

F²MC-16FX FAMILY

16-BIT MICROCONTROLLER

ALL SERIES

I/O-PORT

APPLICATION NOTE

Revision History

Date	Issue
2006-04-20	V1.0, First release, MWi
2006-12-05	V1.1, Reviewed the document and updated with review findings, MPi
2007-02-21	V1.2, some clarifications added, MPi
2007-08-02	V1.3, typos corrected; small clarifications, MPi
2009-09-27	V1.4, Added chapter "Using the same IO-port..." MHz

This document contains 21 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** (e.g. 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 that 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 Data Register (PDR)	8
2.2.2 Data Direction Register (DDR)	8
2.2.3 Port Input Enable Register (PIER)	9
2.2.4 External Pin State Register (EPSR)	9
2.2.5 (Extended) Port Input Level Register (PILR, EPILR)	9
2.2.6 Port Output Drive Register (PODR) and Port High Drive Register (PHDR) ...	9
2.2.7 Pull-up Control Register (PUCR)	10
2.3 Input Mode	10
2.3.1 Digital Port Input	10
2.3.2 Resource Input	11
2.4 Pull-up control register	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	15
4.1 Hysteresis Inputs	15
5 USING THE SAME I/O PORT SIMULTANEOUSLY AS IN- AND OUTPUT	16
6 TIPS AND TRICKS	18
6.1 Initial Value	18
6.2 Bit Instructions	18
6.3 RMW Instructions	18

7 ADDITIONAL INFORMATION.....	19
LIST OF FIGURES	20
LIST OF TABLES.....	21

1 Introduction

The I/O-port functionality is the simplest peripheral function of the Fujitsu 16FX 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_P3” means Port 02 Bit 3.

1.1 Key features

- Port direction settable
- Usage of I/O Port or Resource Pin, both states readable
- Input can be disabled, if corresponding pin is unused
- Internal pull-up resistor can be enabled
- Input level can be set to CMOS (0307), CMOS (0208), Automotive Hysteresis, and TTL
- Output drive strength can be set

2.2 Registers

2.2.1 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} _Pz	Pin Function
0	Pin State Low (V _{SS})
1	Pin State High (V _{DD})

Table 2-1: PDR

The read value of PDR register depends on the following:

- The corresponding bit in DDR register
- The Current status of the resource that is connected to the same pin
- The type of instruction used (Read/Read-modify-write)

The following table describes the above discussed behaviour:

DDR Value	Resource Output	Pin Value	Read Value of 'Read Instructions'	Read Value of 'Read-modify-write Instructions'
0 (Input)	Enabled	Value of Resource Input	Input Pin State	Value of PDR
	Disabled	Port Input	Input Pin State	Value of PDR
1 (Output)	Enabled	Value of Resource Output	Value of PDR	Value of PDR
	Disabled	Value of PDR	Value of PDR	Value of PDR

Table 2-2: Reading PDR

2.2.2 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	Resource Output	Pin Function
0	Disabled	Port Input ¹
0	Enabled	Resource Output
1	Disabled	Port Output
1	Enabled	Resource Output logically OR with Port Output

Table 2-3: DDR

¹ Please note that an input **always must** be enabled by the PIER, regardless whether digital input or resource input is used.

2.2.3 Port Input Enable Register (PIER)

This registers enables the input functionality of a MCU pin. If a corresponding bit is set to “0” (default value) the pin can be left unconnected regardless of the settings of all port and resource registers.

PIER _{xy} _IE _z	Function
0	Digital Input disabled
1	Digital Input enabled

Table 2-4: PIER

2.2.4 External Pin State Register (EPSR)

This register always contains the state of the corresponding external pin.

Please note that the corresponding bit in the Port Input Enable Register must also be set to “1”, otherwise the read-back value is undefined.

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

With these registers one of the following input levels can be chosen. The input level applies to both pin input and resource input (e.g. TINn input of Reload Timer, INn input of Input Capture).

