Tishk International University
Engineering Faculty
Mechatronics Engineering Department
Embedded Systems and Real-time
Week 5

Embedded Systems and Real-time
PIC Microcontroller

Dr. Rand Basil Alhashimie
Email: rand.basil@tiu.edu.iq

Objective

- Understand the general architecture of PIC Microcontroller

Lecture Outline

- General Overview
- General Architecture
- Applications
- Programming

Background

- PIC microcontrollers designed by Microchip Technology are likely the right choice for you if you are the beginner.
- The real name of this microcontroller is PICmicro (Peripheral Interface Controller), but it is better known as PIC. Its first ancestor was designed in 1975 by General Instruments. This chip called PIC1650 was meant for totally different purposes. Not longer than ten years after, by adding
- EEPROM memory, this circuit was transformed into a real PIC microcontroller. Nowadays, Microchip Technology announces a manufacturing of the 5 billionth sample...
- In order you can better understand the reasons for its popularity, we will briefly describe several important things.

PIC Microcontroller Today

- Looking at the range of PIC microcontrollers today. There are literally hundreds of different devices, offered in different packages, for different applications. Let us therefore try to identify the characteristics that all of these have in common. At the time of writing, all PIC microcontrollers are low-cost, self-contained, 8-bit, Harvard structure, pipelined, RISC, single accumulator (the Working or W register), with fixed reset and interrupt vectors.
- Today, Microchip offers five main families of microcontrollers, whose features are summarised in Table below. It is possible to see clear evolution from one family to the other, so knowledge of one readily leads to knowledge of another. Every member of any one family shares the same core architecture and instruction set. The families are identified primarily by the first two digits of the device code.

PIC family	Stack size (words)	Instruction word size	Number of instructions	Interrupt vectors
12CXXX/12FXXX	2	12- or 14-bit	33	None
16C5XX/16F5XX	2	12-bit	33	None
16CXXX/16FXXX	8	14-bit	35	1
17CXXX	16	16-bit	58, including hardware multiply	4
18CXXX/18FXXX	32	16-bit	75, including hardware multiply	2 (prioritised)

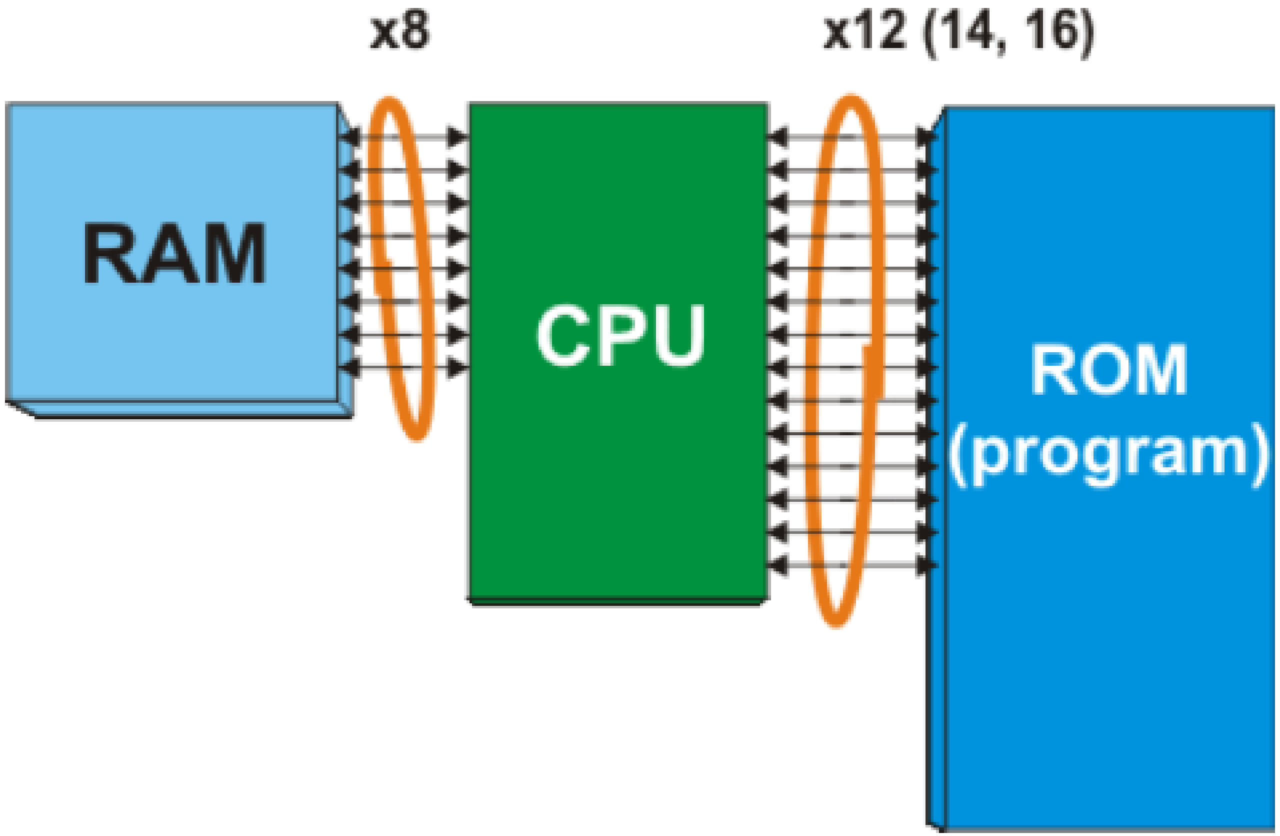
PIC Microcontroller Today

- The alphabetic character that follows gives some indication of the technology used.
- The **‘C’ insert implies CMOS** technology, where CMOS stands for Complementary Metal Oxide Semiconductor, the leading semiconductor technology for implementing low-power logic systems.
- The **‘F’** insert indicates incorporation of Flash memory technology (still using CMOS as the core technology).
- An **‘A’** after the number indicates a technological upgrade on the first issue device.
- An **‘X’** indicates that a certain digit can take a number of values, the one taken being unimportant to the overall number quoted.
- For example, the 16C84 was the first of its kind. It was later reissued as the 16F84, incorporating Flash memory technology. It was then reissued as the 16F84A, with certain further technological upgrades.
- Microchip also used to give each family a name. Thus, their first family, the 16C5XX, was called the ‘baseline’ family. The development of this, with device numbers starting ‘16C’ or ‘16F’ (and a fourth digit that was *not* 5), was called the ‘mid-range’ family. The powerful evolution of this, with codes starting ‘17C’, was called the ‘high-end’ family. As the further families developed, with both very simple and advanced architectures, this naming convention has lost prominence, although the terminology is still found. For simplicity, to identify a PIC family, this book will refer to ‘12 Series’, ‘16 Series’, ‘18 Series’ and so on. Let us survey each family in turn.

Family	ROM [Kbytes]	RAM [bytes]	Pins	Clock Freq. [MHz]	A/D Inputs	Resolution of A/D Converter	Compar-ators	8/16 – bit Timers	Serial Comm.	PWM Outputs	Others
Base-Line 8 - bit architecture, 12-bit Instruction Word Length											
PIC10FXXX	0.375 - 0.75	16 - 24	6 - 8	4 - 8	0 - 2	8	0 - 1	1 x 8	-	-	-
PIC12FXXX	0.75 - 1.5	25 - 38	8	4 - 8	0 - 3	8	0 - 1	1 x 8	-	-	EEPROM
PIC16FXXX	0.75 - 3	25 - 134	14 - 44	20	0 - 3	8	0 - 2	1 x 8	-	-	EEPROM
PIC16HVXXX	1.5	25	18 - 20	20	-	-	-	1 x 8	-	-	Vdd = 15V
Mid-Range 8 - bit architecture, 14-bit Instruction World Length											
PIC12FXXX	1.75 - 3.5	64 - 128	8	20	0 - 4	10	1	1 - 2 x 8 1 x 16	-	0 - 1	EEPROM
PIC12HVXXX	1.75	64	8	20	0 - 4	10	1	1 - 2 x 8 1 x 16	-	0 - 1	-
PIC16FXXX	1.75 - 14	64 - 368	14 - 64	20	0 - 13	8 or 10	0 - 2	1 - 2 x 8 1 x 16	USART I2C SPI	0 - 3	-
PIC16HVXXX	1.75 - 3.5	64 - 128	14 - 20	20	0 - 12	10	2	2 x 8 1 x 16	USART I2C SPI	-	-
High-End 8 - bit architecture, 16-bit Instruction Word Length											
PIC18FXXX	4 - 128	256 - 3936	18 - 80	32 - 48	4 - 16	10 or 12	0 - 3	0 - 2 x 8 2 - 3 x 16	USB2.0 CAN2.0 USART I2C SPI	0 - 5	-
PIC18FXXJXX	8 - 128	1024 - 3936	28 - 100	40 - 48	10 - 16	10	2	0 - 2 x 8 2 - 3 x 16	USB2.0 USART Ethernet I2C SPI	2 - 5	-
PIC18FXXKXX	8 - 64	768 - 3936	28 - 44	64	10 - 13	10	2	1 x 8 3 x 16	USART I2C SPI	2	-

Background

- All PIC microcontrollers use harvard architecture, which means that their program memory is connected to CPU via more than 8 lines. Depending on the bus width, there are 12-, 14- and 16-bit microcontrollers. The table above shows the main features of these three categories.



Overview – Part 1

PIC16F887 is one of the latest products of *Microchip*. It features all the components which upgraded microcontrollers normally have. For its low price, wide range of application, high quality and easy availability, it is an ideal solution in applications such as: control of different processes in industry, machine control device, measurement of different values etc. Some of its main features are listed below:

1. **RISC architecture**

   Only 35 instructions to learn

   All single-cycle instructions except branches

2. **Operating frequency 0-20 MHz Precision internal oscillator**

   Factory calibrated

   Software selectable frequency range of 8MHz to 31KHz

3. **Power supply voltage 2.0-5.5V**

   Consumption: 220uA (2.0V, 4MHz), 11uA (2.0 V, 32 KHz) 50nA (stand-by mode)

Overview – Part 2

4. **Power-Saving Sleep Mode**
5. **Brown-out Reset (BOR) with software control option 35 input/output pins**

   High current source/sink for direct LED drive

   software and individually programmable *pull-up* resistor Interrupt-on-Change pin
6. **8K ROM memory in FLASH technology**

   Chip can be reprogrammed up to 100.000 times
7. ***In-Circuit Serial Programming* Option**

   Chip can be programmed even embedded in the target device
8. **256 bytes EEPROM memory**

   Data can be written more than 1.000.000 times

Overview – Part 3

9. 368 bytes RAM memory A/D converter:

14-channels

10-bit resolution

10. Three independent timers/counters Watch-dog timer

11. Two Analog comparators

12. Fixed voltage reference (0.6V) Programmable on-chip voltage reference

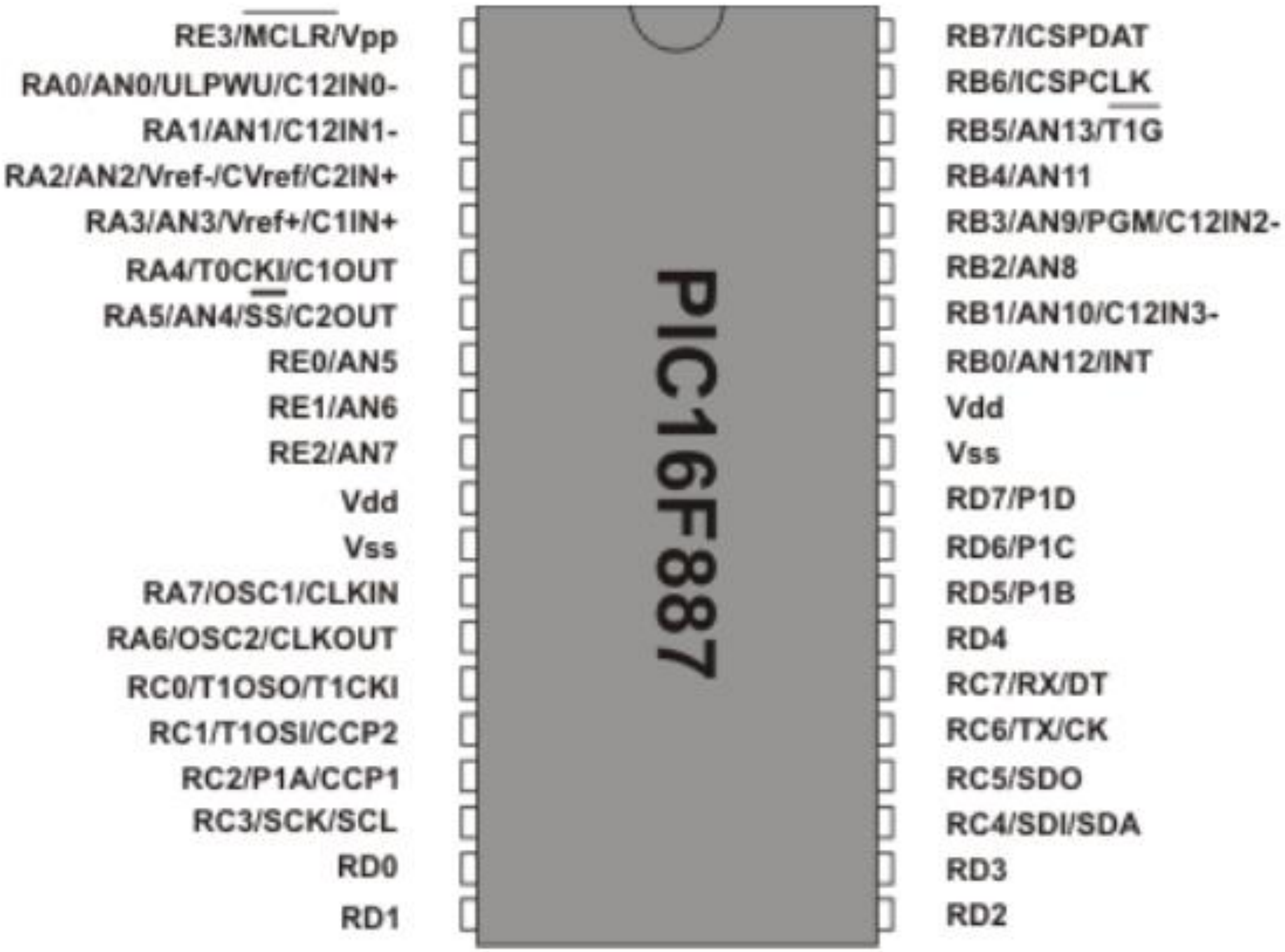
13. PWM output steering control Enhanced USART module

