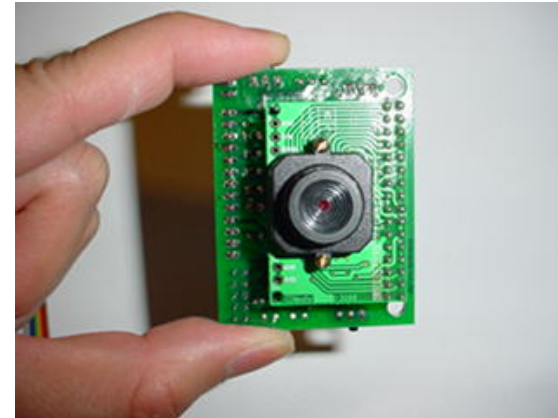


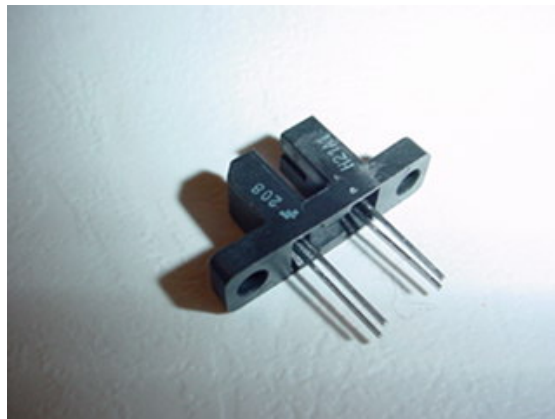
Some sensors



Ultrasonic



دوربین



زوج نوری

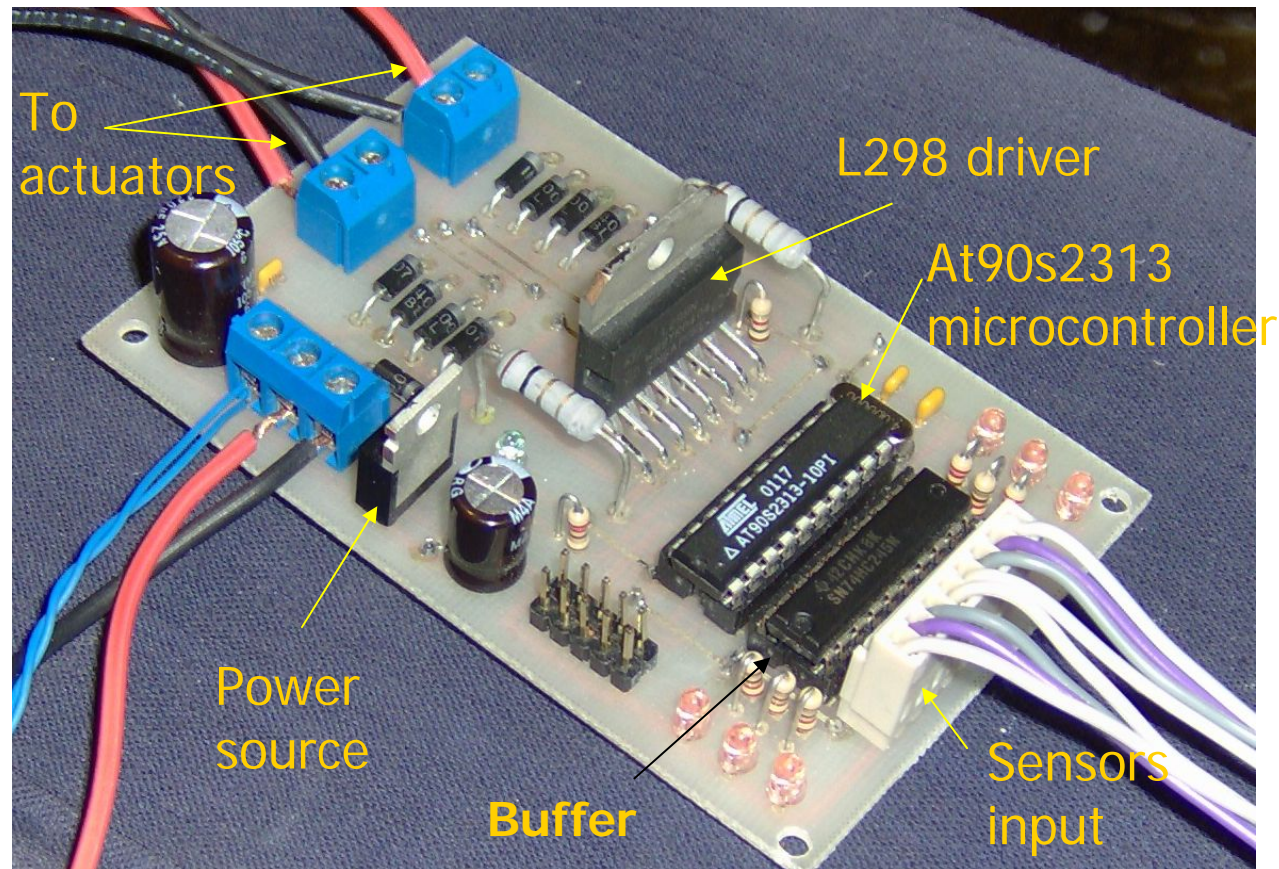


Micro switch

Stepper Motors



Sample Controller Circuit



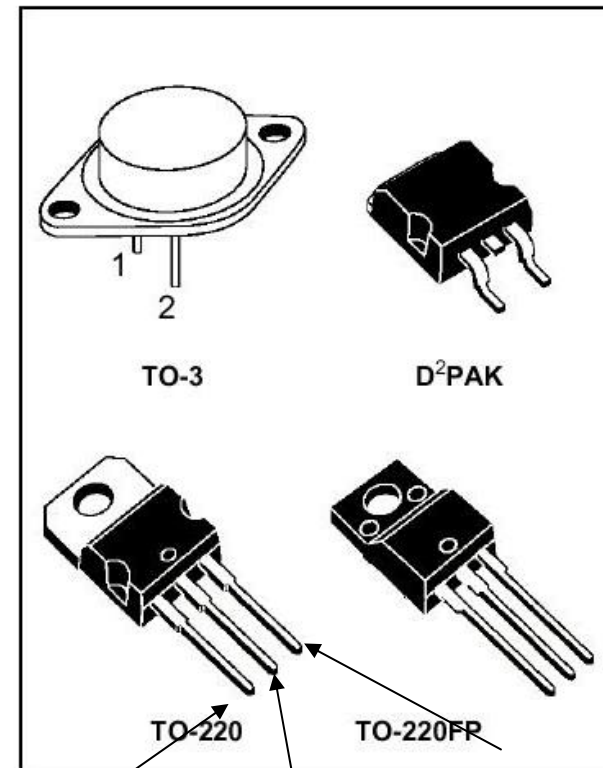
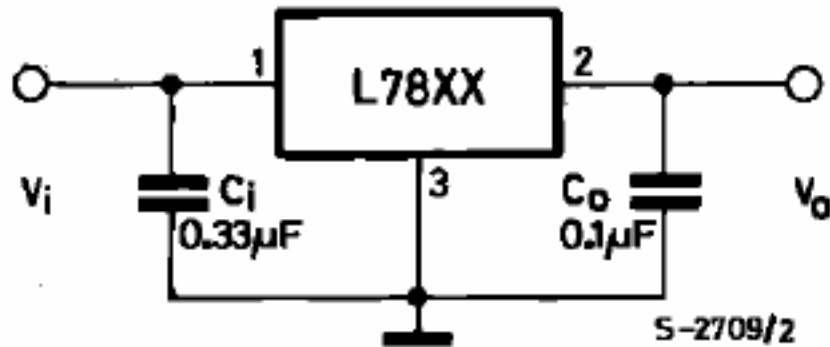


