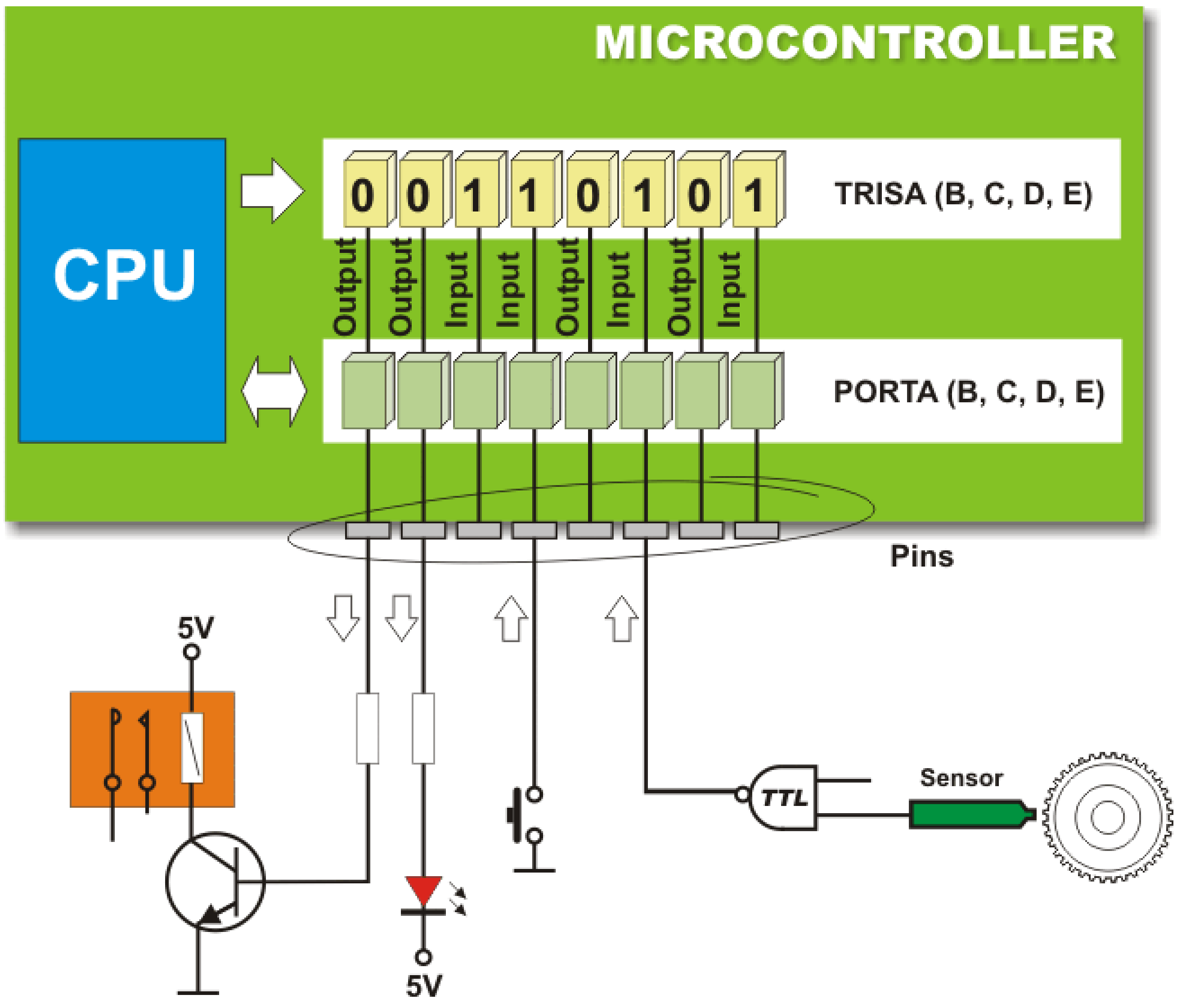
Input/Output ports :

In order to synchronize the operation of I/O ports with the internal 8-bit organization of the microcontroller, they are, similar to registers, grouped into five ports denoted by A, B, C, D and E. All of them have several features in common:

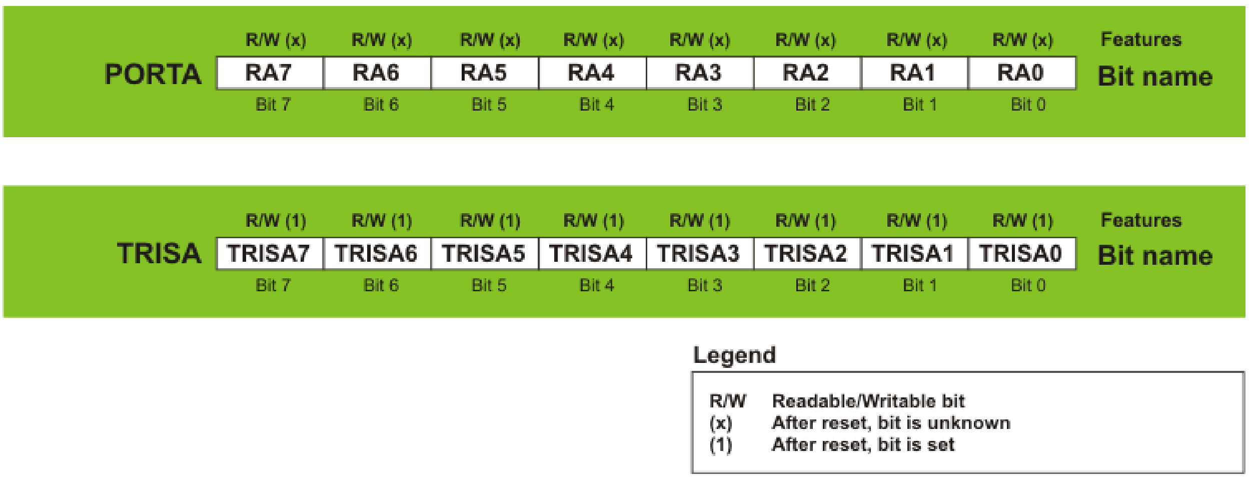
- For practical reasons, many I/O pins are multifunctional. If a pin performs any of these functions, it may not be used as a general-purpose input/output pin.
- Every port has its ‘satellite’, i.e. the corresponding TRIS register: TRISA, TRISB, TRISC etc. which determines the performance of port bits, but not their contents.

By clearing any bit of the TRIS register (bit=0), the corresponding port pin is configured as an output. Similarly, by setting any bit of the TRIS register (bit=1), the corresponding port pin is configured as an input. This rule is easy to remember 0 = **O**utput, 1 = **I**nput.



PORTA and TRISA register

Port A is an 8-bit wide, bidirectional port. Bits of the TRISA and ANSEL registers control the Port A pins. All Port A pins act as digital inputs/outputs. Five of them can also be analog inputs (denoted by AN):



	R/W (x)	R/W (x)	R/W (x)	R/W (x)	R/W (x)	R/W (x)	R/W (x)	R/W (x)	Features
PORTA	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0	Bit name
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	

	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	Features
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	Bit name
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	

Legend

R/W	Readable/Writable bit
(x)	After reset, bit is unknown
(1)	After reset, bit is set

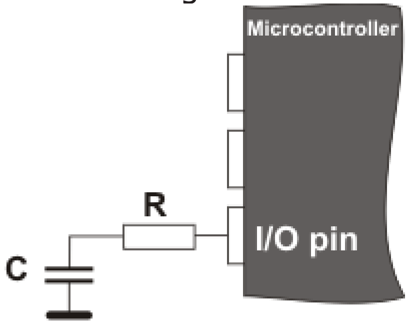
RA0 = AN0 (determined by the ANS0 bit of the ANSELregister) RA1 = AN1 (determined by the ANS1 bit of the ANSELregister) RA2 = AN2 (determined by the ANS2 bit of the ANSELregister) RA3 = AN3 (determined by the ANS3 bit of the ANSELregister) RA5 = AN4 (determined by the ANS4 bit of the ANSELregister) Similar to bits of the TRISA register determine which of the pins are to be configured as inputs and which ones as outputs, the appropriate bits of the ANSEL register determine whether pins are to be configured as analog inputs or digital inputs/outputs. Each bit of this port has an additional function related to some of the built-in peripheral units, which will be described in later chapters. This chapter covers only the RA0 pin’s additional function since it is related to port A and the ULPWU unit. **Let's do it in mikroC...**

```
// The PORTA.2 pin is configured as a digital input.
// All other PORTA pins are digital outputs

ANSEL = ANSELH = 0; // All I/O pins are configured as digital
PORTA = 0;          // All PORTA pins are cleared
TRISA = 0b00000100; // All PORTA pins except PORTA.2 are configured as outputs
...
```

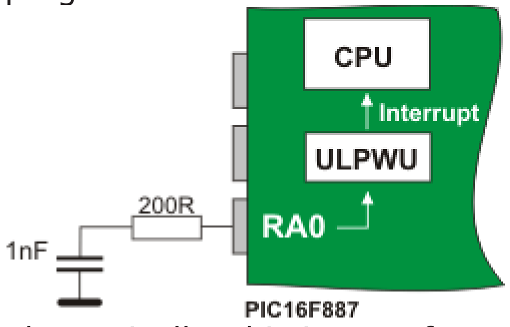
ULPWU UNIT

The microcontroller is commonly used in devices which operate periodically and completely independently using a battery power supply. Minimum power consumption is one of the priorities here. Typical examples of such applications are: thermometers, fire detection sensors and the like. It is known that a reduction in clock frequency reduces the power consumption, thus one of the most convenient solutions to this problem is to slow down the clock, i.e. to use 32KHz quartz crystal instead of 20MHz.

Setting the microcontroller to sleep mode is another step in the same direction. Still, the problem is how to wake up the microcontroller and set it to normal mode? It is obviously necessary to have an external signal to change the logic state of some of the pins. This signal must be generated by additional electronics, which causes higher power consumption of the entire device...
The ideal solution would be that the microcontroller wakes up periodically by itself, which is not impossible at all. The circuit which enables it is shown in figure on the left.