Supports RS-485, RS-232 and LIN2.0

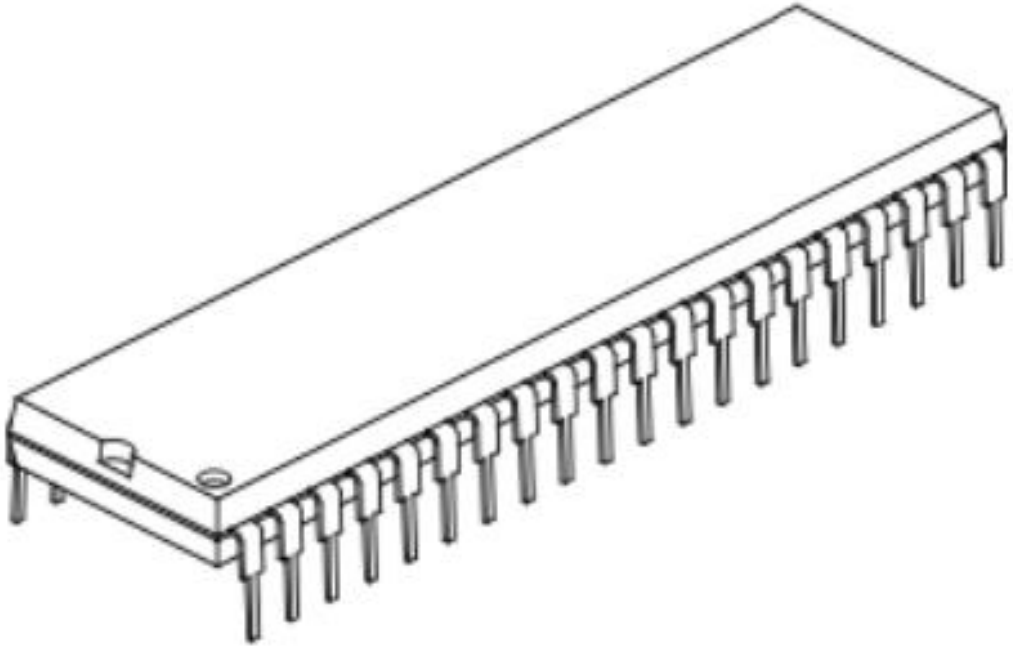
Auto-Baud Detect

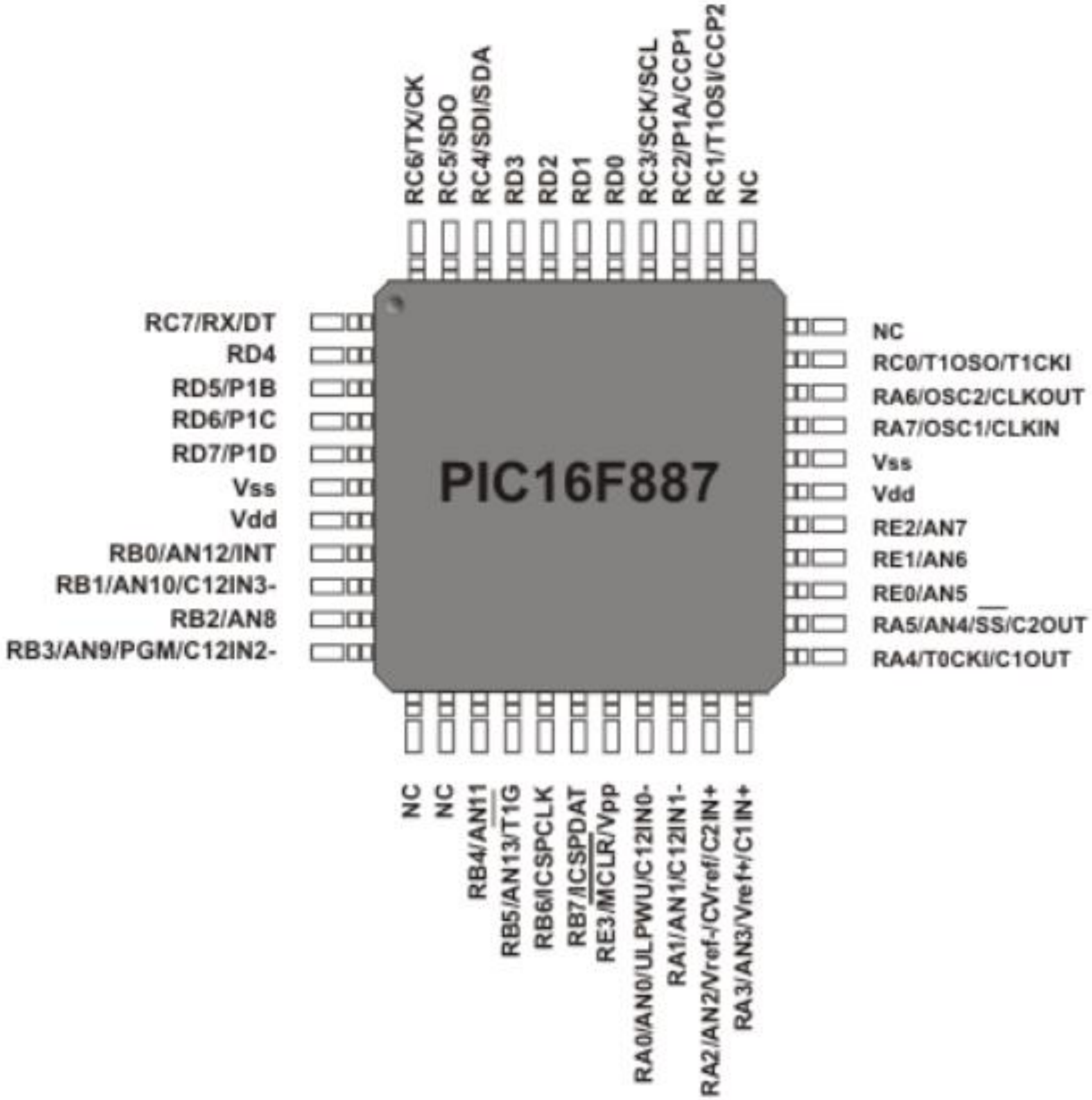
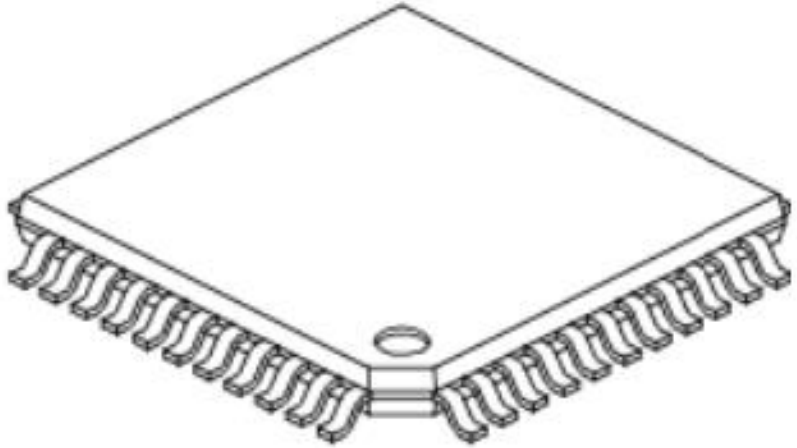
14. Master Synchronous Serial Port (MSSP)

Supports SPI and I2C mode



PIC16F887
RE3/MCLR/Vpp
RA0/AN0/ULPWU/C12IN0-
RA1/AN1/C12IN1-
RA2/AN2/Vref-/CVref/C2IN+
RA3/AN3/Vref+/C1IN+
RA4/T0CKI/C1OUT
RA5/AN4/SS/C2OUT
RE0/AN5
RE1/AN6
RE2/AN7
Vdd
Vss
RA7/OSC1/CLKIN
RA6/OSC2/CLKOUT
RC0/T1OSO/T1CKI
RC1/T1OSI/CCP2
RC2/P1A/CCP1
RC3/SCK/SCL
RD0
RD1
RB7/ICSPDAT
RB6/ICSPCLK
RB5/AN13/T1G
RB4/AN11
RB3/AN9/PGM/C12IN2-
RB2/AN8
RB1/AN10/C12IN3-
RB0/AN12/INT
Vdd
Vss
RD7/P1D
RD6/P1C
RD5/P1B
RD4
RC7/RX/DT
RC6/TX/CK
RC5/SDO
RC4/SDI/SDA
RD3
RD2

