

# APLIKACIJSKI ČLANKI APPLICATION ARTICLES

## MAKING IT EASY WITH MICROCONTROLLERS

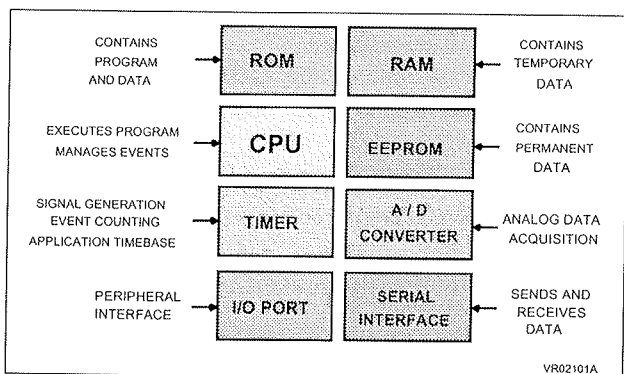
### WHAT IS A MICROCONTROLLER?

A few years ago, system control functions were implemented using **logic components** and were usually large, heavy boxes. Later on, microprocessors were used and the entire controller could fit onto a small circuit board. As the process of miniaturization continued, all of the components needed for a controller were built right onto **one chip**. By only including the features specific to the task, cost is relatively low.

A typical microcontroller has bit manipulation instructions, easy and direct access to I/O, and quick and efficient interrupt processing. Therefore, a microcontroller is a **highly integrated device** which includes, on one chip, all or most of the parts needed to perform an application control function.

Microcontrollers come in many varieties. Depending on the power and features that are needed, customers might choose a 4, 8, 16, or 32 bit microcontroller.

Figure 1. Typical MCU Block Diagram



#### ROM (Read Only Memory)

ROM is usually used to store program instructions. ROM is the least expensive mean of storing a program in a microcontroller, especially for high volume manufacturing.

#### RAM (Random Access Memory)

During in the execution of a program, data have to be saved for later use. Data are usually stored in RAM.

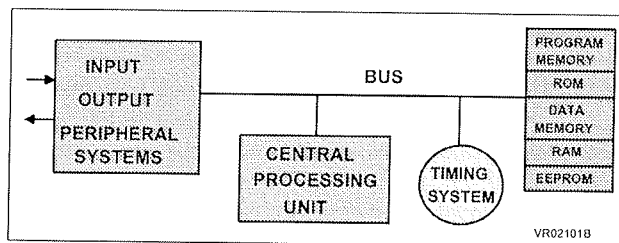
#### EEPROM (Electrically Erasable Programmable Read Only Memory)

There is another way to store data in a microcontroller: EEPROM is used to store data that must be saved through a power down cycle.

#### CPU (Central Processing Unit)

It is the brain of the system that processes all data and their travel along the bus. For example, in order to execute a program, the CPU will read the first instruction from program memory. This instruction is decoded by the CPU and executed. At the completion of the execution of the instruction, the next instruction is fetched from memory and is executed. This procedure is repeated until the end of the program (or an endless loop) is found.

Figure 2. Typical Microcontroller Block Organization



#### TIMERS

The timer or timing system makes it possible to measure and time external and internal events. The power of the timers varies greatly between different MCUs.

#### I / O (INPUT / OUTPUT) PORTS

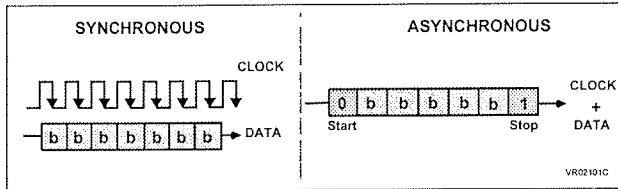
Most microcontrollers have several digital I/O ports that are used to drive a LED or get a keyboard input. Usually, a port consists of eight or less bits, usually programmable as either input or output bits.

#### SERIAL INTERFACE

Serial interface are used to exchange data with the external world. Many microcontrollers have both **asynchronous** and **synchronous** communications peripherals built in. Usually, an asynchronous interface is called a serial communication interface (**SCI** or **UART**) while the synchronous interface is called a serial peripheral interface (**SPI**). A typical SCI application is to connect a PC for debugging purpose while a typical SPI application is to connect an external EEPROM.

