024 (Atmel).
- I²C Graphic LCD: Chip-On-Glass (COG) 96 x 40 Dot Matrix LCD module (Batron with NXP PCF8558 LCD controller)

The associated zip file, AN-AT91SAM-TWI-software.zip, contains the source code example described in Section 6.4 "Associated Source Code Example and Projects" on page 26.

2. Associated Documentation

Before going further into this document, refer to the latest documents for the corresponding AT91SAM (AT91SAM7 or AT91SAM9) device on the Atmel Web site. http://www.atmel.com/products/AT91/

3. Terminology, Abbreviations and Typographical Conventions

Abbreviation	Description
TWI	Two-wire Interface
A	Acknowledge
NA	Non Acknowledge
P	Stop
S	Start
Sr	Repeated Start
SADR	Slave Address
ADR	Any address except SADR
R	Read
W	Write

4. Two-wire Interface Overview

The Two-wire Interface (TWI) interconnects components on a unique two-wire bus, made up of one clock line and one data line with speeds of up to 400 Kbits per second, based on a byte-oriented transfer format. It can be used with any Atmel Two-wire Interface bus Serial EEPROM and I²C compatible device such as Real Time Clock (RTC), Dot Matrix/Graphic LCD Controllers and Temperature Sensor, to name but a few. The TWI is programmable as master transmitter or master receiver with sequential or single-byte access. A configurable baud rate generator permits the output data rate to be adapted to a wide range of core clock frequencies. Below, Table 4-1 lists the compatibility level of the Atmel Two-wire Interface and a full I²C compatible device.

Table 4-1. Atmel TWI Compatibility with I²C Standard

I²C Standard	Atmel TWI
Standard Mode Speed (100 KHz)	Supported
Fast Mode Speed (400 KHz)	Supported
7 or 10 bits Slave Addressing	Supported
START BYTE[(1)]	Not Supported
Repeated Start (Sr) Condition	Not Fully Supported[(2)]
ACK and NACK Management	Supported
Slope control and input filtering (Fast mode)	Not Supported
Clock stretching	Supported

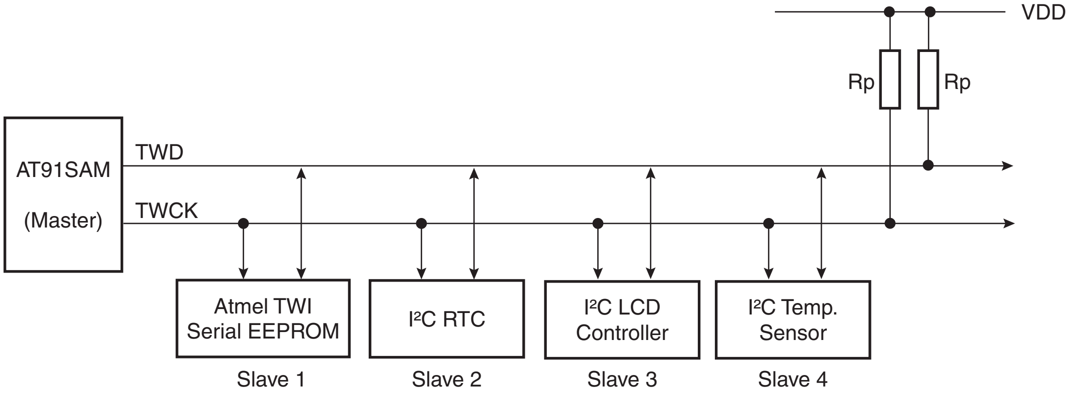
Notes: 1. START + b000000001 + Ack + Sr

2. A repeated start condition is only supported in Master Receiver mode.

5. Hardware Considerations

As shown in Figure 5-1, the Two-wire/I²C Network is made of one Master device, the AT91SAM, and several slave devices. The communication is always between the master and one slave at a time.

Figure 5-1. Application Block Diagram



Rp: Pull up value as given by the I²C Standard

5.1 Bus Capacitance, Pull-up and Rise Time

Due to the architecture of the bus shown below and the bi-directional communication on the data line, the master and the slave devices connected to the bus must have open drain (or open collector) Input/Output buffers. A pull-up resistor on TWD and TWCK is thus needed to drive the clock and data line low. The value of the pull-up resistors is driven by the total capacitance of the bus (input capacitance of each slaves, wire, connectors, PCB layout) and the TWI bit rate. The number of slave devices that can be connected to the bus is limited only by the maximum bus capacitive loading of 400 pF. Figure 5-2 below, shows the Input/output buffers with two slave devices connected to the bus. Since the transition time from low level to high level (rise time) is determined by the RC network formed by Rp resistors and bus capacitance, the Rp value has to be computed to match the rise time required by the I²C Standard, see Table 5-2.

Table 5-1. I/O Lines Description

Pin Name	Pin Description	Type
TWD	Two-wire Serial Data	Open Drain Input/Output
TWCK	Two-wire Serial Clock	Open Drain Output

Table 5-2. TWI/I²C bit rate versus rise time

Bit rate	Rise time (see Figure 5-3)
Standard Mode (100 KHz)	1 µs Max.
Fast Mode (400 KHz)	300 ns Max.

Figure 5-2. Input/Output Buffers

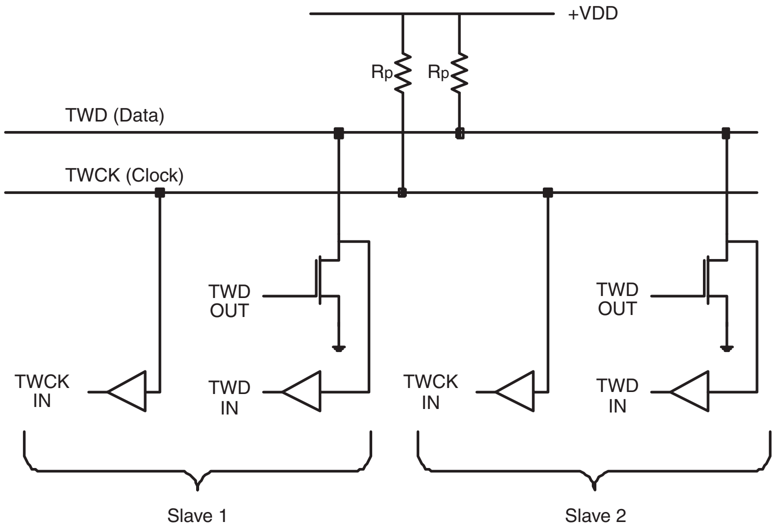
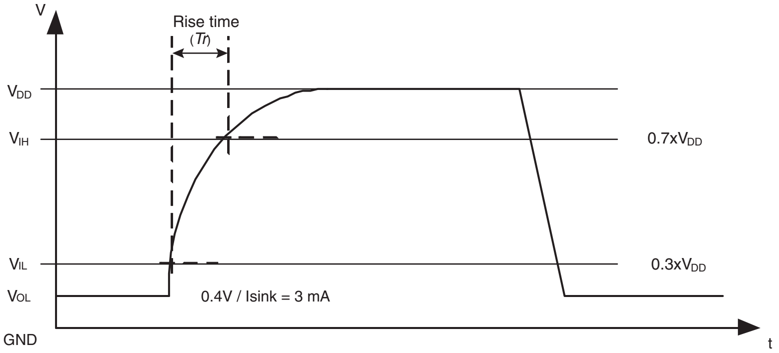


Figure 5-3. Rise Time



The rise time (*Tr*) defined between 30% and 70% of VDD is equal to:

$$Tr = -\tau x \ln\left(\frac{Vdd - Vih}{Vdd - Vil}\right) = 0,847\tau$$

where:

Tr: rise time

τ: *Rp x Cbus (Bus Capacitance)*

Tr = 0.847 x Rp x Cbus

Figure 5-4 and Figure 5-5 below (from the I²C-bus specification), show the Rp pull-up resistor range for both the standard mode and fast mode bit rate.

Figure 5-4. Maximum value of Rp as a function of bus Capacitance for Fast-mode

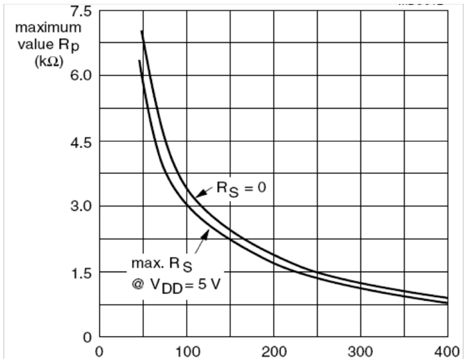
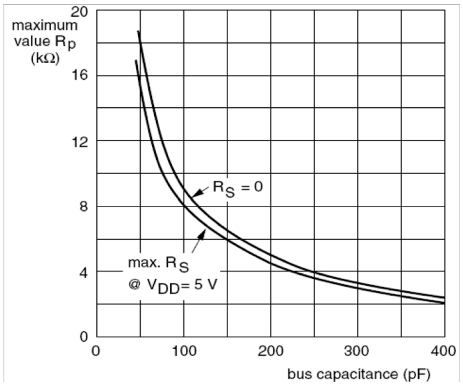


Figure 5-5. Maximum value of Rp as a function of bus Capacitance for Standard-mode



5.2 Mixing 5V and 3.3V Devices

The I²C-bus specification originally focussed on 5V devices. Since the I²C-Bus Specification 2.0, 3.3v devices have been taken into account. Most of the newest TWI/I²C devices (Master or Slave) have their Input/Output buffers powered at 3.3v. Some devices are still using 5V digital logic. Some AT91SAM devices have 5V-tolerant I/Os. This allows combine slave devices operating at 5V and 3.3V on the same bus without using level shifters. Check the corresponding product datasheet of the AT91SAM device. If the AT91SAM device is not 5V tolerant, level shifters must be used.