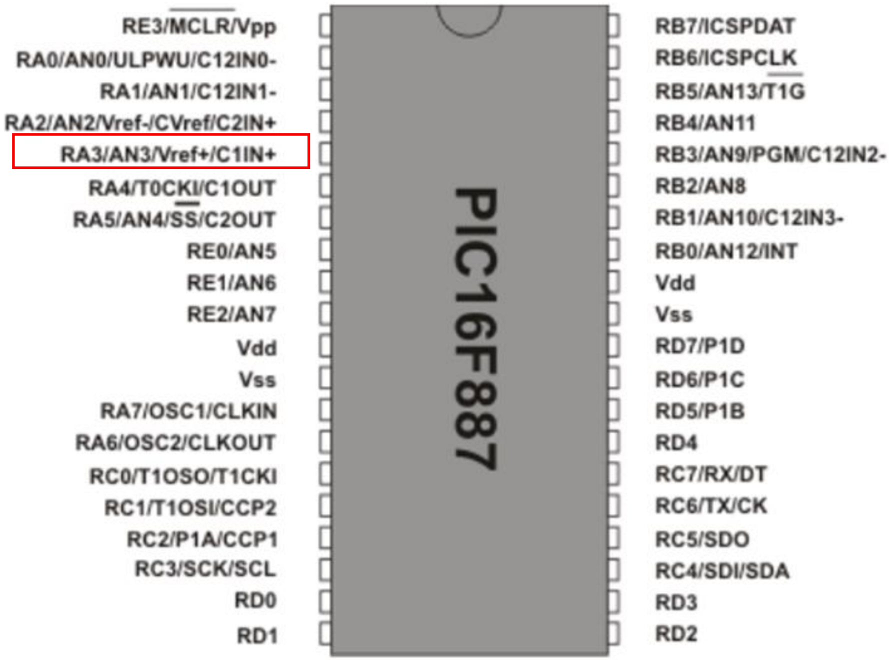


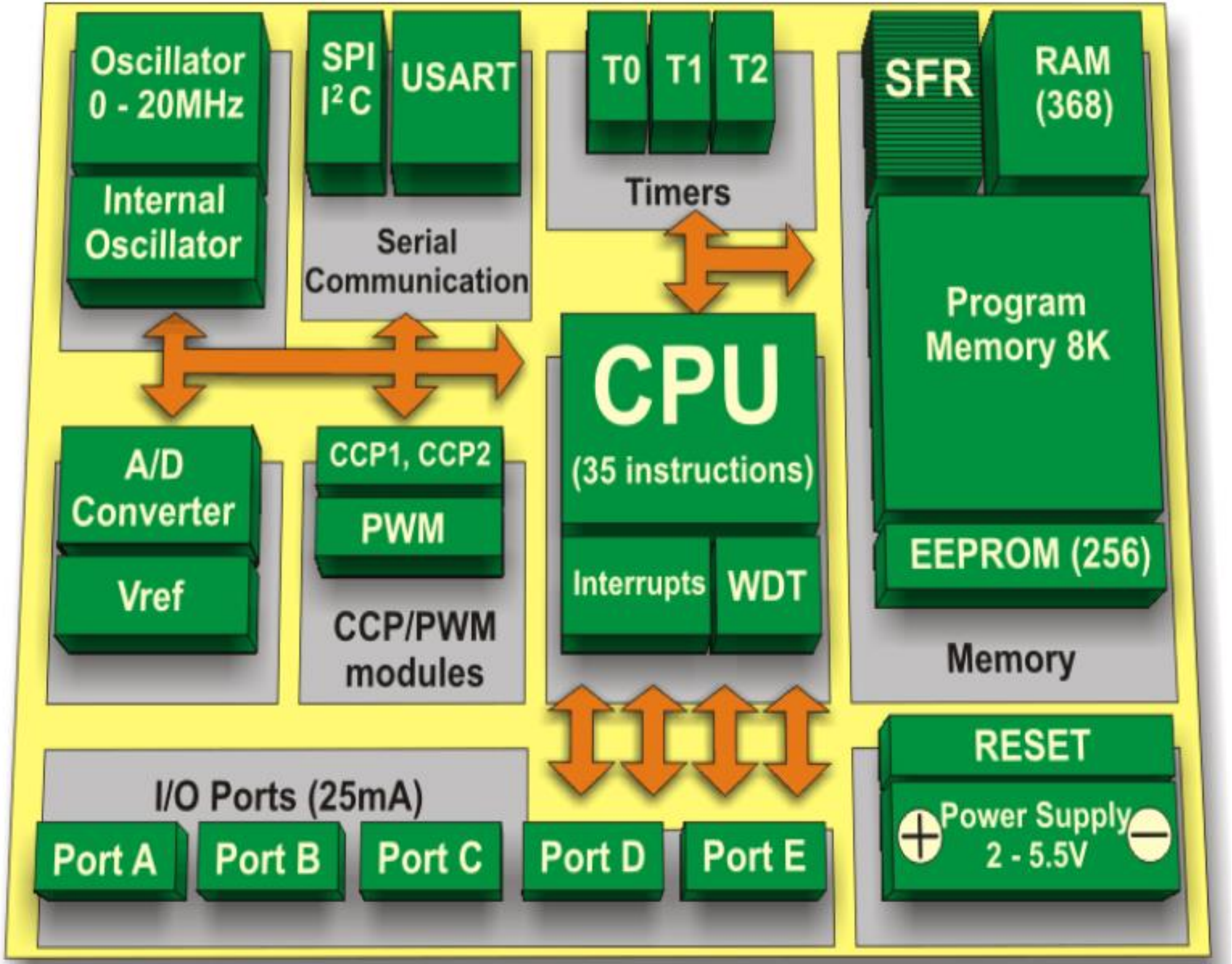

PIC16F887
RC6/TX/CK
RC5/SDO
RC4/SDI/SDA
RD3
RD2
RD1
RD0
RC3/SCK/SCL
RC2/P1A/CCP1
RC1/T1OSI/CCP2
NC
RC7/RX/DT
RD4
RD5/P1B
RD6/P1C
RD7/P1D
Vss
Vdd
RB0/AN12/INT
RB1/AN10/C12IN3-
RB2/AN8
RB3/AN9/PGM/C12IN2-
NC
RC0/T1OSO/T1CKI
RA6/OSC2/CLKOUT
RA7/OSC1/CLKIN
Vss
Vdd
RE2/AN7
RE1/AN6
RE0/AN5
RA5/AN4/SS/C2OUT
RA4/T0CKI/C1OUT
NC
NC
RB4/AN11
RB5/AN13/T1G
RB6/ICSPCLK
RB7/ICSPDAT
RE3/MCLR/Vpp
RA0/AN0/ULPWU/C12IN0-
RA1/AN1/C12IN1-
RA2/AN2/Vref-/CVref/C2IN+
RA3/AN3/Vref+/C1IN+


Pin Description

Most pins are multi-functional. For example, designator RA3/AN3/Vref+/C1IN+ for the fifth pin specifies the following functions:

- RA3 Port A third digital input/output
- AN3 Third analog input
- Vref+ Positive voltage reference
- C1IN+ Comparator C1positive input
- This small trick is often used because it makes the microcontroller package more compact without affecting its functionality. These various pin functions cannot be used simultaneously, but can be changed at any point during operation.




Oscillator
0 - 20MHz
Internal
Oscillator
SPI
I²C
USART
Serial
Communication
T0
T1
T2
Timers
SFR
RAM
(368)
Program
Memory 8K
CPU
(35 instructions)
Interrupts
WDT
A/D
Converter
Vref
CCP1, CCP2
PWM
CCP/PWM
modules
EEPROM (256)
Memory
RESET
Power Supply
2 - 5.5V
I/O Ports (25mA)
Port A
Port B
Port C
Port D
Port E

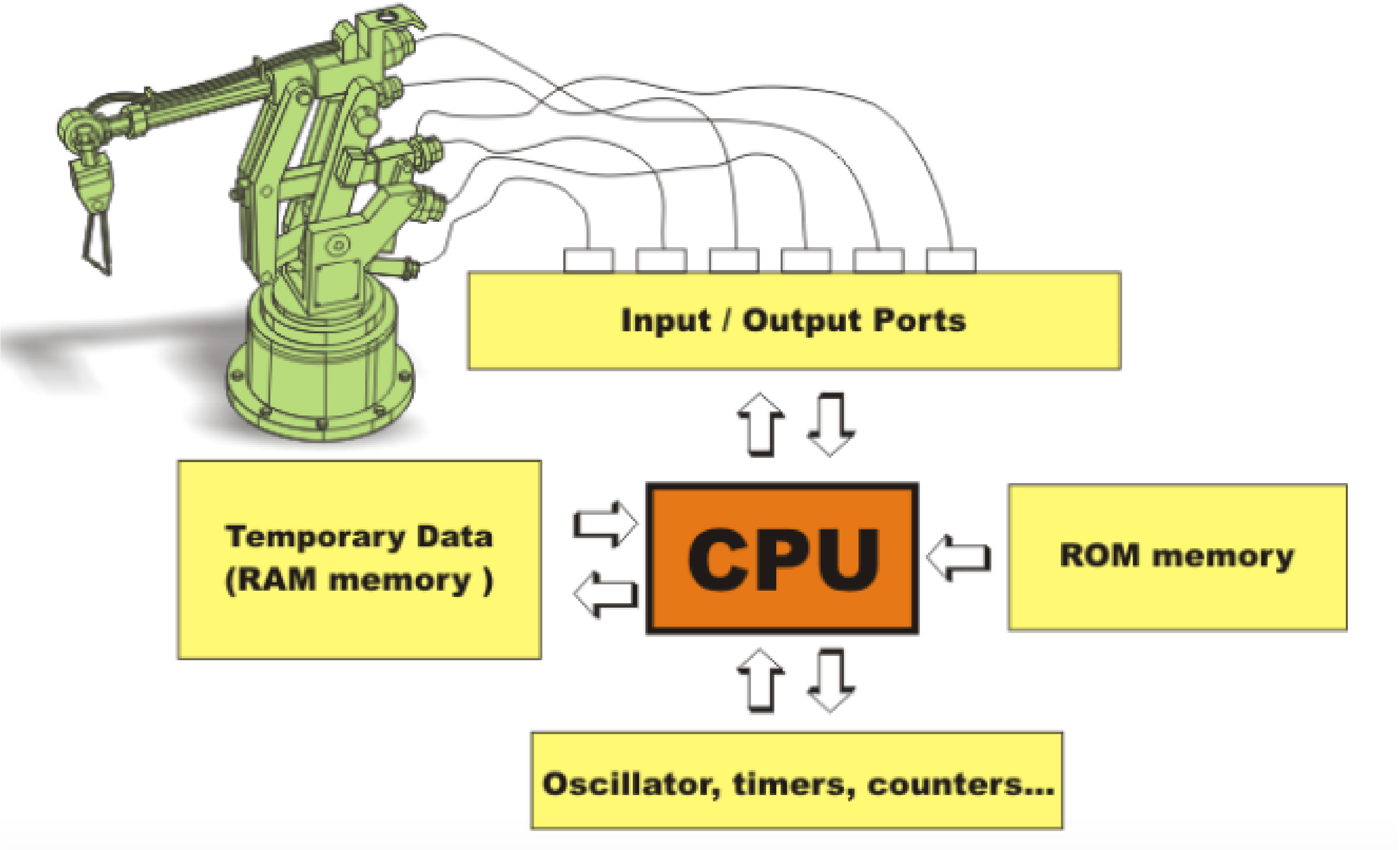
Name	Number (DIP 40)	Function	Description
RE3/MCLR/Vpp	1	RE3	General purpose input Port E
		MCLR	Reset pin. Low logic level on this pin resets microcontroller.
		Vpp	Programming voltage
RA0/AN0/ULPWU/C12IN0-	2	RA0	General purpose I/O port A
		AN0	A/D Channel 0 input
		ULPWU	Stand-by mode deactivation input
		C12IN0-	Comparator C1 or C2 negative input
RA1/AN1/C12IN1-	3	RA1	General purpose I/O port A
		AN1	A/D Channel 1
		C12IN1-	Comparator C1 or C2 negative input
RA2/AN2/Vref-/CVref/C2IN+	4	RA2	General purpose I/O port A
		AN2	A/D Channel 2
		Vref-	A/D Negative Voltage Reference input
		CVref	Comparator Voltage Reference Output
		C2IN+	Comparator C2 Positive Input
RA3/AN3/Vref+/C1IN+	5	RA3	General purpose I/O port A
		AN3	A/D Channel 3
		Vref+	A/D Positive Voltage Reference Input
		C1IN+	Comparator C1 Positive Input
RA4/T0CKI/C1OUT	6	RA4	General purpose I/O port A
		T0CKI	Timer T0 Clock Input
		C1OUT	Comparator C1 Output
RA5/AN4/SS/C2OUT	7	RA5	General purpose I/O port A
		AN4	A/D Channel 4
		SS	SPI module Input (*Slave Select*)
		C2OUT	Comparator C2 Output
RE0/AN5	8	RE0	General purpose I/O port E
		AN5	A/D Channel 5
RE1/AN6	9	RE1	General purpose I/O port E
		AN6	A/D Channel 6
RE2/AN7	10	RE2	General purpose I/O port E
		AN7	A/D Channel 7
Vdd	11	+	Positive supply
Vss	12	-	Ground (GND)

Name	Number (DIP 40)	Function	Description
RA7/OSC1/CLKIN	13	RA7	General purpose I/O port A
		OSC1	Crystal Oscillator Input
		CLKIN	External Clock Input
RA6/OSC2/CLKOUT	14	OSC2	Crystal Oscillator Output
		CLKO	Fosc/4 Output
		RA6	General purpose I/O port A
RC0/T1OSO/T1CKI	15	RC0	General purpose I/O port C
		T1OSO	Timer T1 Oscillator Output
		T1CKI	Timer T1 Clock Input
RC1/T1OSO/T1CKI	16	RC1	General purpose I/O port C
		T1OSI	Timer T1 Oscillator Input
		CCP2	CCP1 and PWM1 module I/O
RC2/P1A/CCP1	17	RC2	General purpose I/O port C
		P1A	PWM Module Output
		CCP1	CCP1 and PWM1 module I/O
RC3/SCK/SCL	18	RC3	General purpose I/O port C
		SCK	MSSP module Clock I/O in SPI mode
		SCL	MSSP module Clock I/O in I^2C mode
RD0	19	RD0	General purpose I/O port D
RD1	20	RD1	General purpose I/O port D
RD2	21	RD2	General purpose I/O port D
RD3	22	RD3	General purpose I/O port D
RC4/SDI/SDA	23	RC4	General purpose I/O port A
		SDI	MSSP module *Data* input in SPI mode
		SDA	MSSP module *Data* I/O in I^2C mode
RC5/SDO	24	RC5	General purpose I/O port C
		SDO	MSSP module *Data* output in SPI mode
RC6/TX/CK	25	RC6	General purpose I/O port C
		TX	USART Asynchronous Output
		CK	USART Synchronous Clock
RC7/RX/DT	26	RC7	General purpose I/O port C
		RX	USART Asynchronous Input
		DT	USART Synchronous Data

Name	Number (DIP 40)	Function	Description
RD4	27	RD4	General purpose I/O port D
RD5/P1B	28	RD5	General purpose I/O port D
		P1B	PWM Output
RD6/P1C	29	RD6	General purpose I/O port D
		P1C	PWM Output
RD7/P1D	30	RD7	General purpose I/O port D
		P1D	PWM Output
Vss	31	-	Ground (GND)
Vdd	32	+	Positive Supply
RB0/AN12/INT	33	RB0	General purpose I/O port B
		AN12	A/D Channel 12
		INT	External Interrupt
RB1/AN10/C12INT3-	34	RB1	General purpose I/O port B
		AN10	A/D Channel 10
		C12INT3-	Comparator C1 or C2 Negative Input
RB2/AN8	35	RB2	General purpose I/O port B
		AN8	A/D Channel 8
RB3/AN9/PGM/C12IN2-	36	RB3	General purpose I/O port B
		AN9	A/D Channel 9
		PGM	Programming enable pin
		C12IN2-	Comparator C1 or C2 Negative Input
RB4/AN11	37	RB4	General purpose I/O port B
		AN11	A/D Channel 11
RB5/AN13/T1G	38	RB5	General purpose I/O port B
		AN13	A/D Channel 13
		T1G	Timer T1 External Input
RB6/ICSPCLK	39	RB6	General purpose I/O port B
		ICSPCLK	Serial programming Clock
RB7/ICSPDAT	40	RB7	General purpose I/O port B
		ICSPDAT	Programming enable pin

CPU

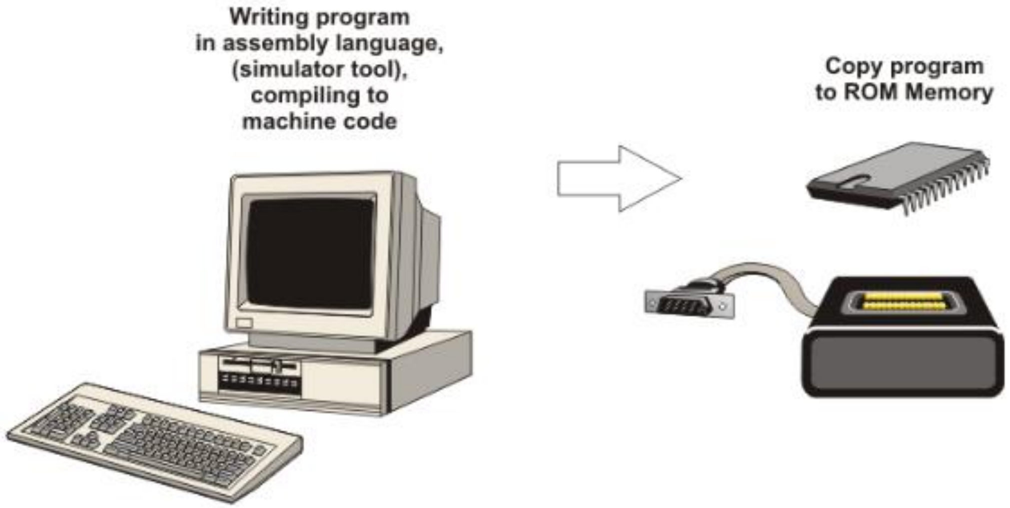
- CPU is made in RISC technology
- RISC stands for *Reduced Instruction Set Computer*, which gives the PIC16F887 two great advantages:
    - Its CPU can recognize and execute only 35 simple instructions (In order to program some other microcontrollers it is necessary to know more than 200 instructions by heart).
    - Execution time is the same for all of them and lasts 4 clock cycles (oscillator whose frequency is stabilized by quartz crystal). The only exceptions are jump and branch instructions whose execution time is twice as long. It means that if the microcontroller's operating speed is 20MHz, execution time of each instruction will be 200ns, i.e. the program will be executed at the speed of 5 million instructions per second!

