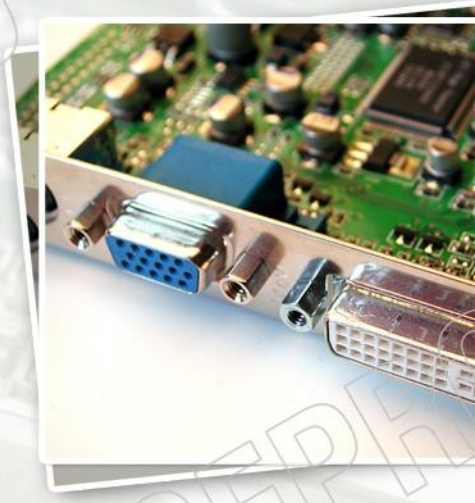
Interfacing Digital Stop Watch Using Atmega32

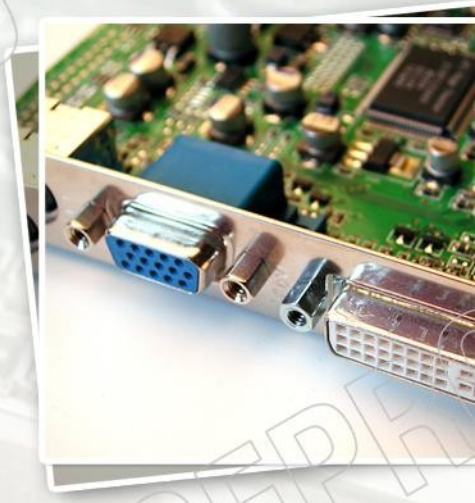
Presented by

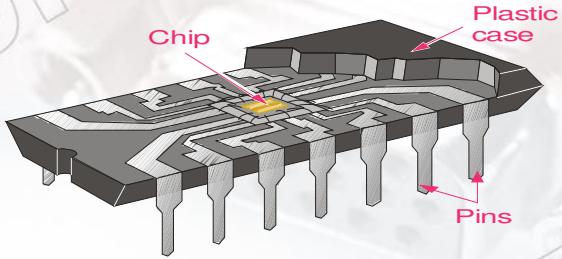
- Tanvir Ahammad-108407
- Md. Sajib Al Mamun-108404
- Md. Alamin-108416
- Md. Afijer Rahman-108428
- Ibn Zobayer Abdullah-108402

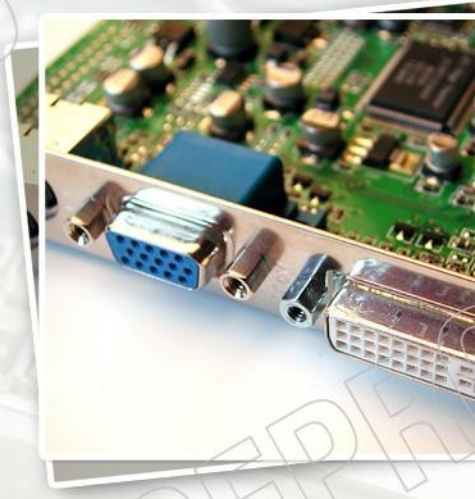


INDEX

- Introduction to Microcontroller
- AVR Microcontroller
- Introduction to Microcontroller Port
- Microcontroller System Development
- Embedded C Programming
- Designing Proposed System - Digital Stop Watch
- Project Demonstration



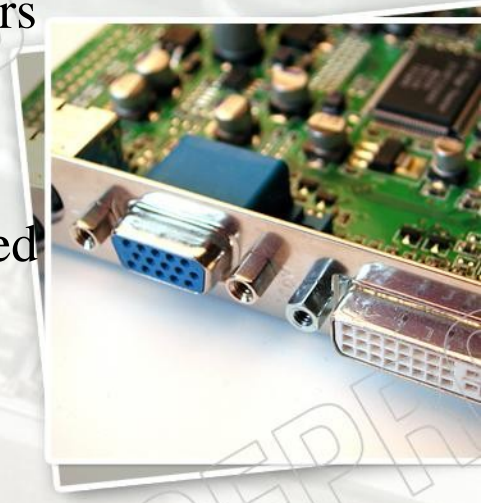


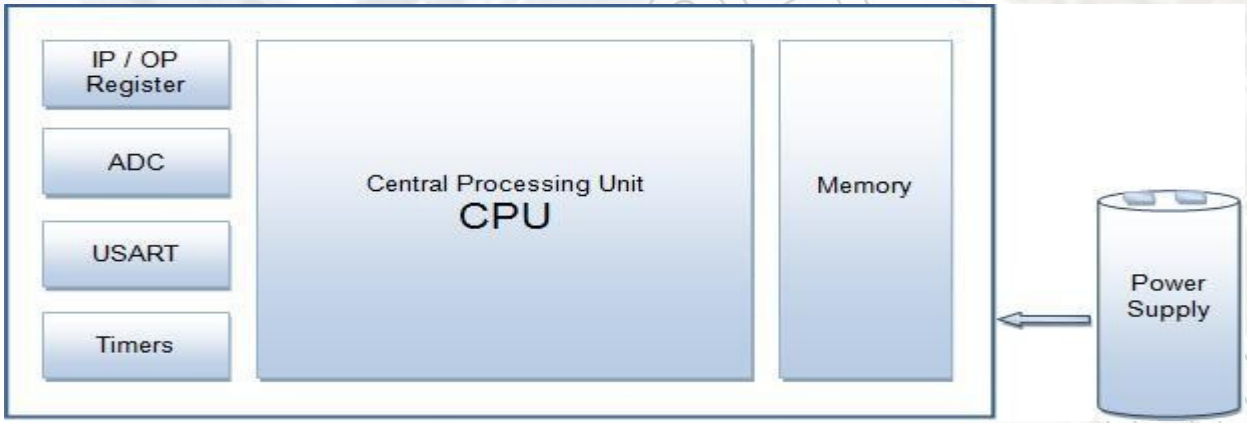


Introduction to Microcontroller

Introduction to Microcontroller

- Includes number of peripherals like RAM, EEPROM, Timers etc.
- Performs some predefined task such as temperature control
- The 8051, AVR and PIC microcontrollers are most renowned

