Panayotis Papazoglou, PhD

An Educational Guide to the AVR Microcontroller Programming

First English edition (translation from Greek): 2018
Kessariani, Athens, Greece
email: **p.m.papazoglou@hotmail.com**

book photos: **Panayotis Papazoglou**

ISBN: 978-1986008396

Follow your dreams!

PREFACE

The type and the number of applications of the modern technology have driven to different solutions and approaches. Some of these solutions are based in a computer system which supports the corresponding requirements. The computers (PCs) that are used nowadays represent only one view of the reality, because they mainly support the requirements regarding the processing power, everyday applications, simplicity of using and programming, etc.

From the other point of view, there is a great number of applications that require an autonomous control system which is usually embedded in a device. In such systems, the requirements are focused on the operation autonomy, the physical size, the power consumption, etc.

Approaching these views from the aspect of architecture and programming, the core of the systems is a microprocessor or a microcontroller.

It must be noticed that the basic differences between microprocessors and microcontrollers, but also of their architecture and programming constitutes an unclear subject for many new engineering students and readers.

This book (volume 1) constitutes a complete basic educational guide which offers important knowledge and demystifies the AVR programming. Moreover, this book has been written by taking in account the real needs of students, teachers and others who want to develop AVR based applications.

All the programs and applications of the book have been developed and tested in a real microcontroller, in contrast with other books where the corresponding material has been developed only theoretically with no tests in practice.

The above lines, state the deep belief of the author that this book will constitute a useful teaching and educational tool for helping anyone understand the AVR applications. On the other hand, the book can be used by the teacher for organizing lectures and presentations as well as the laboratory exercises.

Panayotis Papazoglou, PhD

The book structure

The book (volume 1) consists of 7 chapters which include simple exercises or laboratory activities.

The **first chapter** is a general introduction to microcontrollers regarding their features and capabilities. The chapter also includes application examples that are focused on the corresponding components such as sensors, etc.

The AVR features such as the internal architecture, the memory system and the registers are presented in the **second chapter**. These features are presented only with the needed level of details in order to be easily understandable by the reader without any hard work.

In the **third chapter**, the basic instruction set is analyzed using a great number of figures. The assembly instruction set constitutes the basic tool for developing the corresponding applications. All the instructions are presented by using practical examples as well as special figures in order to demonstrate the corresponding operation. Thus, the reader understands the operation philosophy of every instruction.

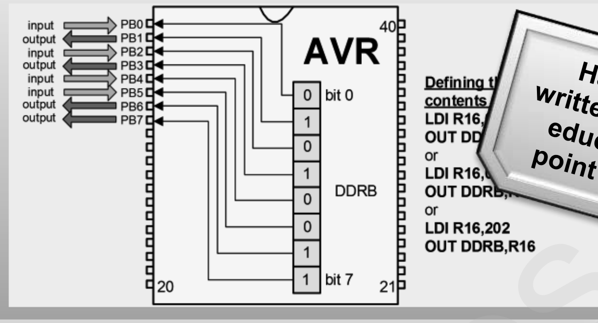
Due to the fact that the algorithm constitutes the most important component for every application and the code synthesis is based on the structured programming concepts, a whole chapter (**fourth chapter**) has been written for presenting all this knowledge. Thus, the reader is guided step by step to the correct way of thinking for developing the desired application.

In the **fifth chapter** the reader learns for the first time how to exploit the digital input/output pins of the microcontroller. More precisely, the chapter begins with the basic knowledge on electrical circuits for understanding basic concepts as the current, voltage, etc. Next, the application of this knowledge in the external circuits of the microcontroller as well as the corresponding programming is analyzed.

The display units (e.g. seven segment displays, LCD) are analyzed in the **sixth chapter**. The display operation is very important for the user interaction with the application as well as for debugging.

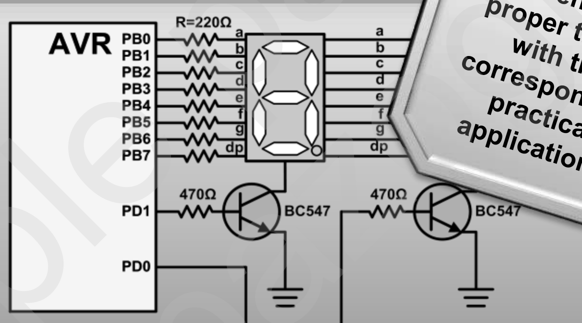
Due to the fact that the modern applications support the input of complex data and instructions by the user, it is very important to present the corresponding switch circuits that can be used as keyboards. The **seventh chapter** analyzes how to develop such a circuit as well as the programming method for using it.

Important features of this book

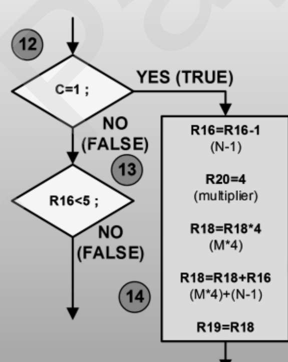


Has been written from an educational point of view

```
Turn off the two SSD units
START
Display the number 1 on the left SSD unit
Turn off the left SSD unit (duration<5 sec)
Display the number 5 on the right SSD
Turn off the right SSD unit (duration
RETURN TO START
```



Presents the proper theory with the corresponding practical applications



The presentation is done via rich analytic figures, while the complex subjects are simplified and demystified; the development of selected codes is analyzed via flow chart diagrams