5.3 Mixing Standard-mode and Fast-mode Devices

Standard-mode (100 kHz) and Fast-mode (400 kHz) devices can be connected on the same bus. But the Fast-mode devices will be clocked at 100 kHz for proper operations. Since the slave address is sent to all devices on the bus, sending a slave address at 400 kHz to 100 kHz devices will not be safe. To work around this, one can use an I²C-bus repeater, such as the NXP PCA9515 or the equivalent, to split the TWI/I²C bus into two buses; Standard Mode bus and Fast Mode bus.

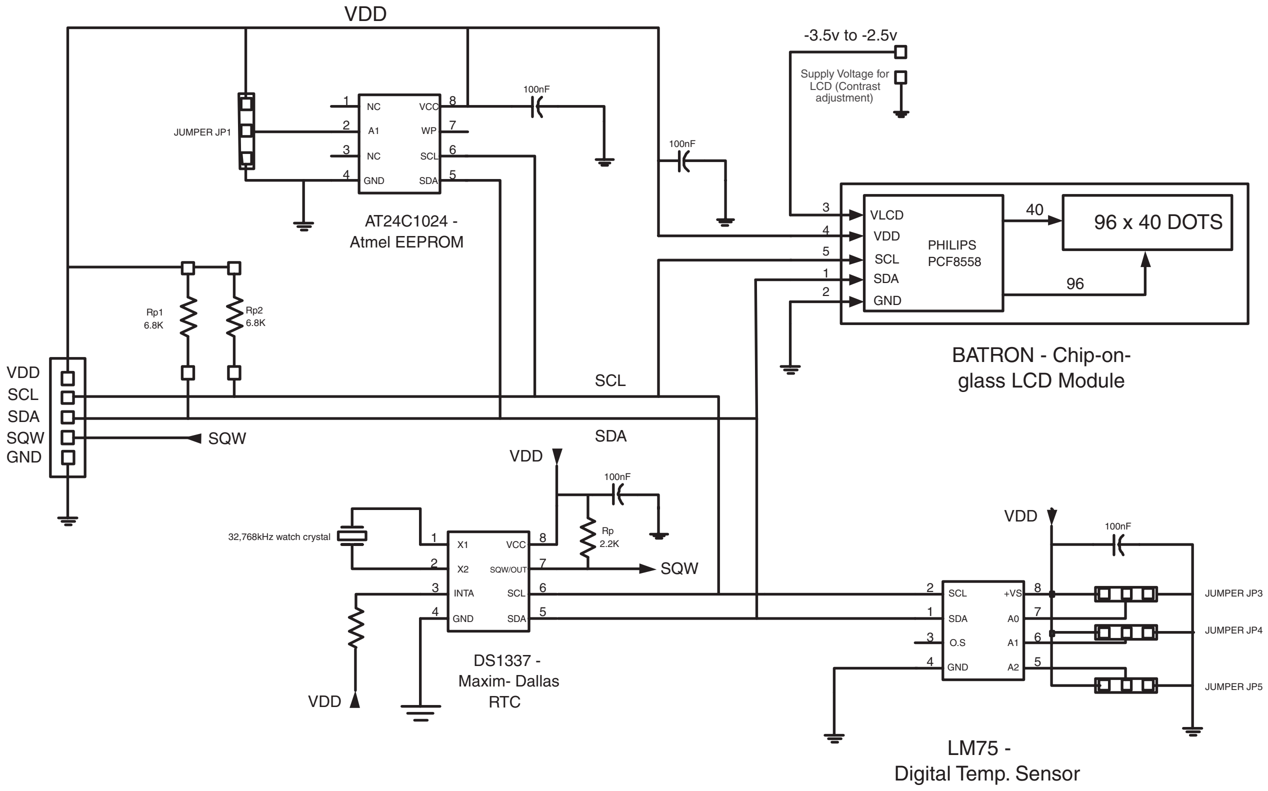
5.4 Communication with Long Cables

A designer might need to build a TWI/I²C network between different boards separated by several meters. (Such as a motherboard of a vending machine or industrial control machine where small LED or Graphic Display and/or keypad that are separated from the main unit and communicating with the main unit via I²C.) Due to the 400 pF bus capacitance limit, using long cables (80 pF/m) rapidly decreases the number of slave devices that can be connected to the bus. To work around this limitation, a bus extender can be used such as the NXP PCA951x, P82B715, P82B96, or the equivalent. Possible distances range from 50 meters at 85 kHz to 1 kilometer at 31 kHz over twisted-pair phone cables. Up to 400 kHz over short distances.

5.5 TWI Network Board Schematic

The schematic given below shows the hardware connection of the slave devices listed in Section 1.

Figure 5-6. TWI Network Board Schematic



5.5.1 Pull-up and Bus Capacitance Computation

The input capacitances given in the table below are the maximum input capacitances per line, i.e. for TWCK (SCL) and TWD (SDA)

Table 5-3. Input Capacitance

Slave Device	Max. Input Capacitance (Cin)
Maxim-Dallas DS1337 RTC	10 pF
NS LM75 Temperature Sensor	20 pF
GLCD Batron (NXP PCF8558)	7 pF
Atmel AT24C1024	8 pF
PCB, Wires, Connectors	5 pF
Total (Cbus):	50 pF

As seen in Figure 5-3 on page 4, the rise time, *Tr,* is equal to 0,847 x Rp x Cbus. For a 400 kHz data rate, Rp equals 7.1K ohms whereas the closest standard value for 5% resistor is 6.8K ohms. All the slave devices used in this application are able to run at this frequency.

5.5.2 Resetting a TWI/I²C Slave Device

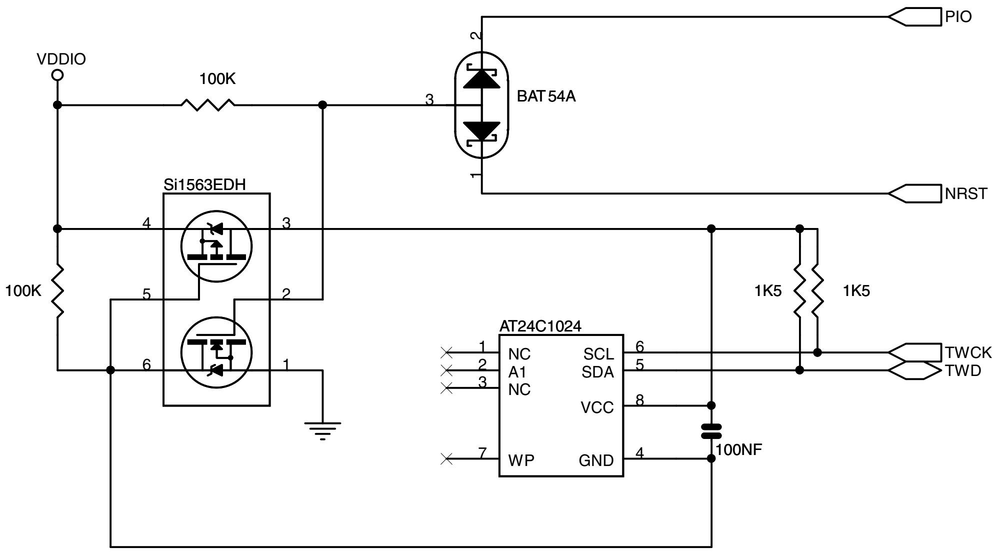
The TWI/I²C slave devices used in this application note do not have a hardware reset input pin. Most of the other slave devices on the market do not have a reset input. In case of a general system reset, failure or the master is interrupted by a higher priority task while transmitting a TWI/I²C frame, the master will end the transfer. The slave device may not finish the transfer due to the fact that the master has stopped the communication. It may happen that the slave holds the TWD/SDA line low and waits for the next clock pulse from the master. In this situation, when the master will resume the TWI/I²C transfer, the master will not be able to start the communication again due to the low state on the data line, which means that the bus is hung up. Nevertheless, solutions exist to recover the communication between master and slave. The first one is purely a software work around. The master must send from one to nine clock pulses at a maximum and monitor the TWD/SDA line to check if the line goes high. If the line goes high, a stop condition must be sent by the master. Since the TWI peripheral cannot resume when its TWD line is kept low, the user will have to reprogram the TWD and TWCK pins into PIO mode to perform the recovery procedure. The procedure described above, called “bus recovery procedure”, is detailed below.

The bus recovery procedure is as follows:

1. Master tries to assert a Logic 1 on the SDA line
2. Master still sees a Logic 0 and then generates a clock pulse on SCL (1-0-1 transition)
3. Master examines SDA. If SDA = 0, go to Step 2; if SDA = 1, go to Step 4
4. Generate a STOP condition

This procedure is a generic one. Check your slave device datasheet as another bus recovery procedure may be available.

Another way to reset a slave device, is to drive the VDD and/or GND pin of a slave device via the main system reset signal driven by a transistor and with a PIO pin of the AT91SAM device to perform a warm reset. A schematic example is shown below for a serial EEPROM.