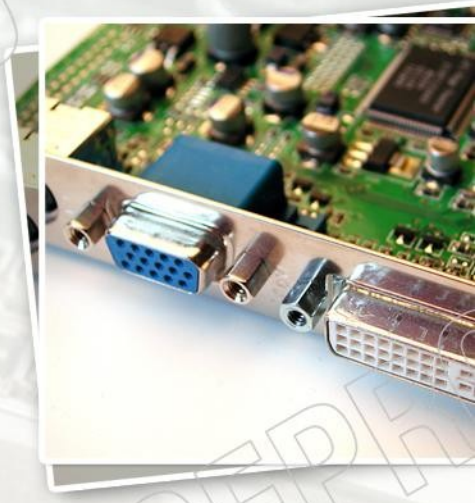




Courtesy: http://www.engineersgarage.com/articles/avr-microcontroller

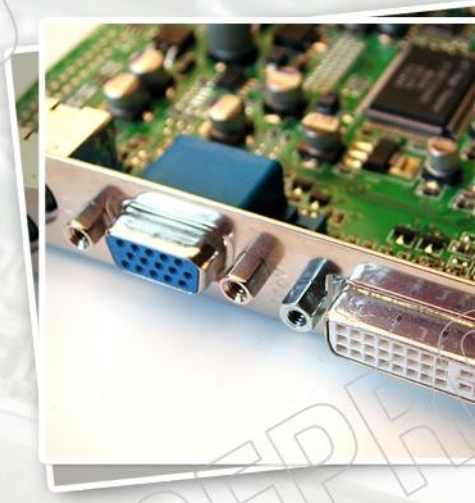
Why use Microcontrollers?

- Simpler construction and control.
- Lower cost.
- Low total area occupied on the circuit board.
- Flexibility – Changes can be made by simply changing the software (code used to program the microcontroller).
- Software implementation (firmware) is usually easier than its hardware counterpart.



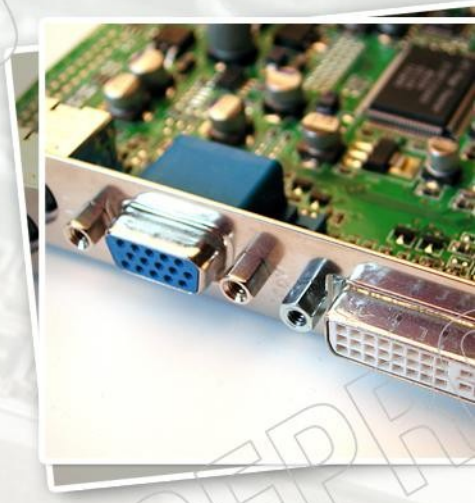
MPU Vs. MCU

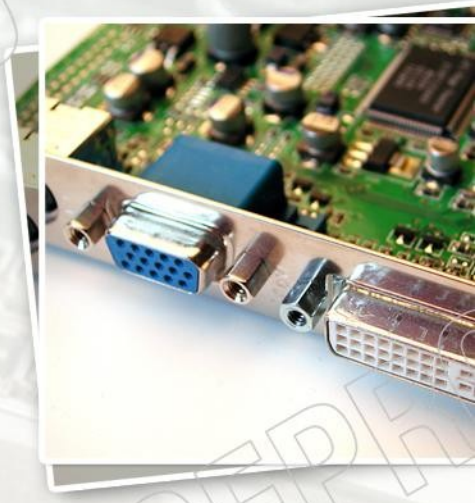
Microprocessor	Microcontroller
1) Off chip memory.	1) On Chip Memory.
2) CPU Stands alone.	2) CPU Standing with on chip peripherals,Timers, interrupts etc.
3) Ex: 8085, Core 2 Duo	3) 8051, PIC, AVR



Different Families of Microcontrollers

- ARM Cortex-M
- Atmel AVR (8-bit), AVR32 (32-bit), and AT91SAM (32-bit)
- Cypress Semiconductor's M8C Core used in their PsoC
- Freescale ColdFire (32-bit) and S08 (8-bit)
- Freescale 68HC11 (8-bit)
- Intel 8051
- Microchip Technology PIC
- NXP Semiconductors LPC Series (8-bit)
- Parallax Propeller
- Rabbit 2000 (8-bit)
- Renesas Microcontrollers
- STMicroelectronics STM8 (8-bit), ST10 (16-bit) and STM32 (32-bit)
- Texas Instruments TI MSP430 (16-bit) C2000 (32-bit)
- Toshiba TLCS-870 (8-bit/16-bit).

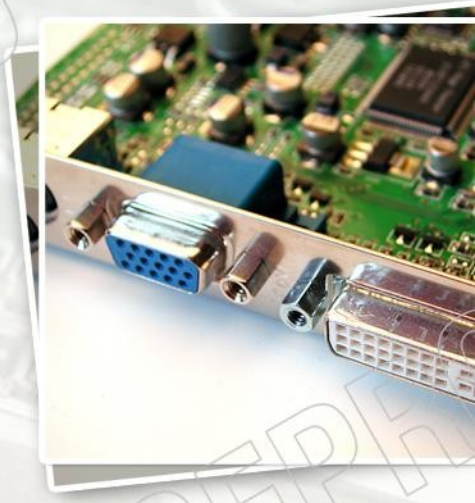




AVR Microcontroller

AVR Microcontroller

- Developed in the year 1996 by Atmel Corporation
- The architecture was developed by Alf-Egil Bogen and Vegard Wollan.
- AT90S8515 was the first microcontroller based on AVR architecture
- 8-bit microcontroller belonging to the family of Reduced Instruction Set Computer (RISC)



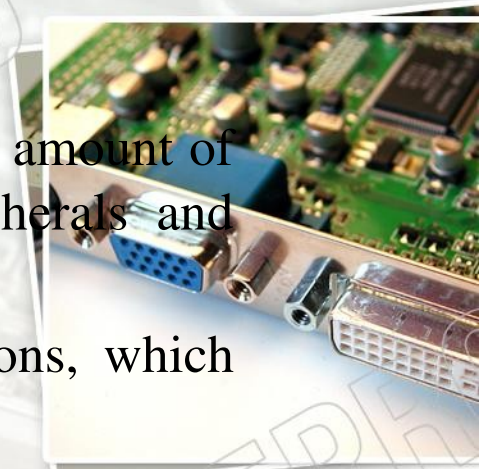
What's special about AVR?