Important features of this book

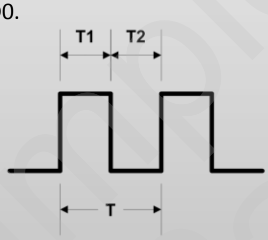
```
;**************************
;Main program
;**************************
main:
  SBI PORTB,Button;Pull Up re
                  ;activation
                  ;at pin PB0
  SBI DDRB,LED    ;The pin 1 of port B
                  ;will be set as output
scan:
      SBIC PINB,Button ;If the pin 0 (button)
                       ;is activated
                       ;(PB0=LOW=0V),
                       ;then the next
                       ;instruction
                       ;is bypassed

      RJMP scan      ;Return to check button
      SBI PORTB,LED ;LED activation
```

Well documented source code

Step 2

Fill the following signal attributes based on the expected measurements on pin PD0.



T1 =

T2 =

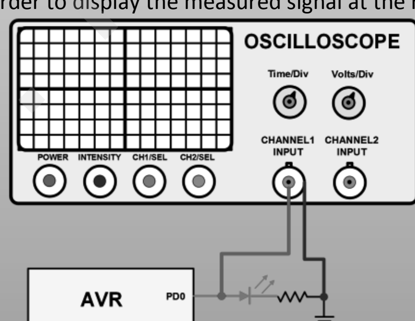
T =

Frequency F =

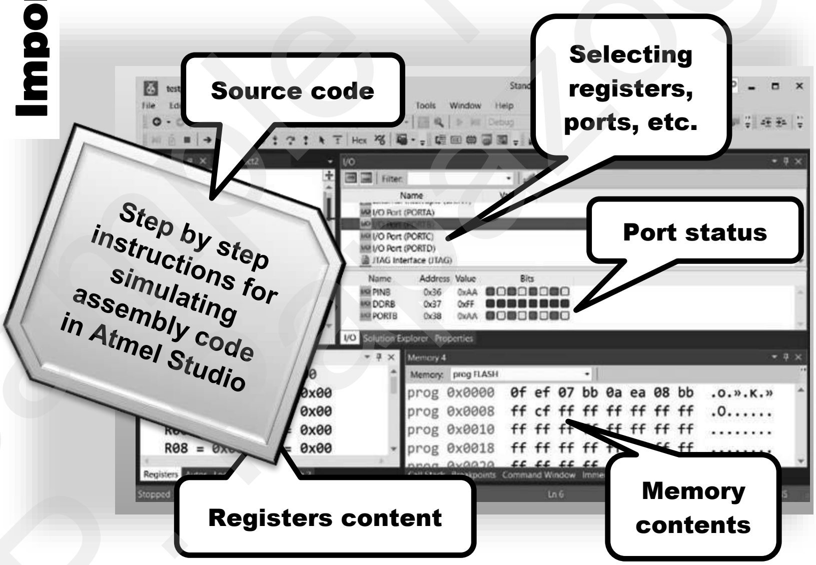
Laboratory activities and simple exercises are included

Step 3

Connect one channel of the oscilloscope as described in the next figure. Set properly the time base (Time/Div) and the voltage scale (Volts/Div) on the oscilloscope in order to display the measured signal at the right size.



Important features of this book



Happy Reading!

Table of Contents

1 Introduction

Content-Goals

This chapter presents the operation of microcontrollers in general as well as some corresponding indicative applications. The above operation is analyzed in comparison to the classical microprocessor and the conventional computer as a system. The goal is to clarify the microcontroller's role in the application development process of specific purpose.

Chapter contents

1.1 General

Microcontrollers are usually embedded in autonomous systems which support specific applications. An electronic thermometer, a robotic vehicle, a smart television, the electronic controller of a car, a mobile medical instrument, the bus ticket controller, the metro security system, the automated transaction systems, the traffic lights controller and many more, constitute systems where an embedded microcontroller supports the corresponding service. It is unlikely in such a system for a traditional PC to be used. This happens due to hard restrictions such as the physical size, the operation autonomy, the energy consumption, the difficult installation conditions, the real time response to external events, the reboot time, etc. The above constraints cannot be met by a traditional PC.

Thus, the microcontroller as a single electronic chip is embedded in such a system in order to satisfy the above constraints. A microcontroller is usually called "small computer" due to the fact that it can autonomously support the above applications.

On the other hand, humans are more familiar with traditional PCs, and thus the corresponding introduction will start from them. Using initially the microprocessor as a reference point, the presentation will proceed to the microcontroller in order to be able to make a comparison. With this approach, many of the basic microcontroller features (which constitute the core of the book) will be more understandable.

1.2 Microprocessors and Microcontrollers

A computer system (Computer-Personal Computer-PC) consists of various components (internal and external) in order to be exploited by the users that select the corresponding applications. A computer does not only contain a microprocessor and a memory system but also has output units such as a monitor to display information, input units such as a keyboard for entering data, etc. Moreover, the computer consists of supplementary circuits for implementing the communication between the main components (e.g. microprocessor, memory) in order to support the instruction execution.

Figure 1.1 shows a typical structure of a computer. The computer operation as a system is based on the interaction between the three main components (internal units such as microprocessor, memory and external units such as monitor, keyboard). The above components constitute independent units which are interconnected through the bus system.

