

FR FAMILY
32-BIT MICROCONTROLLER
MB91460

I/O PORTS

APPLICATION NOTE

Revision History

Date	Issue
2008-01-25	V1.0, First release, MPi
2008-03-20	V1.1, MSt some typos corrected
2008-04-24	V1.2, Type in Title page corrected, MSt
2009-09-27	V1.3, Added chapter "Using the same IO-port..." MHz

This document contains 20 pages.

Warranty and Disclaimer

To the maximum extent permitted by applicable law, Fujitsu Microelectronics Europe GmbH restricts its warranties and its liability for **all products delivered free of charge** (eg. software include or header files, application examples, target boards, evaluation boards, engineering samples of IC's etc.), its performance and any consequential damages, on the use of the Product in accordance with (i) the terms of the License Agreement and the Sale and Purchase Agreement under which agreements the Product has been delivered, (ii) the technical descriptions and (iii) all accompanying written materials. In addition, to the maximum extent permitted by applicable law, Fujitsu Microelectronics Europe GmbH disclaims all warranties and liabilities for the performance of the Product and any consequential damages in cases of unauthorised decompiling and/or reverse engineering and/or disassembling. **Note, all these products are intended and must only be used in an evaluation laboratory environment.**

1. Fujitsu Microelectronics Europe GmbH warrants that the Product will perform substantially in accordance with the accompanying written materials for a period of 90 days from the date of receipt by the customer. Concerning the hardware components of the Product, Fujitsu Microelectronics Europe GmbH warrants that the Product will be free from defects in material and workmanship under use and service as specified in the accompanying written materials for a duration of 1 year from the date of receipt by the customer.
2. Should a Product turn out to be defect, Fujitsu Microelectronics Europe GmbH's entire liability and the customer's exclusive remedy shall be, at Fujitsu Microelectronics Europe GmbH's sole discretion, either return of the purchase price and the license fee, or replacement of the Product or parts thereof, if the Product is returned to Fujitsu Microelectronics Europe GmbH in original packing and without further defects resulting from the customer's use or the transport. However, this warranty is excluded if the defect has resulted from an accident not attributable to Fujitsu Microelectronics Europe GmbH, or abuse or misapplication attributable to the customer or any other third party not relating to Fujitsu Microelectronics Europe GmbH.
3. To the maximum extent permitted by applicable law Fujitsu Microelectronics Europe GmbH disclaims all other warranties, whether expressed or implied, in particular, but not limited to, warranties of merchantability and fitness for a particular purpose for which the Product is not designated.
4. To the maximum extent permitted by applicable law, Fujitsu Microelectronics Europe GmbH's and its suppliers' liability is restricted to intention and gross negligence.

NO LIABILITY FOR CONSEQUENTIAL DAMAGES

To the maximum extent permitted by applicable law, in no event shall Fujitsu Microelectronics Europe GmbH and its suppliers be liable for any damages whatsoever (including but without limitation, consequential and/or indirect damages for personal injury, assets of substantial value, loss of profits, interruption of business operation, loss of information, or any other monetary or pecuniary loss) arising from the use of the Product.

Should one of the above stipulations be or become invalid and/or unenforceable, the remaining stipulations shall stay in full effect

Contents

REVISION HISTORY	2
WARRANTY AND DISCLAIMER	3
CONTENTS	4
1 INTRODUCTION	6
1.1 Key features	6
2 THE I/O-PORT	7
2.1 Block Diagram	7
2.2 Registers	8
2.2.1 Port Input Enable (PORTEN)	8
2.2.2 Port Data Register (PDR)	8
2.2.3 Data Direction Register (DDR)	8
2.2.4 Port Data Register Direct Read (PDRD)	9
2.2.5 (Extra) Port Input Level Register (PILR, EPILR)	9
2.2.6 Port Output Drive Register (PODR)	9
2.2.7 Port Pull-up/Pull-down Enable and Control Register (PPER, PPCR)	9
2.2.8 (Extra) Port Function Register (PFR, EPFR)	9
2.3 Input Mode	10
2.3.1 Digital Port Input	10
2.3.2 Resource Input	11
2.4 Configuring Pull-up or Pull-down resistor	11
2.5 Output-mode	12
2.5.1 Digital Port Output	12
2.5.2 Resource Output	12
3 PORT INPUT / UNUSED PINS	13
3.1 Port Input / Unused Pins	13
4 TECHNICAL INFORMATION	14
4.1 Hysteresis Inputs	14
5 USING THE SAME I/O PORT SIMULTANEOUSLY AS IN- AND OUTPUT	15
6 TIPS AND TRICKS	17
6.1 Initial Value	17
6.2 Bit Instructions	17
6.3 RMW Instructions	17