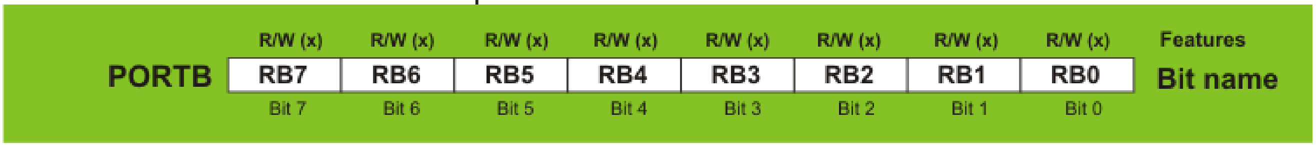
The principle of operation is simple: A pin is configured as an output and a logic one (1) is brought to it. This causes the capacitor to be charged. Immediately after this, the same pin is configured as an input. The change of logic state enables an interrupt and the microcontroller is set to *Sleep* mode. All that's left now is to wait for the capacitor to discharge by the leakage current flowing out through the input pin. When it occurs, an interrupt takes place and the microcontroller proceeds with the program execution in normal mode. The whole procedure is repeated.



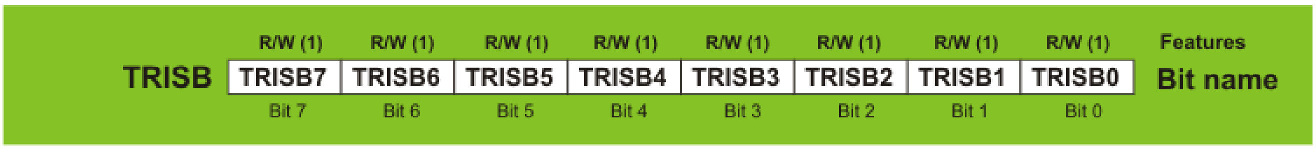
Theoretically, this is a perfect solution. The problem is that all pins able to cause an interrupt in this way are digital and have relatively large leakage current when their voltage is not close to the limit values Vdd (1) or Vss (0). In this case, the condenser is discharged for a short time since the current amounts to several hundreds of microamperes. This is why the ULPWU circuit, capable of registering slow voltage drops with minimum power consumption, was designed. Its output generates an interrupt, while its input is connected to one of the microcontroller pins. It is the RA0 pin. Referring to figure (R=200 ohms, C=1nF), discharge time is approximately 30mS, while a total consumption of the microcontroller is 1000 times lower (several hundreds of nanoamperes).

PORTB and TRISB register

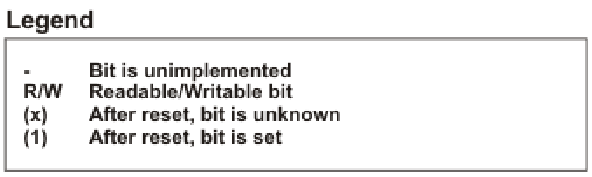
Port B is an 8-bit wide, bidirectional port. Bits of the TRISB register determine the function of its pins.



	R/W (x)	R/W (x)	R/W (x)	R/W (x)	R/W (x)	R/W (x)	R/W (x)	R/W (x)	Features
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	Bit name
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	



	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	Features
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	Bit name
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	

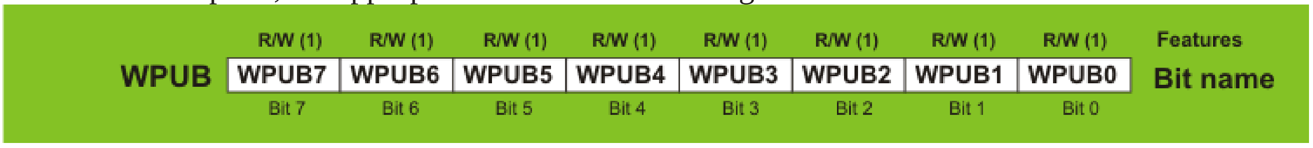


Legend

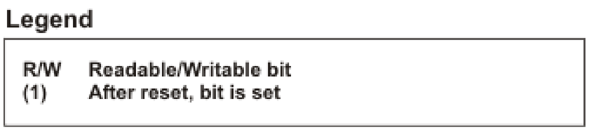
-	Bit is unimplemented
R/W	Readable/Writable bit
(x)	After reset, bit is unknown
(1)	After reset, bit is set

Similar to port A, a logic one (1) in the TRISB register configures the appropriate portB pin as an input and vice versa. Six pins of this port can act as analog inputs (AN). The bits of the ANSELH register determine whether these pins are to be configured as analog inputs or digital inputs/outputs: RB0 = AN12 (determined by the ANS12 bit of the ANSELH register) RB1 = AN10 (determined by the ANS10 bit of the ANSELH register) RB2 = AN8 (determined by the ANS8 bit of the ANSELH register) RB3 = AN9 (determined by the ANS9 bit of the ANSELH register) RB4 = AN11 (determined by the ANS11 bit of the ANSELH register) RB5 = AN13 (determined by the ANS13 bit of the ANSELH register) Each port B pin has an additional function related to some of the built-in peripheral units, which will be explained in later chapters. This port has several features which distinguish it from other ports and make its pins commonly used:

- All the port B pins have built in *pull-up* resistors, which make them ideal for connection to push buttons (keyboard), switches and optocouplers. In order to connect these resistors to the microcontroller ports, the appropriate bit of the WPUB register should be set.*



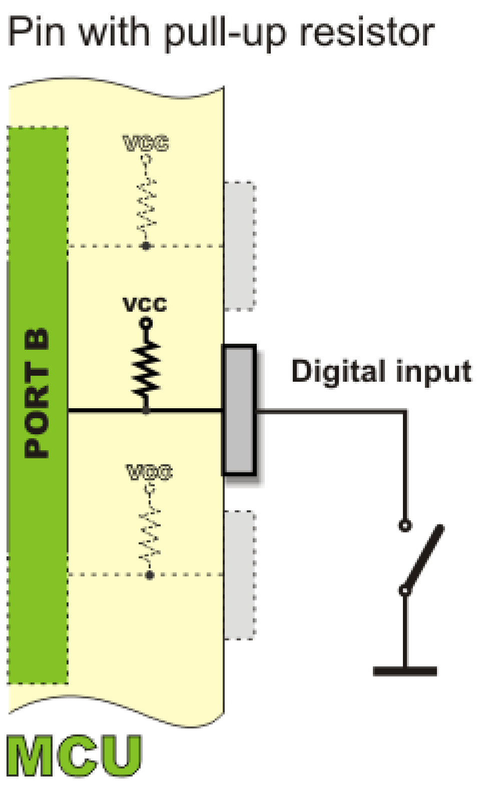
	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	Features
WPUB	WPUB7	WPUB6	WPUB5	WPUB4	WPUB3	WPUB2	WPUB1	WPUB0	Bit name
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	

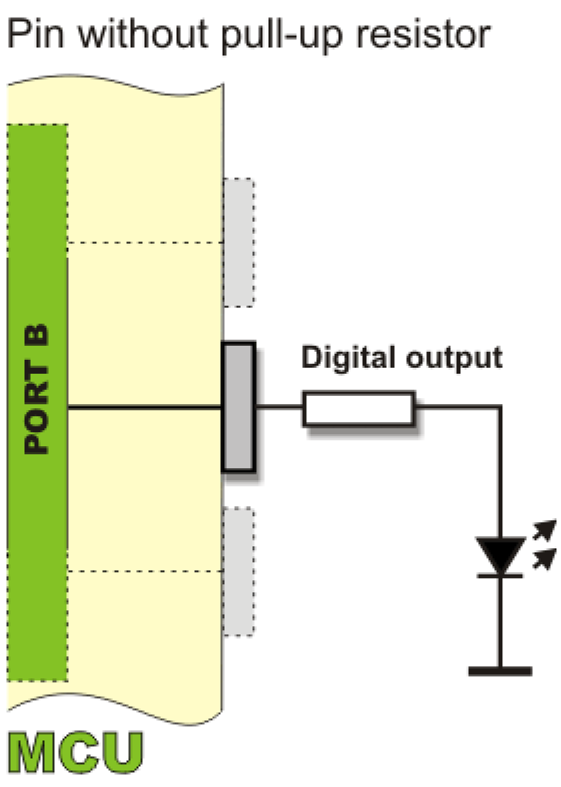


Legend

R/W	Readable/Writable bit
(1)	After reset, bit is set

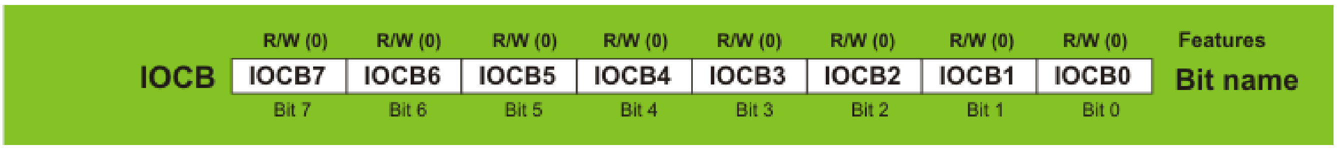
Having a high level of resistance (several tens of kiloohms), these ‘virtual’ resistors do not affect pins configured as outputs, but serves as a useful complement to inputs. As such, they are connected to the inputs of CMOS logic circuits. Otherwise, they would act as if they are floating due to their high input resistance.



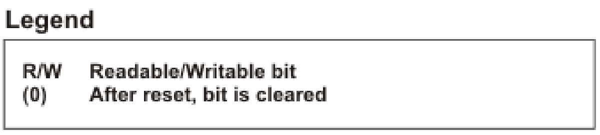


** Apart from the bits of the WPUB register, there is another bit affecting the installation of all pull-up resistors. It is the RBPU bit of the OPTION_REG.*

- If enabled, each port B bit configured as an input may cause an interrupt by changing its logic state. In order to enable pins to cause an interrupt, the appropriate bit of the IOCB register should be set.