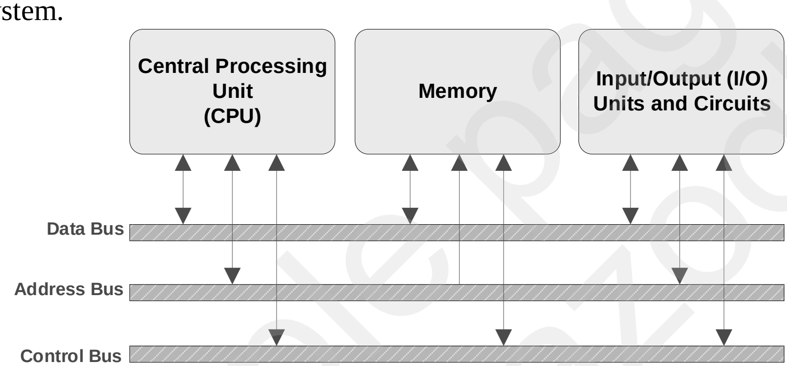


Figure 1.1 A typical computer structure

More precisely, the goal of each component (unit) of figure 1.1, is:

Central Processing Unit -CPU. The CPU accepts instructions from the main memory and implements the execution process for producing the corresponding results. If for example the programmer has used an instruction for addition (code in memory), then this instruction is transferred in the CPU for performing the addition (instruction execution) and the result is returned back to memory. The CPU is implemented in an integrated circuit (IC-chip) which is called **microprocessor**.

Memory. The memory (main memory) stores the programs and the corresponding data that are processed by the programs. The memory constitutes an one dimensional array with identical locations in terms of capacity. On the other hand, every memory location has a unique identifier which is called address. For reading or writing to a memory location the corresponding address has to be activated.

I/O units. The I/O units ensure the system functionality and the communication with the outer environment (outside world) in order for the computer to be useful to the user. A well-known example is the keyboard for entering data to the computer.

Data bus. The system bus is common among the computer components in order to control and exchange data between the system units. The data are transferred via the data bus. For example, the result of the mentioned addition is transferred to the memory via the data bus.

Address bus. Addresses are unique numbers that are assigned to all computer components for supporting the corresponding communication. The result of the previous addition will be used again as an example. The result storage location in the main memory is activated through an address which is transferred via the address bus.

Control bus. The data exchange between computer components is achieved via special control signals that are transferred through the control bus. These signals ensure that a data packet will be delivered at the correct address (destination) and the corresponding units will be activated in the right priority.

From all the above it is obvious that the microprocessor is an integrated digital circuit which only contains processing elements for executing instructions from the memory. That means that the microprocessor cannot operate autonomously. In other words, the exploitation of the microprocessor cannot be done without the support of additional components such as the memory, I/O units, etc. Figure 1.2 shows from a different point of view the independent components (units) of the computer beginning from the microprocessor level towards to the outer environment (outside world). Figure 1.2 shows also that the exploitation of the microprocessor is a complicated process due to the fact that much more components have to be combined in order to form a complete computer system. The above constraints increase the system cost, the management complexity and the installation difficulties for an autonomous application.

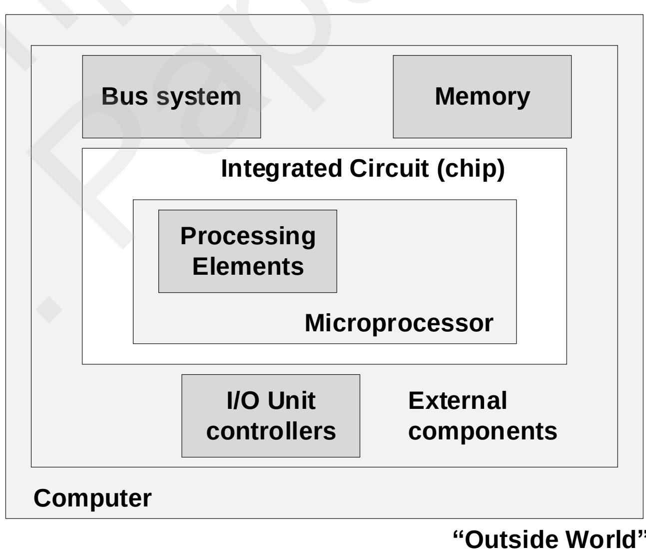


Figure 1.2 The microprocessor and the system components

Figures 1.3a and 1.3b, show the front and rear views of a real microprocessor IC respectively.

Figure 1.3a Front view of a microprocessor

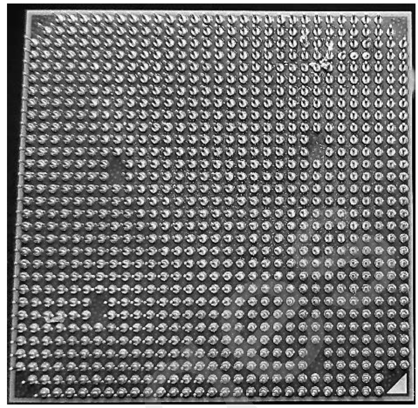


Figure 1.3b Rear view of a microprocessor

A microprocessor offers high processing performance by executing instructions in short times. The fact that the microprocessor only executes instructions means that there are many supplementary components and circuits needed to build a functional system.

Figure 1.3c shows a main board (motherboard) which hosts the microprocessor as well as the supplementary components and circuits for the system operation.