What is GND and VCC?

- **GND** :The Base for indicating voltage
- Usually the negative pole of DC power source or battery .
- **VCC , VDD , VS** : Positive pole of DC Power source or battery .
- Usually we indicate +5v with VCC and other voltages with VS and VDD.

Voltage regulator

- 7805 voltage regulator:
- Input: 7.5 to 12 V
- Output 5V – 800mA



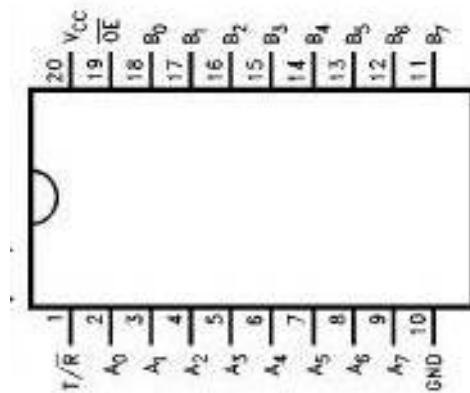
Input

GND

Output

TTL buffer

74HC245 TTL Buffer



Pin Descriptions

Pin Names	Description
\overline{OE}	Output Enable Input (Active LOW)
T/\overline{R}	Transmit/Receive Input
A ₀ -A ₇	Side A Inputs or TRI-STATE Outputs
B ₀ -B ₇	Side B Inputs or TRI-STATE Outputs

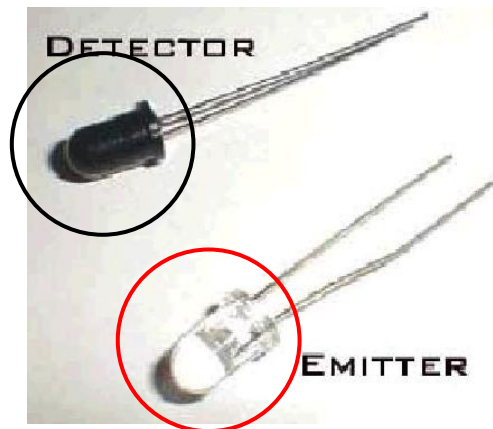
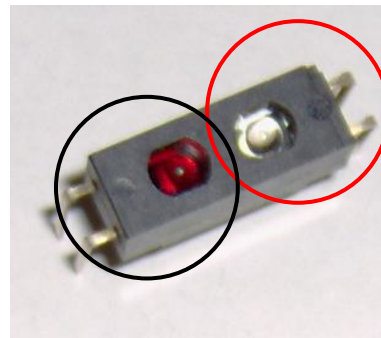
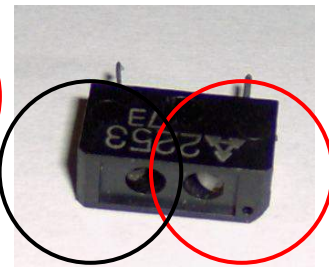
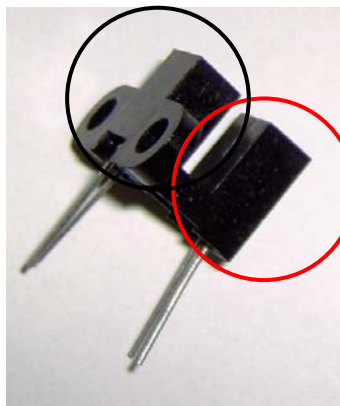
Truth Table

Inputs		Output
\overline{OE}	T/\overline{R}	
L	L	Bus B Data to Bus A
L	H	Bus A Data to Bus B
H	X	High Z State

H - HIGH Voltage Level
 L - LOW Voltage Level
 X - Immaterial

IR sensors

- Detect black and white surfaces
 - Sender (IR LED)
 - Receiver (Photo transistor)



How IR sensors works?

- Sender issue IR signal
- Receiver sense the reflect of the signal

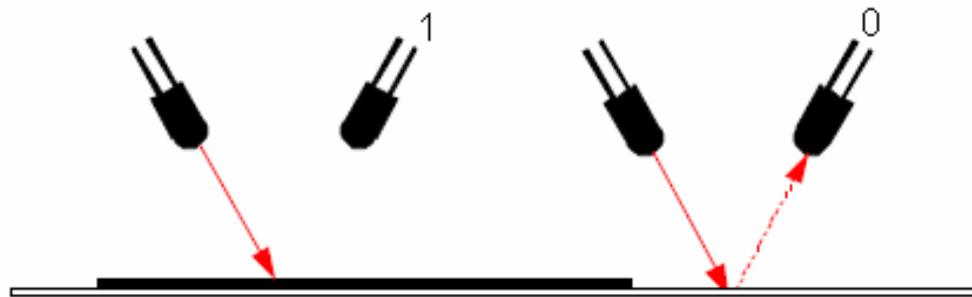
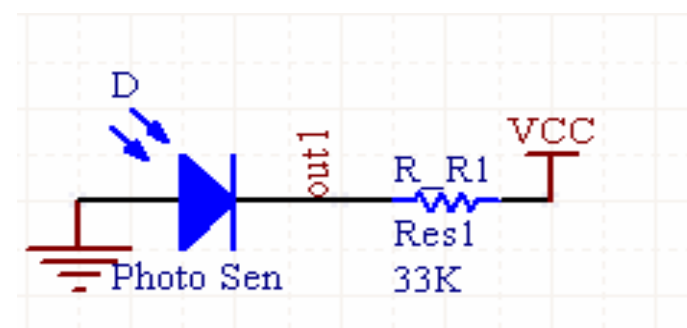
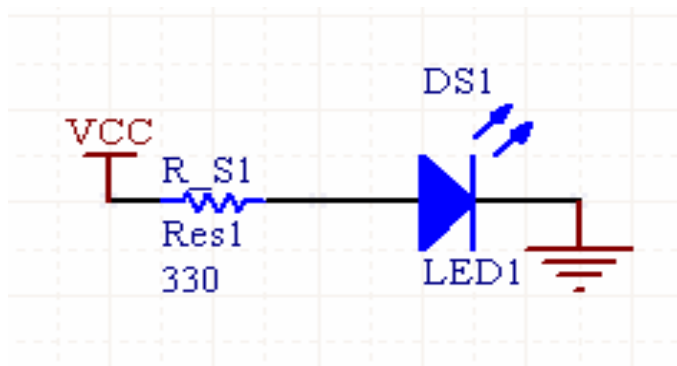
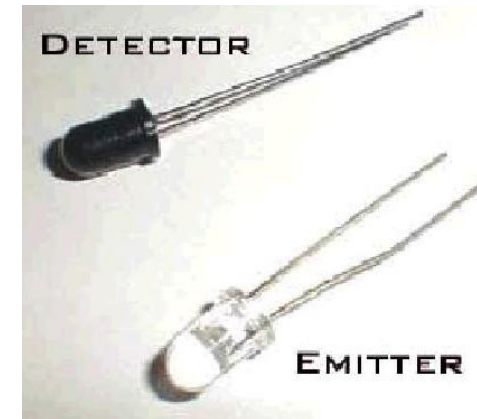