7 ADDITIONAL INFORMATION.....	18
LIST OF FIGURES	19
LIST OF TABLES.....	20

1 Introduction

The I/O-port functionality is the simplest peripheral function of the Fujitsu FR microcontroller. Nevertheless, some details should be considered while programming. This application note reflects the functionality and describes the different modes.

Please note that in this document each port number is given with the 2-digit placeholder “xy”. “z” always means the bit position 0 – 7.

Example: “PDR02_D3” means Port 02 Bit 3.

1.1 Key features

- Port direction settable
- Global port enable for port inputs
- Usage of I/O Port or Resource Pin 1/2, both states readable
- Input can be disabled, if corresponding pin is unused
- Internal pull-up/pull-down resistor can be enabled
- Input level can be set to CMOS Hysteresis A (0307), CMOS Hysteresis B (0208), Automotive Hysteresis and TTL
- Output drive strength can be set
- In STOP-HIZ state all pins are switched to input high impedance state

2.2 Registers

2.2.1 Port Input Enable (PORTEN)

This registers globally enable the inputs.

Bit No.	Name	Explanation	Initial Value	Value	Operation
7-2	-	Undefined	X	0	Reserved Bit , always write "0" to it
1	CPORTEN	Bootloader Communication Port Enable	0	0	Inputs of bootloader communication ports are disabled
				1	Inputs of bootloader communication ports are enabled
0	GPORTEN	General Port Enable	0	0	Inputs of all ports are disabled
				1	Inputs of all ports are enabled

Table 2-1: PORTEN

2.2.2 Port Data Register (PDR)

This register contains the data bits, if the corresponding port acts as a simple digital output. The contents are output, if the Port Direction Register is set to output mode.

Please note that a resource output control bit overwrites the PDR bit value.

PDR _{xy} _Dz	Pin Function
0	Pin State Low (V _{SS})
1	Pin State High (V _{DD})

Table 2-2: PDR

The read value of PDR register depends on the following:

- The corresponding bit in DDR and PFR register
- The type of instruction used (Read/Read-modify-write)

The following table describes the above discussed behaviour:

PFR Value	DDR Value	PDR		
		Write	Read	Read-modify-write
0	0 (Input)	Writes the PDR setting value, has no effect on the pin value	Reads the sampled pin data	Always writes the PDR setting value
0	1 (Output)	Writes the PDR setting value to the corresponding external pins	Reads the PDR register value	Always writes the PDR setting value

Table 2-3: Reading PDR

2.2.3 Data Direction Register (DDR)

This register contains the bit information of the corresponding pins if they should act as input or output.

DDR _{xy} _Dz	Pin Function
0	Port Input
1	Port Output

Table 2-4: DDR

2.2.4 Port Data Register Direct Read (PDRD)

This register samples the pin data with CLKP. It is read-only.

2.2.5 (Extra) Port Input Level Register (PILR, EPILR)

The input levels of each port can be programmed bit-wise using PILR and EPILR registers.

PILR _{xy} _Dz	EPILR _{xy} _Dz	Input Level	V _{IL}	V _{IH}
0	0	CMOS Hysteresis A (0307)	0.3 V _{DD}	0.7 V _{DD}
1	0	Automotive Hysteresis	0.5 V _{DD}	0.8 V _{DD}
0	1	TTL	0.8 V	2.1 V
1	1	CMOS Hysteresis B (0208)	0.2 V _{DD}	0.8 V _{DD}

Table 2-5: PILR, EPILR

2.2.6 Port Output Drive Register (PODR)

With these registers the strength of the output current of a pin can be adjusted to improve the EMI behaviour. This setting does not limit the output drive to the given values. The output must not exceed these values to guarantee the specified Output levels. See also Datasheet of MB91F46x series.

PODR _{xy} _Dz	Output Current
0	5 mA output drive
1	2 mA output drive

Table 2-6: PODR

2.2.7 Port Pull-up/Pull-down Enable and Control Register (PPER, PPCR)

These registers enable and connect an internal pull-up or pull-down resistor to a port pin.