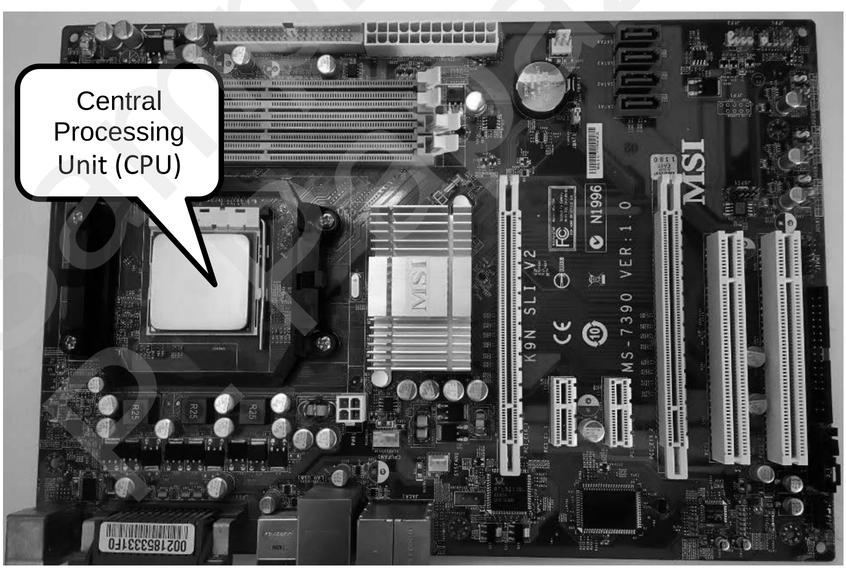


Figure 1.3c Main board (motherboard) for hosting a microprocessor and other supplementary units and circuits

On the other hand, the integrated circuit of a microcontroller (fig. 1.4) contains internally all the components that in the case of the microprocessor were built externally (e.g. memory, I/O controllers).

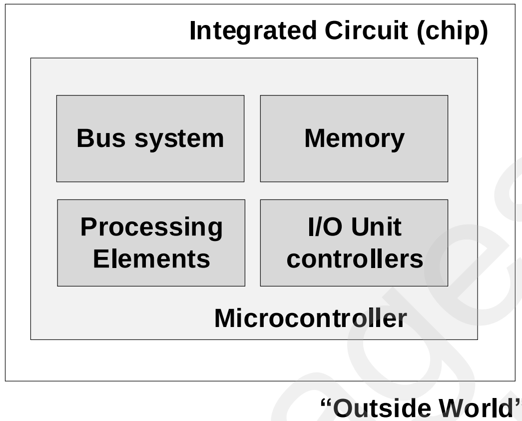


Figure 1.4 General microcontroller structure

The existence of the needed components inside the microcontroller, makes possible the autonomous operation of the corresponding system by combining the low cost, the limited complexity as well as the installation simplicity in the final applications.

Thus, a microcontroller is implemented in a single integrated circuit with some dozens of pins. Figures 1.5a and 1.5b show the microcontroller models ATmega328 and ATmega32 respectively.

Due to the fact that the microcontroller contains the needed components for autonomous operation (fig. 1.6), the corresponding external circuits (which are much simpler than the supplementary units and circuits of a microprocessor) consist only of electronic components for the power supply, the reset circuit and the optional clock oscillator (the microcontroller can be operated only with an internal clock in a lower frequency).

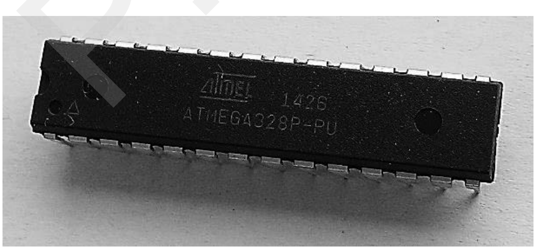


Figure 1.5a
The microcontroller ATmega328

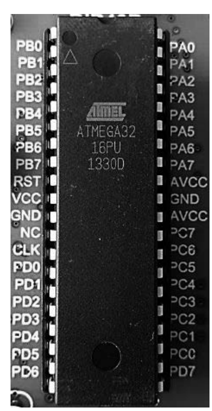


Figure 1.5b
The microcontroller ATmega32

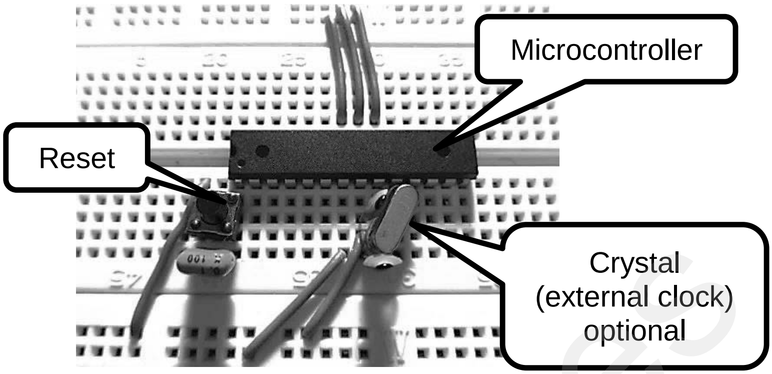


Figure 1.6 External microcontroller circuit

This simplicity makes the microcontroller an ideal solution for mobile and other applications where it can be embedded to support autonomous operation by implementing the control tasks of the target application.

From the previous brief presentation it is obvious that the microprocessors and the microcontrollers start from different approaches and are developed for applications and systems of different purpose.

Table 1.1 summarizes the differences between microprocessors and microcontrollers.