- They are fast: executes most of the instructions in single execution cycle
- About 4 times faster than PICs
- They consume less power and can be operated in different power saving modes

	8051	PIC	AVR
SPEED	Slow	Moderate	Fast
MEMORY	Small	Large	Large
ARCHITECTURE	CISC	RISC	RISC
ADC	Not Present	Inbuilt	Inbuilt
Timers	Inbuilt	Inbuilt	Inbuilt
PWM Channels	Not Present	Inbuilt	Inbuilt

Different types of AVR Microcontrollers

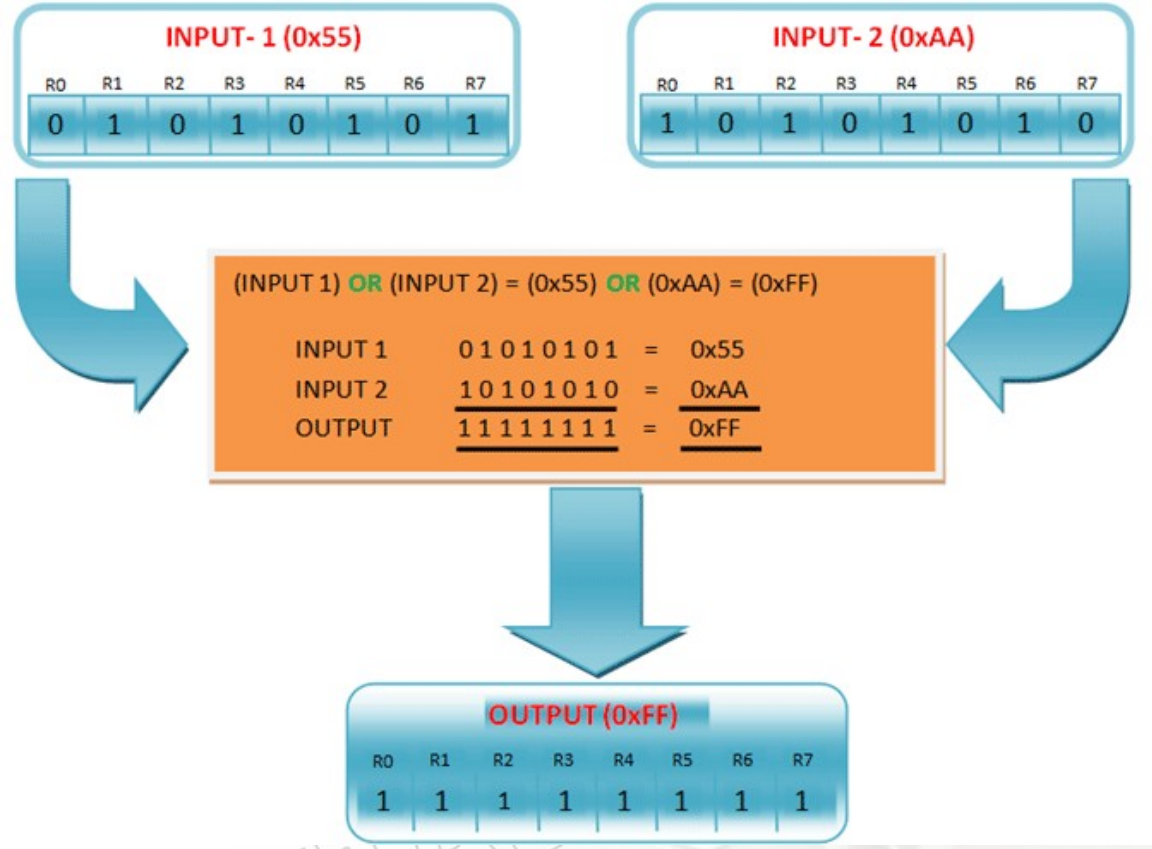
- **TinyAVR** – Less memory, small size, suitable only for simpler applications
- **MegaAVR** – These are the most popular ones having good amount of memory (upto 256 KB), higher number of inbuilt peripherals and suitable for moderate to complex applications.
- **XmegaAVR** – Used commercially for complex applications, which require large program memory and high speed.



Series Name	Pins	Flash Memory	Special Feature
TinyAVR	6-32	0.5-8 KB	Small in size
MegaAVR	28-100	4-256KB	Extended peripherals
XmegaAVR	44-100	16-384KB	DMA , Event System included

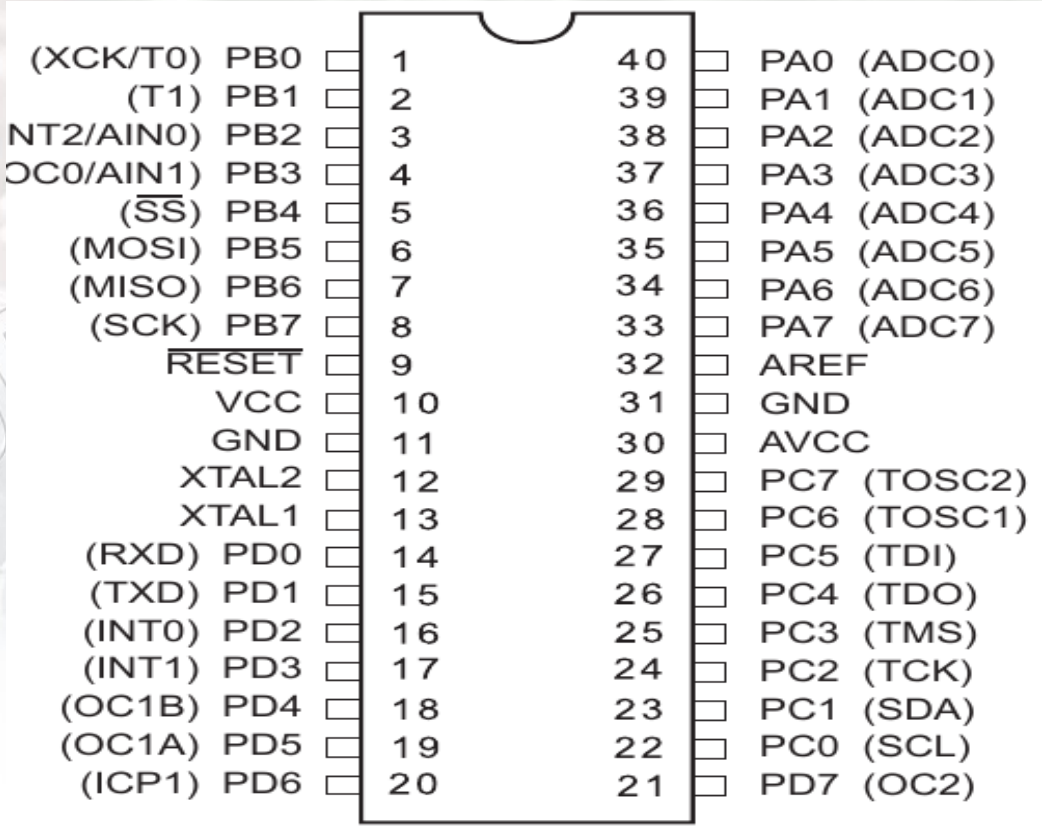
What is 8-bit MicrAocontroller?