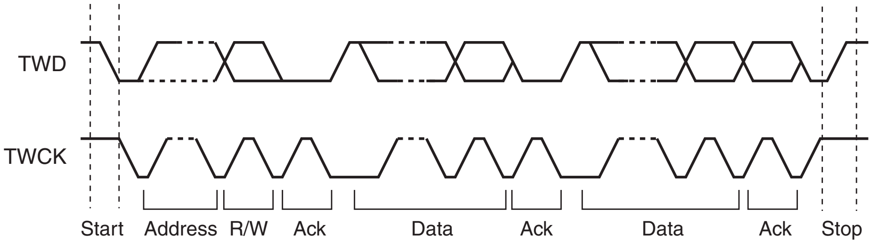
6. Software and Application Considerations

The Transfer Format and Modes of Operation sections below give a brief reminder of the TWI operation. For further details and register definitions refer to the corresponding AT91SAM7 or AT91SAM9 datasheet.

6.1 Transfer Format

The data put on the TWD line must be 8 bits long. Data is transferred MSB first; each byte must be followed by an acknowledgement. The number of bytes per transfer is unlimited (see Figure 6-1).

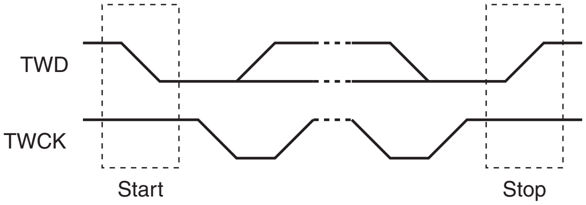
Figure 6-1. Transfer Format



Each transfer begins with a START condition and terminates with a STOP condition (see Figure 6-2).

- A high-to-low transition on the TWD line while TWCK is high defines the START condition.
- A low-to-high transition on the TWD line while TWCK is high defines a STOP condition.

Figure 6-2. START and STOP Conditions



6.2 Modes of Operation

The TWI has two modes of operation:

- Master Transmitter Mode: Master Writes Data into the Slave
- Master Receiver Mode: Master Reads Data from the Slave

For both modes, the TWI is able to perform single or multiple (sequential) read/write operations. Depending on the slave device, single read or write accesses will be used. This is the case for the LM75 temperature sensor where most of the time, only the temperature register will be read. For slave devices such as the graphic LCD and the serial EEPROM, sequential read and write accesses will be done. The RTC used in this application note, is also accessed sequentially to read the second, minute, hour, day registers and so on. Each of the devices having sequential accesses for their registers or memory locations, can of course be accessed in single access mode, but this will increase the overall access time. This is particularly true when accessing a memory page in the serial EEPROM.

6.2.1 Internal Address

When addressing slave devices, the internal address bytes are used to perform random address (read or write) accesses to reach one or more data bytes from an offset location, within a memory page location in a serial memory, for example. When performing read operations with an internal address, the TWI performs a write operation to set the internal address in the slave device, and then switch to Master Receiver mode. Note that the second start condition (after sending the IADR) is sometimes called "repeated start" (Sr) in I²C fully-compatible devices. See Figure 6-3 and Figure 6-4 below.

Figure 6-3. Single Data Byte Write with Internal Address

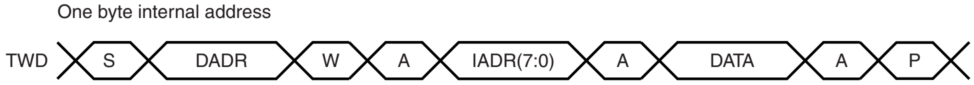
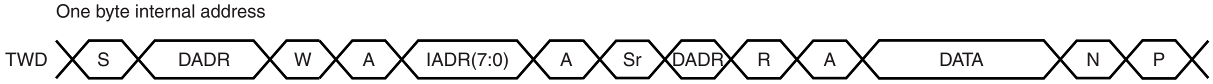


Figure 6-4. Single Data Byte Read with Internal Address



6.2.1.1 TWI Operations

To summarize, the TWI can perform the following operations.

- Master Transmitter Mode (Master writes to the Slave)
  - Single Data byte Write
  - Single Data byte Write with Internal Address (Slave register offset)
  - Multiple Data byte Write
  - Multiple Data byte Write with Internal Address (Slave register offset)
- Master Receiver Mode (Master reads from the Slave)
  - -Single Data byte Read
  - Single Data byte Read with Internal Address (Slave register offset)
  - Multiple Data byte Read
  - Multiple Data byte Read with Internal Address (Slave register offset)

6.3 Slave devices

6.3.1 National LM75 Digital Temperature Sensor

The LM75 is a temperature sensor with an accuracy of ±2° C(max) for -25° C to 100° C temperature range, and ±3° C(max) for -55° C to 125° C range. It has an open-drain Over-temperature output which will be active when the temperature exceeds a programmable limit configured via the Temperature Output Shutdown (TOS) Register. This is not used in this application note. The LM75 has four internal (hardwired) slave address bits and three external (user defined) slave address bits. Up to eight sensors can be connected on the bus.

Slave address format: 1-0-0-1 + A2-A1-A0 (A2 to A0 are user defined bits).

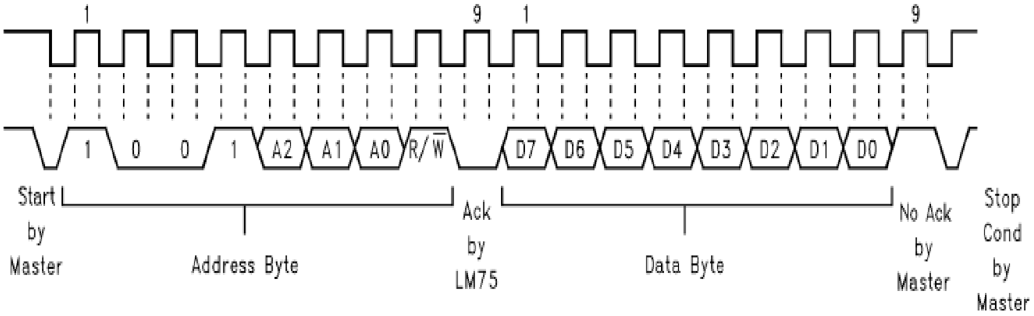
In this application note, the following address is used: "1-0-0-1" + A0=A1=A2= +VDD = 0x4F. The LM75 has four data registers selectable by a 5th register, the pointer register. This means that each time one of the four data registers need to be read or written, the pointer register must be set beforehand.

- Pointer Register: 8-bit (only 2 LSB's are significant)
- Configuration Register: 8-bit
- Temperature Register: 16-bit (only 9 MSBs are significant)
- Temperature Output Shutdown Register: 16-bit (only 9 MSBs are significant)
- Temperature Hysteresis Register: 16-bit (only 9 MSBs are significant)

6.3.1.1 8-bit Register Access

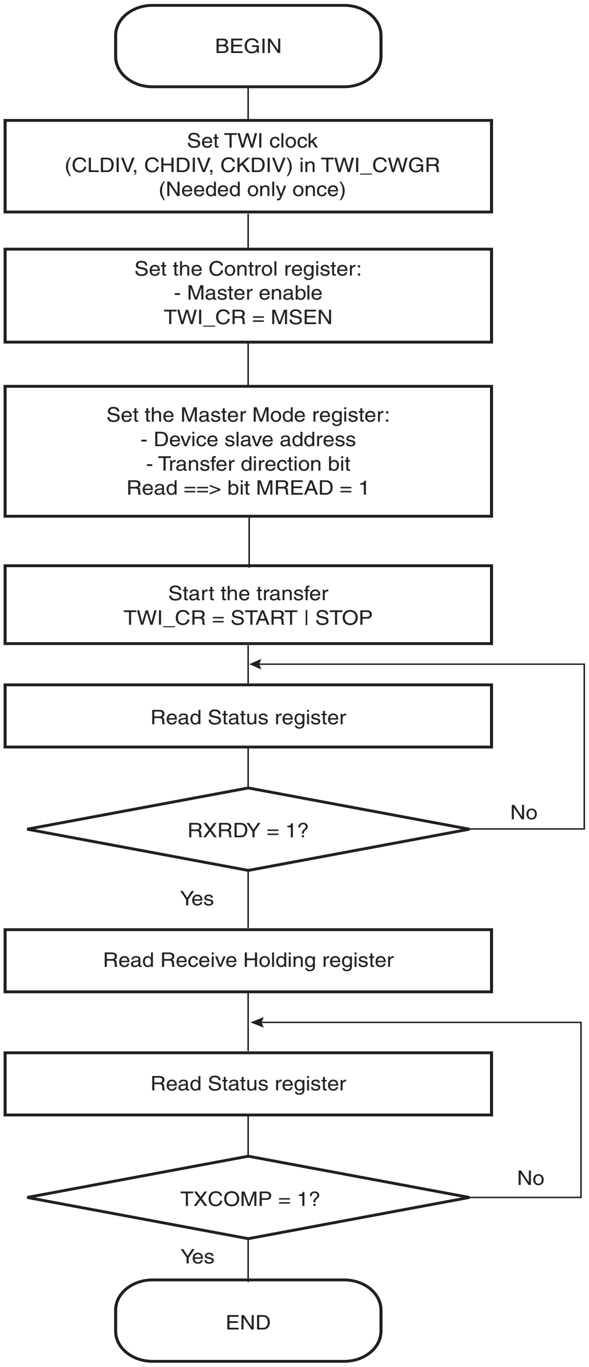
To access the 8-bit registers, a single byte read or write access is needed if the pointer register is already set to the correct register to access. Figure 6-5 shows the transfer format when reading from the configuration register when the pointer has already been set.