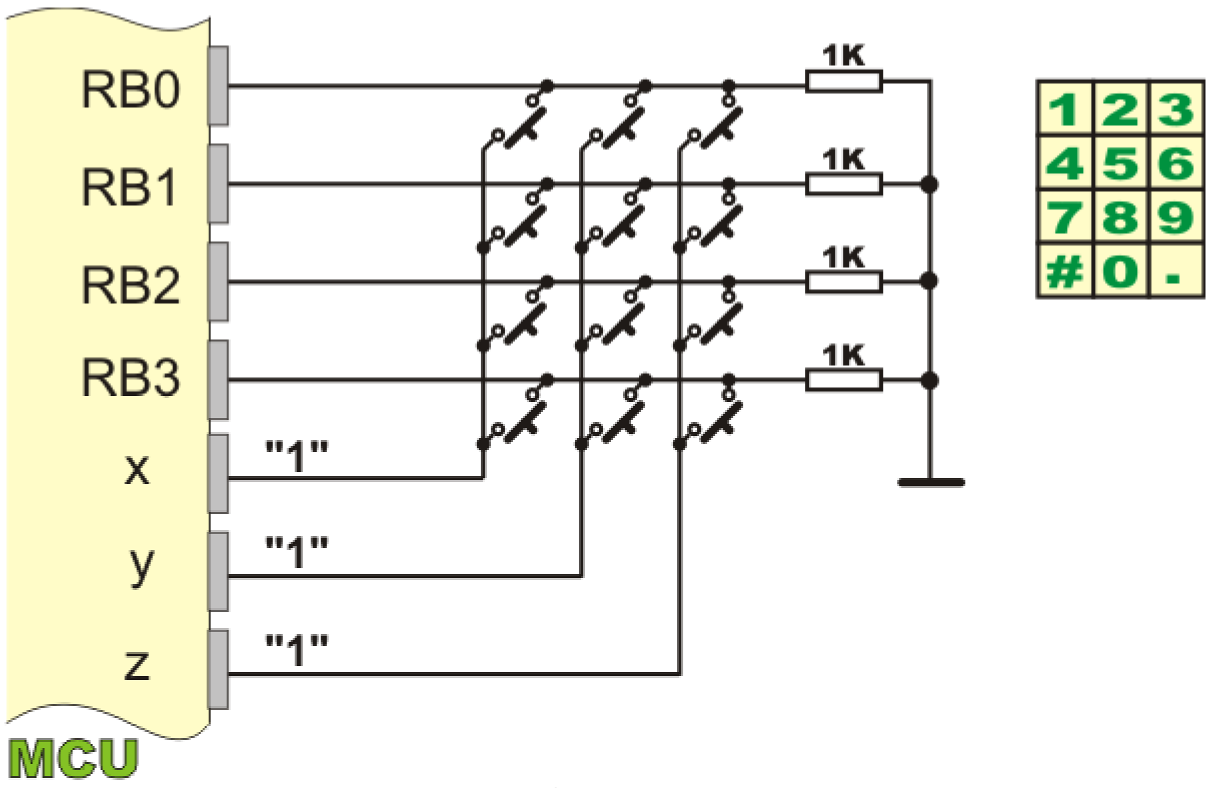
	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	Features
IOCB	IOCB7	IOCB6	IOCB5	IOCB4	IOCB3	IOCB2	IOCB1	IOCB0	Bit name
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	



Legend

R/W	Readable/Writable bit
(0)	After reset, bit is cleared

Thanks to these features, the port B pins are commonly used for checking push buttons on the keyboard because they unerringly register any button press. Thus, there is no need to ‘scan’ these inputs all the time.



When the X, Y and Z pins are configured as outputs set to logic one (1), it is only necessary to wait for an interrupt request which arrives upon any button press. After that, by combining zeros and ones on these outputs it is checked which push button is pressed. **Let's do it in mikroC...**

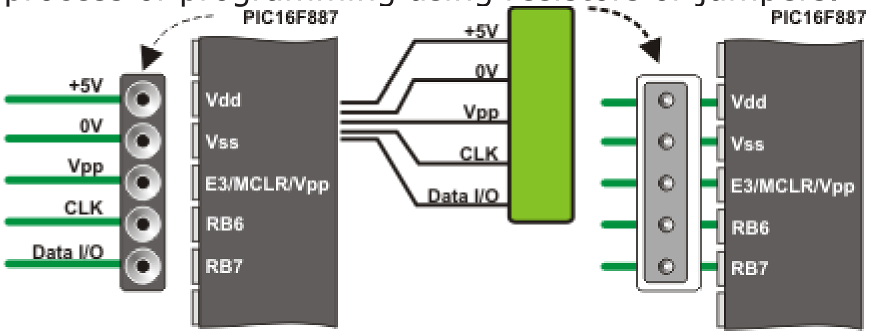
```
/* The PORTB.1 pin is configured as a digital input. Any change of its logic state will cau
se
an .i.n.terrupt. It also has a pull-up resistor. All other PORTB pins are digital outputs.*
/


ANSEL = ANSELH = 0; // All I/O pins are configured as digital
PORTB = 0;          // All PORTB pins are cleared
TRISB = 0b00000010; // All PORTB pins except PORTB.1 are configured as outputs
RBPU = 0;           // Pull-up resistors are enabled
WPUB1 = 1;          // Pull-up resistor is connected to the PORTB.1 pin
IOCB1 = 1;          // The PORTB.1 pin may cause an interrupt on logic state change
RBIE = GIE = 1;     // Interrupt is enabled
...
```

PIN RB0/INT

The RB0/INT pin is the only ‘true’ external interrupt source. It can be configured to react to signal raising edge (zero-to-one transition) or signal falling edge (one-to-zero transition). The INTEDG bit of the OPTION_REG register selects the appropriate signal.

