

F²MC-16LX FAMILY

16-BIT MICROCONTROLLER

ALL SERIES

I/O-PORT

APPLICATION NOTE

Revision History

Date	Issue
09.03.2004	V1.0, First release, HWe
09.03.2004	V1.1, minor changes

This document contains 12 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	5
2 THE I/O-PORT	6
2.1 Input-mode	6
2.2 Pull-up control register.....	6
2.3 Output-mode	7
2.4 Software-Examples	7
2.5 Bit-instructions.....	7
2.5.1 Example	7
2.5.1.1 Example using RMW-instructions	7
2.5.1.2 Example using byte-instructions.....	8
3 I/O-PORT SHARED WITH OTHER PERIPHERAL RESOURCES	9
3.1 Analogue-Inputs	9
3.2 UART	9
4 UNUSED PINS	10
5 TECHNICAL INFORMATION.....	11
5.1 Hysteresis-inputs.....	11
6 TIPS & TRICKS	12
6.1 Initial value	12
6.2 Bit-instructions.....	12

1 Introduction

The I/O-port functionality is the simplest peripheral function of the Fujitsu 16LX microcontroller.

Nevertheless, some details should be considered while programming.

This application note reflects the functionality and describes the different modes.

Further, some software-examples are given as well as some tips & tricks.

Note:

This application is intended to give an overview about the peripheral function "I/O-port". Although this resource is given in all 16LX devices, detailed differences may occur between different microcontroller series.

All data given in this application note shall give a feeling to the customer, but will not replace the specified data given by the datasheet and hardware-manual of the corresponding microcontroller.

2 The I/O-port

This chapter describes the basic functionality of the I/O-port.

Figure 2-1 shows the internal block diagram of an external I/O-pin.

For each external I/O-pin a data-register-bit and direction-register-bit exist. Up to 8 I/O-pins may be encapsulated within one Direction-register (DDRx) and Data-register (PDRx), where 'x' refers to the related port-number.

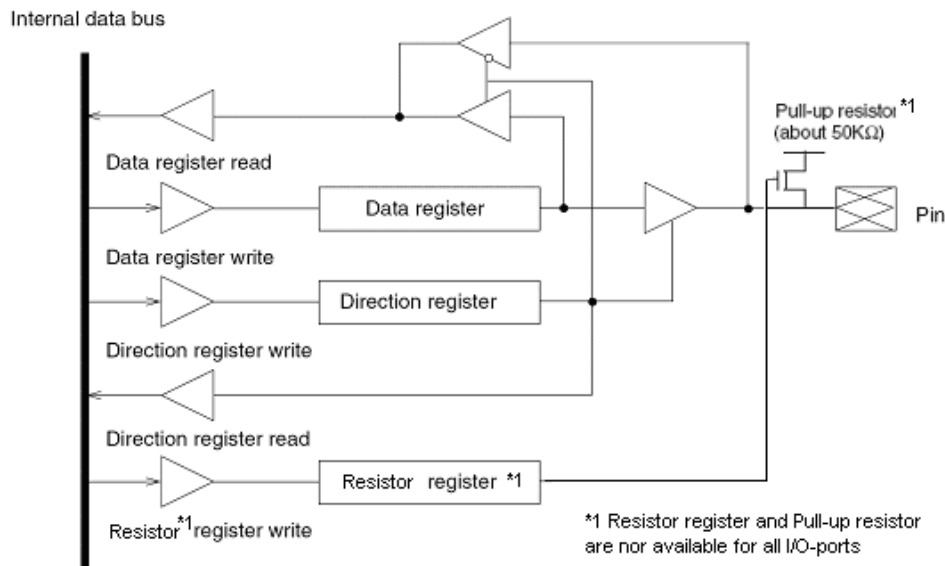


Figure 2-1: I/O-port block-diagram

2.1 Input-mode

If the corresponding direction-bit is set to '0' in the DDRx-register, which normally is the initial value after power-on reset, the pin functions as an input.

Reading the corresponding bit-position in the PDRx-register delivers the level of the related external pin.

Writing to the corresponding bit-position in the PDRx-register, writes the data to an output-latch, but the level is not driven outside. In this case, take care of read-modify-write instructions (see chapter 2.5), especially when using bit-operations.

2.2 Pull-up control register

Some ports, while in input-mode, may have the possibility to enable an internal pull-up resistor (~ 50Kohm) by programming the PUCRx register.

The initial value of '0' disconnects the internal pull-up resistor, writing '1' to the corresponding bit-position in the PUCRx-register enables the resistor.

If the port-pin is used as an output the value of the register-bit has no meaning and the pull-up resistor is disabled.

Enabled pull-up resistors will be disabled while the microcontroller is in "stop"-mode.

If the external pin is used by the external bus-interface, the internal pull-up register can not be used, too.

2.3 Output-mode

If the corresponding direction-bit is set to '1' in the DDRx-register the pin functions as an output.