Figure 6-5. 1-byte Configuration Register Read with Pointer Register Pre-set



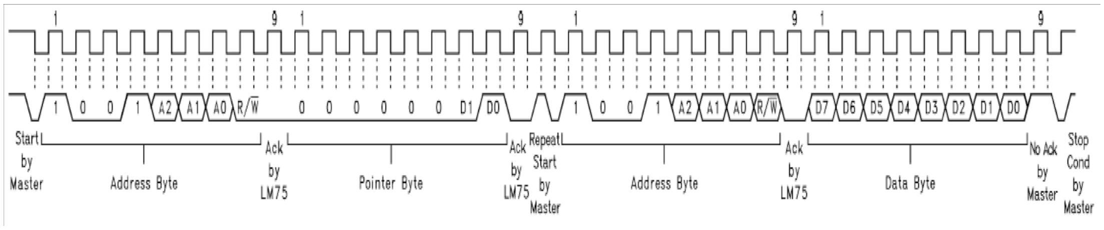
The TWI must be programmed as shown in Figure 6-6 on page 11.

Figure 6-6. Single Data Byte Read



If the pointer register is not pre-set to the correct register to access, an extra “Pointer Byte” must be sent before writing to or reading from the register during the same access. Figure 6-7 shows the transfer format when reading from the Configuration register when the pointer has not already been set. This can be done by using the Internal Address register (TWI_IADR) of the TWI. The value of the Pointer Register will be the value in TWI_IADR.

Figure 6-7. 1-byte Configuration Register Read with Pointer Byte Not Pre-set



In this case, the TWI must perform a single data byte read with use of the internal address register (TWI_IADR). The TWI will have to be programmed as shown in Figure 6-8 on page 13 for reading and Figure 6-9 on page 14 for writing.

Figure 6-8. Single Data Byte Read with Internal Address

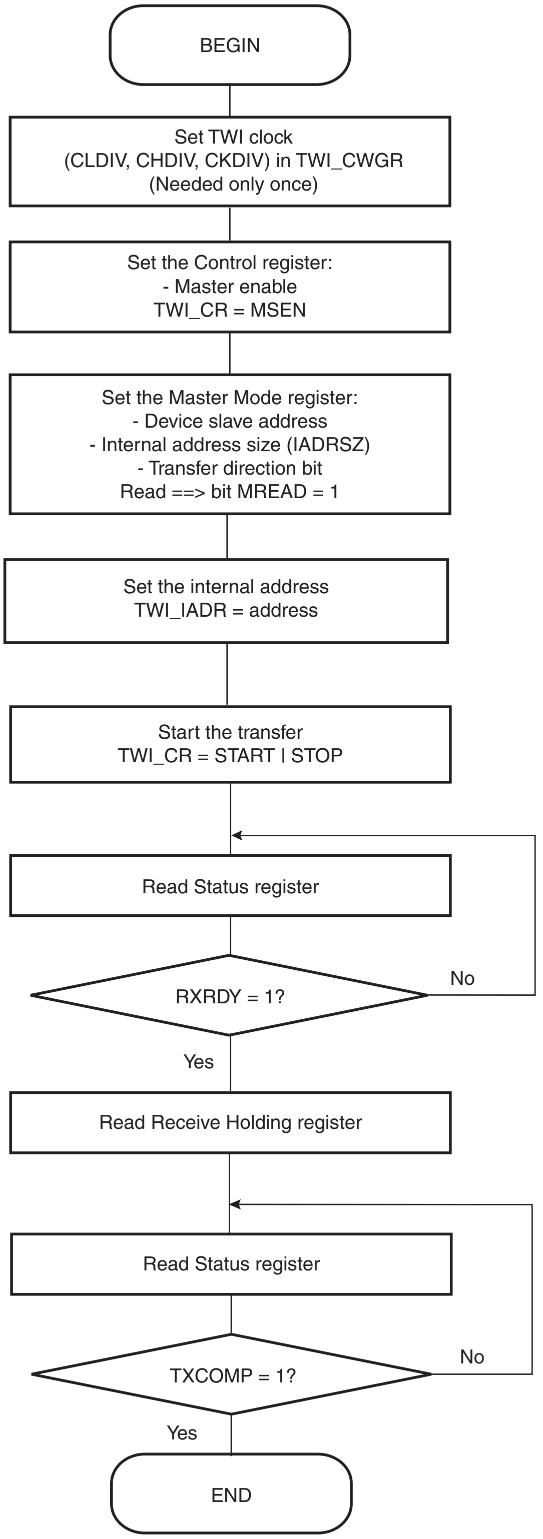
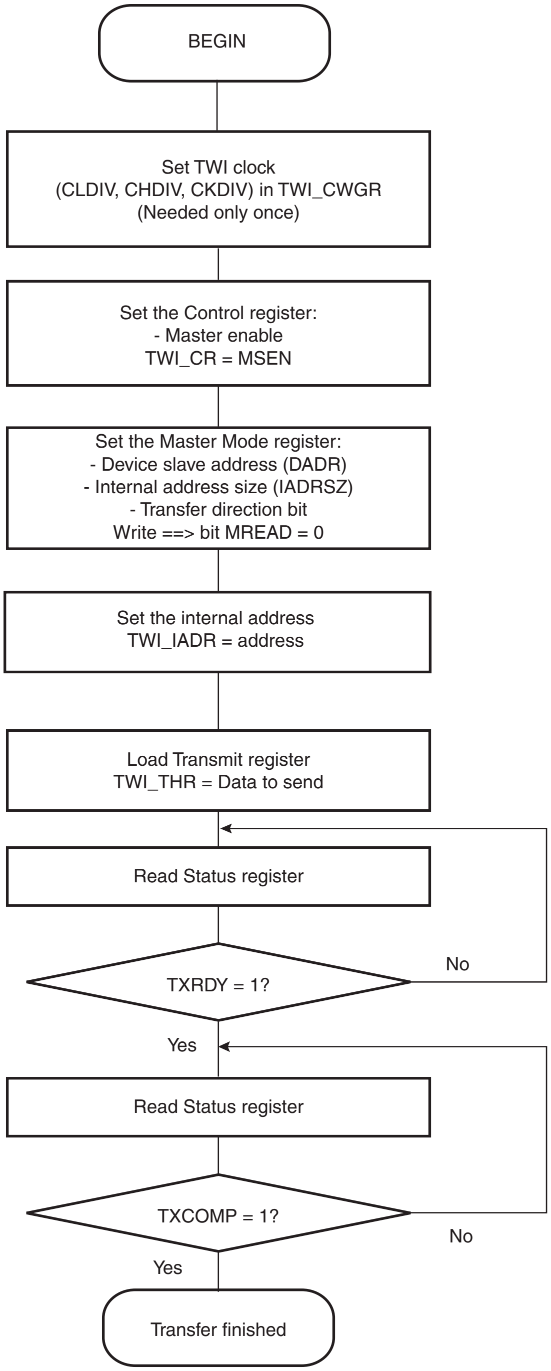


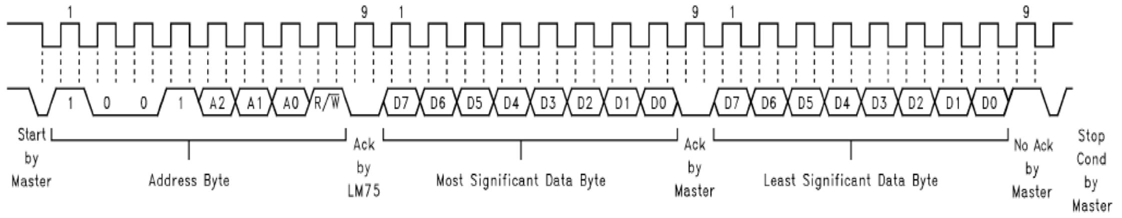
Figure 6-9. Single Data Byte Write with Internal Address



6.3.1.2 *16-bit Register Access*

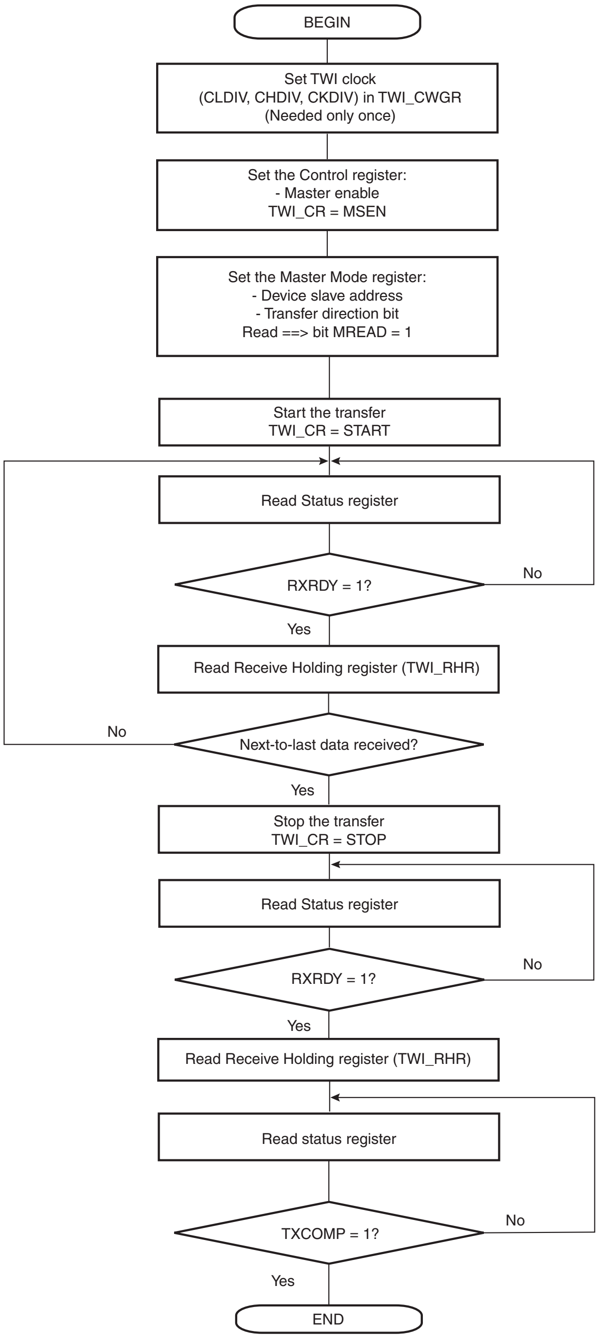
To access the 16-bit registers, a multiple byte read or write access is needed if the pointer register is already set to the correct register to access. Figure 6-10 shows the transfer format when reading from the temperature register (16-bit) when the pointer has already been set.