Table 1.1 Basic differences of microprocessors and microcontrollers

Microprocessor	Microcontroller
An integrated circuit which only implements the central processing unit (CPU) and does not constitute a computer	Constitutes a "small computer" because it contains processing unit, memory, I/O controllers, etc.
Is used in general purpose systems that support many and complicated applications	Is used in systems where control is needed as well as small application construction size, low energy consumption, low cost, fast response in external events, etc.
Is used in cases where a high processing performance is needed. Offers high performance where complicated tasks and calculations are needed	Performs simple tasks and operations, while the processing performance is much lower as compared to microprocessors
Constitutes the heart of the modern computers (PCs)	Is ideal for the embedded systems (constitutes the heart of the autonomous systems like the everyday appliances)
Supports complicated mathematical calculations (e.g. contains a floating point unit)	Does not support such capabilities. Some calculations are only supported by special software
It has high cost (is also based on many complementary units and circuits to form a computer system)	The cost is extremely low (starting from \$2-\$3)
High power consumption (also due to the complementary units and circuits)	Very low power consumption (can be supplied from a single battery)
High clock frequency (e.g. 1-3GHz)	Low clock frequency (e.g. 1-30 MHz)

Higher delay to external event processing due to multiple software levels existence (e.g. operating system)	Has been designed especially for fast response to external events. Is ideal for processing sensor and other device data in real time
Does not used in real time systems due to software delay issues	Is ideal for real time systems where the results and the decisions have to be produced in limited time slices
The final system has a large physical size and thus cannot be used in size limited applications	The final system has a small size and thus constitutes an ideal solution where the available physical size of installation is limited
Long booting time	Very short booting time

Note
The previous table refers to typical cases of microprocessors and microcontrollers. There are cases of advanced microcontrollers with higher cost which offer higher processing performance as compared to some microprocessors.

1.3 Microcontroller structural components

Due to the fact that a microcontroller supports a great range of autonomous applications, it is obvious that it has the capability of communicating with the external environment ("outside" world) for exchanging and processing the corresponding signals in order to satisfy the applications requirements. For efficiently supporting the above capabilities, the microcontroller contains a number of components (units) such as Control Unit (MCU), memory, interrupt controller, signal converter, etc. As mentioned before, microprocessors are used in general purpose systems, while microcontrollers are used in special purpose systems. Figure 1.7 shows with more detail the components (units) that constitute a typical microcontroller.

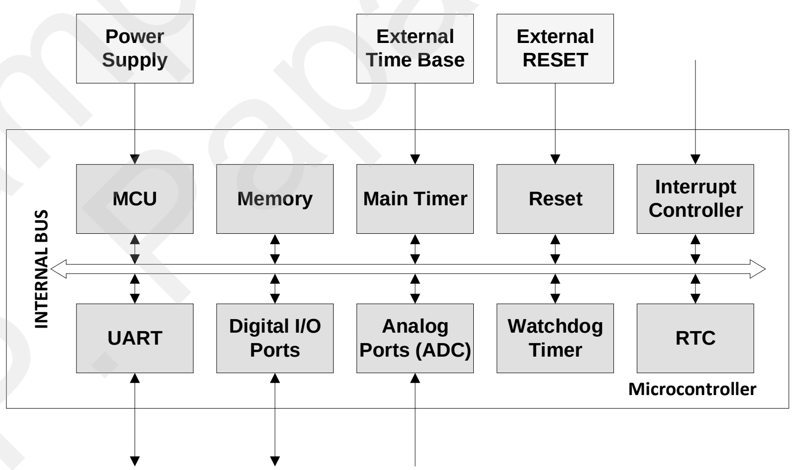


Figure 1.7 Basic microcontroller units

The basic operation of the microcontroller units in figure 1.7, is:

- **MCU (Microcontroller Control Unit)**. The MCU executes the program instructions that are stored in the program memory. Based on the internal architecture, there is a corresponding Assembly instruction set. The MCU operation requires registers (general and special purpose), arithmetic and logic unit (ALU), etc, like a typical computer CPU.

- **Memory**. The memory stores programs and data and can also be connected with an external memory unit. There are many types of memories that are used in microcontrollers such as:
    - EEPROM (reprogrammable with electrical signals)
    - Flash (fast storage)
    - Classic RAM (random access)
- **Main Timer**. The instructions execution as well as the internal units interaction for supporting a functional microcontroller is implemented using a common clock signal which is generated by a timer (oscillator). This oscillator can be internal or external (using a crystal for higher frequencies).
- **Reset**. Operation restart via an external circuit or internal program
- **Interrupt controller**. This controller implements the external event manipulation by interrupting the main program execution flow for a tiny time slice in order to execute the corresponding interrupt service routine which supports the event. If for example a switch for lighting a LED is momentary activated while the main program is under execution, then the interrupt controller activates the corresponding code section in order to light the LED and after a tiny time slice the execution flow returns to the main program.
- **UART (Universal Asynchronous Receiver Transmitter)**. This unit supports the asynchronous serial communication which is used for controlling external devices as well as data exchanging between the microcontroller and other devices. The serial communication includes the data preparation and framing for ensuring the correct reception at the destination.
- **Digital I/O Ports**. Every microcontroller has a set of ports that can be used as inputs or outputs. The signals that can be recognized by these ports are exclusively digital, that is signals of a 0 or 5V level (fig. 1.8).