Writing to the corresponding bit-position in the PDRx-register, writes the data to an output-latch, and the value drives directly the external pin-level.

Reading the PDRx-register delivers the content of the output-latch.

2.4 Software-Examples

```
PDR0 = 0xAA;    // value 0xAA is written to output-latch
                // Note: if DDR0 already is set to output
                // the output-latch drives the external pins

DDR0 = 0xF0;    // Port 0: Bit 7...4 is set to output
                // Port 0: Bit 3...0 is set to input

PDR0_P05 = 0;   // Port 0: Bit 5 is cleared to 0
                // read-modify-write (RMW) instruction is used

DDR0_D01 = 1;   // Port-pin P01 is switched to output
                // read-modify-write (RMW) instruction is used
```

2.5 Bit-instructions

Bit instructions like `PDR0_P05 = 0` will be handled by the 16LX core as read-modify-write (RMW) instruction. This means, before the new value of the bit will be written, the whole byte of the register will be (R)ead, the corresponding bit will be (M)odified and the whole byte is (W)ritten back.

In case that the port-direction will be changed within the program flow, this might cause a problem, if two bits will be modified sequentially within the output-latch, while the port-pins are set to input-mode. The second RMW-instruction may overwrite the first one, because the real level of the input pin will be read.

The following example shows the danger using RMW-instruction. Further, RMW-instructions are slower than a byte-transfer.

2.5.1 Example

Port 0 is set to input-mode. Pull-up resistors are connected to Port 0 to define a '1'-level. Bit 0 and bit 1 should be 'prepared' in order to output '00' at a further program-state, when the port 0 will be set to output mode.

2.5.1.1 Example using RMW-instructions

```
DDR0 = 0x00;    // Set port 0 as input
                // External pull-up resistors are connected to P00-P07

PDR0_P00 = 0;  // Write '0' to the output-latch bit 0
                // because of RMW-instruction the whole port 0 is read
                // (0xff because of external pull-up resistor)
                // and bit 0 will be set to '0'
                // content of output-latch: '11111110'
                // Because the port-direction is still set to input
                // the content of the output-latch is not output
```

```
PDR0_P01 = 0; // Write '0' to the output-latch bit 1
              // because of RMW-instruction the whole port 0 is read
              // (0xff because of external pull-up resistor)
              // and bit 1 will be set to '0'
              // content of output-latch: '11111101'

DDR0 = 0x03; // Set bit 0 and bit 1 to output-mode
              // The content of the output-latch will be displayed:
              // bit0=1 bit1=0
              // Bit 0 was overwritten by the second RMW-instruction
              // This result DOES NOT cover the programmer's expectation
```

2.5.1.2 Example using byte-instructions

```
DDR0 = 0x00; // Set port 0 as input
              // External pull-up resistors are connected to P00-P07

PDR0 = 0x00; // Write '0' to the output-latch bit 0 and bit 1
              // content of output-latch: '00000000'
              // Because the port-direction is still set to input
              // the content of the output-latch is not output

DDR0 = 0x03; // Set bit 0 and bit 1 to output-mode
              // The content of the output-latch will be displayed:
              // bit0=0 bit1=0
              // This result covers the programmer's expectation
```


3 I/O-port shared with other peripheral resources

This chapter gives advices for I/O-pin that are shared with peripheral-functions

In order to prevent malfunction the corresponding port-direction-register of all I/O-pins should be set regarding its function (input or output) within the application.

Normally, in case that an external-pin is used by a peripheral resource, the pin-direction is controlled by the resource-module itself. Some exceptions may exist, that will be mentioned in the chapter of the corresponding peripheral-resource in the hardware of the microcontroller-series.

3.1 Analogue-Inputs

Take care of I/O-pins that may be used by the A/D-converter as Analogue-inputs.

For these pins (ports) an additional register ADER_x exist (x means the port-number).

Normally the initial value of this register is 0xff, which means the external pin will be used as an analogue-input. In order to use the pin as a digital I/O-pin, the corresponding bit in the ADER_x-register has to be cleared.

Example: Use port-pin P60 and P62 of MB90340series as digital I/O-pin

Depending on the FLASH-derivate the ports 5, 6 and 7 may be used as analogue-inputs. The initial value of the registers ADER₅, ADER₆ and ADER₇ are 0xff. In order to select P60 and P62 for digital-I/O set ADER₆=0xfa (11111010)

3.2 UART

Some microcontroller-series do not switch automatically the pin-direction of the TXD-pin to output, if the UART-module is used. This has to be handled manually, by setting the corresponding port-bit to '1' in the DDR_x register.

4 Unused pins

Do not leave input-pins unconnected. Floating input-pins can cause latch-up effects that may defect the device.

Use a pull-up or pull-down resistor to define input-level or set unused I/O-pins to output-mode within the software.