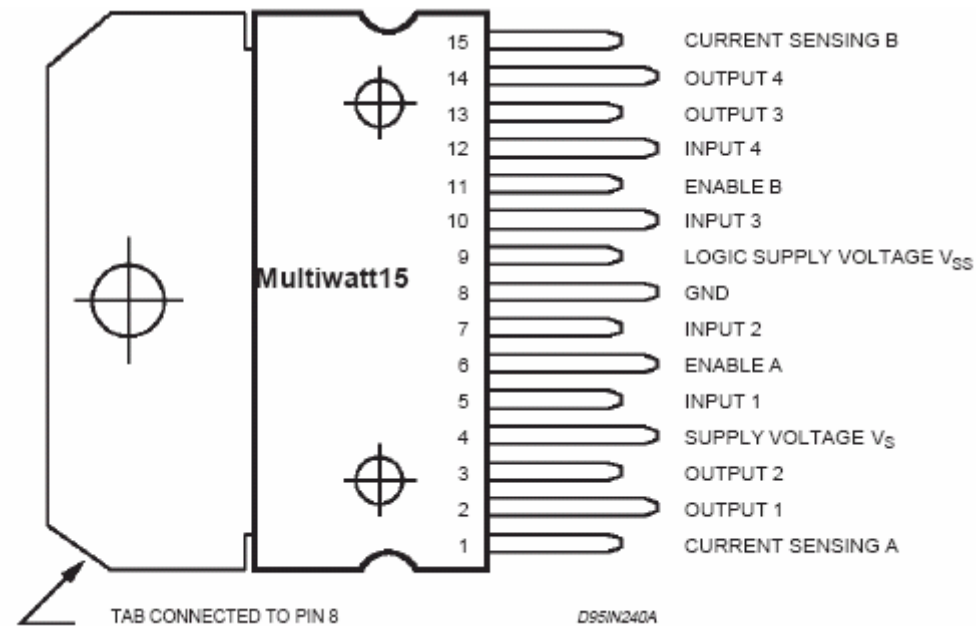
Figure 5 - Sensor operation

More detecting distance

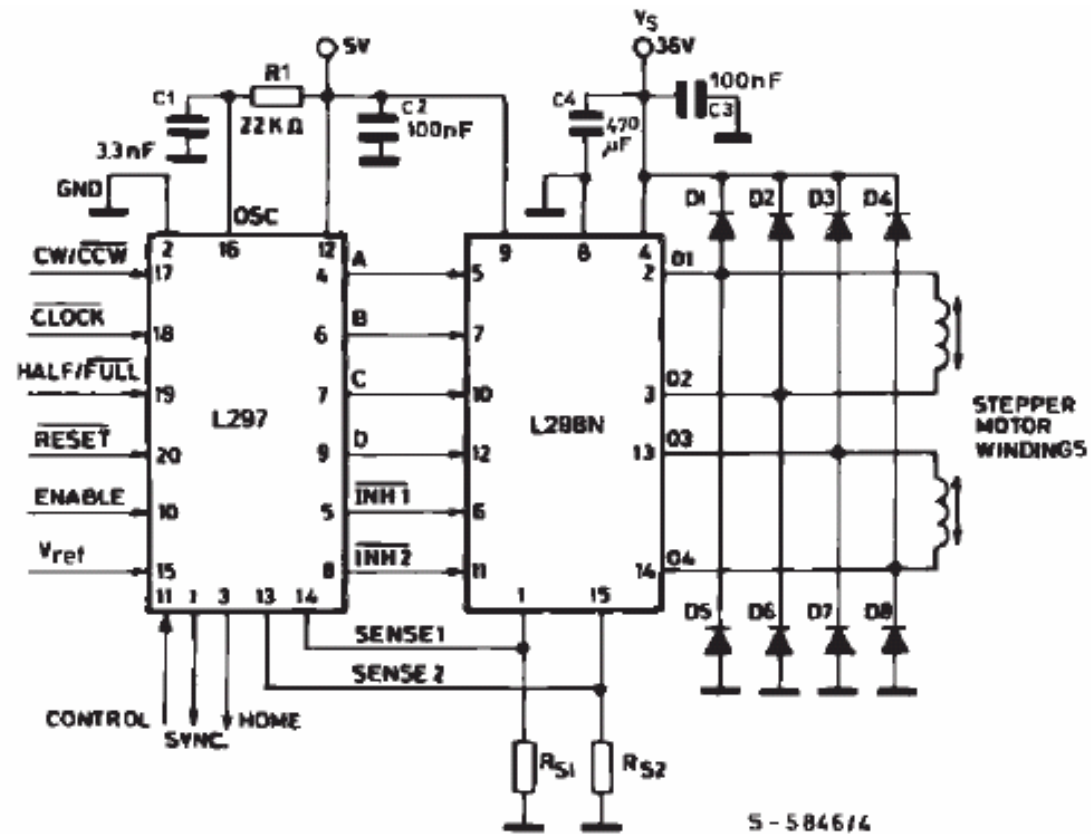
Detecting distance :
about 1 ~ 10 cm



Motor Driver -L298



Motor Driver - L298



$$R_{S1} = R_{S2} = 0.5 \Omega$$

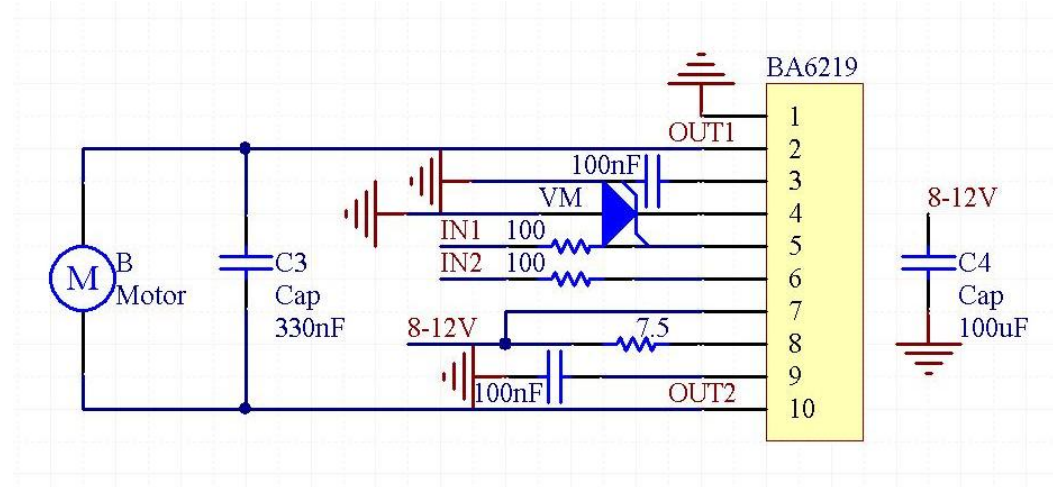
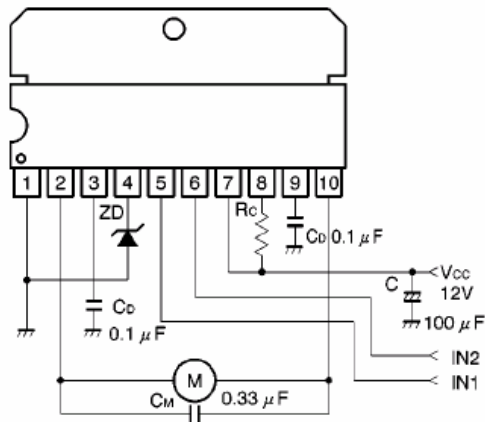
D1 to D8 = 2 A Fast diodes $\left\{ \begin{array}{l} V_F \leq 1.2 \text{ V @ } I = 2 \text{ A} \\ \text{trr} \leq 200 \text{ ns} \end{array} \right.$

Motor drivers- BA6219

- BA6219(output : 2A ,18V)
- Application :1XDC motor ,left ,right ,break