- Capable of handling 8-bit data
- The registers are of 8-bits
- The CPU takes values from two INPUT registers, performs the logical operation and stores the value into the OUTPUT register
- All this happens in 1 execution cycle.



ATMEGA 32

- 32Kbytes of In-System Programmable (ISP)
- Flash program memory
- 32 × 8 General Purpose Working Registers
- 2KBytes of SRAM
- 1024 Bytes of EEPROM
- Available in 40-Pin DIP
- 8-Channel 10-bit ADC
- External and Internal Interrupt Sources
- Two 8-bit Timers/Counters
- One 16-bit Timer/Counter
- Real Time Counter with Separate
- Oscillator
- 4 PWM Channels
- Programmable Serial USART
- Master/Slave SPI Serial Interface



More About ATMEGA 32

Operating Voltages

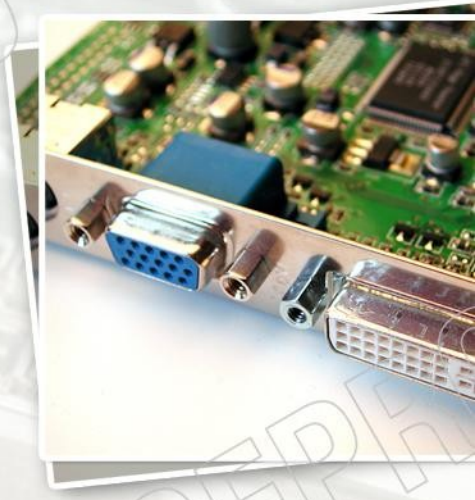
- 2.7V - 5.5V for ATmega32L
- 4.5V - 5.5V for ATmega32

Speed Grades

- 0 - 8MHz for ATmega32L
- 0 - 16MHz for ATmega32

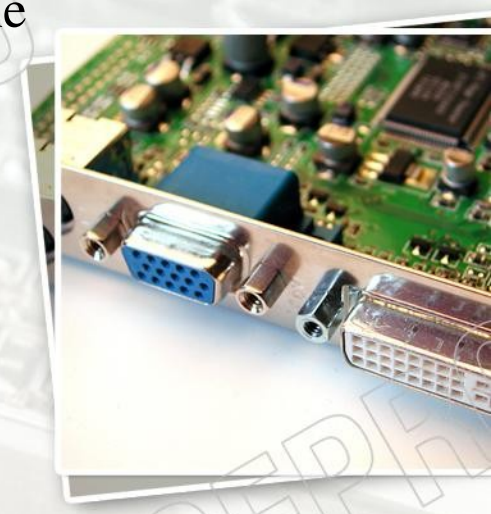
Power Consumption at 1MHz, 3V, 25°C

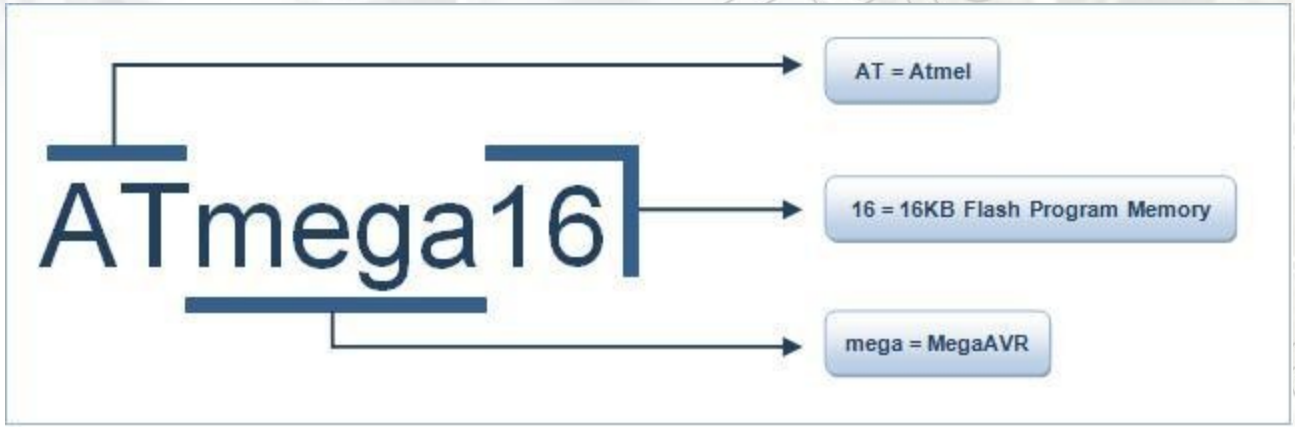
- Active: 1.1mA
- Idle Mode: 0.35mA
- Power-down Mode: < 1µA