Figure 6-10. 2-byte Temperature Register Read with Pointer Register Pre-set



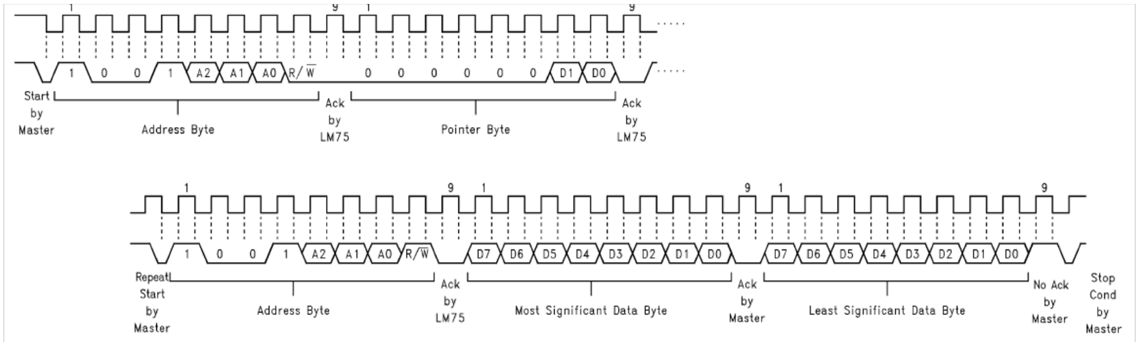
In this case, the TWI must be programmed as shown in Figure 6-11 on page 16.

Figure 6-11. Multiple Data Byte Read



As in 8-bit register accesses, if the pointer register is not already set to the correct register to access, the “Pointer Byte” of the LM75 must be sent before writing to or reading from the register during the same access. Figure 6-12 shows the transfer format when reading from the hysteresis or output shutdown temperature register when the pointer has not already been set. This can be done by using the Internal Address register (TWI_IADR) of the TWI. The value of the Pointer Register will be the value in TWI_IADR.

Figure 6-12. 2-byte Temperature Hysteresis/Tos Register Read with Pointer Register not Pre-set



In this case, the TWI must be programmed as shown in Figure 6-13 for reading and Figure 6-14 on page 19 for writing.

Figure 6-13. Multiple Data Byte Read with Internal Address

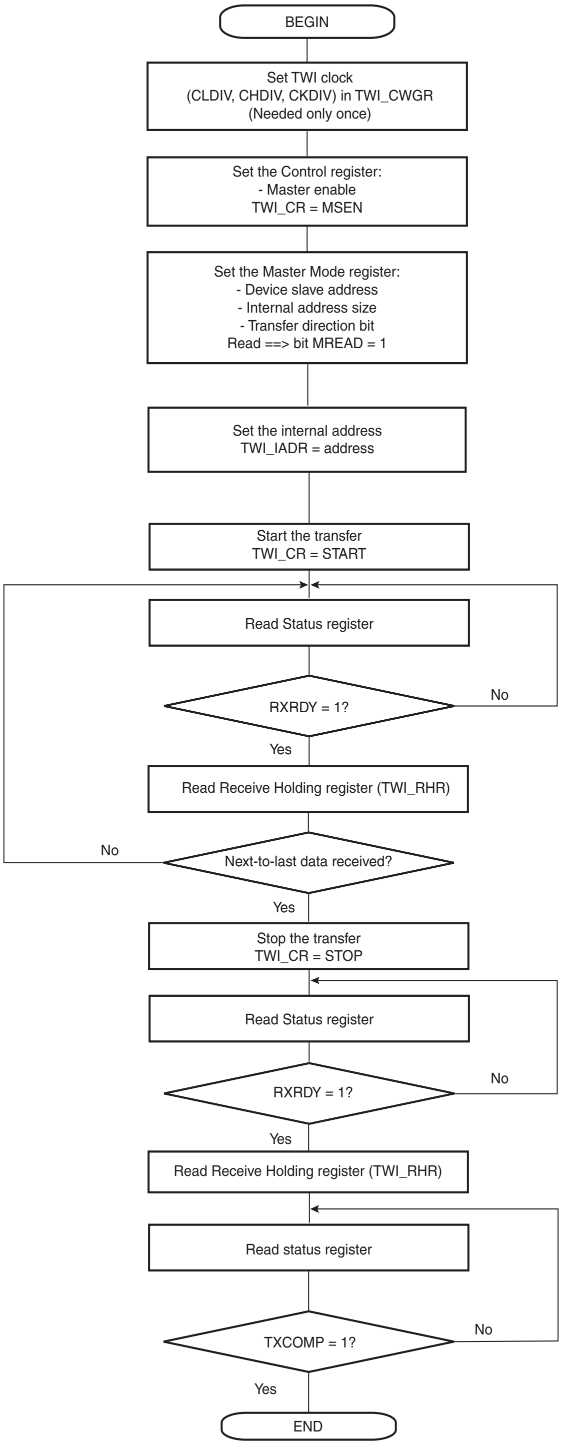
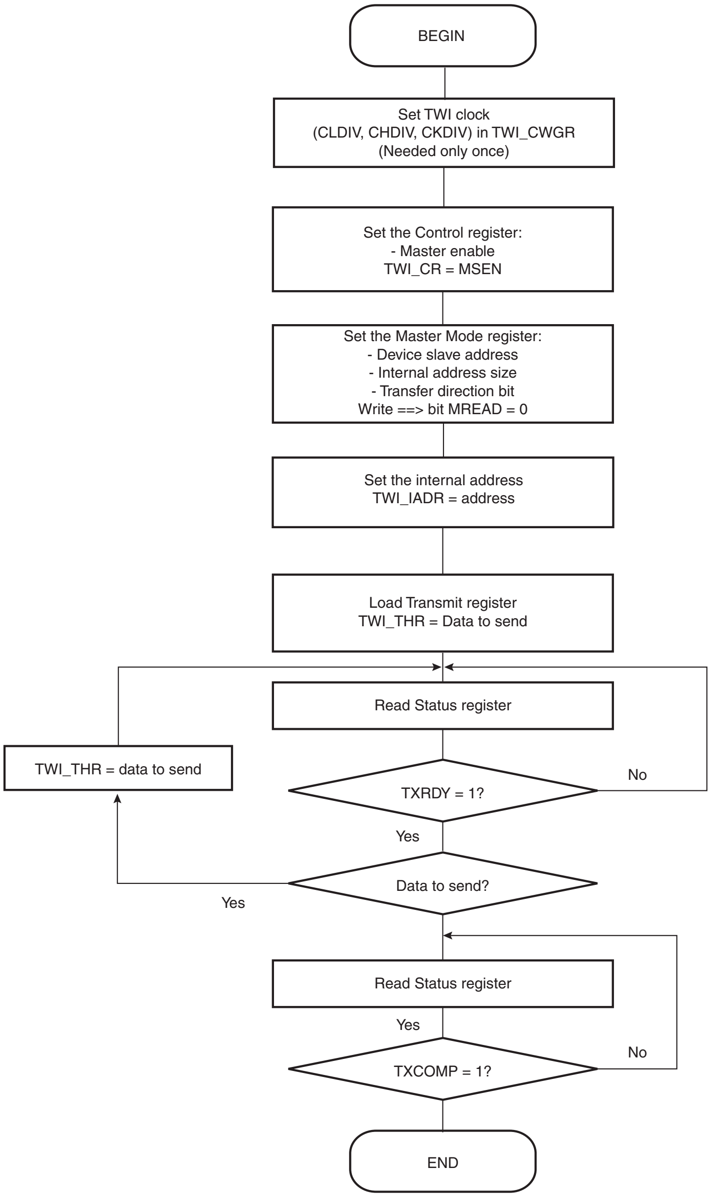


Figure 6-14. Multiple Data Byte Write with Internal Address



6.3.1.3 *Source Code Example*

The following C functions are provided with this application note. (File: *lm75.c* and *lm75.h)*

```
AT91F_LM75_ReadConfigurationRegister
AT91F_LM75_SetConfigurationRegister
AT91F_LM75_ReadThystRegister
AT91F_LM75_SetThysRegister
AT91F_LM75_ReadTosRegister
AT91F_LM75_SetTosRegister
AT91F_LM75_ReadTempRegister
```

How to use these functions is shown in source code *GLCD-TempSensor.c* and in *All-in-one-Demo.c*