PILR _{xy} _IL _z	EPILR _{xy} _EIL _z	Input Level	V _{IL}	V _{IH}
0	0	CMOS (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 (0208)	0.2 V _{DD}	0.8 V _{DD}

Table 2-5: PILR, EPILR

2.2.6 Port Output Drive Register (PODR) and Port High Drive Register (PHDR)

With these registers the strength of the output current of a pin can be adjusted:

PODR _{xy} _OD _z	PHDR _{xy} _HD _z	Output Current
0	0	Normal Current
1	0	Reduced Current
X	1	High Current

Table 2-6: PODR, PHDR

High current outputs are only available with ports those provide the stepper motor functionality i.e. Port 8, 9 and 10. It is driven by DV_{CC}. It should be noted that the Output Current also influences the slew rate and therefore EMI emission. The higher the output current the more the slew rate and hence higher the EMI noise.

Please see datasheet for current values. The current values depend on V_{CC} and DV_{CC} respectively.

2.2.7 Pull-up Control Register (PUCR)

These registers connect

PIER _{xy} _IE _z	Function
0	Digital Input disabled
1	Digital Input enabled

an internal pull-up resistor to a port pin.

PUCR _{xy} _PU _z	Pull-Up Resistor
0	Disabled
1	Enabled

Table 2-7: PUCR

The nominal value for this pull-up resistor is 50 kΩ. It may vary between 25 kΩ to 100 kΩ depending upon the doping process and temperature.

2.3 Input Mode

In general, if a pin should acts as a digital input, the corresponding bit in the Port Input Enable Register (2.2.3) *must* be set to “1”.

2.3.1 Digital Port Input

The following example shows the register configuration that needs to be done on MB96340 Series, if a pin should act as a digital input:

```
PIER06_IE0 = 1;    // port input enable
DDR06_D0 = 0;     // data direction - input
ADER0_ADE0 = 0;   // disable AN0 input that is shared with port 06
                  // pin 0
```

As shown above, the resource input AN0 (Analog Input 0 of ADC) which shares port 06 pin 0 input has to be disabled for digital port input functionality.

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

```
if ( 1 == PDR06_P0 )    // if pin high?
{
    // do something
}
else                    // pin low
{
    // do something
}
```

Additionally the level of the input pin can also be read out via the External Pin State Register (2.2.4) as follows:

```
if ( 1 == EPSR06_PS0 ) // if pin high?
{
    // do something
}
else // pin low
{
    // do something
}
```

Optionally the input level can be set via the (Extended) Port Input Level Register (2.2.5) as follows:

```
PILR06_IL0 = 1; // set input detection level as CMOS (0208)
EPILR06_EIL0 = 1; //
```

If the connected external source may change to high-Z state, please use an external pull-up or –down resistor or set the corresponding bit in the Pull-up Control Register (2.2.7).

2.3.2 Resource Input

The following example shows the register configuration that needs to be done on MB96340 Series, if a pin should act as resource input (ADC input AN0 in this case):

```
PIER06_IE0 = 1; // port input enable
ADER0_ADE0 = 1; // AN0 pin as analog input
```

Optionally the input level can be set via the (Extended) Port Input Level Register (2.2.5).

If the connected external source may change to high-Z state, please use an external pull-up or –down resistor or set the corresponding bit in the Pull-up Control Register (2.2.7).

2.4 Pull-up control register

All ports, while in input-mode, have the possibility to enable an internal pull-up resistor (about 50 kΩ) by programming the Pull-up Control Register (2.2.7).

The initial value of “0” disconnects the internal pull-up resistor, writing “1” to the corresponding bit-position in the PUCR_{xy} 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 (Exception: For I²C pins SDA and SCL, the setting remains. Also for UART output SOT the internal pull-up can be used if not provided by line driver).

Enabled pull-up resistors will be disabled while the microcontroller is in stop mode or timer mode, if the SPL bit of SMCR register is configured as 1 before entering these modes. The resistor is also disabled if the pin is used as ADC input.

If the external pin is used by the external bus-interface, the internal pull-up resistor cannot be used too.

2.5 Output-mode

2.5.1 Digital Port Output

The following example shows the register configuration that needs to be done on MB96340 Series, if a pin should act as a digital output:

```
PIER08_IE1 = 1; // this configuration is required only if the external
                // pin status needs to be read via EPSR
PDR08_P1 = 0; // clear output before setting the data direction,
              // this is required to guarantee the initial value
              // when the data direction is set as output
DDR08_D1 = 1; // data direction - output
TMCSR0_OUTE = 0; // disable TOT0 output that is shared with port 08
                // pin 1
```

As shown above, the resource output TOT0 (Output of RLT0) which shares port 08 pin 1 output has to be disabled for digital port output functionality.