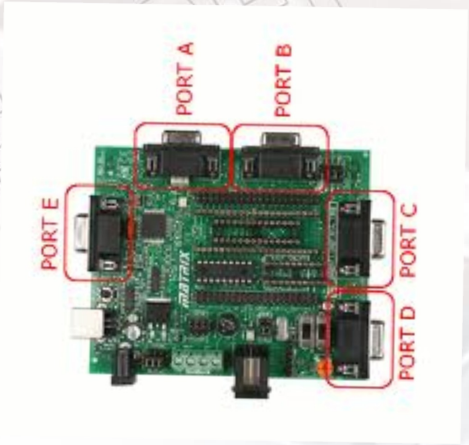
Naming Convention

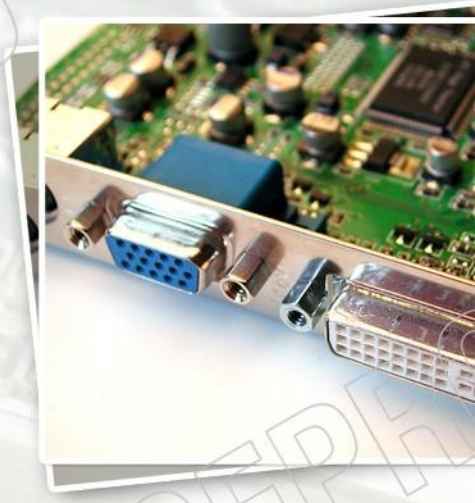
The AT refers to **Atmel** the manufacturer, **Mega** means that the microcontroller belong to MegaAVR category, e.g 16 signifies the memory of the controller, which is 16KB.





Courtesy: http://www.engineersgarage.com/articles/avr-microcontroller

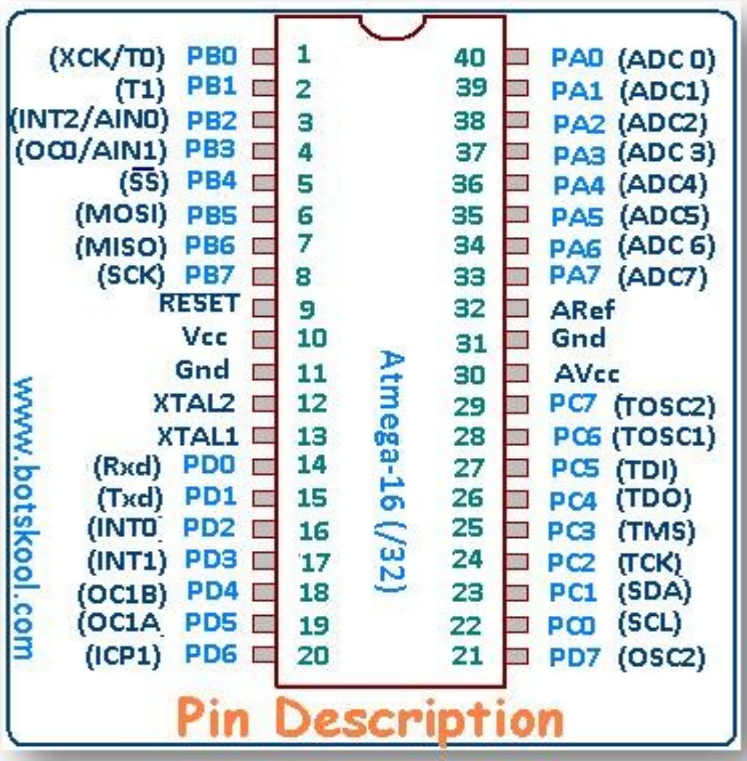




Introduction to Port of Microcontroller

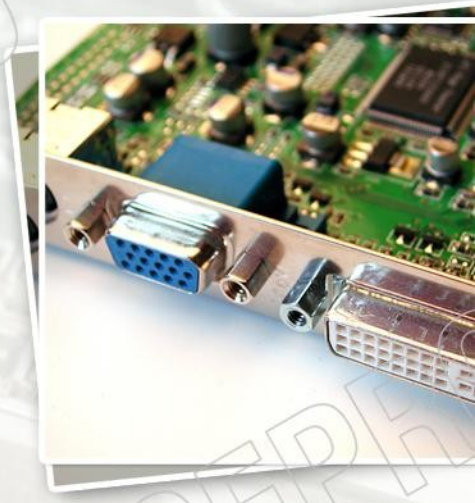
Microcontroller Pins

- Each pin of a microcontroller can serve multiple purposes
- For a particular function of a pin, the microcontroller must be configured to use that function



Configuring Pins for Particular Operation

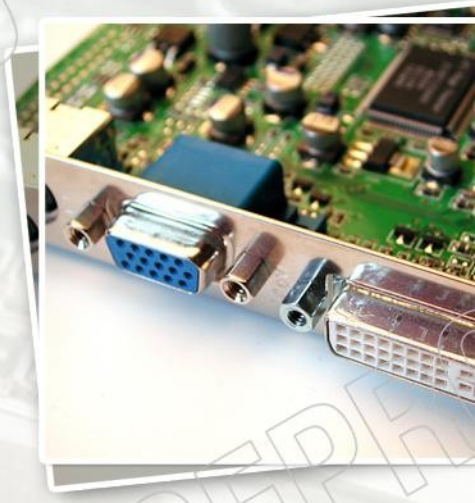
- The Pins are configured through registers
- A Register is a special high speed memory location that contains the status or configuration information of a microcontroller
- Information regarding the register and its configurations can be most reliably found in the manufacturer's datasheet of any microcontroller.



I/O Programming

I/O port has 3 registers associated with each port.
These three registers are

- DDRx (data direction register)
- PINx
- PORTx



DDR (Data Direction Register)

From this register we can decide the particular port as input or output.