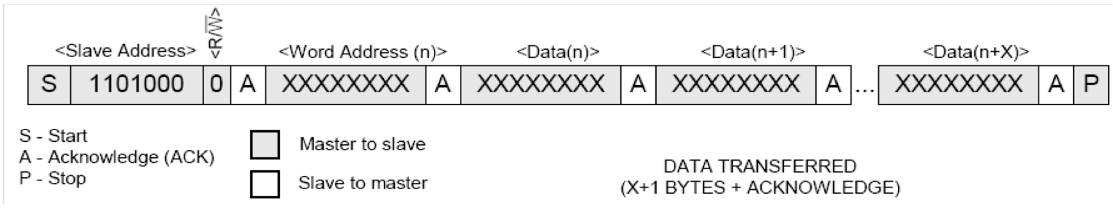
6.3.2 Maxim-Dallas DS1337 Real Time Clock

The DS1337 Real-Time Clock (RTC) provides clock and calendar information with seconds, minutes, hours, day, date, month, and year information. It also features two time-of-the day alarms. The DS1337 has an open-drain, programmable Square-Wave Output signal (SQW). This signal is used in this TWI Network Demo as 1Hz tick time for reading clock and calendar information each seconds. The clock operates in either the 24-hour or 12-hour format with AM/PM indicator. The DS1337 has sixteen registers. The first seven are for second, minute, hour, day, date, month, and year information and the remaining registers for alarm settings, control and status. Each register is 8 bits wide. Before accessing one of the registers, the register pointer (noted as "Word Address" in the figures below) must be set to the right register offset, i.e. seconds or minutes or day, etc. The DS1337's slave address is b110 1000 (0x68).

6.3.2.1 Data Write - Slave Receiver Mode

This mode is used when writing one or several registers. A word address (pointer register offset) must be sent before sending the value to write in the corresponding register(s). Each time a register is written, the pointer register offset is automatically incremented.

Figure 6-15. Data Write - Slave Receiver Mode

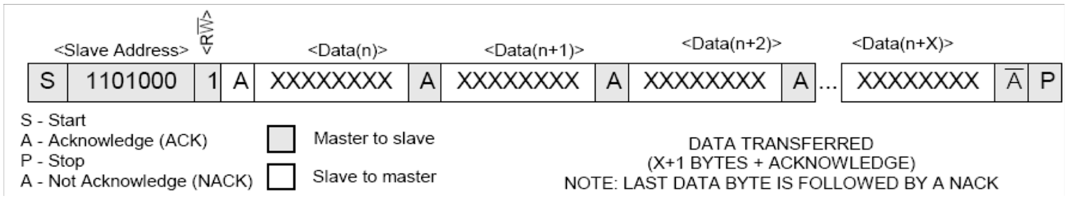


To write one or more registers, the TWI must be used in single or multiple data byte write operation with one byte internal address as shown in Figure 6-9 on page 14 for single data and Figure 6-14 on page 19 for multiple data.

6.3.2.2 Data Read - (from Current Pointer Location) - Slave Transmitter Mode

This mode is the same as Data Write mode, but with no word address to program in the DS1337. If only one register needs to be read, the TWI must be programmed as shown in Figure 6-6 on page 11. If Several registers must be read during the same access, the TWI must be programmed as shown in Figure 6-11 on page 16.

Figure 6-16. Data read - (from Current Pointer Location) - Slave Transmitter Mode

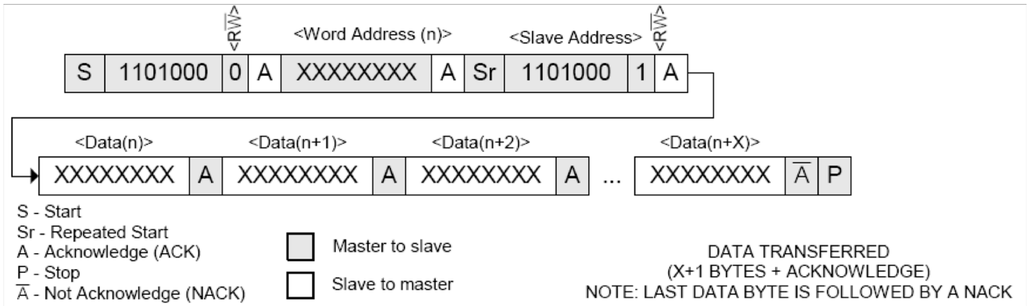


6.3.2.3 Data Read (Write Pointer, then Read) - Slave Receive and Transmit

In this mode, the Word Address (Pointer) is not pre-set to point to the register to be accessed. A write into the pointer register must be done before reading. For this, performing a read access (Single or Multiple Data Byte) with internal address set, the TWI will first execute a write access, to set the pointer register and then will start reading from the slave. In the Figure 6-17 below, the

Sr (repeated start) is automatically handled by the TWI. To program the TWI in Single or Multiple data byte read with internal address, refer to Figure 6-8 on page 13 and Figure 6-13 on page 18.

Figure 6-17. Data Read (Write Pointer, then Read) - Slave Receive and Transmit



6.3.2.4 *Source Code Example*

The following C functions are provided with this application note. (File: *ds1337.c* and *ds1337.h)*

```
AT91F_DS1337_SetSeconds
AT91F_DS1337_ReadSeconds
AT91F_DS1337_SetMinutes
AT91F_DS1337_ReadMinutes
AT91F_DS1337_SetHours
AT91F_DS1337_ReadHours
AT91F_DS1337_SetDay
AT91F_DS1337_ReadDay
AT91F_DS1337_SetDate
AT91F_DS1337_ReadDate
AT91F_DS1337_SetMonth
AT91F_DS1337_ReadMonth
AT91F_DS1337_SetYear
AT91F_DS1337_ReadYear
AT91F_DS1337_SetDateTime
AT91F_DS1337_ReadDateTime
AT91F_DS1337_SetControlRegister
```

How to use these functions is shown in source code *GLCD-RTC-DS1337.c* and in *All-in-one-Demo.c*

6.3.3 AT24C1024 TWI Serial EEPROM

The AT24C1024 provides 1,048,576 bits of serial electrically erasable and programmable read only memory (EEPROM) organized as 131,072 words of 8 bits each. The 1024K is internally organized as 512 pages of 256 bytes each. Random word addressing requires a 17-bit data word address. The AT24C1024 has five fixed internal address bits plus one external user defined bit, A1, and one memory page address bit, P0.

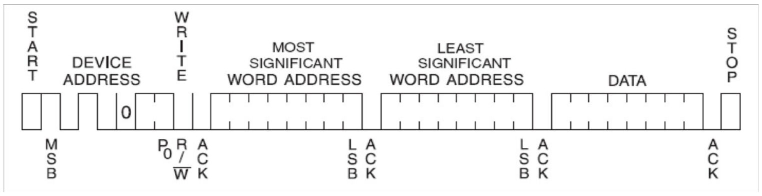
Slave address format: 1-0-1-0-0 + A1 + P0

This memory page address bit, P0, is the most significant bit of the data word address as shown in Figure 6-18. The slave address used above in this example with an AT24C1024 is A1 = 1 (+VDD). The AT24C1024 features two write operations and three read operations as described in the sections that follow.

6.3.3.1 *Byte Write Operation*

The byte write operation permits writing a single data at any address. Before sending the data, a word address must be sent to the EEPROM as shown in Figure 6-18 below. The TWI has to be programmed as described in Figure 6-9 on page 14 with two internal address bytes.

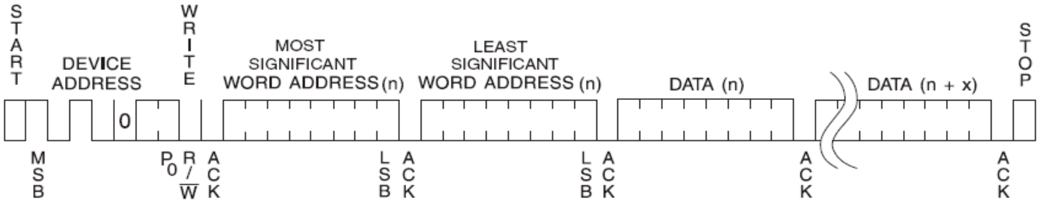
Figure 6-18. Byte Write



6.3.3.2 *Page Write Operation*

The page write operation is almost exactly the same as a byte write operation, but it must be used in multiple data byte write operations, to transmit up to 256 data words (Page Size) without sending a stop condition between each byte. In this operation the EEPROM automatically increments its internal word address counter and roll-over after the 256th byte received from the master. The word address can be any page address. The TWI must be used in multiple data byte write operation as described in Figure 6-9 on page 14.

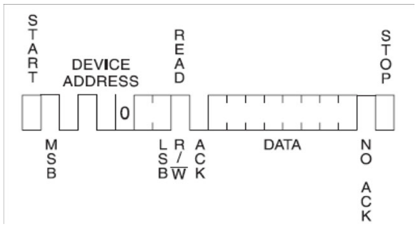
Figure 6-19. Page Write Operation



6.3.3.3 *Current Address Read Operation*

For this read operation, the TWI must be programmed as shown in Figure 6-6 on page 11. Please note that in this operation, the internal word address counter of the EEPROM is incremented by one and will roll over after the last byte of the last memory page.