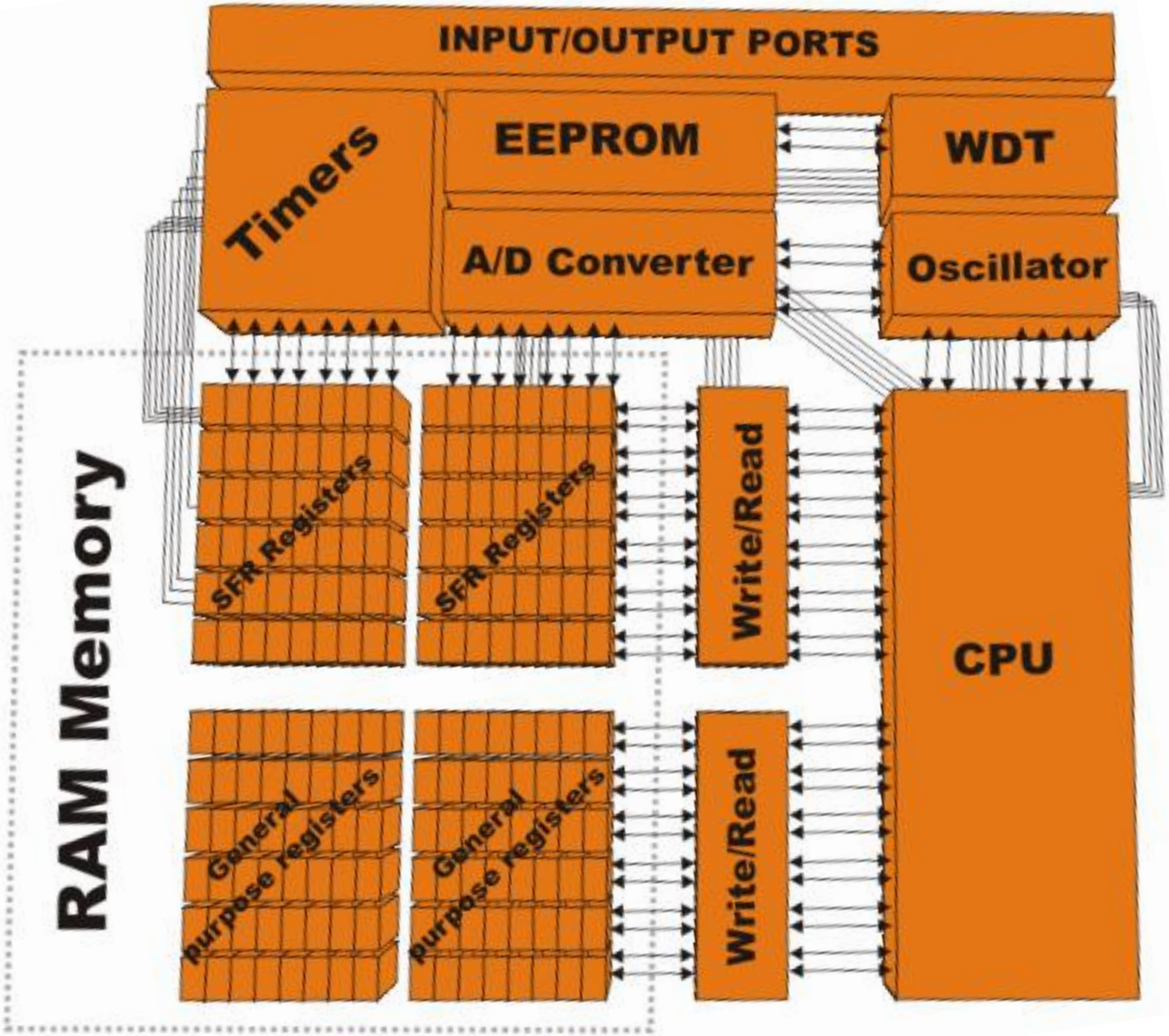

Input / Output Ports
Temporary Data
(RAM memory)
CPU
ROM memory
Oscillator, timers, counters...

Memory

PIC microcontroller has three types of memory- ROM, RAM and EEPROM. All of them will be separately discussed since each has specific function, features and organization.

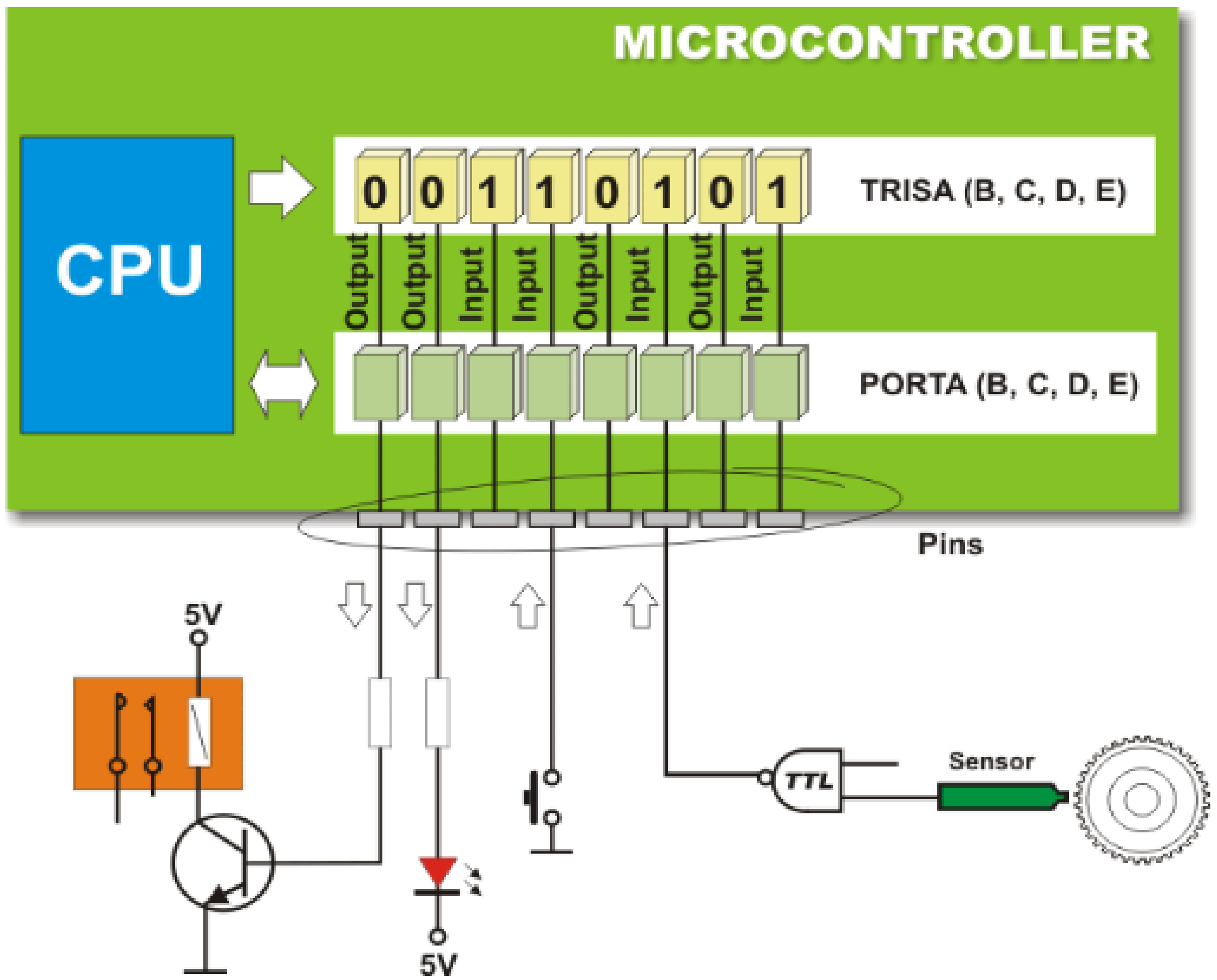
- **ROM Memory**: is used to permanently save program being executed. That is why it is often called "program memory". The PIC16F887 has 8Kb ROM (in total of 8192 locations). Since, in this very case, ROM is made in FLASH technology, its contents can be changed by providing special programming voltage (13V).
- **EEPROM Memory**: similar to program memory, the contents of EEPROM is permanently saved, even upon the power goes off. However, unlike ROM, the contents of EEPROM can be changed during operation of the microcontroller. That is why this memory (256 locations) is a perfect one for permanently saving results created and used during the operation.
- **RAM Memory**: which is the third and the most complex part of microcontroller memory, it consists of two parts: general-purpose registers and special-function registers (SFR).


Writing program
in assembly language,
(simulator tool),
compiling to
machine code
Copy program
to ROM Memory


INPUT/OUTPUT PORTS
Timers
EEPROM
WDT
A/D Converter
Oscillator
RAM Memory
SFR Registers
SFR Registers
Write/Read
CPU
General purpose registers
General purpose registers
Write/Read

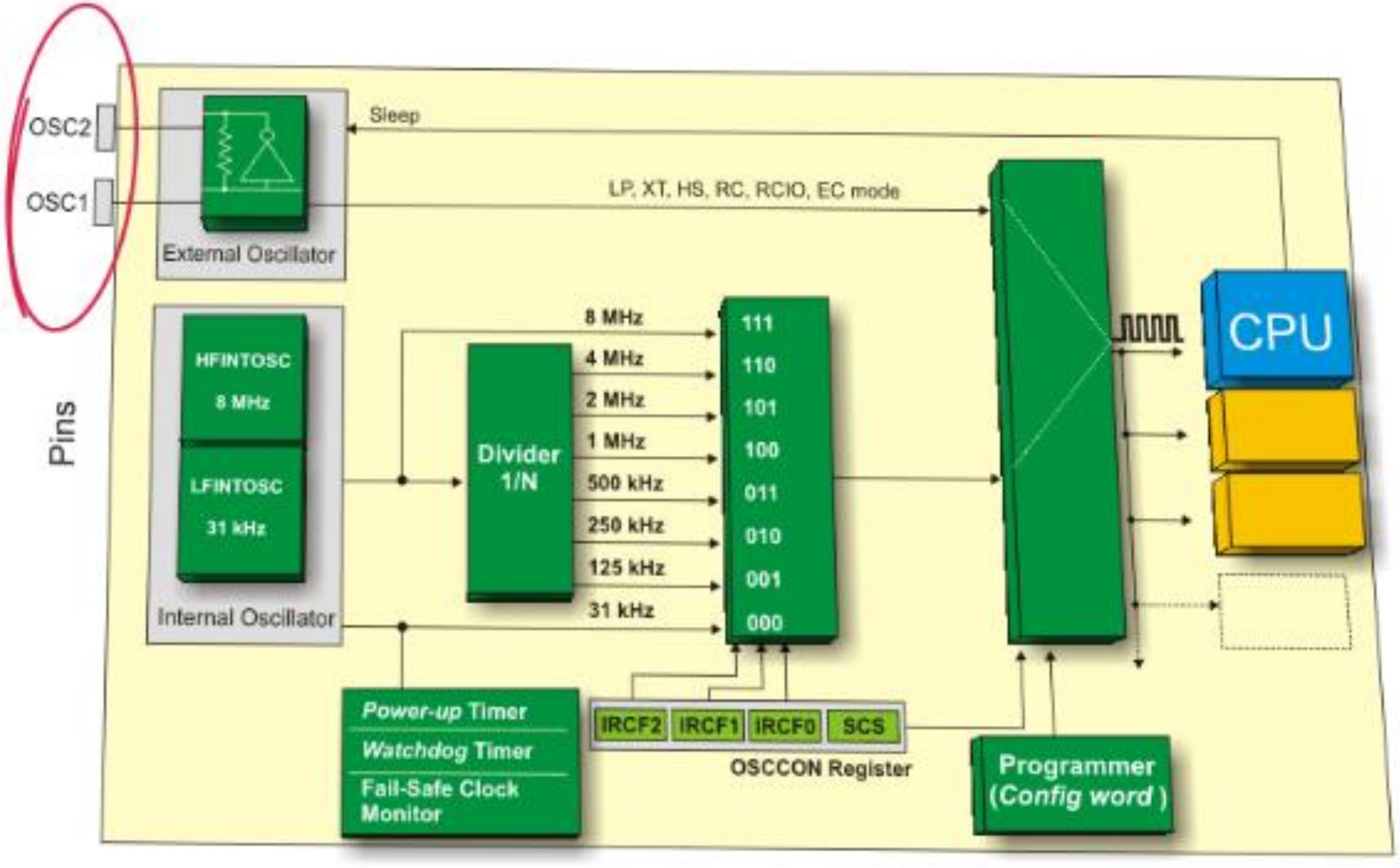
I/O Ports

- One of the most important feature of the microcontroller is a number of input/output pins used for connection with peripherals. In this case, there are in total of thirty-five general purpose I/O pins available, which is quite enough for the most applications.
- In order pins' operation can match internal 8-bit organization, all of them are grouped into five ports denoted by A, B, C, D and E. They have several features in common:
  - For practical reasons, many I/O pins have two or three functions. In case any of these alternate functions is currently active, that pin may not simultaneously use as a general purpose input/output pin.
  - Every port has its "satellite", i.e. the corresponding TRIS register: TRISA, TRISB, TRISC etc. which determines performance, but not the contents of the port bits.
  - By clearing some bit of the TRIS register (bit=0), the corresponding port pin is configured as output. Similarly, by setting some bit of the TRIS register (bit=1), the corresponding port pin is configured as input. This rule is easy to remember 0 = Output, 1 = Input.