- DDRx = 0xFF; //Output Port (Source)
- DDRx = 0x00; // Input Port (Sink)

PORT-B	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
Function	Output	Output	Input	Output	Input	Input	Input	Output
DDRB	1	1	0	1	0	0	0	1

PORT

If pin is defined as output

- PORTx = 1; //high output
- PORTx = 0; // Low Output

PORT-A	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
Value	High(+5V)	High(+5V)	Low(0V)	Low(0V)	Low(0V)	High(+5V)	High(+5V)	Low(0V)
PORTA	1	1	0	0	0	1	1	0

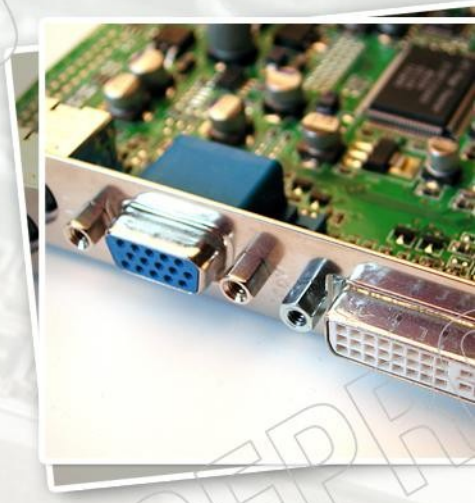
- If pin is Input pin

  Set bit 0 → Tri-Stated

  Set bit 1 → Pull Up

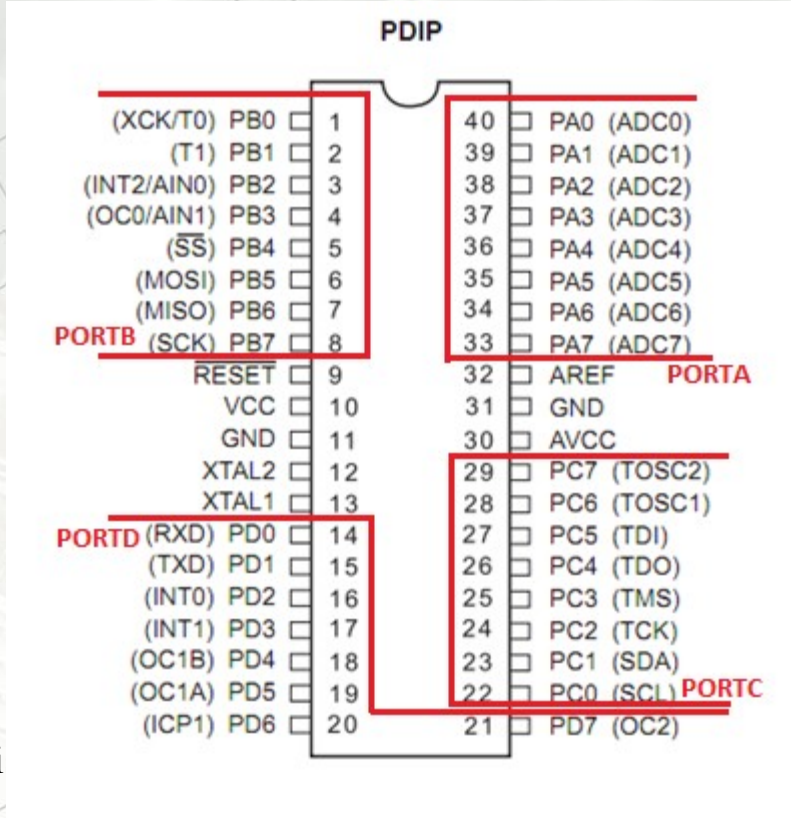
  Example

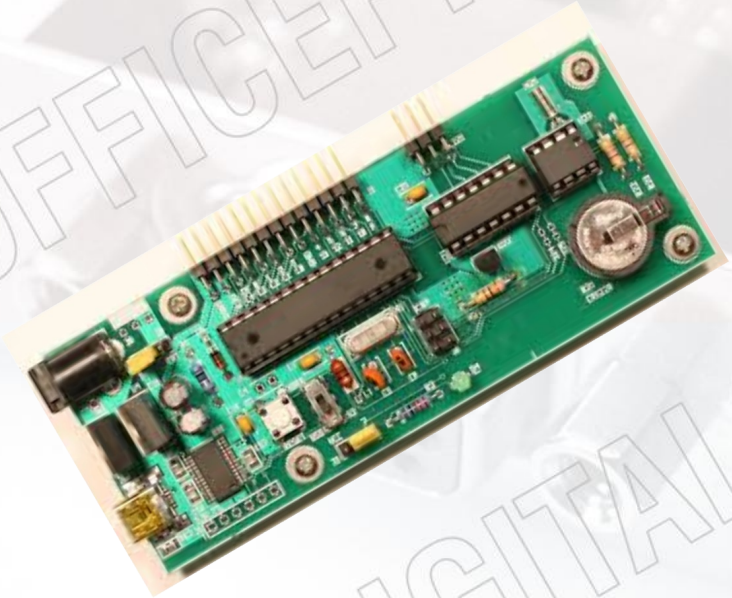
  To make PORTA as I/P with pull-up enable
- DDRA = 0b00000000
- PORTA = 0b11111111

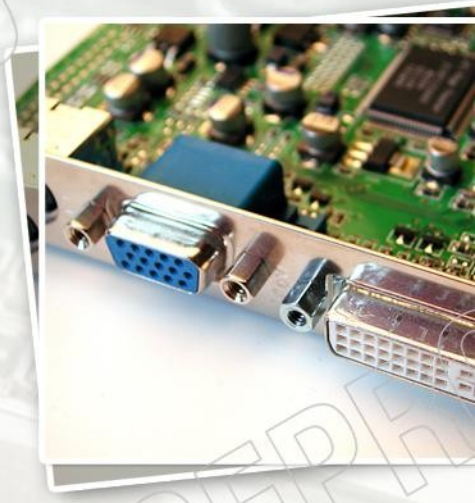


ATMEGA32 I/O Ports

- Port A (PA7..PA0)– Serves as the analog inputs to the
- A/D Converter– Pins are tri-stated when a reset condition becomes active
- Port B(PB7..PB0)– Pins are tri-stated when a reset condition becomes active– Serves the functions of various special Features
- Port C (PC7..PC0)– Pins are tri-stated when a reset condition becomes active– Port C also serves the functions of the JTAG interface and other special features
- Port D (PD7..PD0)– Pins are tri-stated when a reset condition becomes active
  – Serves the functions of various special features
- Ports are 8-bit bi-directional I/O port with internal pull-up resi