PPER _{xy} _Dz	PPCR _{xy} _Dz	Pull-Up Resistor
0	x	Pull-up/ Pull-down Disabled
1	0	Pull-down is selected
1	1	Pull-up is selected

Table 2-7: PPER, PPCR

The nominal value for this pull-up resistor is 50 kΩ.

2.2.8 (Extra) Port Function Register (PFR, EPFR)

PFR along with EPFR configures the certain port as general purpose I/O or resource 1 (input/output) or resource 2 (input/output), if available. It should be noted that PFR is available for all the ports where as EPFR is available for selected ports.

The following table gives an example of this.

PFR19_D2	EPFR19_D2	Pin Function
0	X	General Purpose I/O
1	0	SCK5 input/output
1	1	CK5 input

Table 2-8: PFR, EPFR

For more details please refer the hardware manual.

2.3 Input Mode

In general, if a pin should act as a digital input, the `PORTEN` register should be set to `0x03` in order to globally enable the inputs

2.3.1 Digital Port Input

The following example shows the register configuration that needs to be done on MB91460 Series, if a pin should act as a digital input:

```
PFR19_D4 = 0;    // Port 19 pin 4 as general purpose i/o
DDR19_D4 = 0;    // data direction - input
PORTEN = 0x03;  // global port input enable
```

As shown above, first the `PORTEN` register should be set to `0x03`. Then the corresponding bit of `PFR` register should be set to 0 in order to configure port 19 pin 4 as general purpose i/o and finally the corresponding bit of `DDR` register should be set to 0.

After configuring a pin as digital port input, the level of the input pin can be determined as follows:

```
if ( 1 == PDR19_D4 )    // if pin high?
{
    // do something
}
else                    // pin low
{
    // do something
}
```

Additionally the level of the input pin can also be read out via the `PDRD` register (2.2.4) as follows:

```
if ( 1 == PDRD19_D4 )  // if pin high?
{
    // do something
}
else                    // pin low
{
    // do something
}
```

Optionally the input level can be set via the `PILR` and `EPILR` registers (2.2.5) as follows:

```
PILR19_D4 = 1;    // set input detection level as CMOS hysteresis
                // B(0208)
EPILR19_D4 = 1;  //
```

If the connected external source may change to high-Z state, please use an external pull-up or –down resistor or configure the internal pull-up or pull-down resistor using `PPER` and `PPCR` (2.2.7) registers.

2.3.2 Resource Input

The following example shows the register configuration that needs to be done on MB91460 Series, if a pin should act as resource input (clock input for Free-run timer 4):

```
PFR19_D2 = 1;      // Port 19 pin 2 as resource function
EPFR19_D2 = 1;    // Port 19 pin 2 as CK4 input
PORTEN = 0x03;    // global port input enable
```

In addition to the above the `ECLK` bit of `TCCS4` register should be set as 1 in order to select the external clock for Free-run timer 4.

Optionally the input level can be set via the `PILR` and `EPILR` registers (2.2.5), if required and also the pull-up or pull-down resistor can be configured using `PPER` and `PPCR` registers, if required.

2.4 Configuring Pull-up or Pull-down resistor

All ports while in input-mode (digital input as well as resource input) have the possibility to enable and configure an internal pull-up/pull-down resistor (about 50 k Ω) by programming the `PPER` and `PPCR` (2.2.7).

It should be noted that the `PPCR` can only be changed while the corresponding bit of `PPER` 0 i.e. pull-up/pull-down disabled. If the `PPCR` is written with a different value while pull-up/pull-down enabled then the new `PPCR` setting would not be effective.

The following example code shows how to do the same:

```
. . .
. . .
PPER19_D4 = 0;    // disable pull-up/pull-down on port 19 pin 4
PPCR19_D4 = 0;    // configure pull-down on port 19 pin 4
PPER19_D4 = 1;    // re-enable pull-down on port 19 pin 4
PORTEN = 0x03;    // global port input enable
. . .
. . .
```

If the port-pin is used as an output, the values of corresponding bits of these registers have no meaning and the pull-up/pull-down resistor is disabled.

Enabled pull-up resistors will be disabled while the microcontroller is in `STOP-HIZ` state.