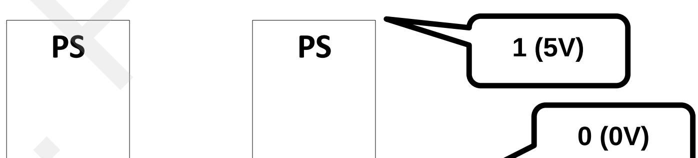


Figure 1.8 Digital Signal (PS=Positive Segment=5V)

- **PWM.** Any device that can be activated or deactivated with a digital signal, can also be controlled by a microcontroller. Another feature of the microcontrollers is the pseudoanalog signal generation for controlling analog devices. These signals are generated using the PWM (Pulse Width Modulation) capability of the microcontroller which is based on controlling the duration of the PS (Positive Segment) in relation with the signal period (Duty Cycle-DC). Figure 1.9 shows some waveforms with different DC that are generated by the digital outputs. By setting the DC percentage in combination with a high signal frequency (e.g. 500Hz), a different mean output voltage value can be achieved. Thus, intermediate voltage levels in the range [0,5]V can be produced. For example, a motor's control can be achieved using the above technique.

○

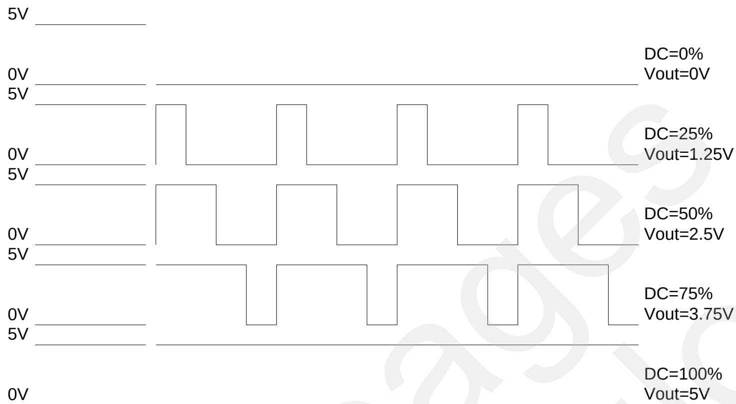


Figure 1.9 Some PWM indicative signals

- **Analog Ports (ADC)**. Many microcontrollers contain special converters (Analog to Digital Converter-ADC) which convert analog signals to digital. These ports usually read analog signals in the range [0,5]V, while this can be extended using a voltage divider. The resulted digital signal is represented in bits where the corresponding number of bits defines the conversion accuracy or resolution. If for example, the microcontroller uses 10bits resolution, then the resolution step is 5V/1024=0.0048V (2^{10}=1024).

- **Watchdog Timer**. Due to the fact that the systems that are based on a microcontroller have been designed to operate autonomously in the physical field of installation there must be a way the application operation to be recovered from a severe error (deadlock). For achieving this, a special counter which is reset periodically (through the application if there is no problem) is activated. Otherwise, if the counter exceeds a predefined limit (this means that the counter has not been initialized by the normal operation of the application) then it implements a reset to the microcontroller to recover the normal operation.

- **RTC (Real Time Clock)**. In some microcontroller models, special circuits are used for keeping the real time. Despite the existence of such circuits, the corresponding time is not accurate. Thus, external circuits or devices are used.

1.4 Microcontroller applications

As mentioned before, a microcontroller operates autonomously like a "small computer". The development of a full functional application which is fully controlled by a microcontroller, requires the combination of the most suitable external circuits, sensors, etc. Figure 1.10 shows a set of external "components" that can be combined in order to build the desired application.

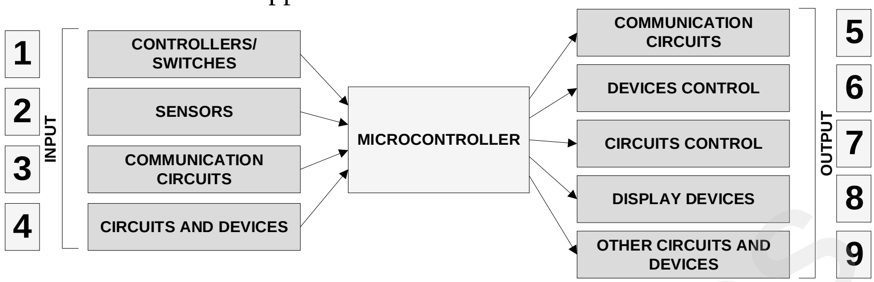


Figure 1.10 Component types for application development

Application example 1: Temperature display

There are already known features of a microcontroller and thus, this example is focused on the features and development of the application. This application measures the room temperature which is displayed through LEDs (the LEDs correspond to different temperature levels). The application of the first example consists of the following components:

(a) Temperature sensor (connected to an analog input of the microcontroller)- component type 2 (fig. 1.10)

(b) 3 LEDs (connected to digital outputs of the microcontroller through resistors) –component type 8 or 9 (fig. 1.10)

(c) A program which reads the sensor and controls the LEDs

Figure 1.11 shows the application circuit. For simplicity reasons, it is assumed that the three LEDs are lit according to the temperature ranges [10,19.99], [20,29.99] and [30,39.99] ^{o}C degrees respectively.