Optionally the output current strength can be set by the Port Output/High Drive Registers (2.2.6) as follows:

```
PODR08_OD1 = 1; // reduced current output
PHDR08_HD1 = 0; //
```

2.5.2 Resource Output

The following register settings have to be done, if a pin should act as a resource output:

```
PIER08_IE1 = 1; // this configuration is required only if the external
                // pin status needs to be read via EPSR
TMCSR0_OUTE = 1; // enable TOT0 output
```

Optionally the output current strength can be set by the Port Output/High Drive Registers (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. 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 can happen.

Use the internal pull-up resistors in this case. If not, use external pull-up or pull-down resistors to define the input-level.

The recommended way is to set the port input enable to “0”, if a port pin is unconnected.

Never connect a potential divider with almost same resistor values.

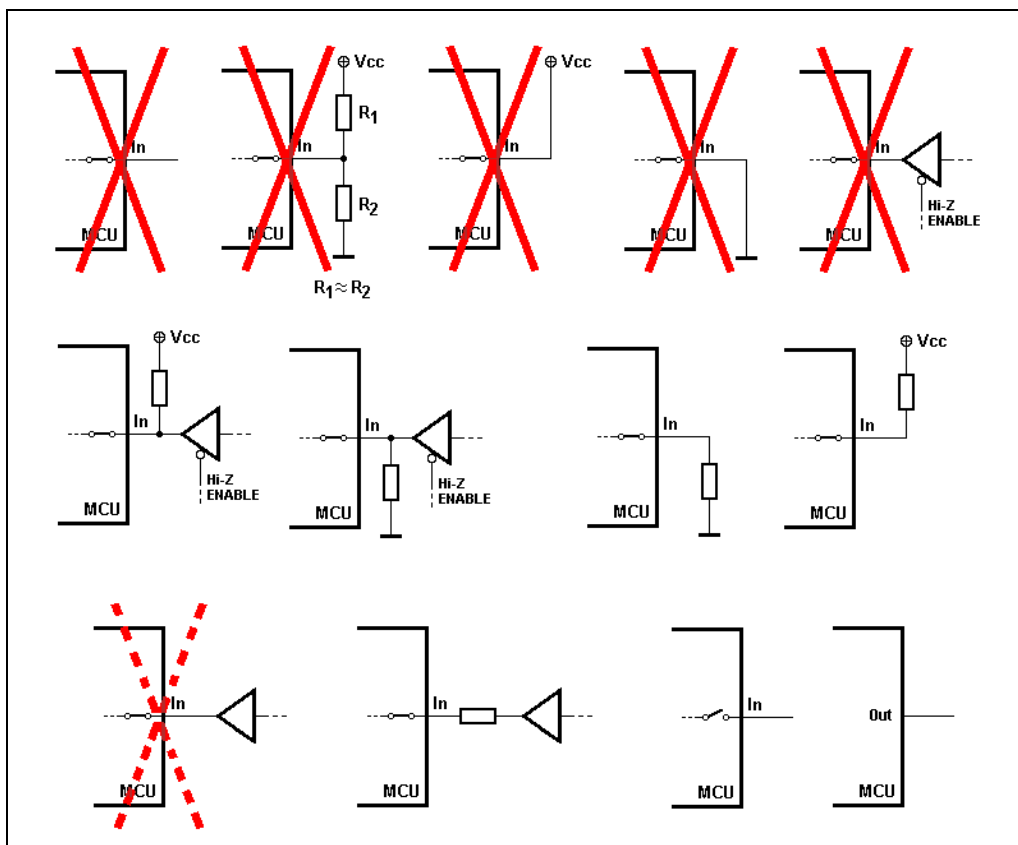


Figure 3-1: Recommended Connections for Port Input and Unused Pins

Be careful with connection of input pins to other devices, which can go into High-Z states. Always use internal pull-up or external pull-up or pull-down resistors in this case.

Outputs from external circuits should always be connected via a serial resistor to a MCU input pin to prevent latch-up effects caused by under- or overshoots.