RB6 AND RB7 PINS

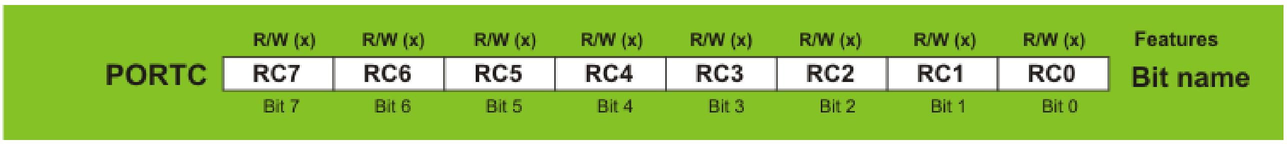
The PIC16F887 does not have any special pins for programming (the process of writing a program to ROM). Port pins, normally available as general-purpose I/O pins, are used for this purpose. To be more precise, it is about port B pins used for clock (RB6) and data transfer (RB7) during program loading. Besides, it is necessary to apply power supply voltage Vdd (5V) as well as appropriate voltage Vpp (12-14V) for FLASH memory programming. During programming, Vpp voltage is applied to the MCLR pin. You don’t have to think of all details concerning this process, nor which one of these voltages is applied first since the programmer’s electronics is in charge of that. What is very important here is that the program may be loaded to the microcontroller even after soldering it onto the target device. Normally, the loaded program can also be changed in the same way. This function is called ICSP (*In-Circuit Serial Programming*). In order to use it properly, it is necessary to plan ahead. A piece of cake! It is only necessary to install a miniature 5-pin connector onto the target device so as to provide the microcontroller with necessary programming voltages. In order to prevent these voltages from interfering with other device electronics connected to microcontroller pins, all additional peripheral devices should be disconnected during the process of programming using resistors or jumpers.



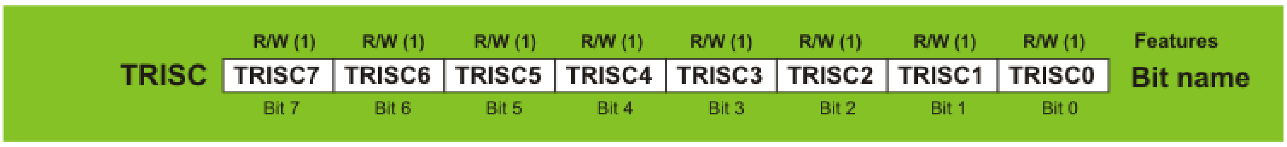
As you can see, voltages applied to programmer's socket pins are the same as those used during ICSP programming

PORTC and TRISC register

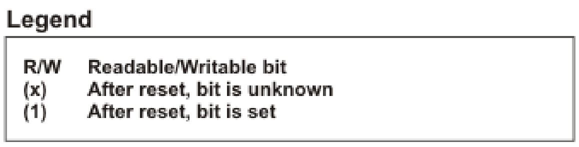
Port C is an 8-bit wide, bidirectional port. Bits of the TRISC register determine the function of its pins. Similar to other ports, a logic one (1) in the TRISC register configures the appropriate portC pin as an input.



	R/W (x)	R/W (x)	R/W (x)	R/W (x)	R/W (x)	R/W (x)	R/W (x)	R/W (x)	Features
PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	Bit name
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	



	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	Features
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	Bit name
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	



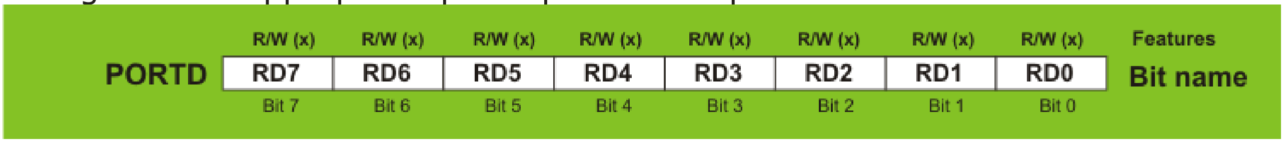
Legend

R/W	Readable/Writable bit
(x)	After reset, bit is unknown
(1)	After reset, bit is set

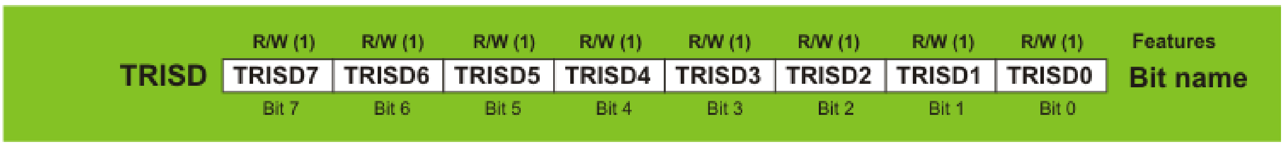
All additional functions of port C bits will be explained later.