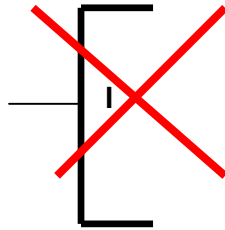


Figure 4-1: Do not leave open unused input-pins

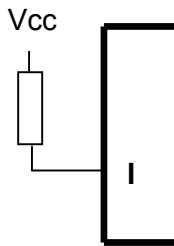


Figure 4-2: Use pull-up resistor ...

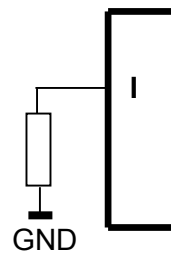


Figure 4-3: ... or pull-down resistor ...

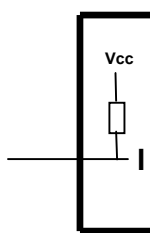


Figure 4-4: ... or use internal pull-up resistor (if available) ...

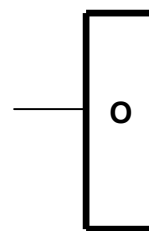


Figure 4-5: ... or set pin to output-mode

5 Technical information

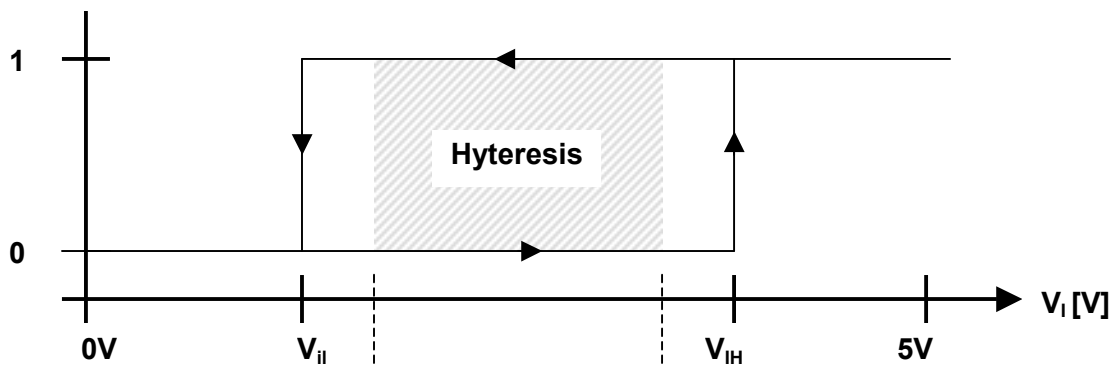
5.1 Hysteresis-inputs

“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.

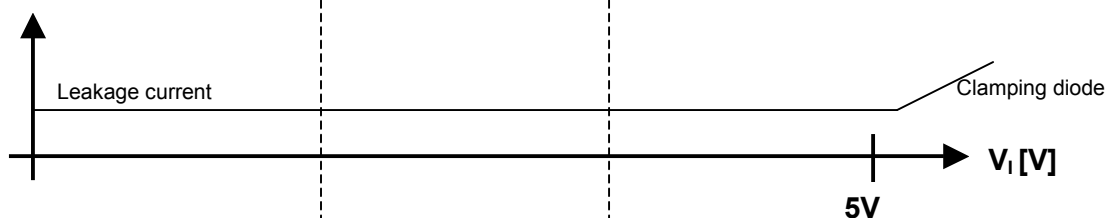
The datasheet of the corresponding microcontroller specifies the level. For some microcontroller-series the levels might be software-selectable (CMOS, TTL, automotive).

Take care that the power-consumption of the device but not the input-current of the I/O-pin may increase while the input-voltage is within the Hysteresis-area.

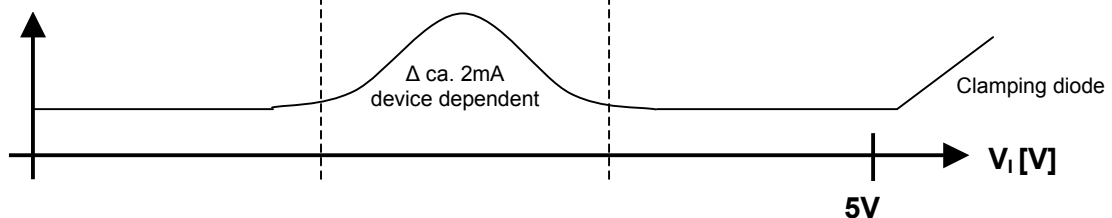
Detected Level



Input-current



Power supply-current



- 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

6 Tips & Tricks

6.1 Initial value

Take care 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.

```
PDRx = initial_value; // define initial value before port is set to output
DDRx = 0xff;          // set port x to output, after initial value is defined
```

6.2 Bit-instructions

Do not use bit-instructions, if the port-/pin-direction is changed sometimes within the application; e.g. I²C-emulation. Use byte-instructions instead, which also will be executed faster. See chapter 2.5.