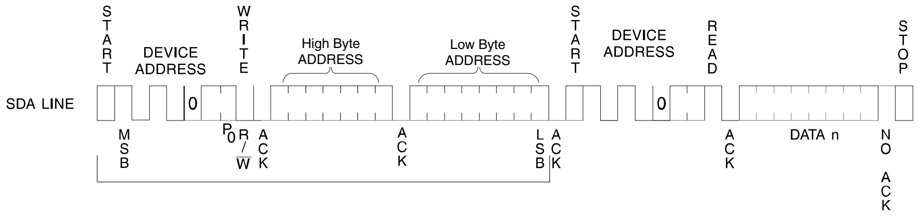
Figure 6-20. Current Address Read Operation



6.3.3.4 *Random Read Operation*

The random read operation is similar to the current address read operation, but with word address setting in the same read operation. It can be needed when reading one data from different memory pages. The random read operation is shown in Figure 6-21 below.

Figure 6-21. Random Read Operation

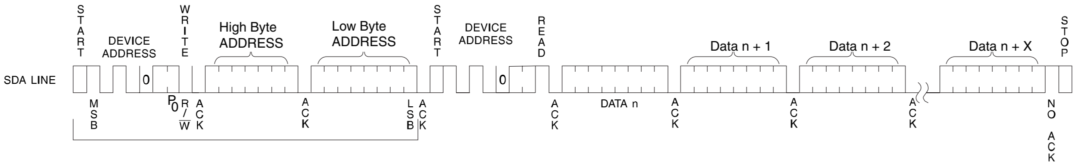


As shown in this waveform, a write operation is needed for word address setting. This is automatically done when using the TWI's internal address register (TWI_IADR). The Second start condition (after low byte address) is also called "Repeated Start". For programming sequence, see Figure 6-8 on page 13.

6.3.3.5 *Sequential Read Operation*

Sequential read operation is similar to the random read operation, but it allows reading the whole EEPROM when the word address is zero, or from any page number up to the end of the memory, without stopping the communication with the master.

Figure 6-22. Sequential Read Operation



As shown in this waveform, a write operation is needed for word address setting. This is automatically done when using the TWI's internal address register (TWI_IADR). The Second start condition (after low byte address) is also called "Repeated Start". For programming sequence, see Figure 6-13 on page 18.

6.3.3.6 *Source Code Example*

The following C functions are provided with this application note. (File: *at24c1024.c* and *at24c1024.h)*

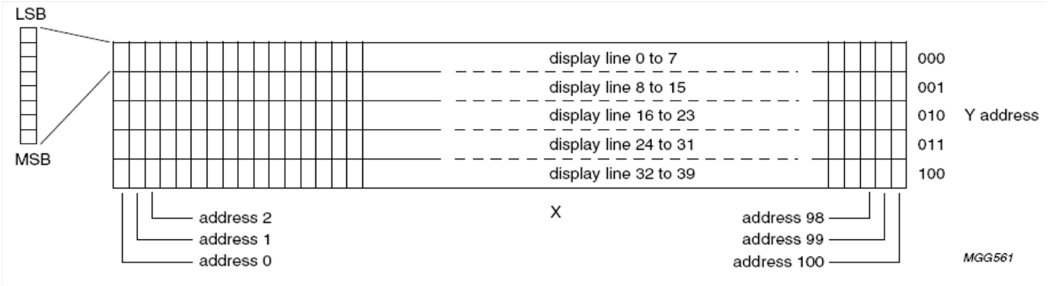
```
AT91F_AT24C_WriteByte
AT91F_AT24C_WritePage
AT91F_AT24C_ReadByte
AT91F_AT24C_ReadPage
AT91F_AT24C_SequentialRead
AT91F_AT24C_MemoryReset
```

How to use these functions is shown in source code; *EEPROM-TEST.c* and in *All-in-one-Demo.c*

6.3.4 Graphic LCD Module (PCF8558 Controller)

The BT96040 is a STN dot Matrix Graphic LCD module with 96 columns and 40 rows. The GLCD module is based on the NXP PCF8558 Universal LCD driver for small graphic panels. It is internally organized as five pages of 96 x 8 dots each. Each page is independently addressable, so the user can write characters, clear a page, clear a pixel and so on without modifying the other pages. The PCF8558 embeds a Display RAM organized as shown in Figure 6-23.

Figure 6-23. Display RAM Matrix



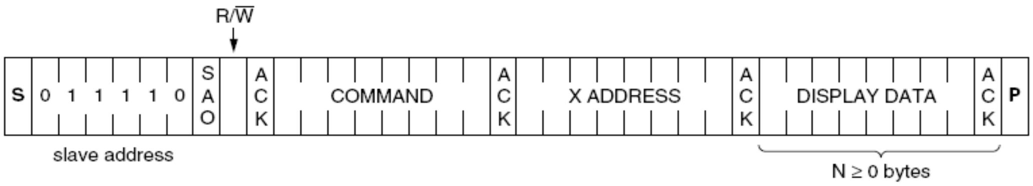
The display has several modes such as vertical or horizontal addressing, power-down mode, inverse video, blank mode and all-segments-on. More detailed information can be found in Section 8. "Resources" on page 28. The PCF8558 controller has six fixed internal address bits plus one external user defined bit, SA0. But since it is embedded in the BATRON module, SA0 is not available to the user.

6.3.4.1 *Slave Address Format:*

- PCF8558 Internal hardcoded address: b011110
- External address SA0, internal wired to logic level 1 in the module.
- The complete slave address is 0x3D.

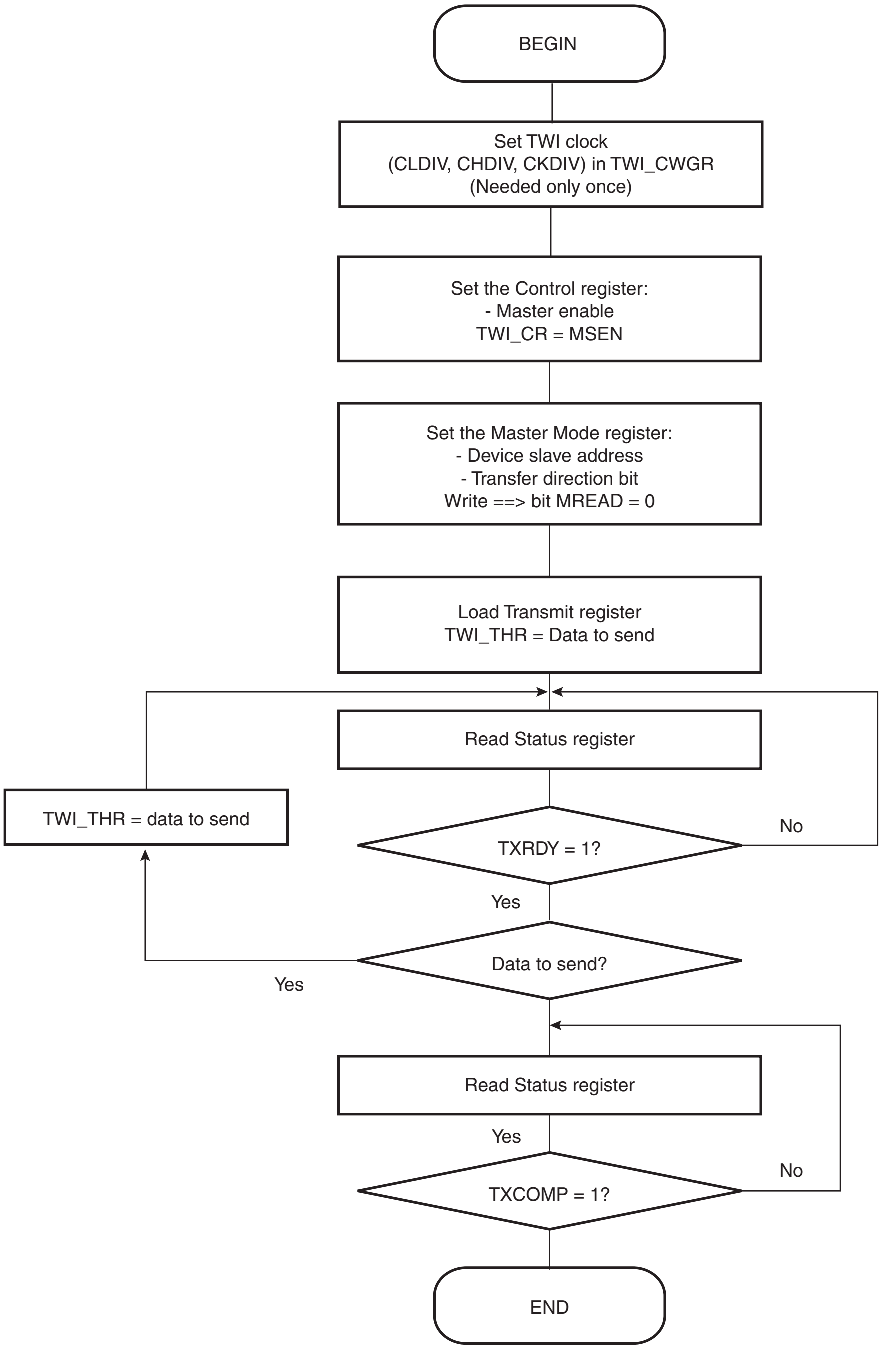
To write data onto the display, commands need to be sent to the controller to set the mode, the page number to access and the X address (column address). Then the display data can be sent to build up a character. Details of a typical frame are shown below.

Figure 6-24. Slave Address Format Frame



After each display data received, the X Address is automatically incremented. This allows sending a stream of data to display without setting the X Address again. To send new display data, the command and the X address will have to be set again. To achieve a frame transfer such as described above, the TWI must be used in multiple data byte operation with no internal address. Refer to Figure 6-25 below. The GLCD module does not have a read mode.