2.5 Output-mode

2.5.1 Digital Port Output

The following example shows the register configuration that needs to be done on MB91460 Series, if a pin should act as a digital output:

```
PFR19_D4 = 0;    // Port 19 pin 4 as general purpose i/o
PDR19_D4 = 0;    // clear output before setting the data direction,
                 // this is required to guarantee the initial value
                 // when the data direction is set as output
DDR19_D4 = 1;    // data direction - output
PORTEN = 0x03;   // this configuration is required only if the external
                 // pin status needs to be read via PDRD
```

Optionally the output current strength can be set by the PODR register (2.2.6) as follows:

```
PODR19_D4 = 1;   // 2mA output drive
```

2.5.2 Resource Output

The following example shows the register configuration that needs to be done on MB91460 Series, if a pin should act as resource output (output for Reload Timer 7):

```
PFR15_D7 = 1;    // Port 15 pin 7 resource output
EPFR15_D7 = 1;   // Port 15 pin 7 as TOT7 output
PORTEN = 0x03;   // this configuration is required only if the external
                 // pin status needs to be read via PDRD
```

Optionally the output current strength can be set by the PODR register (2.2.6).

3 Port Input / Unused Pins

How to connect Input Port Pins and how to proceed with unused Pins

3.1 Port Input / Unused Pins

It is strongly recommended not to leave the pins unconnected, while they are switched to input and are enabled (by `PORTEN` register). In this case those pins can enter a so-called *floating state*. This can cause a high I_{CC} current, which is adverse to low power modes. Also damage of the MCU may happen.

In such cases, it is highly recommended to use the internal pull-up/pull-down resistors.

4 Technical information

ELECTRICAL CHARACTERISTICS OF THE INPUT HYSTERESIS

4.1 Hysteresis Inputs

A hysteresis describes the behaviour of an input pin where the input level at which '1' is detected, and the level at which '0' is detected are different.

The levels are described in chapter 2.2.5.

Kindly note that the power supply current i.e. the power consumption of the device may increase, while the input voltage is within the hysteresis area, however the input current of the I/O pin remains constant.