A synchronous bus includes a separate line for the clock signal which simplifies the transmitter and receiver but is more susceptible to noise when used over long distances. With an asynchronous bus the transmitter and receiver clocks are independent, and a resynchronization is performed for each byte at the start bit.

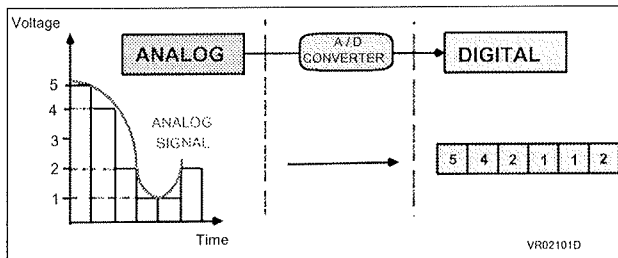
Figure 3. Synchronous and Asynchronous Communication Principles



**A/D CONVERTER**

Converts an external analog signal (typically relative to voltage) into a digital representation. Microcontrollers that have this feature can be used for instrumentation, environmental data logging, or any application that lives in an analog world.

Figure 4. A/D Converter Principle



**1 TYPICAL MICROCONTROLLER APPLICATIONS**

Microcontrollers are frequently found in home appliances (microwave oven, refrigerators, television and VCRs, stereos), computers and computer equipment (laser printers, modems, disk drives), cars (engine control, diagnostics, climate control), environmental control (greenhouse, factory, home), instrumentation, aerospace, and thousands of other uses. In many items, more than one processor can be found.

While microprocessors target the maximum processing performance, the purpose of microcontrollers is to implement a set of control function in the most cost effective way. Although controlling a microwave oven with a Pentium (TM) might seem an attractive idea, it can be easily accomplished with an ST6.

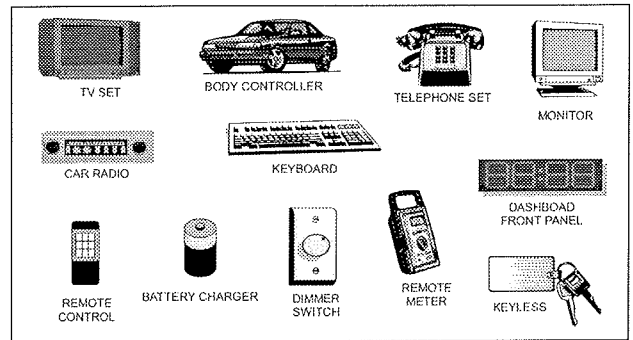
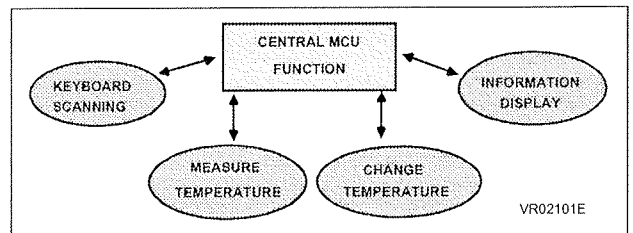


Figure 5. Typical MCU Applications

In a typical application, the MCU has to manage several tasks according to their priority or to the occurrence of external events (new command send by the keyboard, external temperature rise,...)

Figure 6. Example of MCU Task Management.

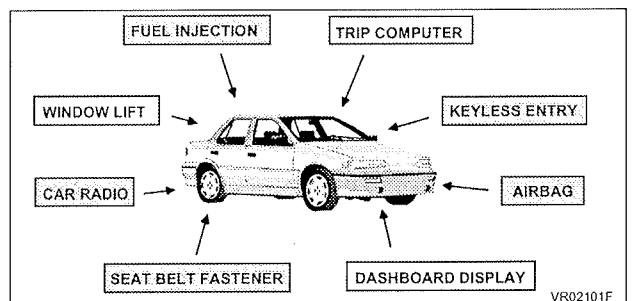


The ability to manage control tasks by hardware or by software is the main performance indicator for MCUs.

**EXAMPLE: THE AUTOMOTIVE MARKET**

The automotive market is the most important single driving force in the microcontroller market, especially at its high end. Several microcontroller families were developed specifically for automotive applications and were subsequently modified to serve other embedded applications.