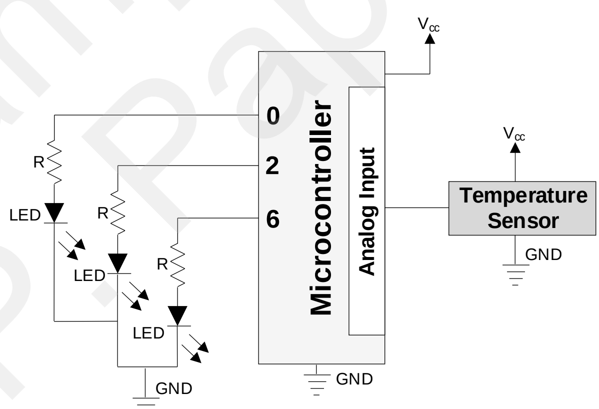


Figure 1.11 Application for displaying temperature

The following algorithm describes the logic of the corresponding program for displaying the temperature.

```
REPEAT
Turn off all the LEDs (write to digital outputs 0b00000000)
```

```
Read temperature sensor (T) from the analog port
If (T) belongs to [10,19.99]
    then write to digital outputs 0b00000001
Otherwise
    If (T) belongs to [20,29.99]
          then write to digital outputs 0b00000100
    Otherwise
    If (T) belongs to [30,39.99]
          then write to digital outputs 0b01000000
WHILE There is power supply
```

Of course, a scale conversion has to be done from the range [0,1023] (which is the output of the ADC with 10bits) to the corresponding levels which will be checked by the program. The symbol `0b` means that the number is binary, where each bit represents the state of a pin in the output port of the microcontroller. Thus, if the outputs are 8, numbered from 7 to 0, then the sequence 0b00000001 means that the pin 0 where the first LED is connected will be set to 5V (active for the temperature range [10,19.99]). Due to the 5V voltage, a current will flow in the LED circuit and the LED will be lit.

In a little different application, the temperature can be displayed in seven segment display units (fig. 1.12). The microcontroller reads the analog output of the sensor and converts the signal to digital form. The digital data (temperature measurement) are converted to discrete digits which are sent to the seven segment display units in real time.

Figure 1.12 Application for displaying temperature using seven segment display units

Application example 2: Controlling the air-condition

An application for controlling the air-condition remotely (wireless communication) can be developed (fig. 1.13) by combining the proper components. The corresponding settings are defined by the user. The current temperature (measured by the application) is compared with the user temperature settings. Based on this comparison, the air-condition is adapted to the needed operation requirements. Thus, the room temperature remains stable or is adapted to the new temperature level which is defined by the user.

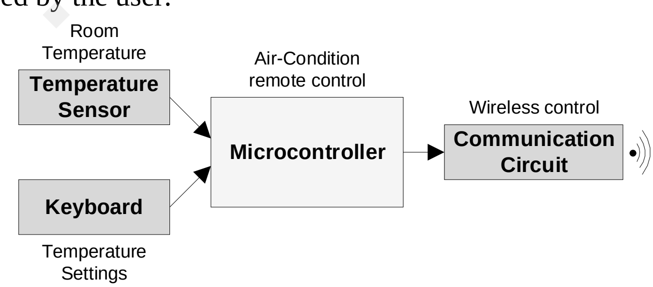


Figure 1.13 Controlling an air-condition

Application example 3: Controlling a robot

Due to autonomous operation with simple circuits and the small size, the microcontrollers, are ideal to be embedded in mobile systems such as an autonomous robot. In this example, the "control center" of the robot controls the motor's activation according to signals that are sent by the light and distance sensors (fig. 1.14). The microcontroller represents the "control center" and is programmed to make a combined processing of the sensors measurements and the remote control instructions in order to decide for the robot movement. The movement is controlled by small signals that are sent from the microcontroller to the motor control circuits.

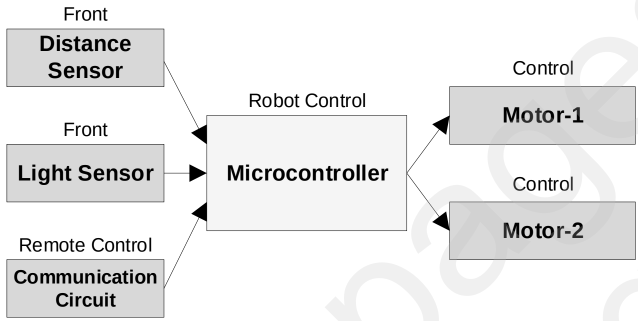


Figure 1.14 Robot control

Application example 4: Car controller/computer

Modern cars display illuminated indications to the driver about the car status (fig. 1.15). The above status is related to special adapted sensors that are installed in critical points of interest in the car. Exploiting the low cost of the microcontrollers and based on the requirements for updating the status indications in real time, a different microcontroller can be used for every sensor group. Thus, the processed measurements are sent to the central microcontroller which constitutes the core unit of the car controller/computer in order to display indications and to store faults for later processing by the service personnel. Of course, a different architecture with only one microcontroller and many sensors can be implemented.