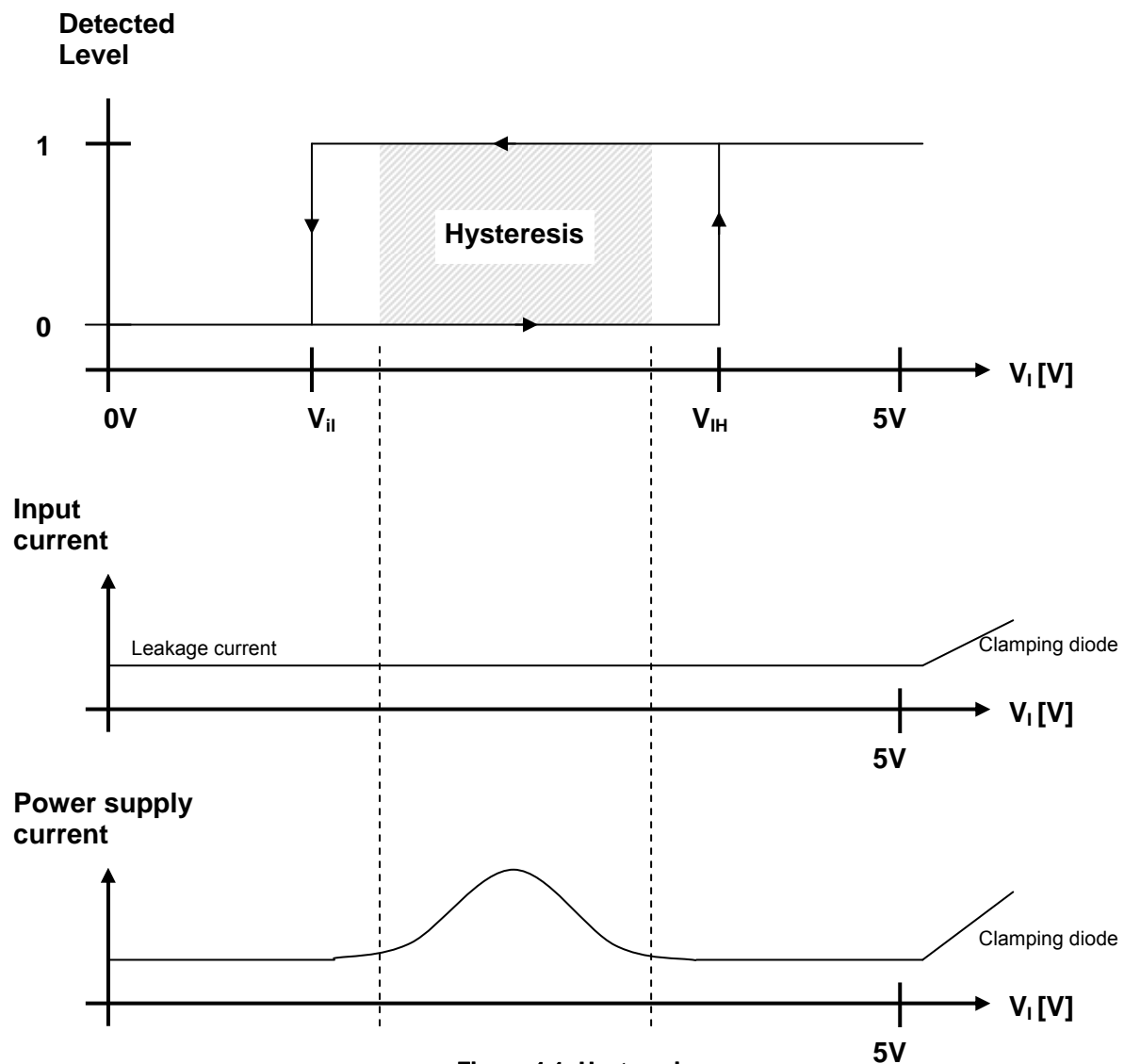


Figure 4-1: Hysteresis

- V_{iL} Port hysteresis input level (low-level) specified in the datasheet
- V_{iH} Port hysteresis input level (high-level) specified in the datasheet
- Real hysteresis area

5 Using the same I/O port simultaneously as in- and output

THIS CHAPTER EXPLAINS USING ONE PIN SIMULTANEOUSLY AS IN- AND OUTPUT

With the circuit shown in figure 6.1, enabled internal pull-up resistor, pin state set low and some small considerations in the software it's possible to use the same port simultaneously as input and output – for polling the key button and driving a LED at the same pin.

If the port is used as output the LED is on. If the port is used as input the LED is off and the key button's status can be polled.