MICROCONTROLLER
CPU
0 0 1 1 0 1 0 1
TRISA (B, C, D, E)
Output
Output
Input
Input
Output
Input
Output
Input
PORTA (B, C, D, E)
Pins
5V
5V
TTL
Sensor

Oscillator

- Crystal Oscillator is used in Microprocessor and Microcontroller to generate clock pulses required for the synchronization of all the internal operations.
- It has two types: **Internal and External**, each type has several modes, these modes are selected by the device configuration bits.



Oscillator

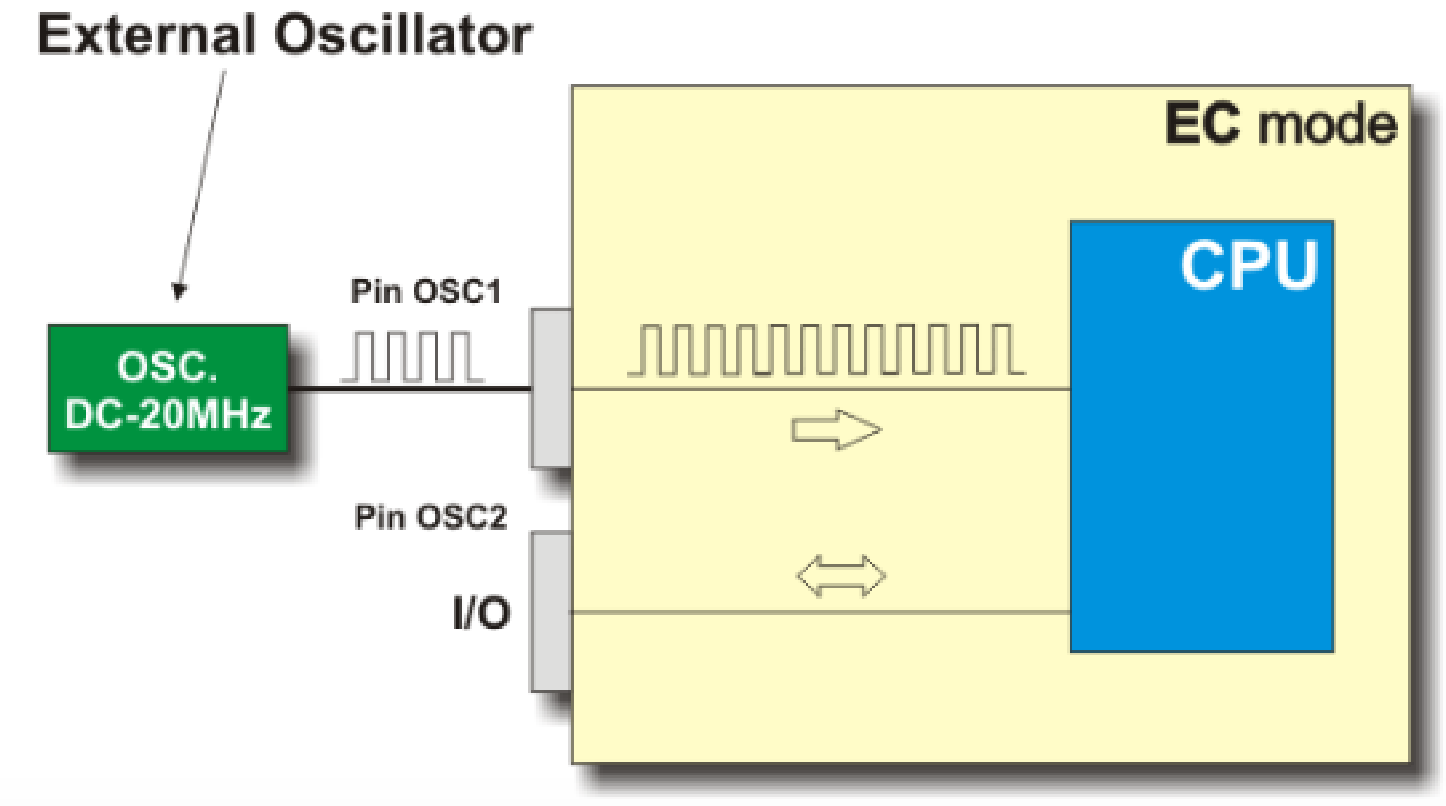
- **External oscillator** is installed within the microcontroller and connected to the OSC1 and OSC2 pins. It is called “external” because it relies on external circuitry for the clock signal and frequency stabilization, such as stand-alone oscillator, quarts crystal, ceramic resonator or resistor-capacitor circuit. It can be stand-alone oscillator, quarts crystal, ceramic resonator or resistor-capacitor circuit. The oscillator mode is selected by bits of bytes sent during programming, so called **Config Word.**
- **Internal oscillator** consists of two separate, internal oscillators:
  - The **HFINTOSC** is a **high-frequency internal oscillator which operates at 8MHz**. The microcontroller can use clock source generated at that frequency or after being divided in pre-scaler.
  - The **LFINTOSC** is a **low-frequency internal oscillator which operates at 31 kHz**. Its clock sources are used for watch-dog and power-up timer but it can be also used as a clock source for the operation of entire microcontroller.

External Oscillator

The external clock (EC) mode uses the system clock source configured from external oscillator. The frequency of this clock source is unlimited (0- 20MHz). This mode has the following advantages:

- The external clock source is connected to the OSC1 input and the OSC2 is available for general purpose I/O.
- It is possible to synchronize the operation of the microcontroller with the rest of on board electronics.
- In this mode the microcontroller starts operating immediately after the power is on. There is no delay required for frequency stabilization.
- Temporary stopping the external clock input has the effect of halting the device while leaving all data intact. Upon restarting the external clock, the device resumes operation as if nothing has happened.

External Oscillator

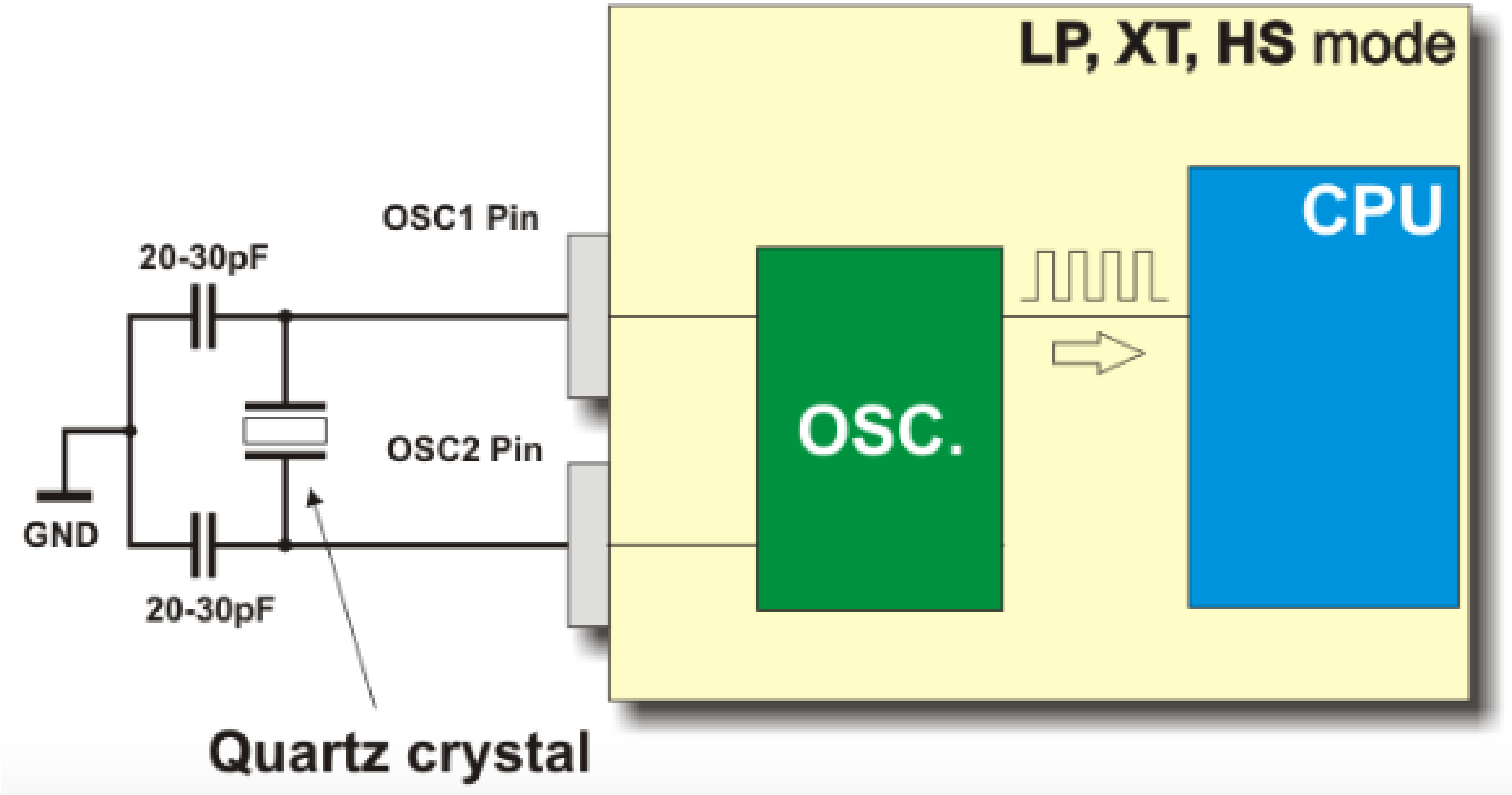


External Oscillator Modes (LP, XT, HS)

The LP, XT and HS modes support the usage of internal oscillator for configuring clock source. The frequency of this source is determined by quartz crystal or ceramic resonators connected to the OSC1 and OSC2 pins. Depending on features of the component in use, select one of the following modes:

- **LP mode** (Low Power) is used for low-frequency quartz crystal only. This mode is designed to drive only 32.768 kHz crystals usually embedded in quartz watches. It is easy to recognize them by small size and specific cylindrical shape. The current consumption is the least of the three modes.
- **XT mode** is used for intermediate-frequency quartz crystals up to 8 MHz. The current consumption is the medium of the three modes.
- **HS mode** (High Speed) is used for high-frequency quartz crystals over 8 MHz. The current consumption is the highest of the three modes.

External Oscillator Modes



Internal Oscillator

The internal oscillator circuit consists of two separate oscillators that can be selected as the system clock source:

- The **HFINTOSC** oscillator is factory calibrated and operates at 8 MHz. Its frequency can be user- adjusted via software using bits of the OSCTUNE register.
- The **LFINTOSC** oscillator is not factory calibrated and operates at 31kHz.

Similar to the external oscillator, the internal one can also operate in several modes. The mode is selected in the same way as in case of external oscillator- using bits of the Config Word register. In other words, everything is performed within PC software, immediately before program writing to the microcontroller starts.