Debouncing and decoupling capacitors should always be chosen as smallest as possible. For detailed information please refer the chapter 3.2 of Hardware Setup application note MCU-AN-300223.

All pins are set to input disabled (i.e. corresponding DDR and PIER bits are 0) after any reset.

Do not connect any input ports directly to V_{CC} or V_{SS} (GND)! Always use pull up or down resistors.

4 Technical information

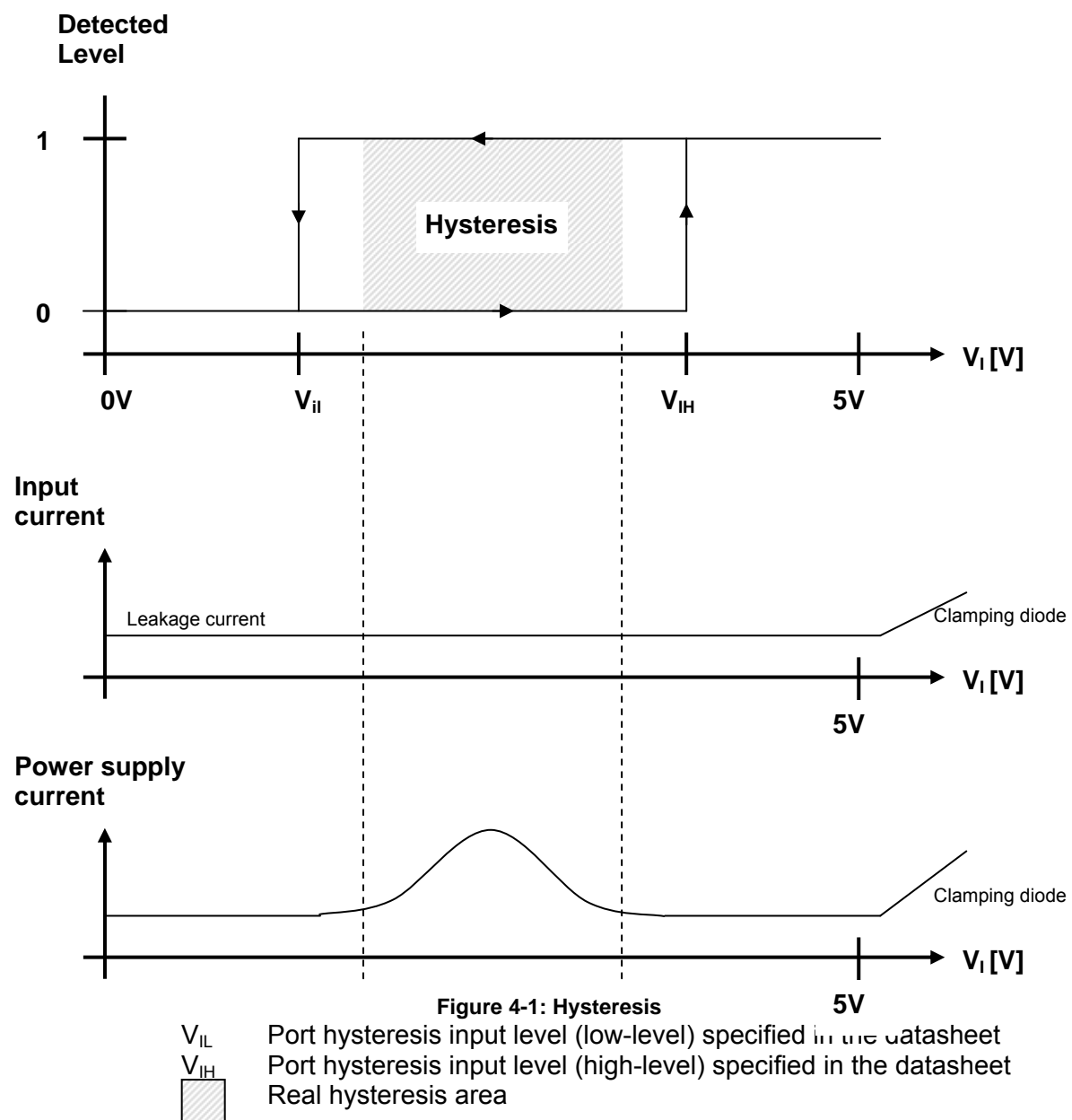
ELECTRICAL CHARACTERISTICS OF THE INPUT HYSTERESIS

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.