System Development

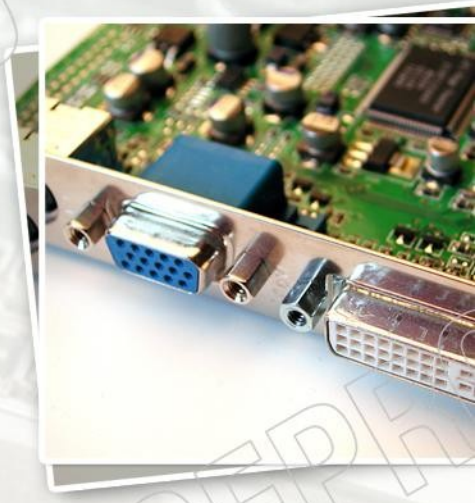
Development Software and Tools

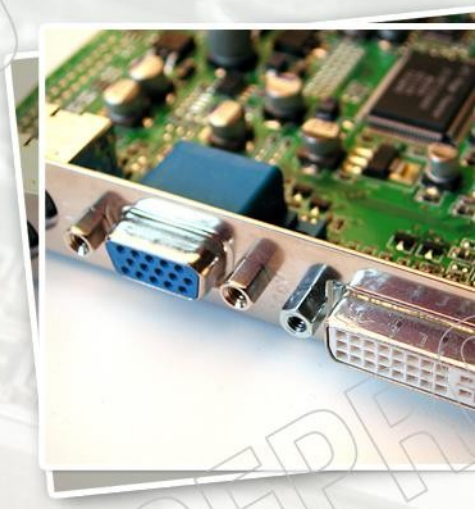
Program Development:

- WinAVR (IDE, Compiler)
- AVR Studio (IDE, Assembler)

Circuit Design and Simulation Software:

- Proteus ISIS
- Orcad Capture

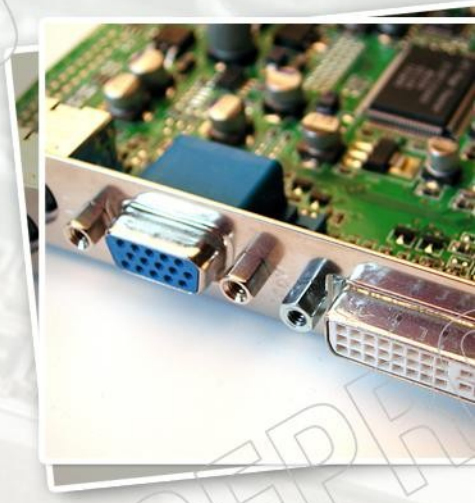




Embedded C Programming

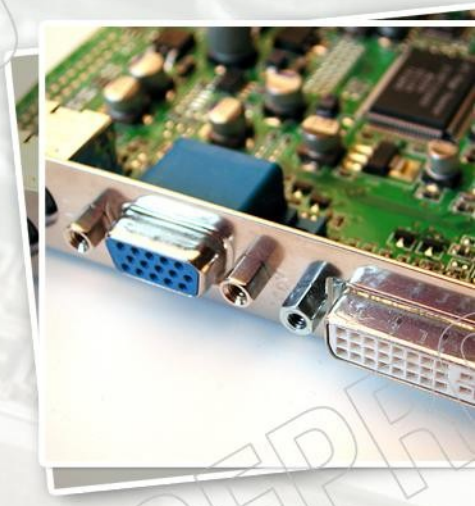
C VS EMBEDDED C

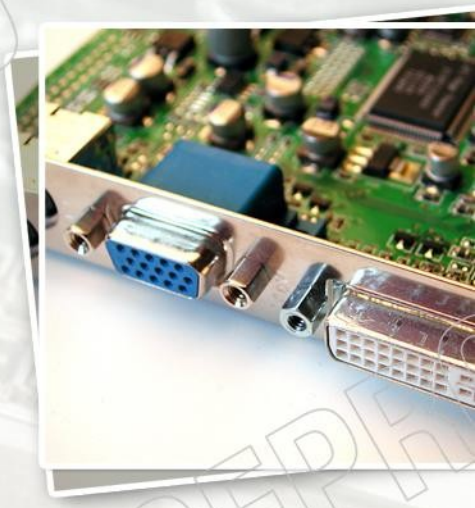
- C is for desktop computers, but embedded C usually is for microcontroller based applications.
- Embedded C is designed to be minimalistic, and does not have standard libraries of
- Embedded C does not produce Executable (exe) but produces memory content for microcontroller.



Writing C Program

- Write text of program (source code) using an editor.
- Run the compiler to check for errors and warningin the edited
- Errors must be removed from the program to make it run successfully
- Use the convenience of creating hex using the compiler from the executable file
- Burn the code on the microcontroller and check the output.