●Application examples

(1) BA6219B





Processor & Microcontroller

- Processor :
 - Device that fetch program codes from memory and execute .
 - A processor circuit contain ports , RAM , ROM, timer , UART ,...
- Microcontroller :
 - Device contain a processor and some minimum peripherals such as few memory , timer , UART, PORT ,...



AVR microcontrollers

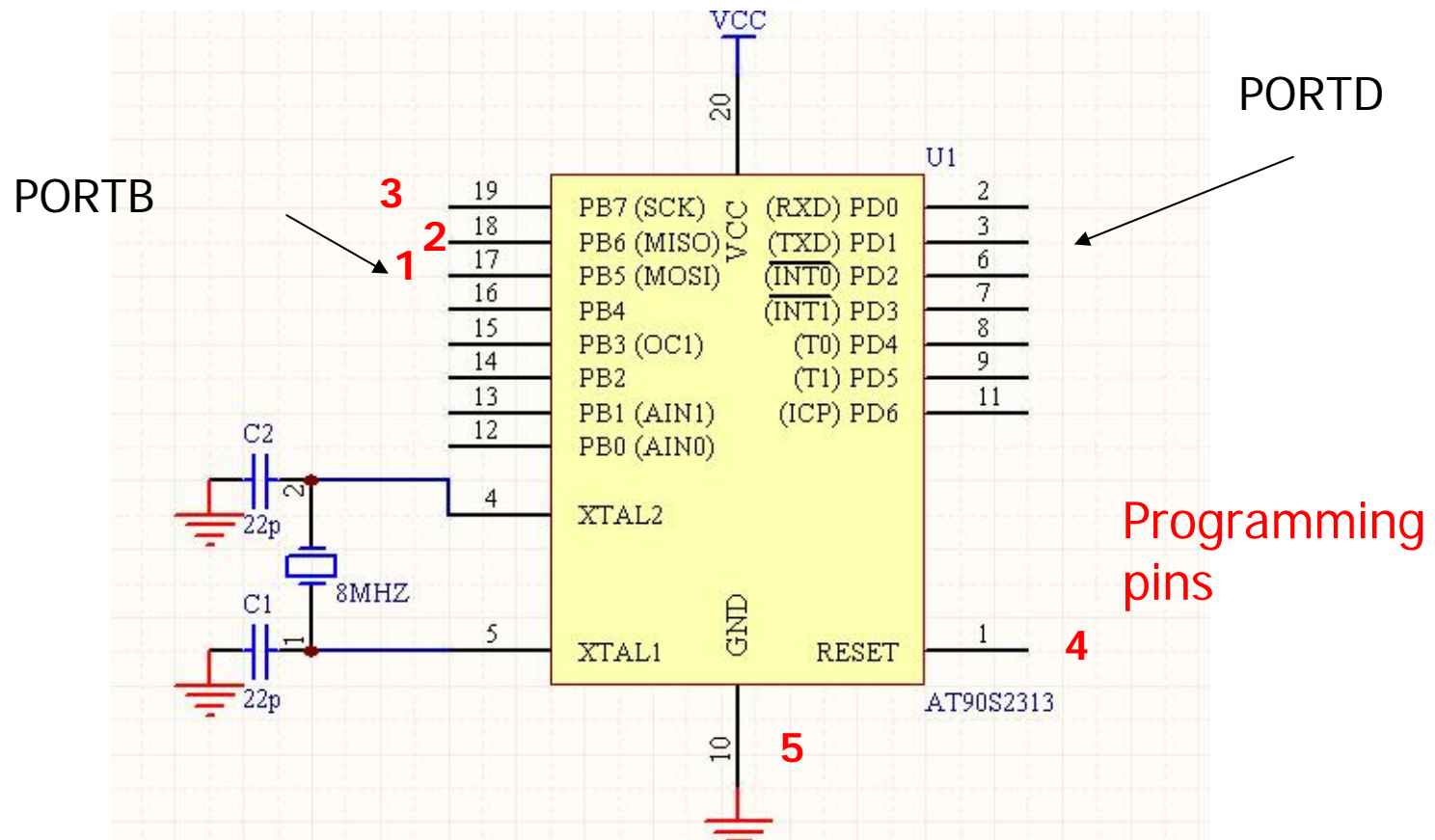
- AT90s2313:
 - 15 IO PINS
 - 1 UART
 - 2 Timer
 - ISP
 - 2KB flash
 - 128 B EEPROM
 - 128 KB SRAM
 - 2 External interrupt
 - 0~12 MHZ operating frequency