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 5.1, 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 a 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.

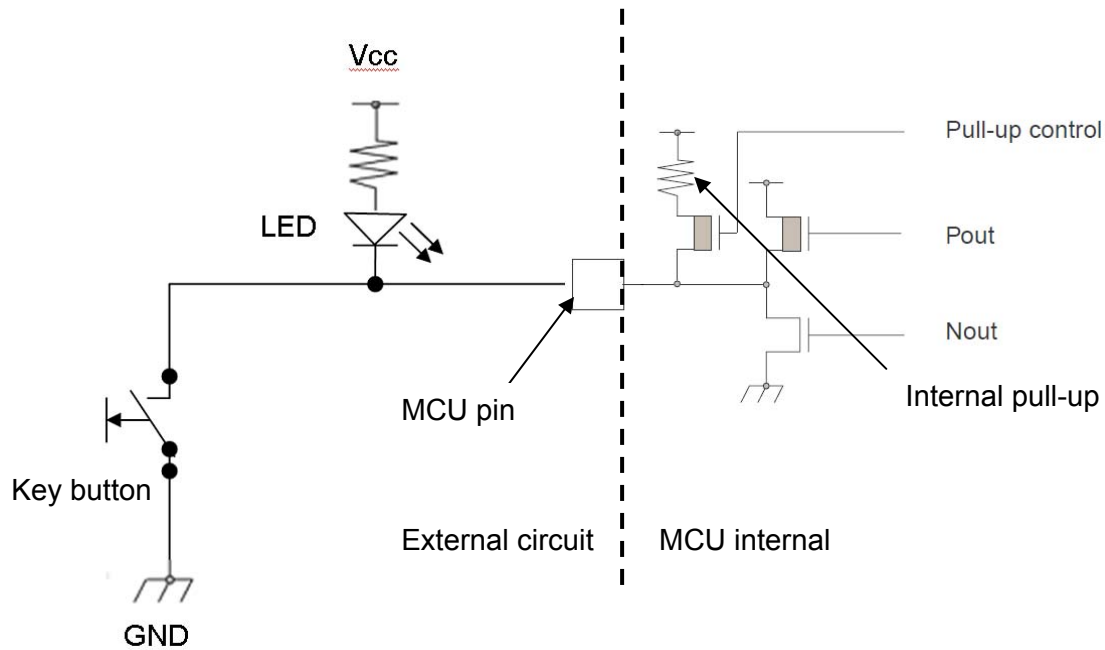


Figure 5-1: I/O-port example circuit

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 kΩ) as soon as the port is set to input. The pin reaches high-level after a typical charging time of 1 to 2 μs.

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

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 5.2 shows a code example for using port 09 pin 0 as in- and output.

```

void KeyLED_Init(void)
{
    PDR09_P0    = 0;    // Preset Port register (never output high, this might cause
                       // shortcircuit if key button is pressed)
    PUCR09_PU0 = 1;    // Enable Pull-Up resistor to be used while in input state
    PIER09_IE0 = 1;    // Enable Port Input
}

void LED_on(void)
{
    DDR09_D0 = 1;    // Switch port to output
}

void LED_off(void)
{
    DDR09_D0 = 0;    // Switch port to input (never output high, this might cause
                       // shortcircuit if key button is pressed)
}

unsigned char KeyPressed(void)
{
    if (PDR09_P0 == 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 switch 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 Port Data Register (2.2.1) 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 whether the resource output of the corresponding pin is enabled or not.

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:

96340_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 3-1: Recommended Connections for Port Input and Unused Pins	13
Figure 4-1: Hysteresis	15
Figure 5-1: I/O-port example circuit	16
Figure 5-2: Example code	17

List of Tables

Table 2-1: PDR	8
Table 2-2: Reading PDR	8
Table 2-3: DDR	8
Table 2-4: PIER	9
Table 2-5: PILR, EPILR	9
Table 2-6: PODR, PHDR	9
Table 2-7: PUCR	10