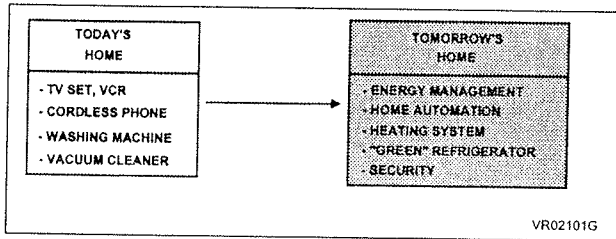
Figure 7. MCU Applications in Automotive



The automotive market is demanding in term of device performance and component reliability. Electronics must operate under **extreme temperatures** and be able to withstand **vibration, shock, and EMI**. The electronics must be reliable, because a failure that causes an accident can (and does) result in multi-million dollar lawsuits. Reliability standards are high - but because these electronics also compete in the consumer market - they have a low price tag.

Automotive is not the only growing market. According to DataQuest, a research institute, the average North American's home contains 35 microcontrollers. By the year 2000, that number will grow to around 240. Consumer electronics is a booming business.

Figure 8. MCU Applications in Today's and Tomorrow's Home



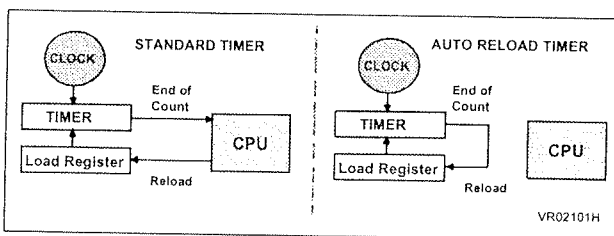
## 2 ADDITIONAL MICROCONTROLLER FEATURES

### 2.1 Timers

**Watchdog timer.** A watchdog timer provides a means of graceful recovery from a system problem. This could be a program that goes into an endless loop, or a hardware problem that prevents the program from operating correctly. If the program fails to reset the watchdog at some predetermined interval, a hardware reset will be initiated. The bug may still exist, but at least the system has a way to recover. This is especially useful for unattended systems.

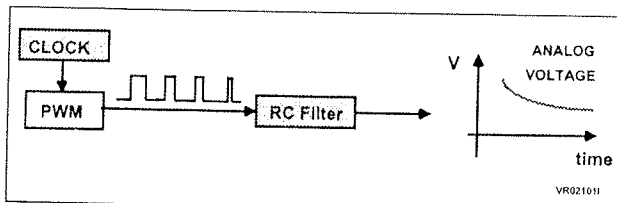
**Auto Reload Timer.** Compared to a standard timer, this timer automatically reloads its counting value when the count is over, therefore sparing a waste of CPU resource.

Figure 9. Standard Timer and Auto-Reload Timer Principle



**Pulse Width Modulator.** Often used as a digital-to-analog conversion technique. A pulse train is generated and regulated with a low-pass filter to generate a voltage proportional to the duty cycle.

Figure 10. PWM Principle

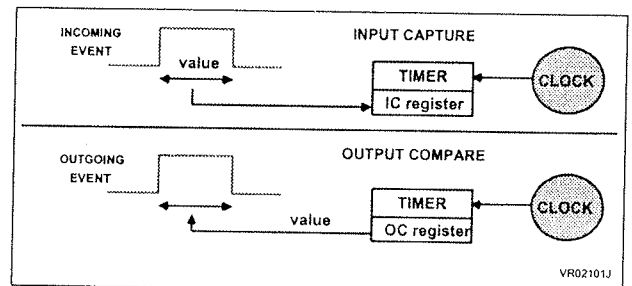


**Pulse Accumulator.** A pulse accumulator is an event counter. Each pulse increments the pulse accumulator register, recording the number of times this event has occurred.

**Input Capture.** Input Capture can measure external frequencies or time intervals by copying the value from a free running timer into the input capture register when an external event occurs.

**Output Compare.** Output Compare can time an external event by sending a value stored inside the output compare register.

Figure 11. Input Capture and Output Capture Timer Functions



### 2.2 Digital signal processors (DSP)

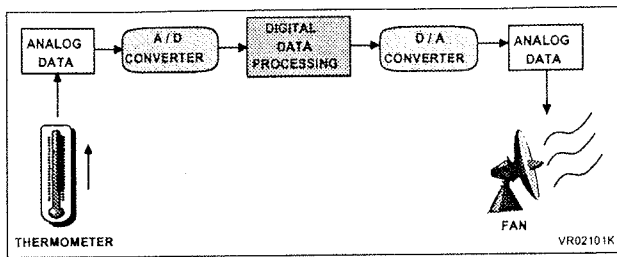
Microcontrollers react to and control events, whereas DSPs execute repetitive math-intensive algorithms. Today many embedded applications require both types of processors, and semiconductor manufacturers have responded by introducing microcontrollers with on-chip DSP capability and DSPs with on-chip microcontrollers.