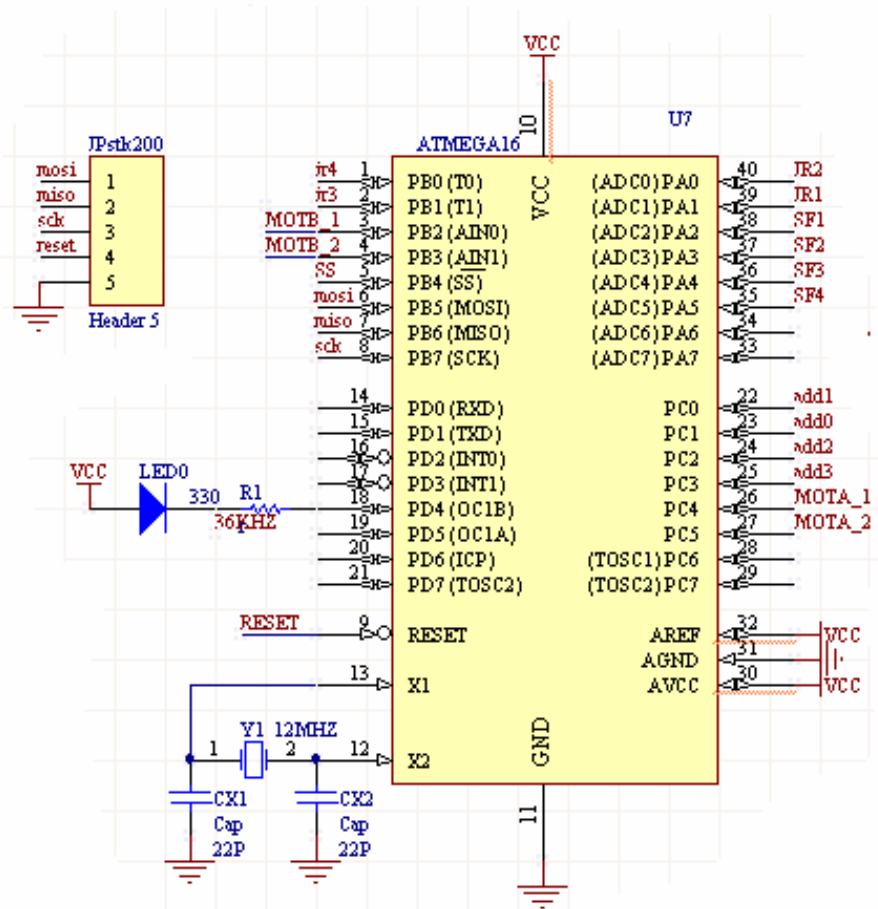
AVR Microcontrollers

- ATMEGA16:
 - 32 IO PINS (4 port)
 - 1 UART
 - 3 Timer
 - ISP
 - 16KB flash
 - 512 B EEPROM
 - 1 KB SRAM
 - 2 External interrupt
 - 0~16 MHZ operating frequency
 - Analog input ports