PORTD and TRISD register

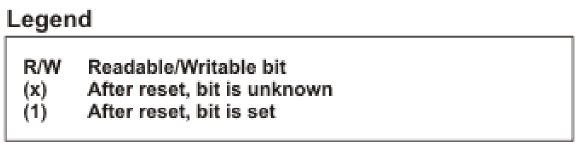
Port D is an 8-bit wide, bidirectional port. Bits of the TRISD register determine the function of its pins. A logic one (1) in the TRISD register configures the appropriate portD pin as an input.



	R/W (x)	R/W (x)	R/W (x)	R/W (x)	R/W (x)	R/W (x)	R/W (x)	R/W (x)	Features
PORTD	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	Bit name
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	



	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	Features
TRISD	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	Bit name
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	

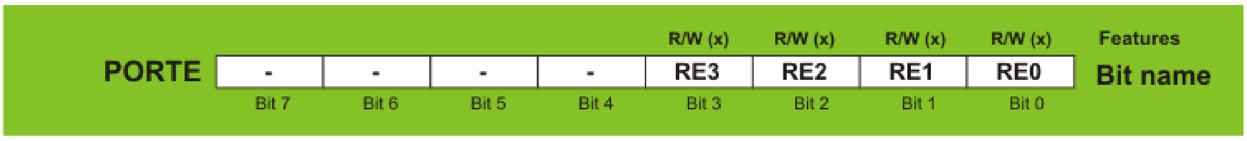


Legend

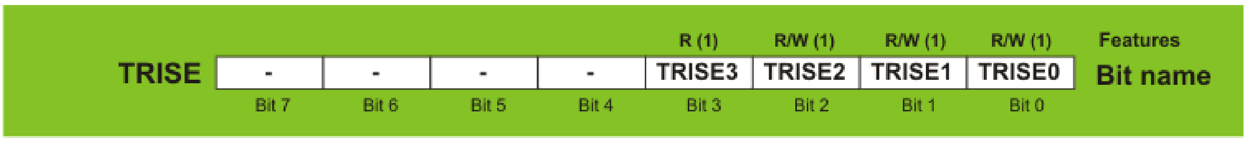
R/W	Readable/Writable bit
(x)	After reset, bit is unknown
(1)	After reset, bit is set

PORTE and TRISE register

Port E is a 4-bit wide, bidirectional port. The TRISE register's bits determine the function of its pins. Similar to other ports, a logic one (1) in the TRISE register configures the appropriate portE pin as an input. The exception is the RE3 pin which is always configured as an input.



					R/W (x)	R/W (x)	R/W (x)	R/W (x)	Features
PORTE	-	-	-	-	RE3	RE2	RE1	RE0	Bit name
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	



					R (1)	R/W (1)	R/W (1)	R/W (1)	Features
TRISE	-	-	-	-	TRISE3	TRISE2	TRISE1	TRISE0	Bit name
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	

Legend

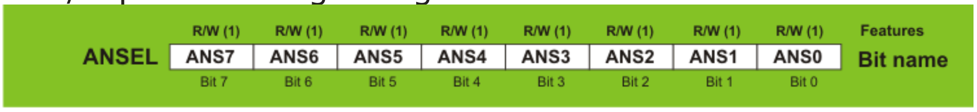
-	Bit is unimplemented
R/W	Readable/Writable bit
R	Readable bit
(x)	After reset, bit is unknown
(1)	After reset, bit is set

Similar to ports A and B, three pins can be configured as analog inputs in this case. The ANSELH register bits determine whether a pin will act as an analog input (AN) or digital input/output: RE0 = AN5 (determined by the ANS5 bit of the ANSELregister); RE1 = AN6 (determined by the ANS6 bit of the ANSELregister); and RE2 = AN7 (determined by the ANS7 bit of the ANSELregister). **Let's do it in mikroC...**

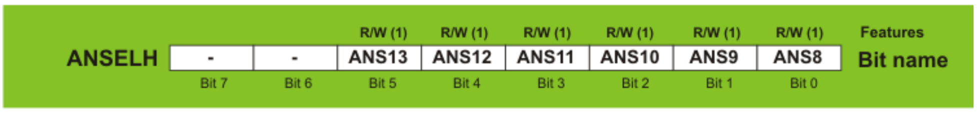
```
/* The PORTE.0 pin is configured as an analog input while another three pins of the same
   port are configured as digital. */
...
ANSEL = 0b00100000; // The PORTE.0 pin is configured as analog
ANSELH = 0;         // All other I/O pins are configured as digital
TRISE = 0b00000001; // All PORTE pins except PORTE.0 are configured as outputs
PORTE = 0;          // All PORTE pins are cleared
...
```

ANSEL and ANSELH register

The ANSEL and ANSELH registers are used to configure the input mode of an I/O pin to analog or digital.