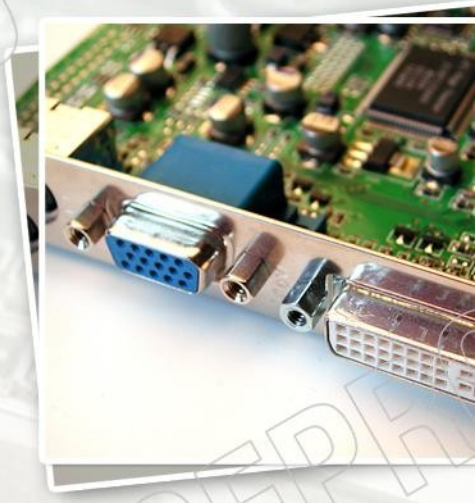




Designing Proposed System- Digital Stop Watch

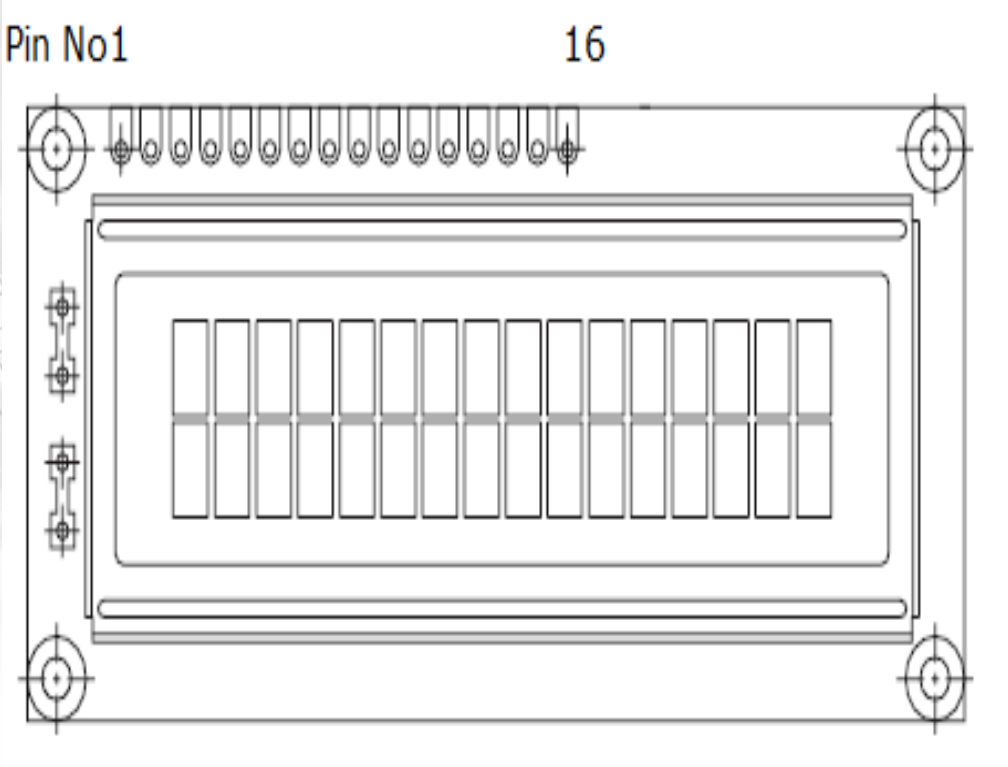
The Problem

Using Atmega32 microcontroller, a digital stop watch is to be continued to count time after pressing a start button and display time on LCD screen. It will also be paused or stopped by pressing push button. The system is to be designed and simulated.



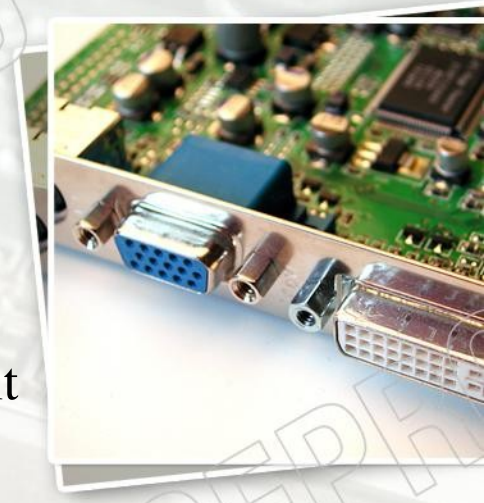
LCD Operation

- 16x2 LCD display is very basic module and is very commonly used in various devices and circuits
- 16 characters per line and there are 2 such lines
- 5 x 8 dots with cursor
- Has Command and Data registers

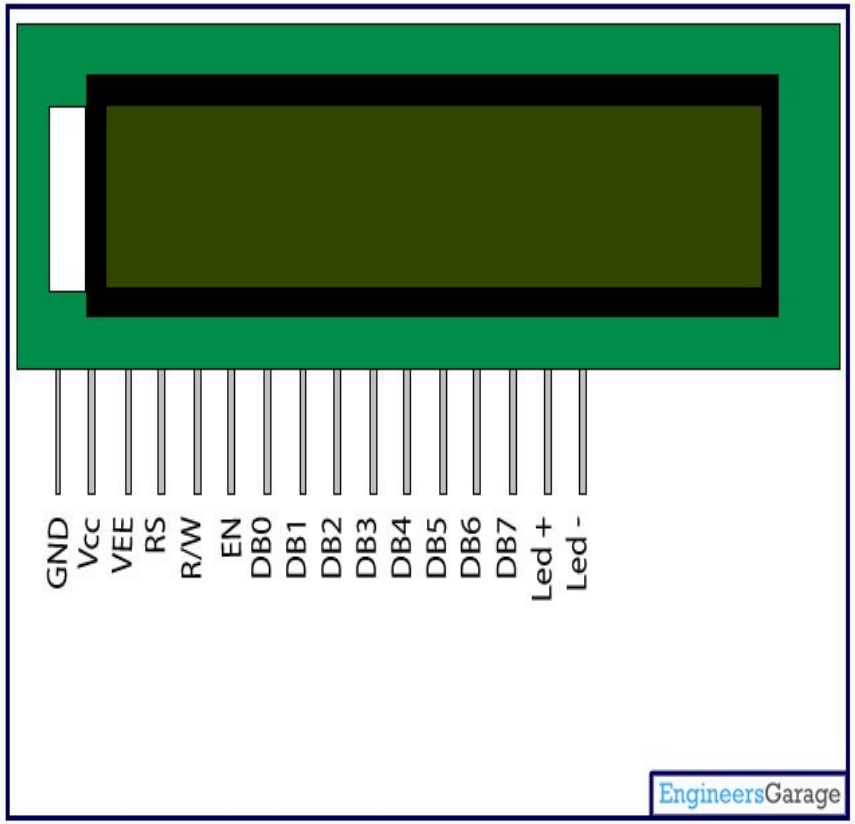


LCD Operation(cont'd)

- Command register stores the command instructions given to the LCD e.g clearing its screen, setting the cursor position etc.
- Data(ASCII value of the character to be displayed) register stores the data to be displayed on the LCD.
- Can be interfaced with the micrcontroller in two modes, 8 bit and 4 bit.
- In 8 bit mode, all of the datalines DB0 to DB7 are connected from the micrcontroller to a LCD module.
- In 4 bit mode, only data lines D4 to D7 are used.



LCD Interfacing Pin Description



PIN No	Name	Function
1	VSS	Ground voltage
2	VCC	+5V
3	VEE	Contrast voltage
4	RS	Register Select 0 = Instruction Register 1 = Data Register
5	R/W	Read/ Write, to choose write or read mode 0 = write mode 1 = read mode
6	E	Enable 0 = start to latch data to LCD character 1= disable
7	DB0	Data bit 0 (LSB)
8	DB1	Data bit 1
9	DB2	Data bit 2
10	DB3	Data bit 3
11	DB4	Data bit 4
12	DB5	Data bit 5
13	DB6	Data bit 6
14	DB7	Data bit 7 (MSB)
15	BPL	Back Plane Light +5V or lower (Optional)
16	GND	Ground voltage (Optional)

Steps to be taken

- Writing and Compiling Program
- Drawing Circuit Using Simulation Software
- Loading Executible into Simulator
- Execution