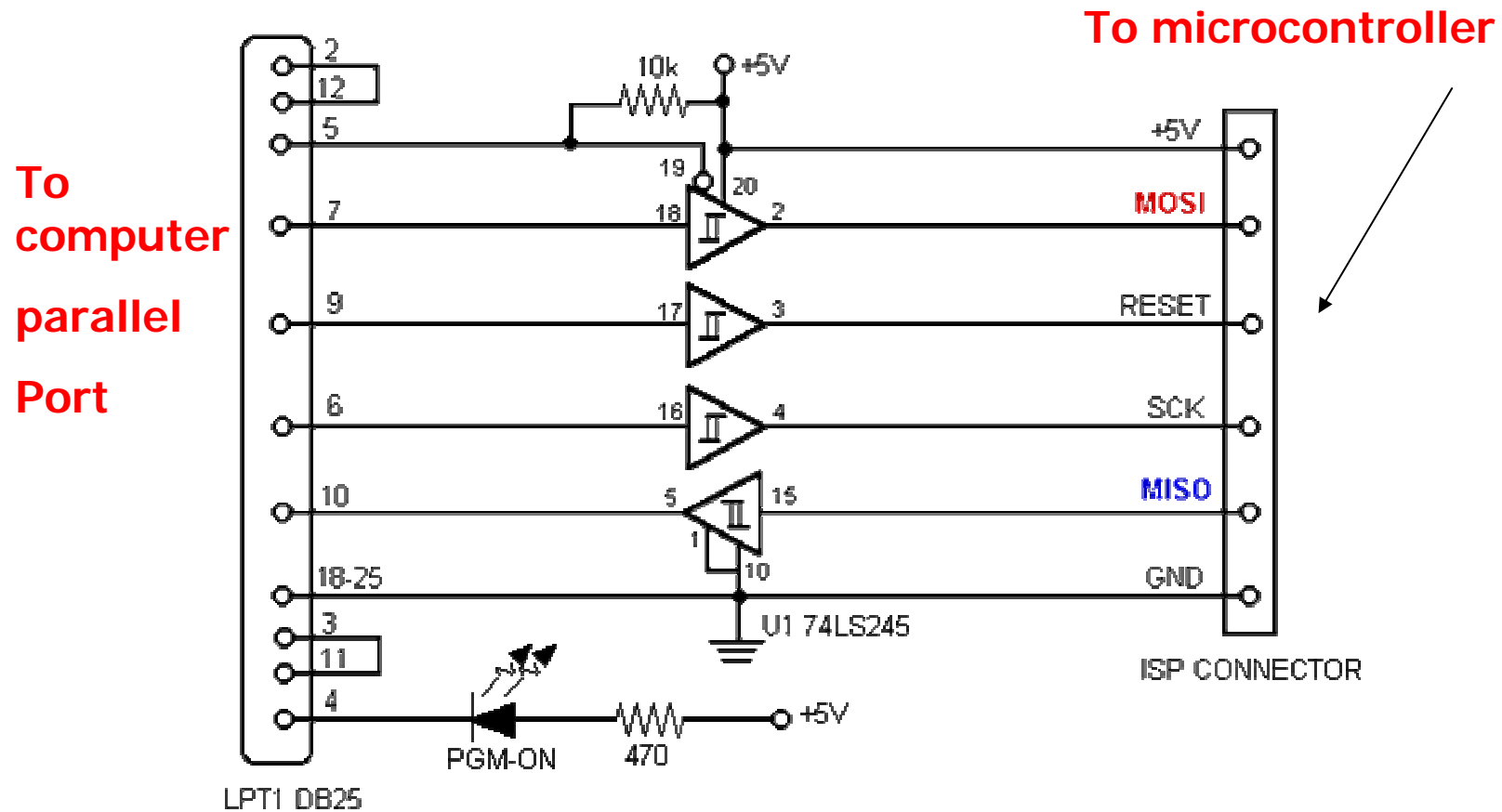
Minimum system for AT90S2313

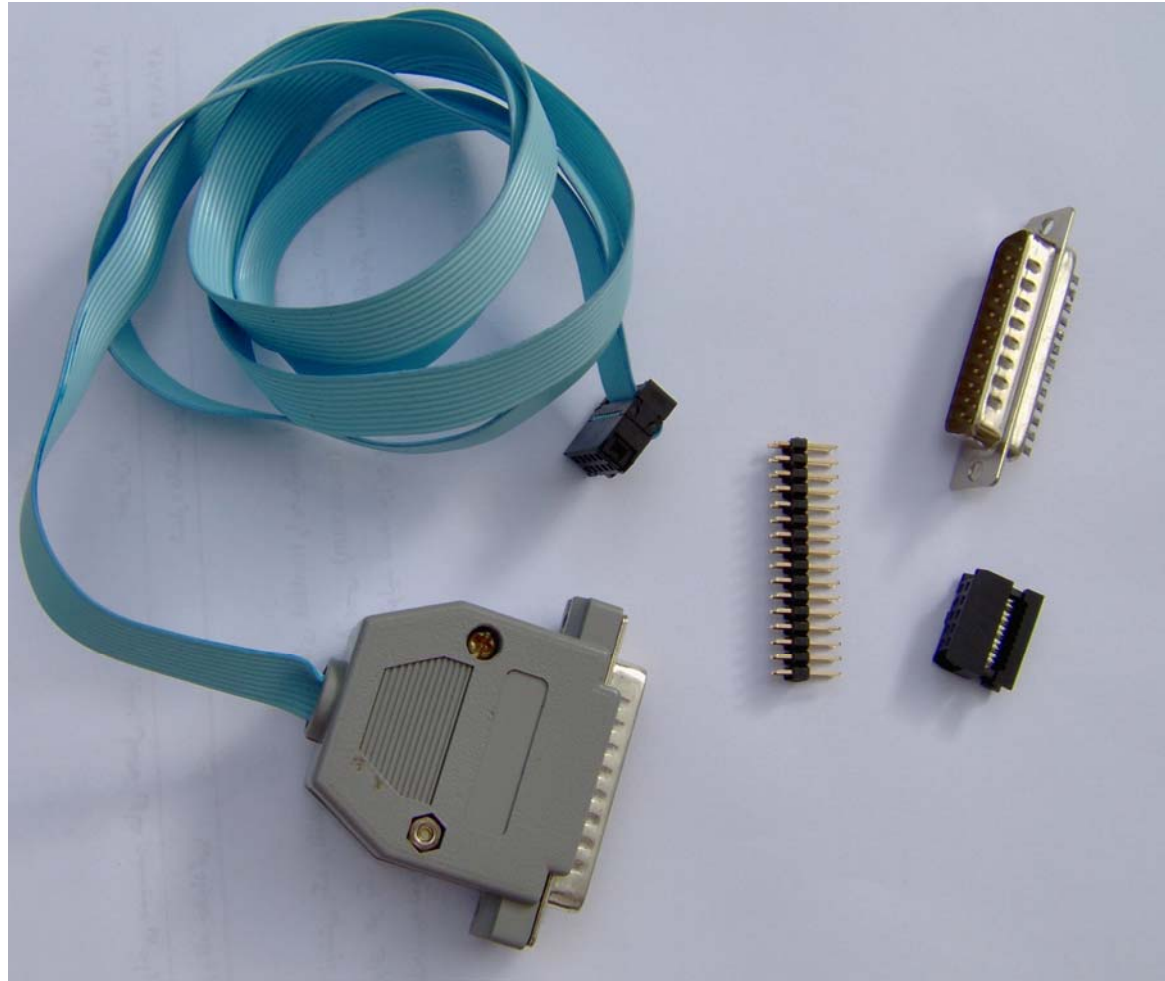
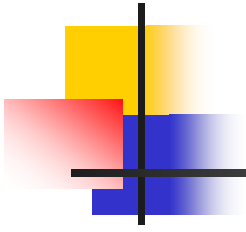