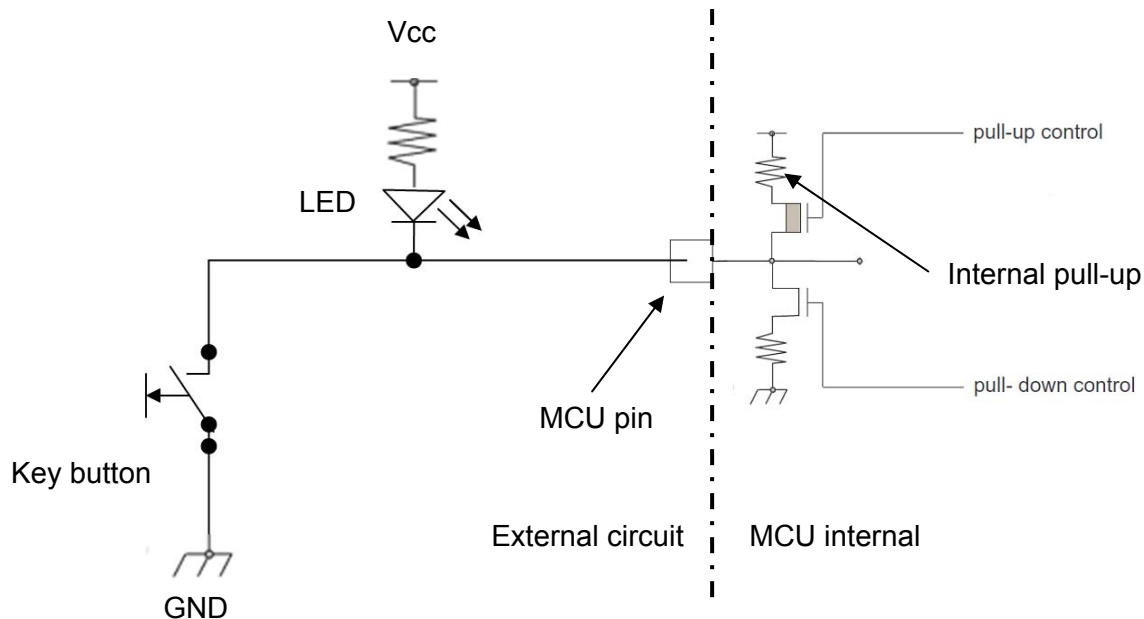


Figure 5-1: I/O-port example circuit

When the port direction is changed from output to input, the pin's level becomes high. The high-level is not reached immediately after the port's DDR register is set to input. Some minor parasitic capacities (~ 30 pF) caused by chip-internal capacities and by the PCB are connected to the pin. These capacities are loaded via the internal pull-up (50 k Ω) as soon as the port is set to input. The pin reaches high-level after a typical charging time of 1 to 2 μ s.

$$\text{Charging time: } \tau = R * C = 50 \text{ k}\Omega * 30 \text{ pF} = 1,5 \text{ us}$$

Polling the port within this time may return a false value. As workaround we recommend to implement a short delay loop after the port's direction is switched to input and before the port polling is started. Figure 6.2 shows a code example for using port 29 pin 0 as in- and output.

```
void KeyLED_Init(void)
{
    PFR29_D0 = 0;      /* Port 29 pin 0 as general purpose I/O */
    PDR29_D0 = 0;      /* Preset Port 29 pin 0 pin state low (never output high */
                       /* this might cause shortcircuit if switch is pressed) */

    PPER29_D0 = 0;     /* disable pull-up/pull-down on port 29 pin 0*/
    PPCR29_D0 = 1;     /* configure pull-up on port 29 pin 0*/
    PPER29_D0 = 1;     /* re-enable pull-up on port 29 pin 0*/
}

void LED_on(void)
{
    DDR29_D0 = 1;      /* Switch Port 29 pin 0 to output */
}

void LED_off(void)
{
    DDR29_D0 = 0;      /* Switch Port 29 pin 0 to input (never output high) */
}

unsigned char KeyPressed(void)
{
    if (PDR29_D0 == 0)
        return (1);    /* return '1' in case that the key button is pressed */
    else
        return (0);
}

void main(void)
{
    ...

    KeyLED_Init();

    LED_on();          /* switch the LED on */
    for (delay=0; delay<500000; delay++) /* keep the LED on for some time delay */
        __asm("\tNOP");
    LED_off();         /* switch the LED off */

    for (delay=0; delay<10; delay++)      /* short delay to get pull-up resistor */
        __asm("\tNOP");                  /* active */

    while (!KeyPressed())                 /* wait until key button is pressed */
        __asm("\tNOP");

    ...
}
```

Figure 5-2: Example code

6 Tips and Tricks

THIS CHAPTER GIVES SOME HINTS ON USING I/O PORTS

6.1 Initial Value

Ensure that the port-data is defined before the pin-direction is changed to output. Otherwise undefined data might be output to the I/O-pin, until `PDR00` is written.

```
PDR00 = 0x00; // define initial value before port 0 is set to output
DDR00 = 0xFF; // set port 0 to output, after initial value is defined
```

6.2 Bit Instructions

Use byte-instructions which will be executed faster instead of using bit instructions since all bit instructions are essentially read-modify-write instructions.

6.3 RMW Instructions

Accessing to the `PDR` register (2.2.2) via a read-modify-write instruction always returns the contents of the register itself during read cycle (of the same read-modify-write instruction) regardless of the `DDR` register setting.

7 Additional Information

Information about FUJITSU Microcontrollers can be found on the following Internet page:

<http://mcu.emea.fujitsu.com/>

The software example related to this application note is:

91460_io

It can be found on the following Internet page:

http://mcu.emea.fujitsu.com/mcu_product/mcu_all_software.htm

List of Figures

Figure 2-1: I/O-port block diagram	7
Figure 4-1: Hysteresis	14
Figure 5-1: I/O-port example circuit	15
Figure 5-2: Example code	16

List of Tables

Table 2-1: PORTEN.....	8
Table 2-2: PDR	8
Table 2-3: Reading PDR	8
Table 2-4: DDR	8
Table 2-5: PILR, EPILR	9
Table 2-6: PODR	9
Table 2-7: PPER, PPCR.....	9
Table 2-8: PFR, EPFR.....	9