The most basic thing a DSP will do is a MACC (Multiply and ACCumulate). The number of data bits a DSP can Multiply and ACCumulate will determine the dynamic range (and therefore the application).

### 2.3 Analog and digital signals

We live in an analog world where the information we see, hear, process, and exchange with each other, and with our mechanical and electronic systems, is always an **analog quantity**: pressure, temperature, voltage and current are always analog entities. They can be **digitized** for more efficient storage and transmission, but the interface (input and output) is almost always analog. Thus the essence of analog electronics lies in sensing continuously varying information, converting it to digital and reshaping the digital data to an analog signal at the other end. Mixed analog-digital devices are being used increasingly to integrate the complex functions of high-speed telecommunications, or the real-time data processing demanded by industrial control systems and automotive systems.

Figure 12. Mixed A/D System Example



**Analog to digital conversion (A / D).** Converts an external analog signal (typically relative to voltage) and converts it to a digital representation. Microcontrollers that have this feature can be used for instrumentation, environmental data logging, or any application that lives in an analog world. Various types of A/D converters that can be found.

**Comparator.** One or more standard comparators can sometimes be placed on a microcontroller die. These comparators operate much like standard comparators however the input and output signals are available on the microcontroller bus.

**Digital to analog conversion (D/A).** This feature takes a digital number and converts it to an analog output. The number 50 would be changed to the analog output of  $(50 / 256 * 5 \text{ Volts}) = .9765625 \text{ V}$  on a 8 bit / 5 Volt system.

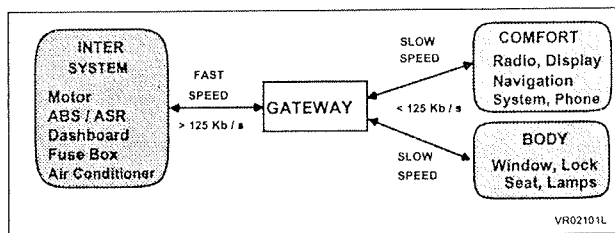
## 2.4 Communication

### CAN & J1850

CAN (Controller Area Network) is a multiplexed wiring scheme that was developed jointly by BOSCH and Intel for wiring in automobiles. J1850 is the SAE (Society of Automotive Engineers) multiplexed automotive wiring standard that is currently in use in North America.

The CAN specification seems to be the one that is being used in industrial control both in North American and Europe. With lower cost microcontrollers that support CAN, CAN has a good potential to take off.

Figure 13. CAN Principle



### I2C BUS - Inter-Integrated Circuit Bus (PHILIPS)

The I2C bus is a simple 2 wire serial interface developed by Philips. It was developed for 8 bit applications and is widely used in consumer electronics, automotive and industrial applications. In addition to microcontrollers, several peripherals also exist that support the I2C bus. The

I2C bus is a two line, multi-master, multi-slave network interface with collision detection. Up to 128 devices can exist on the network and they can be spread out over 10 meters. Each node (microcontroller or peripheral) may initiate a message, and then transmit or receive data. The two lines of the network consist of the serial data line and the serial clock line. Each node on the network has a unique address which accompanies any message passed between nodes. Since only 2 wires are needed, it is easy to interconnect a number of devices.

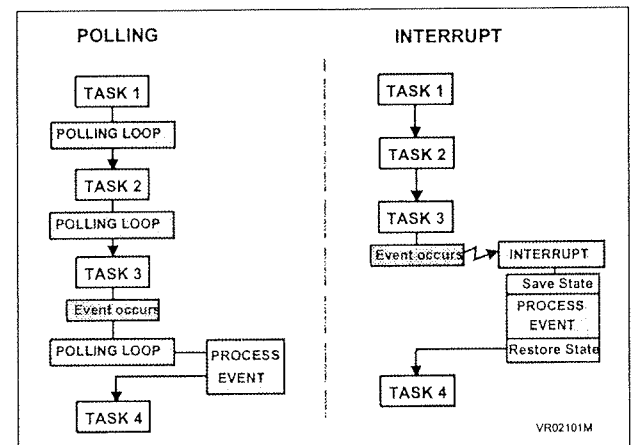
**UART.** A UART (Universal Asynchronous Receiver Transmitter) is a serial port adapter for asynchronous serial communications.