Figure 6-25. Multiple Data Byte Write with No Internal Address



6.3.4.2 *Source Code Example.*

The following C functions are provided with this application note. (File: *glcd_i2c_batron.c* and *glcd_i2c_batron.h)*

```
AT91F_Glcd_PutChar
AT91F_Glcd_PutString
AT91F_Glcd_ClearLcd
```

```
AT91F_Glcd_ClearLcdColumn
AT91F_Glcd_ClearLcdPage
AT91F_Glcd_VideoMode
AT91F_Glcd_ClearChar
AT91F_Glcd_DisplayBitmap
AT91F_Glcd_Reset
```

How to use these functions is shown in source code *GLCD-DEMO.c* and in *All-in-one-Demo.c.* Font file for standard ASCII character set and full character set is also provided.

6.4 Associated Source Code Example and Projects

The associated source code provided with this application note shows the detailed programming of the Two-wire Interface registers according to the flowcharts shown in this application note.

6.4.1 IAR Embedded Workbench® Projects

A standalone project is provided for each slave device. In some of them, the Graphic LCD module is also used. The projects have been developed under IAR Embedded Workbench 4.40A on a AT91SAM7S-EK board with a AT91SAM7S256 device. The in-circuit emulator (ICE) probe used is the Atmel SAM-ICE JTAG/ICE interface.

- GLCD-TempSensor-Demo.eww:

  Samples temperature each second and displays it on the LCD as degrees in both celsius and Fahrenheit.

- GLCD-Demo.eww

  It displays the standard ASCII character set, the full ASCII character set, static images such as Atmel Logo and animated images such as a running cat and dancing penguins.

- GLCD- RTC-DS1337-Demo.eww

  This project gets time and calendar information from the host computer at compile time, sets the corresponding RTC registers and then displays it on the LCD each second. It uses the 1Hz square wave signal (SQW) from the RTC as its time base.

- EEPROM-TEST.eww

  This project performs a test of each byte of each page of the EEPROM.

- All-in-one Demo.eww

  The All-in-one Demo mixes all demo projects described above, except the EEPROM test. The animated and the static images are stored in the EEPROM and displayed on the LCD. The All-in-one Demo makes use of interrupts to switch between demos by pushing a button. The 1Hz square wave signal from the RTC generates an IRQ to read the RTC and the temperature sensor.

- EEPROM Image Loader.eww

  This project goes along with the All-in-one Demo. It allows storing animated and static images into the EEPROM.

6.4.2 Common Source Code

Several low-level Two-wire Interface functions are provided. The TWI function names correspond to the TWI operating modes described in Section 6.2.1.1 “TWI Operations” on page 9.

```
AT91F_TWI_Open
AT91F_TWI_WriteSingle
AT91F_TWI_WriteSingleIadr
AT91F_TWI_WriteMultiple
AT91F_TWI_WriteMultipleIadr
AT91F_TWI_ReadSingle
AT91F_TWI_ReadSingleIadr
AT91F_TWI_ReadMultiple
AT91F_TWI_ReadMultipleIadr
AT91F_TWI_BusRecovery
AT91F_TWI_WaitMicroSecond
AT91F_TWI_ProbeDevices
```

6.4.3 Porting Guide

The code provided here, is for one of the first AT91SAM member in which the Two-wire interface only supports the master mode. For some newer AT91SAM devices, it may happen that a new Two-wire Interface peripheral is embedded with support for slave and multi-master TWI/I²C mode. Check the device's datasheet that you are using. This does not prevent to use most of the code provided. The main difference between TWI peripheral with master mode only and TWI peripheral with Master, Multi-Master and Slave mode, is the formula to set the TWI bit rate.

The IAR embedded workbench projects as described in Section 6.4.1 on page 26 are provided for the AT91SAM7SE-EK board in AN-AT91SAM-TWI-software file associated with this application note.

7. Troubleshooting

The following FAQs give some suggestions in case of TWI malfunction.

- "After a system reset, I am not able to initiate a new TWI transfer, only power supply cycling of the whole system solves the problem."

  → As described in Section 5.5.2 "Resetting a TWI/I²C Slave Device" on page 7, the slave device and TWI/I²C bus may be hung up.

- "When I perform a multiple data read, the TWI reads one more byte from the slave."

  → This can happen when the STOP condition has not been set during the next-to-last data received.

- "I set the TWI clock to be 100 KHz or 400 KHz, but I see less than that."

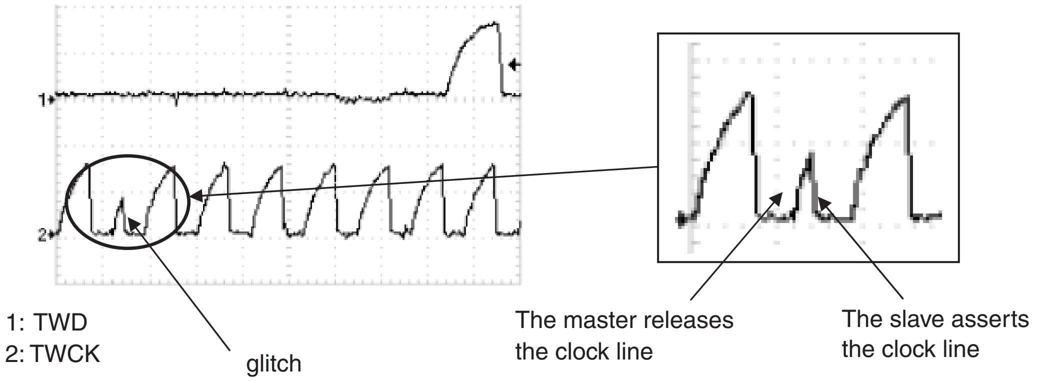
  → Due to the clock stretching capability of the TWI, this may happen if the pull-up resistors (Rp) are not computed correctly (too weak) versus the bus capacitance (Cbus), or the estimated bus capacitance is not good. If the pull-ups are weak, the rise time will be higher than specified and the TWI may interpret it as though a slave device is stretching the clock, thus resulting in a lower frequency.

- In Master transmitter Mode, I sometimes do not have the right number of data transmitted (less data) and the TWI ends the transfer by sending a STOP condition.

  → This may happen when several IRQs are used in the system. It may happen, that the TWI transfer is interrupted by another task of the system. If the Transmit Holding Register

(TWI_THR) is not written fast enough, the TWI ends the transfer by sending a stop condition.

- The communication with the TWI fails. When I probe TWCK and TWD signal with an oscilloscope I observe a glitche(s) on the clock line (TWCK).



→ It might be due to clock stretching. Since clock stretching allows a slave to hold down the clock if it needs to reduce the bus speed (the slave does not respond fast enough to a request). It may happen that the slave does not drive the clock down fast enough after the master has released TWCK.

8. Resources

1. http://i 2c-bus.org/

   – The objective of this site is to provide detailed information on the NXP I²C bus.
2. NXP® Semiconductors I²C-Bus Specification Version 2.1
3. AN10216 I²C MANUAL, NXP Semiconductors
4. Maxim-Dallas DS1337 Real Time Clock Datasheet
5. National Semiconductors® LM75 Digital Temperature Sensor Datasheet
6. Batron 96 x 40 dots STN Yellow Positive Reflective Dot Matrix LCD module BT 96040AV-FSTF-12-I²C-COG
7. NXP, PCF8558 Universal LCD driver for small graphic panels Datasheet
8. Atmel AT24C1024 Two-wire Serial EEPROM Datasheet
9. Analog devices AN-686 APPLICATION NOTE

Revision History

Doc. Rev	Comments	Change Request Ref.
6327B	Figure 5-1“Application Block Diagram” on page 3, repositioned. Section 6.1 “Transfer Format” on page 8, placement of Figure 6-1 and Figure 6-2 swapped and relocated. Section 6.2.1 “Internal Address” on page 9,renamed Figure 6-3 and Figure 6-4 and updated Figure 6-3. Section 6.3.2.1 “Data Write - Slave Receiver Mode”, updated cross references and removed two flowcharts from end of section with subsequent change to figure-number assignments (after 6-15). Section 6.3.3.1 “Byte Write Operation” and Section 6.3.3.2 “Page Write Operation” cross references to flowcharts updated.	4693
6327A	First issue	

Headquarters

Atmel Corporation
2325 Orchard Parkway
San Jose, CA 95131
USA
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

International

Atmel Asia
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimshatsui
East Kowloon
Hong Kong
Tel: (852) 2721-9778
Fax: (852) 2722-1369

Atmel Europe
Le Krebs
8, Rue Jean-Pierre Timbaud
BP 309
78054 Saint-Quentin-en-
Yvelines Cedex
France
Tel: (33) 1-30-60-70-00
Fax: (33) 1-30-60-71-11

Atmel Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581

Product Contact

Web Site
www.atmel.com

Technical Support
Enter Product Line E-mail

Sales Contact
www.atmel.com/contacts

Web Site
www.atmel.com
www.atmel.com/AT91SAM

Technical Support
AT91SAM Support
Atmel techincal support

Sales Contacts
www.atmel.com/contacts/

Literature Requests
www.atmel.com/literature

Disclaimer: The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. **EXCEPT AS SET FORTH IN ATMEL'S TERMS AND CONDITIONS OF SALE LOCATED ON ATMEL'S WEB SITE, ATMEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.** Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel's products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.