	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	Features
ANSEL	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	Bit name
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	

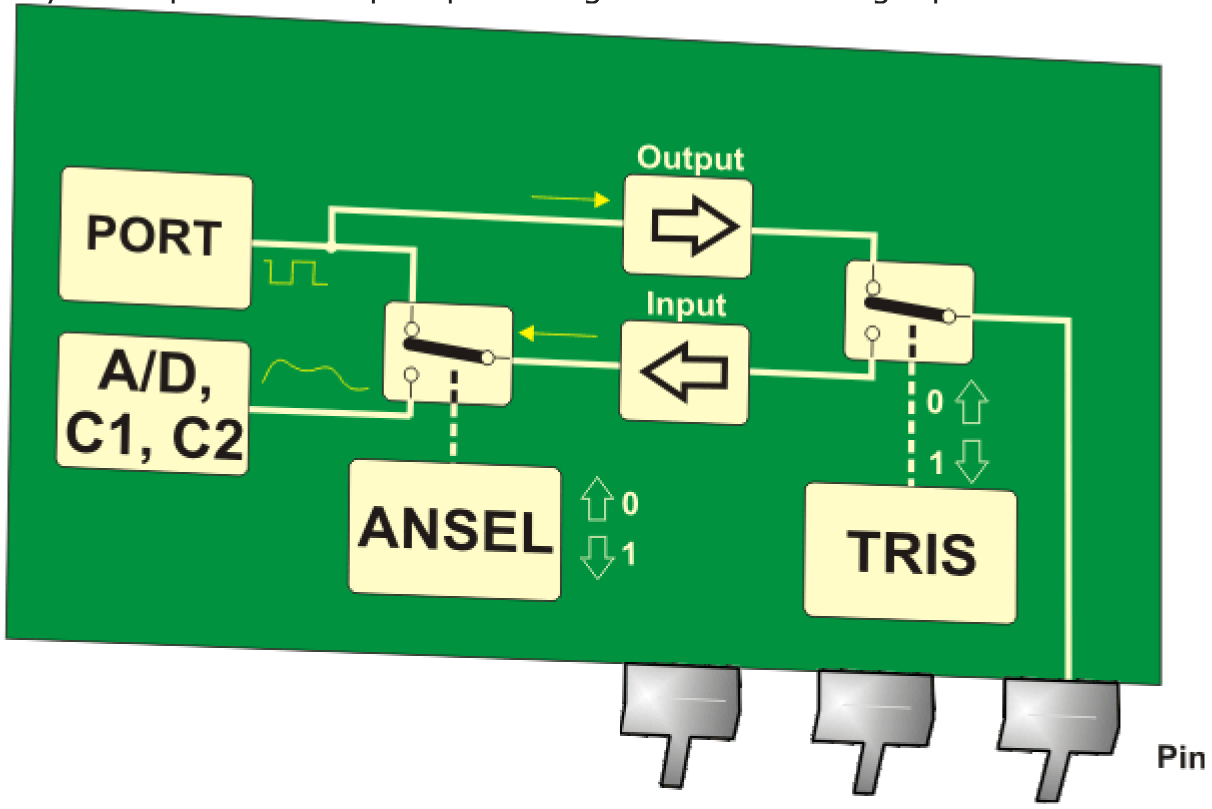


			R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	Features
ANSELH	-	-	ANS13	ANS12	ANS11	ANS10	ANS9	ANS8	Bit name
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	

Legend

-	Bit is unimplemented
R/W	Readable/Writable bit
(1)	After reset, bit is set

The rule is: To configure a pin as an analog input, the appropriate bit of the ANSEL or ANSELH registers must be set (1). To configure a pin as a digital input/output, the appropriate bit must be cleared (0). The state of the ANSEL bits has no influence on digital output functions. The result of any attempt to read a port pin configured as an analog input will be 0.



In Short

You will probably never write a program which doesn't use ports so the effort you make to learn all about them will definately pay off. Anyway, they are probaly the simplest modules within the microcontroller. This is how they are used:

- When designing a device, select a port through which the microcontroller will communicate to peripheral environment. If you use only digital inputs/outputs, select any port you want. If you intend to use some of the analog inputs, select the appropriate ports supporting such a pin configuration (AN0-AN13).
- Each port pin may be configured as either input or output. Bits of the TRISA, TRISB, TRISC, TRISD and TRISE registers determine how the appropriate port pins- PORTA, PORTB, PORTC, PORTD and PORTE will act. Simply...
- If you use some of the analog inputs, it is first necessary to set the appropriate bits of the ANSEL and ANSELH registers at the beginning of the program.

- If you use switches and push buttons as input signal source, connect them to port B pins because they have pull-up resistors. The use of these resistors is enabled by the RBPU bit of the OPTION_REG register, whereas the installation of individual resistors is enabled by bits of the WPUB register.
- It is usually necessary to respond as soon as input pins change their logic state. However, it is not necessary to write a program for checking pins' logic state. It is far simpler to connect such inputs to the PORTB pins and enable an interrupt to occur on every voltage change. Bits of the IOCB and INTCON registers are in charge of that.