Minimum circuit for ATMEGA16

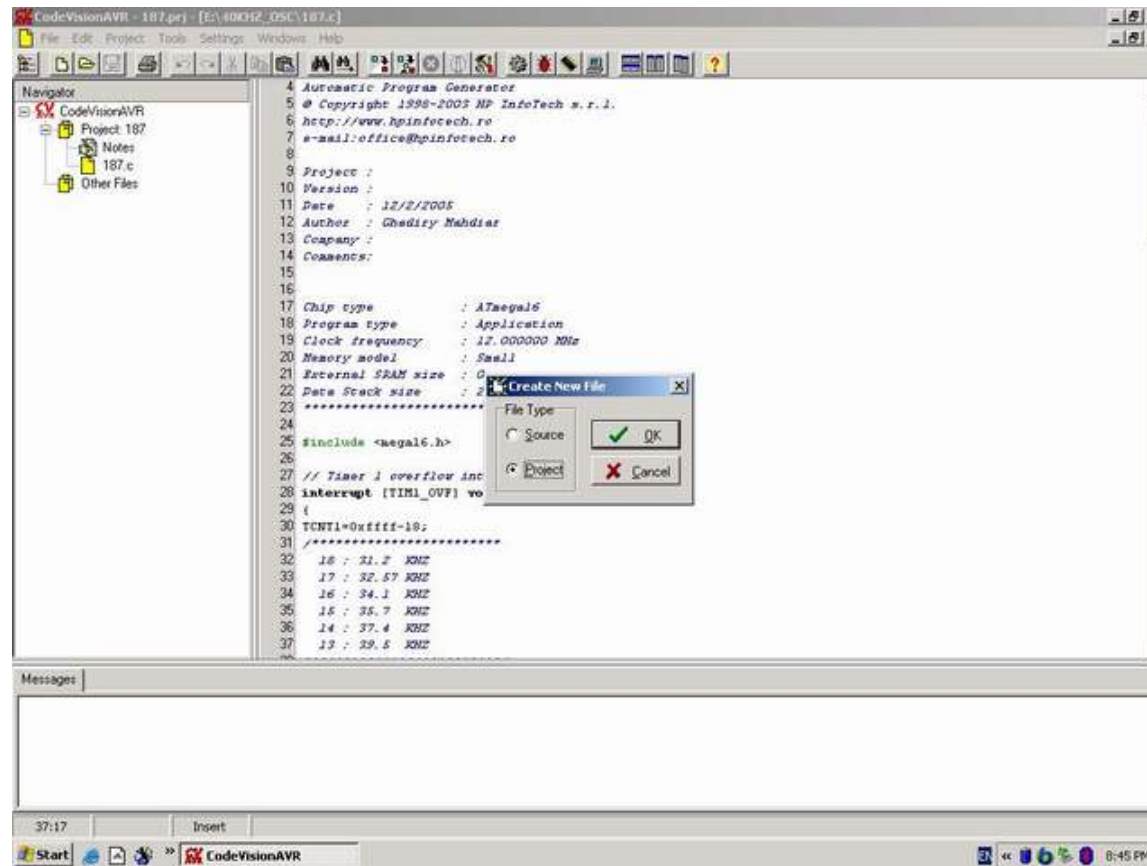


STK200 Programmer cable

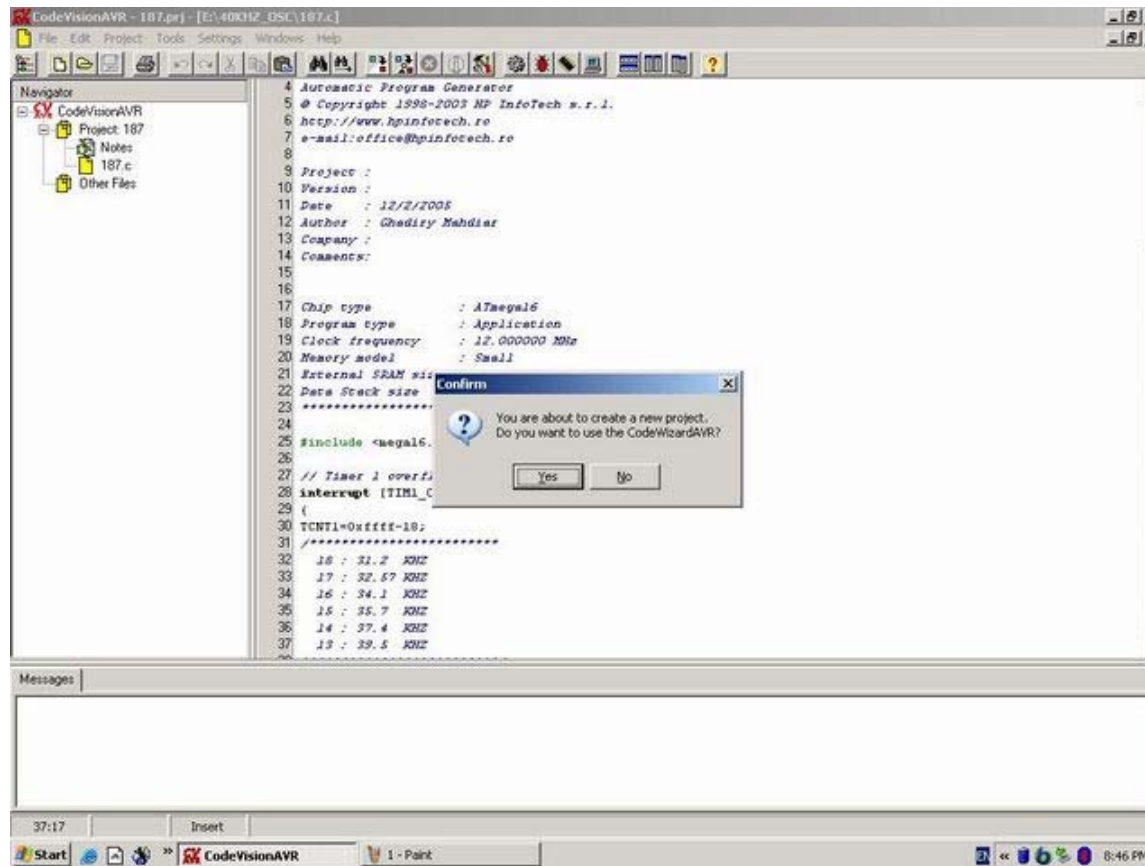




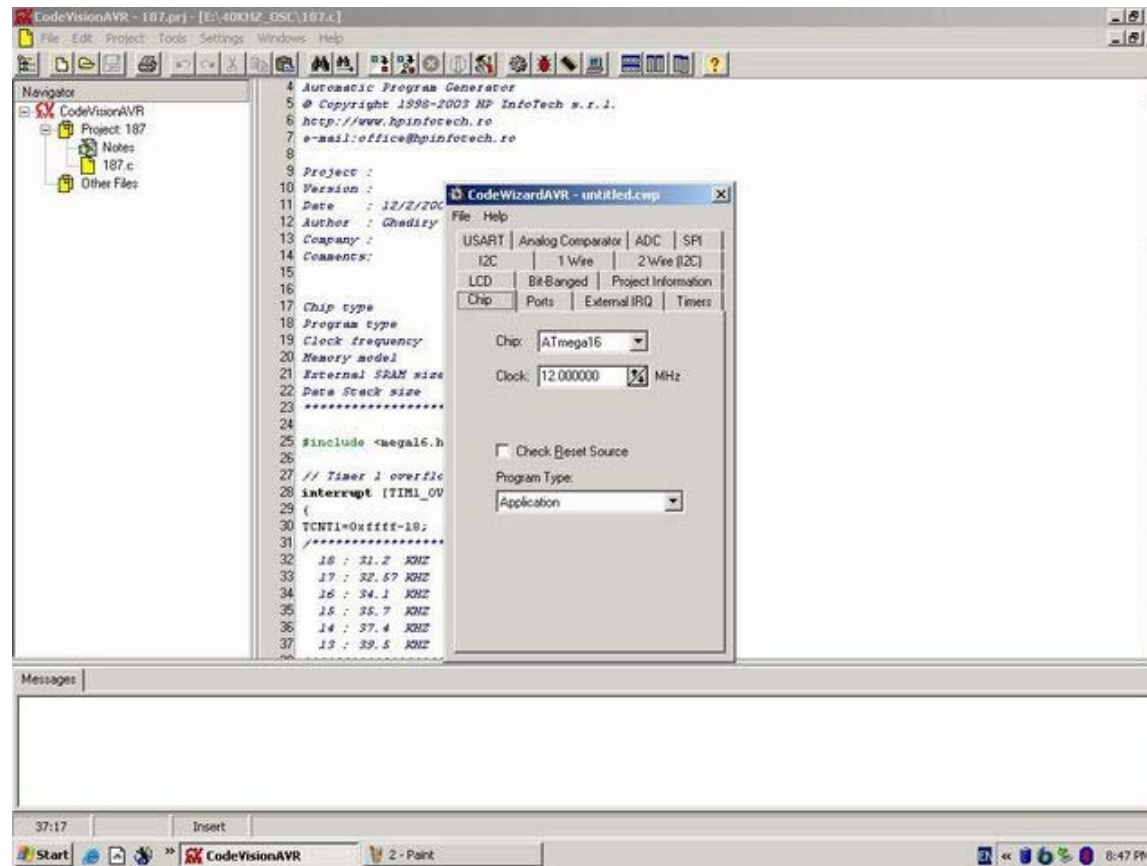
Making a project