**USART.** A USART (Universal Synchronous / Asynchronous Receiver Transmitter) is a serial port adapter for either asynchronous or synchronous serial communications. Communications using a USART are typically much faster (as much as 16 times) than with a UART.

## 2.5 Interrupts

**Polling.** Polling is what you have to do if your microcontroller does not have interrupts or if what you want to do is not time critical. It is a software technique whereby the controller continually asks a peripheral if it needs servicing. The peripheral sets a flag when it has data ready for transferring to the controller, which the controller notices on its next poll. Several peripherals can be polled in succession, with the controller jumping to different software routines, depending on which flags have been set.

Figure 14. Polling versus Interrupt



**Interrupts.** Rather than have the microcontroller continually polling - that is, asking peripherals (timers / UARTS / A/Ds / external components) whether they have any data available (and finding most of the time they do not), a more efficient method is to have the peripherals tell the controller when they have data ready. The controller can be carrying out its normal function, only responding to peripherals when there is data to respond to. On receipt of an interrupt, the controller suspends its current operation, identifies the interrupting peripheral, then jumps to the appropriate

ate interrupt service routine. The advantage of interrupts, compared with polling, is the speed of response to external events and reduced software overhead (of continually asking peripherals if they have any data ready).

Most microcontrollers have at least one external interrupt, which can be edge selectable (rising or falling) or level triggered. Both systems have advantages. Edge is not time sensitive, but it is susceptible to glitches. Level must be held high (or low) for a specific duration (which can be a pain but is not susceptible to glitches).

### 3 POWER SUPPLY ISSUES IN MCUS

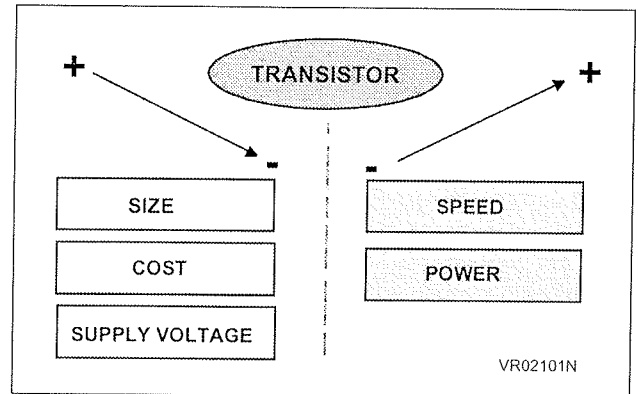
Since automotive applications have been the driving force behind most microcontrollers, and 5 Volts was very easy to do in a car, most microcontrollers only supported 4.5 - 5.5 V operation. In the recent past, as consumer goods were beginning to drive major segments of the microcontroller market, and became portable and lightweight, the requirement for 3 volt (and lower) microcontrollers became urgent. 3 volts means 2 battery solution, lower voltage and longer battery life. Most low voltage parts in the market today are simply 5 volt parts that were modified to operate at 3 volts (usually at a **performance loss**). Some micros being released now are designed from the ground up to operate properly at 3.0 (and lower) voltages, which offer a performance level comparable to 5 volt devices.

But why are voltages going down on ICs? There are a few interesting rules of thumb regarding transistors:

- 1) The amount of power they dissipate is proportional to their size. If you make a transistor half as big, it dissipates half as much power.

- 2) Their propagation delay is proportional to their size. If you make a transistor half as big, it's twice as fast.
- 3) Their cost is proportional to the square of their size. If you make them half as big, they cost one quarter as much.

Figure 15. Transistor Parameter Scheme



For years people have been using 5 Volts to power integrated circuits. Because the transistors were large, there was little danger damaging the transistor putting this voltage across it. However, now that the transistors are getting so small, 5 Volts would now destroy them. The only way around this is to start lowering the voltage. This is also why people are now using 3 (actually 3.3) Volt logic, and this will certainly lead to lower voltages in the next few years.

ST Microelectronics  
 Micro Division  
[www.st.com](http://www.st.com)