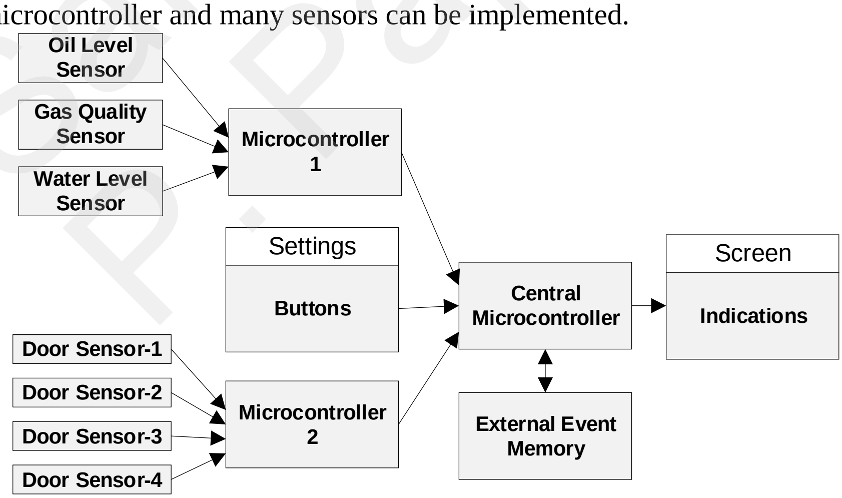


Figure 1.15 Car controller/Computer

1.5 Popular Microcontrollers

From all the above, it is obvious that there is a huge number of microcontroller applications. Due to everyday applications and other advanced services, many companies offer microcontrollers that are available to engineers and hobbyists. An engineer has to choose the most suitable microcontroller for developing the desired application.

Table 1.2 shows the most popular microcontroller models.

Table 1.2 Popular microcontrollers (indicative)

Company	Internet url	Indicative model
Texas Instruments	http://www.ti.com/	MSP430F1x
Microchip	http://www.microchip.com/	PIC16F84A
Zilog	http://www.zilog.com/	S3F80P5
Freescale	http://www.freescale.com/	HC08EY
Atmel (now Microchip)	http://www.atmel.com/	AVR ATmega328
Parallax	http://www.parallax.com/	Basic Stamp
Maxim	http://www.maximintegrated.com/	MAXQ611

The most popular models are offered by Microchip and Parallax.

EXERCISE SHEETS
Knowing the microcontrollers

1. Note every phrase as right or wrong

Phrase	Right	Wrong
The microcontrollers are usually embedded in autonomous systems		
The processing unit in microcontrollers and microprocessors executes the program instructions		
The memory is a two dimensional array for storing addresses and data		
The microprocessor operation requires supplementary external circuits		
The board for supporting a microcontroller contains more complex circuits than the board for supporting a microprocessor		
The autonomous operation of the microcontroller requires an extremely simple external circuit		
Microcontrollers have in general higher cost than the microprocessors		

Microprocessors are ideal for embedded systems		
Microprocessors have in general lower power consumption than microcontrollers		
The digital I/O ports of the microcontroller recognize signals in the range [0-10]V		
The PWM technique is used for controlling analog devices		
More bits in the ADC unit means higher resolution		

2. Search in the literature the frequency and period definition for a periodical signal, as well as the corresponding prefixes for expressing the numbers' magnitudes easier. Based on the above, fill the following tables.

Hz	KHz	MHz	GHz
		1	
16300			
			1.3
	345.8		
		24	
1			
	1		
			1
		4.77	
	20.8		

nSec	μSec	mSec	Sec
			1
	1		
		23.1	
1300			
			5
		4	
	250		
43789			
	1000		
		1000	

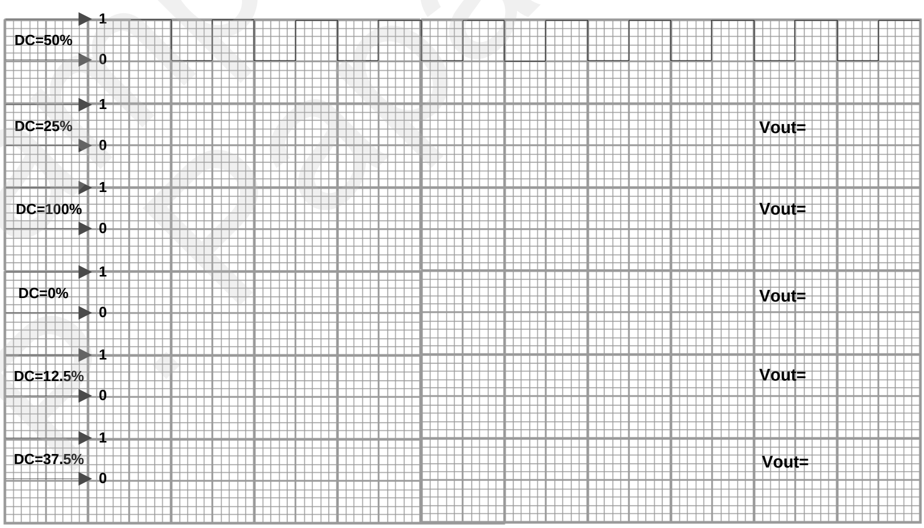
Frequency	Period
15KHz	
	1μSec
16MHz	
	0.3μSec
1GHz	
	1Sec

1Hz	
	1mSec

3. The Duty Cycle is the duration of the positive segment (PS) as compared to the signal period (is expressed as a ratio). If the logical 0 is 0V and the logical 1 is 5V, fill the following tables.

Duty Cycle	Pseudoanalog voltage
32.2%	
	4.45V
50%	
25%	
	4.75V
11%	
	3.75V
100%	
	5V
	0V
0%	
	1.25V
95%	
	2.25V
22.5%	

4. Draw in the scaled paper the waveforms for the following Duty Cycles (DCs) and calculate the corresponding mean voltage V_{out}.



5. Combine the components of figure 1.10 and fill the following table.

Input components	Application	Output components
	Bus ticket control system	
	Microwave oven	
	Television remote control	
	Automated bank transaction machine	
	Autonomous robotic vehicle	
	Mobile phone	
	Digital distance counter for bike	

Download material

http://panospapazoglou.gr/support/

Email

p.m.papazoglou@hotmail.com