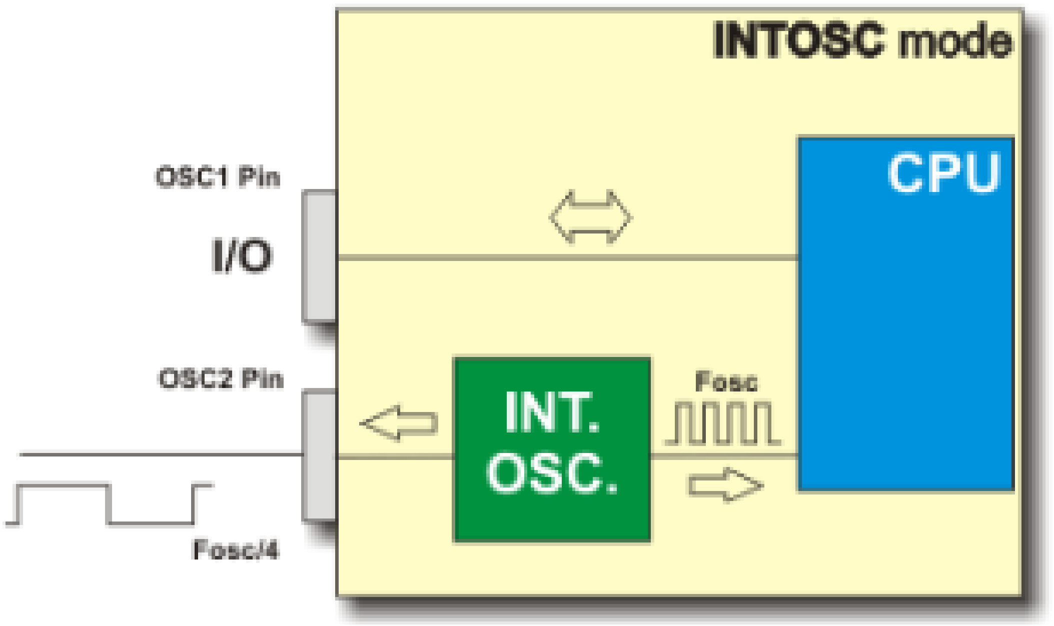
Internal Oscillator modes

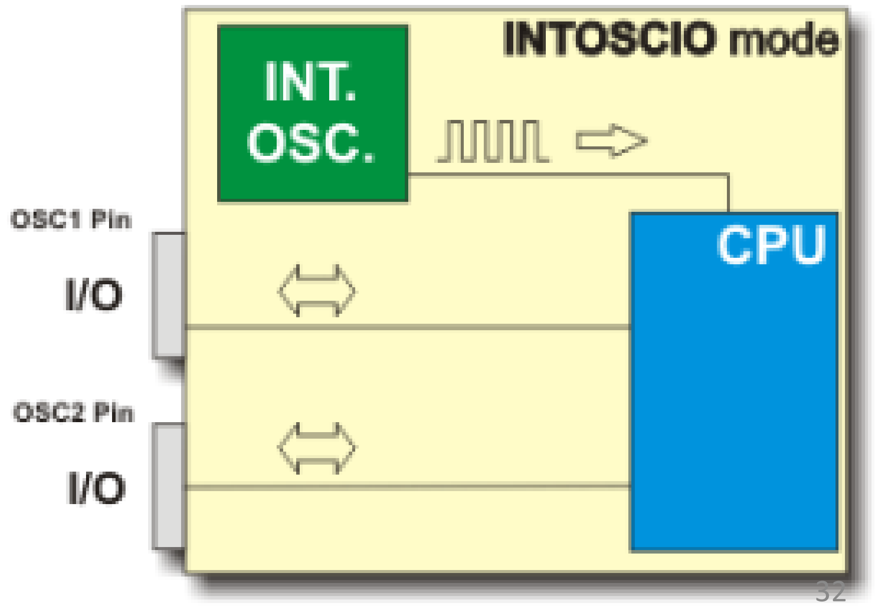
Internal oscillator in INTOSC mode

- In this mode, the OSC1 pin is available as general purpose I/O while the OSC2 pin outputs selected internal oscillator frequency divided by 4.

Internal oscillator in INTOSCIO mode

- In this mode, both pins are available for general purpose I/O.





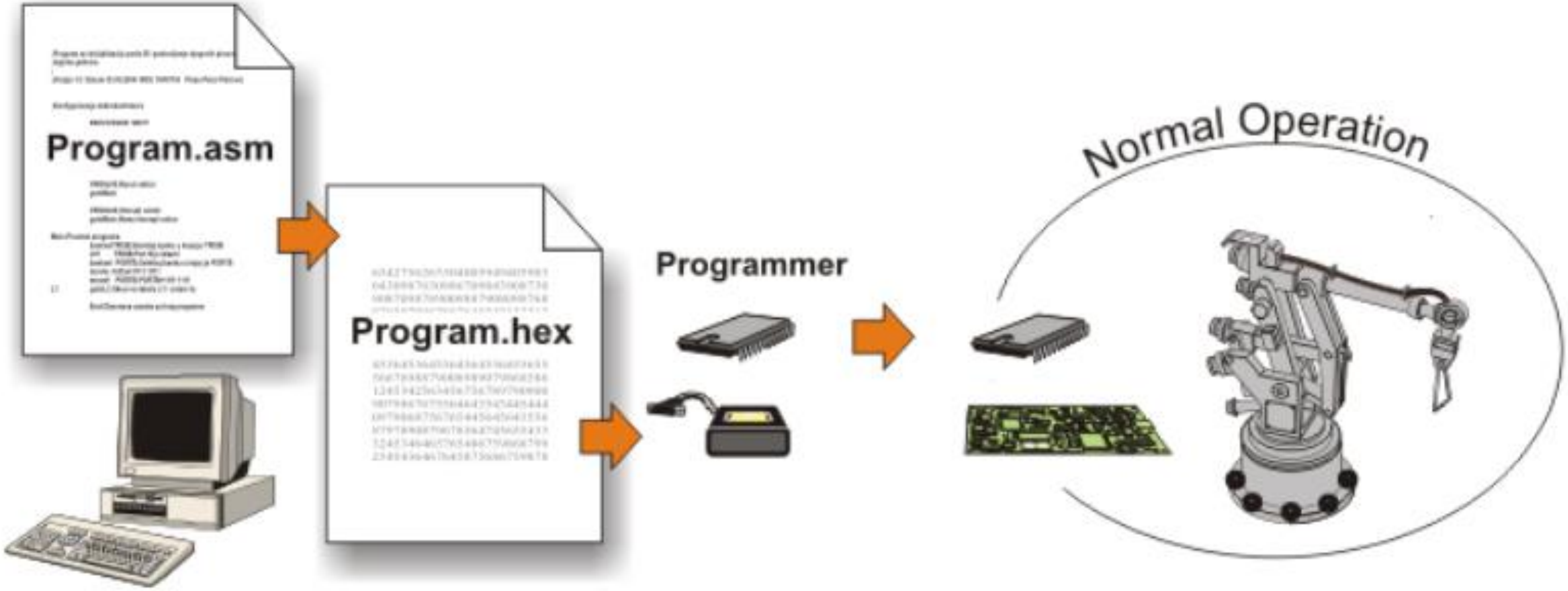
Programming

- Microcontroller and humans communicate through the medium of the programming language called Assembly language. The word “Assembler” itself does not have any deeper meaning, it corresponds to the names of other languages such as English or French.
- In order the microcontroller can understand a program written in assembly language, it must be compiled into a “language of zeros and ones”. **“Assembly language” and “Assembler”** do not have the same meaning. The **Assembly Language** refers to the set of rules used for **writing program for the microcontroller**, while **Assembler** refers to a **program on personal PC used to translate assembly language statements into the language of zeros and ones**.
- A **compiled program** is also called a “**machine code**”. “Program” is a data file stored on a computer hard disc (or in memory of the microcontroller if loaded) and written according to the rules of assembly or some other programming language.

Programming

- **Assembly language** is understandable for the humans because it consists of meaningful words and symbols of alphabet. Let us take for example the command “RETURN” which is, as its name indicates, used to return the microcontroller from a subroutine. In machine code, the same command is represented by a 14-bit array of zeros and ones understandable for the microcontroller.
- All assembly language commands are similarly compiled into the corresponding array of zeros and ones. A data file used for storing compiled program is called “executive file”, i.e. “HEX data file”. The name runs from hexadecimal presentation of data file and suffix “hex” as well, for example “probe.hex”. After has been generated, data file is loaded into the microcontroller using programmer. Assembly language program may be written in any program for text processing (editor) able to create ASCII data file on a hard disc or in a specialized work environment such as MPLAB described later.

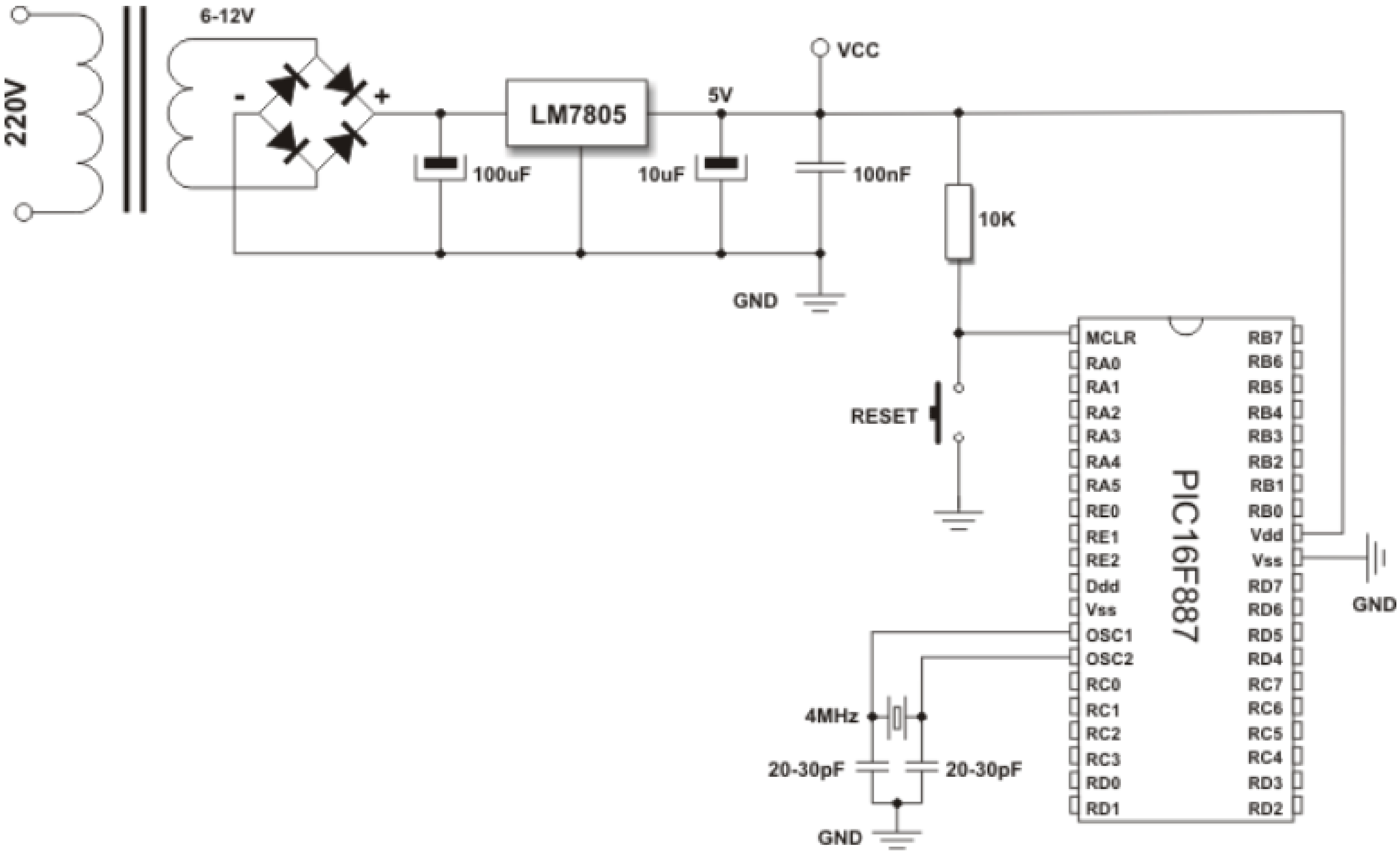
Programming



Basic Operation

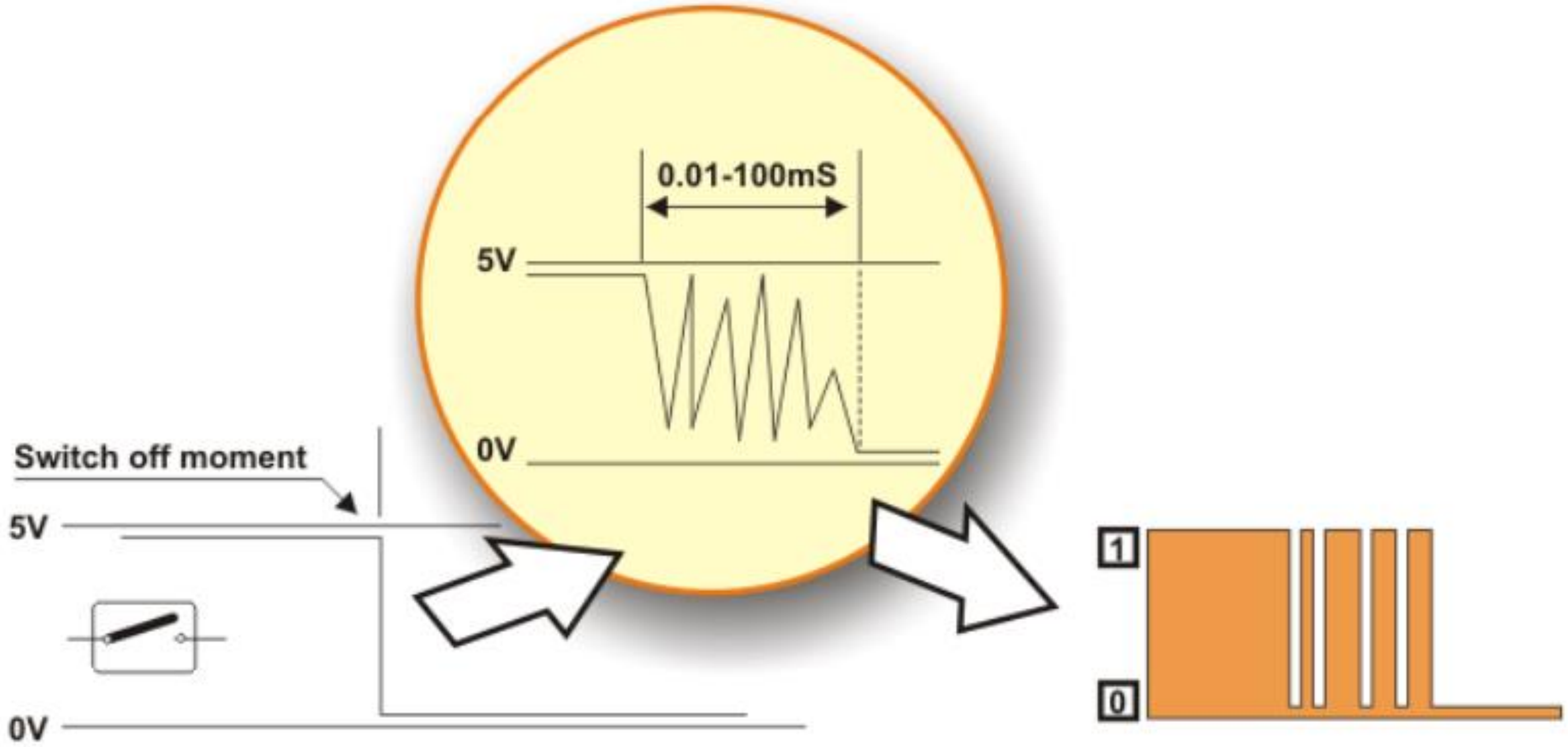
As seen in figure below, in order to enable the microcontroller to operate properly it is necessary to provide:

- Power Supply
- Reset Signal
- Clock Signal



1. SWITCHES AND PUSH-BUTTONS

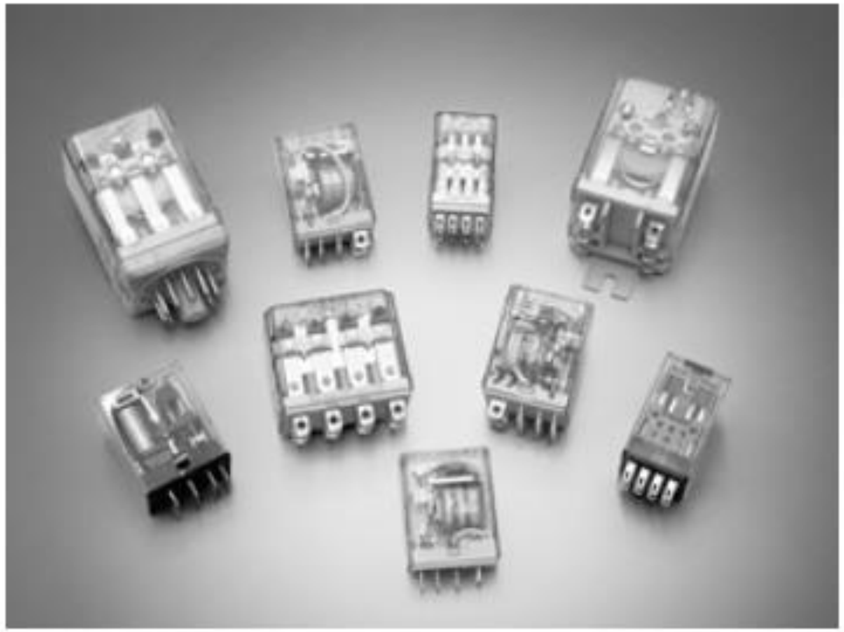
There is nothing simpler than switches and push-buttons! This is definitely the simplest way of detecting appearance of some voltage on the microcontroller input pin and there is no need for additional explanation of how these components operate.



2. RELAY

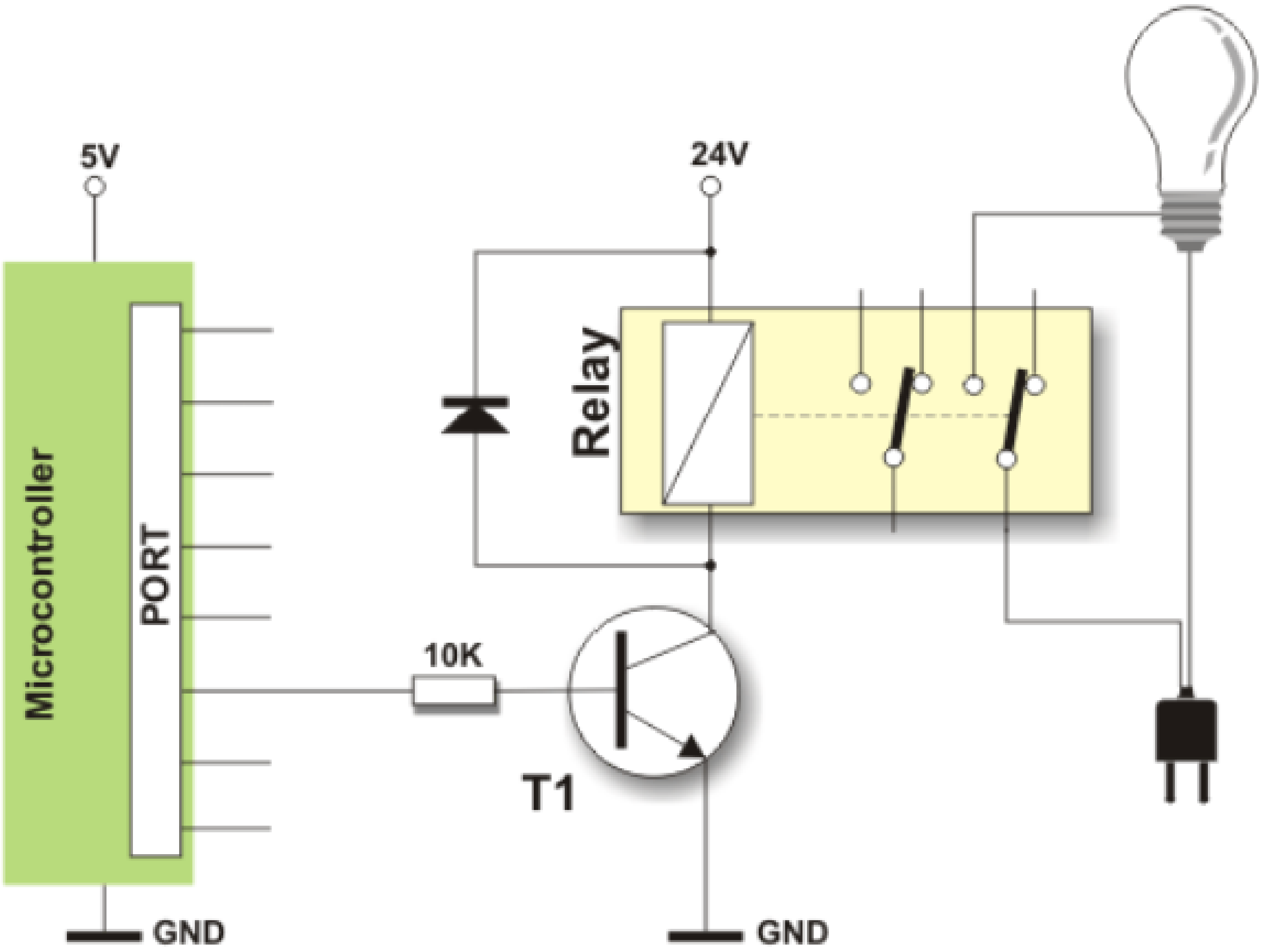
A relay is an electrical switch that opens and closes under the control of another electrical circuit. It is therefore connected to output pins of the microcontroller and used to turn on/off high-power devices such as motors, transformers, heaters, bulbs, etc.

These devices are almost always placed out of the board with sensitive components. There are various types of relays, but all of them operate in the same way***. When a current flows through the coil, the relay is operated by an electromagnet to open or close one or many sets of contacts.*** Similar to optocouplers, there is no galvanic connection (electrical contact) between input and output circuits. Relays usually demand both higher voltage and current to start operation but there are also miniature ones that can be activated by a low current directly obtained from a microcontroller pin.



2. RELAY

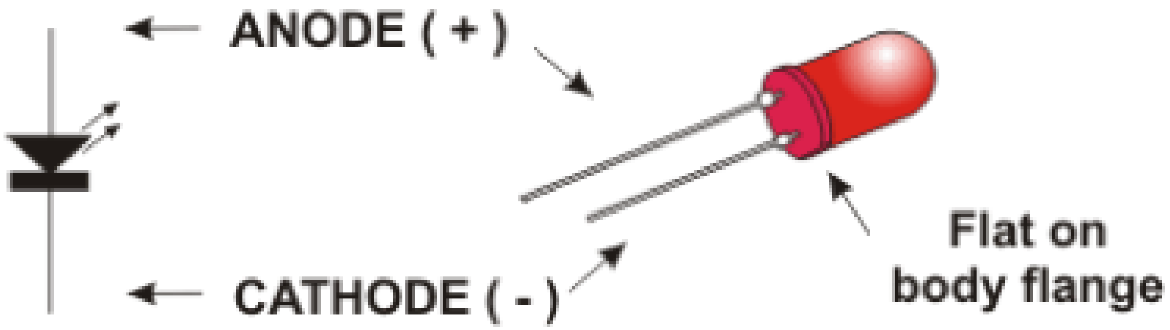
In order to prevent appearance of high voltage of self-induction caused by a sudden stop of current flow through the coil, an inverted polarized diode is connected in parallel to the coil. The purpose of this diode is to “cut off” the voltage peak.

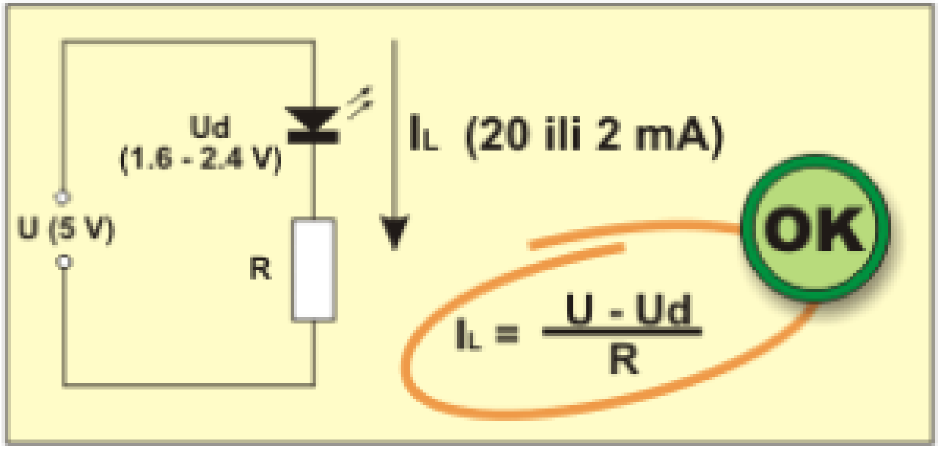


3. LED DIODES

LED has two ends- anode and cathode. Place it properly and bring power supply voltage. The diode will happily emit light. Turn it upside down and bring power supply voltage (even for a moment).

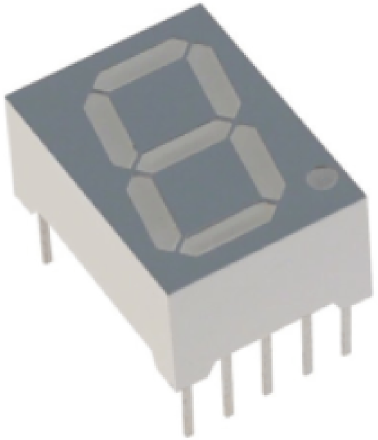
- **Slow burning:** There is a nominal, i.e. maximum current determined for every LED which should not be exceeded. If it happens, the diode will emit more intensive light, but not for a long time!
- **Something to remember:** Similar to the previous example, all you need to do is to discard a current limiting resistor. Depending on power supply voltage, the effect might be spectacular!

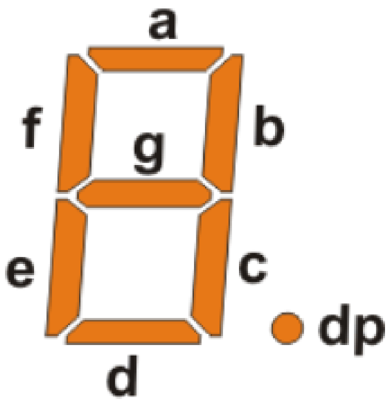




4. LED DISPLAY

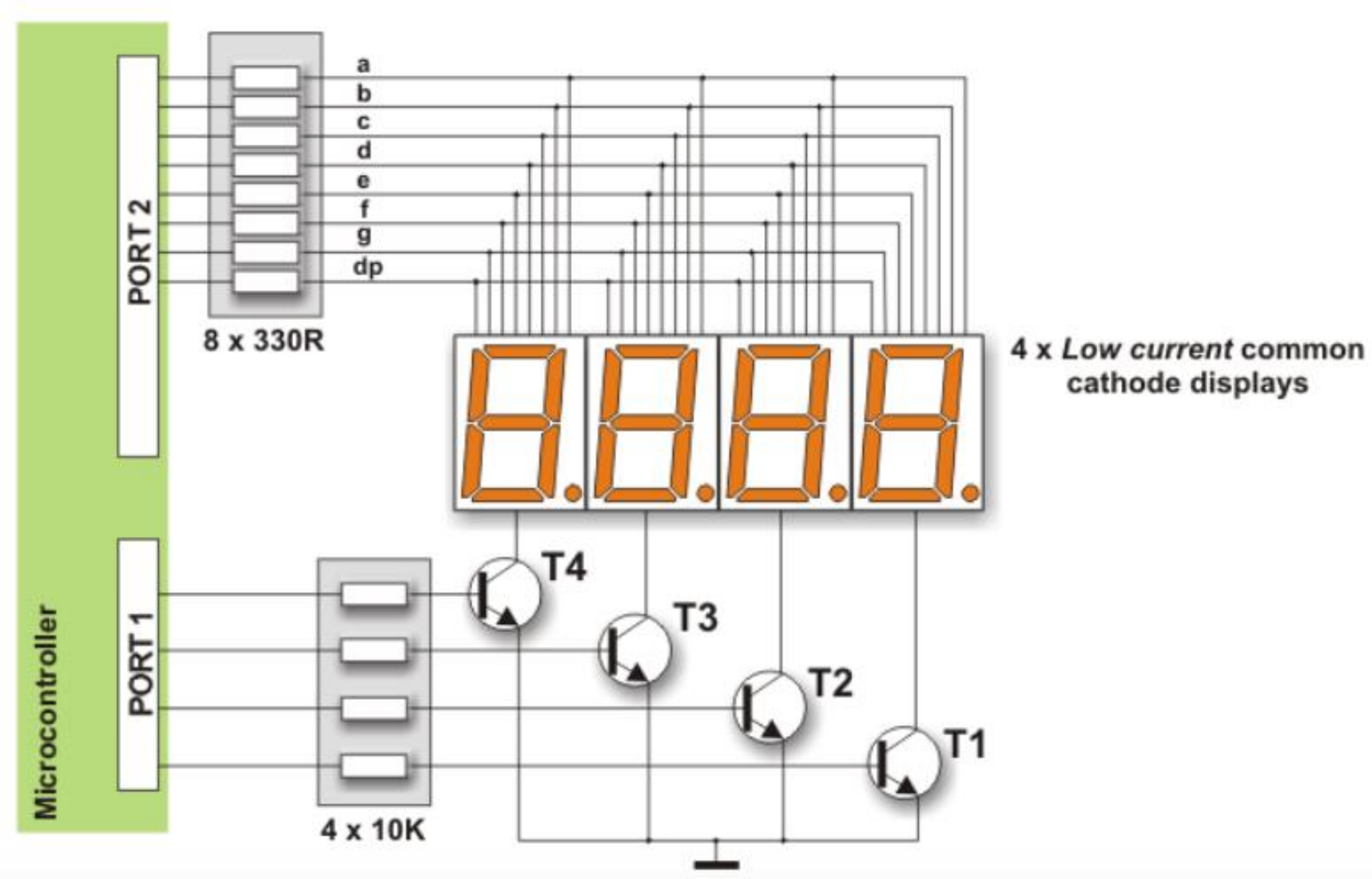
Basically, LED display is nothing else but several LEDs molded in the same plastic case. Diodes are arranged in a way that different marks commonly digits- 0, 1, 2,...9- are displayed by activating them. There are many types of displays composed of several dozens of built in diodes which can display different symbols. The most commonly used is so called 7-segment display. It is composed of 8 LEDs- 7 segments are arranged as a rectangle for symbol displaying and there is an additional segment for decimal point displaying. In order to simplify connection, anodes or cathodes of all diodes are connected to the common pin so that there are common anode displays and common cathode displays, respectively. Segments are marked with the **letters from a to g, plus dp**. On connecting, each diode is treated separately, which means that each must have its own current limiting resistor.





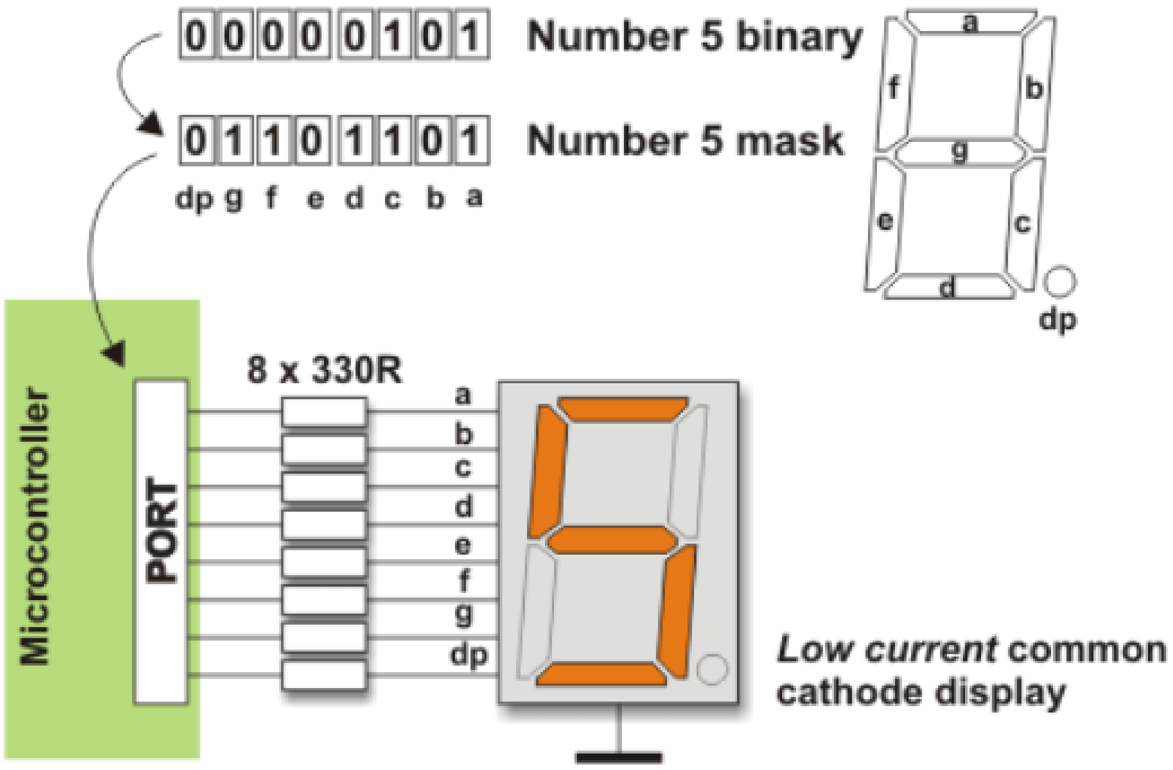
MULTIPLEXING

Only one digit at a time is active, but they change their state so quickly that one gets impression that all digits of a number are active simultaneously. First a byte representing units is applied on a microcontroller port and a transistor T1 is activated simultaneously. After a while, the transistor T1 is turned off, a byte representing tens is applied on a port and transistor T2 is activated. This process is being cyclically repeated at high speed for all digits and corresponding transistors.



LED DISPLAY

- First of all, in a particular subroutine a multi-digital number must be split into units, tens etc. Then, these must be stored in special bytes each. Digits get recognizable format by performing “masking”. In other words, a binary format of each digit is replaced by different combination of bits using a simple subroutine. For example, the digit 8 (0000 1000) is replaced by binary number 0111 1111 in order to activate all LEDs displaying digit 8. The only diode remaining inactive in this case is reserved for decimal point.
- If a microcontroller port is connected to display in a way that bit 0 activates segment “a”, bit 1 activates segment “b”, bit 2 segment “c” etc., then the table below shows the mask for each digit.



LED DISPLAY

Digits to display	Display Segments							
	dp	a	b	c	d	e	f	g
0	0	1	1	1	1	1	1	0
1	0	1	1	0	0	0	0	0
2	0	1	1	0	1	1	0	1
3	0	1	1	1	1	0	0	1
4	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1
6	0	1	0	1	1	1	1	1
7	0	1	1	1	0	0	0	0
8	0	1	1	1	1	1	1	1
9	0	1	1	1	1	0	1	1

5. LCD DISPLAY

This component is specialized to be used with the microcontrollers, which means that it cannot be activated by standard IC circuits. It is used for displaying different messages on a miniature liquid crystal display. A model described here is for its low price and great capabilities most frequently used in practice. It is based on the HD44780 microcontroller (*Hitachi*) and can display messages in two lines with 16 characters each. It displays all letters of alphabet, greek letters, punctuation marks, mathematical symbols etc. In addition, it is possible to display symbols made up by the user. Other useful features include automatic message shift (left and right), cursor appearance, LED backlight etc.

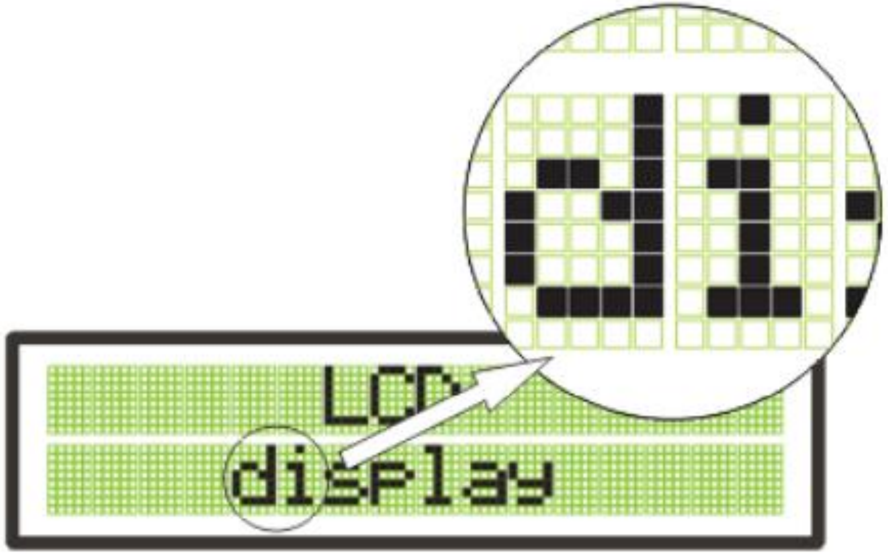
5. LCD DISPLAY

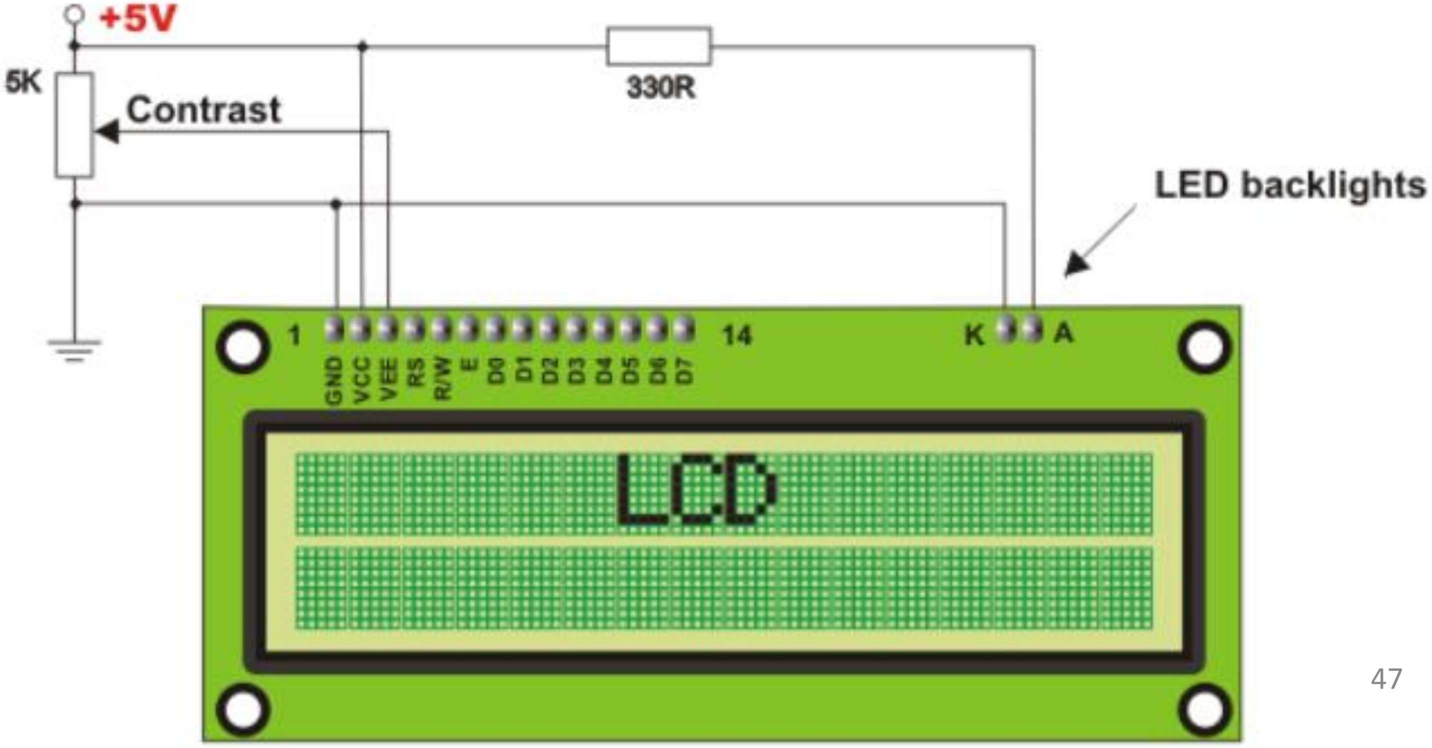
Along one side of a small printed board there are pins used for connecting to the microcontroller. There are in total of 14 pins marked with numbers (16 in case the backlight is built in). Their function is described in this table:

Function	Pin Number	Name	Logic State	Description
Ground	1	Vss	-	0V
Power supply	2	Vdd	-	+5V
Contrast	3	Vee	-	0 - Vdd
Control of operating	4	RS	0 1	D0 – D7 are interpreted as commands D0 – D7 are interpreted as data
	5	R/W	0 1	Write data (from controller to LCD) Read data (from LCD to controller)
	6	E	0 1 From 1 to 0	Access to LCD disabled Normal operating Data/commands are transferred to LCD
Data / commands	7	D0	0/1	Bit 0 LSB
	8	D1	0/1	Bit 1
	9	D2	0/1	Bit 2
	10	D3	0/1	Bit 3
	11	D4	0/1	Bit 4
	12	D5	0/1	Bit 5
	13	D6	0/1	Bit 6
	14	D7	0/1	Bit 7 MSB

6. LCD SCREEN

LCD screen consists of two lines with 16 characters each. Every character consists of 5x8 or 5x11 dot matrix. This book covers 5x8 character display, which is indeed the most commonly used one.





Summary