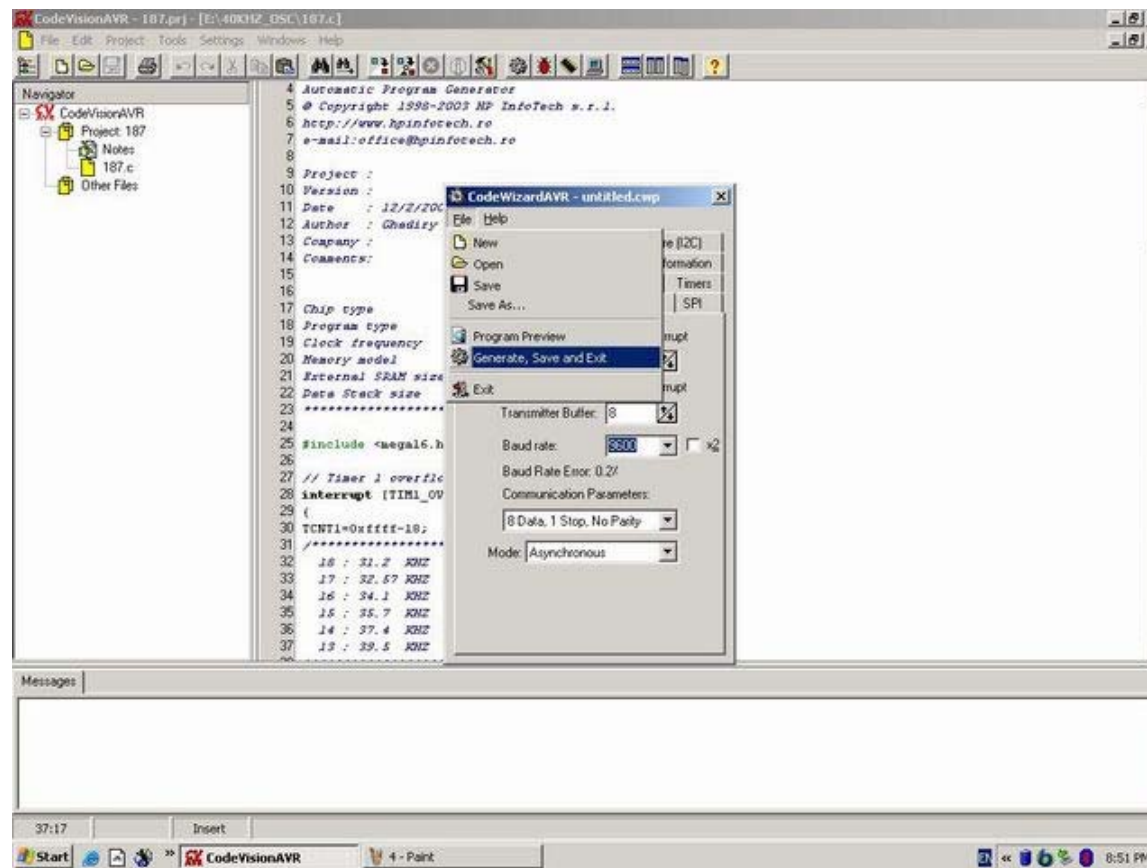
Making a project



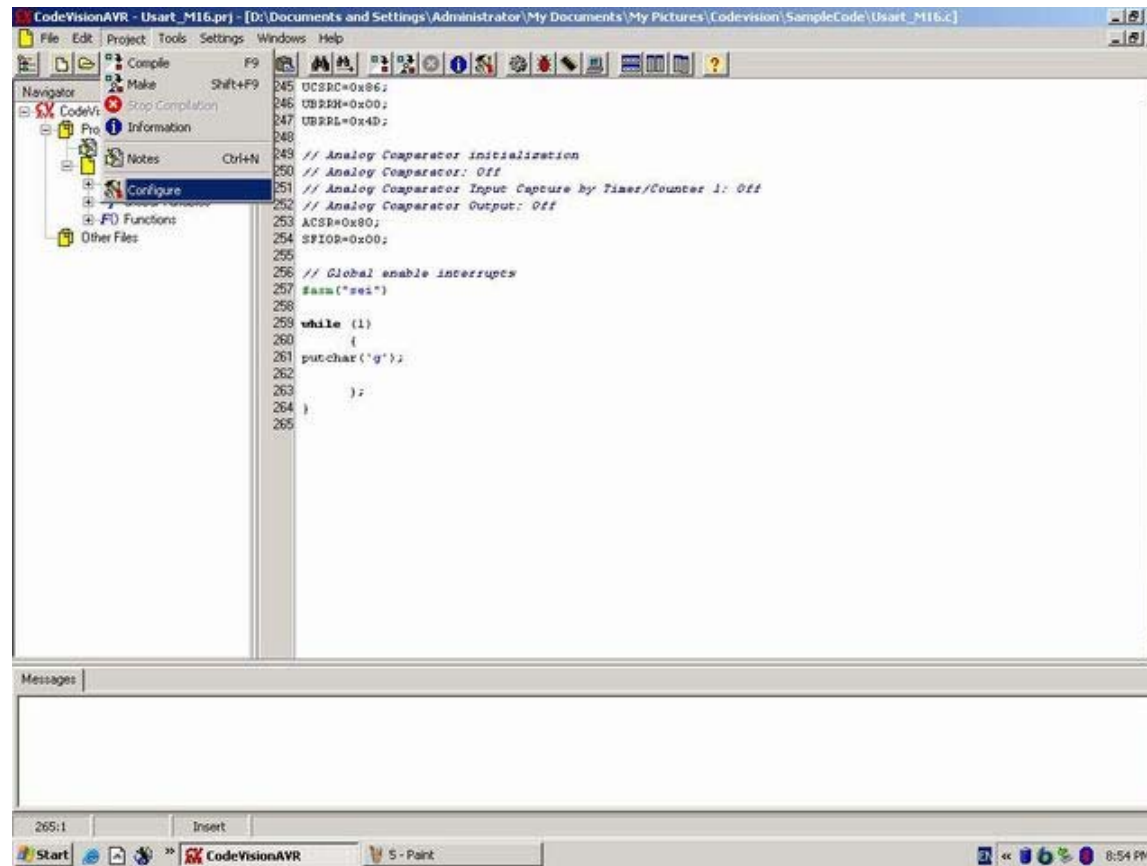
Making a project



Making a project



Configuration



The screenshot shows the CodeVisionAVR IDE interface. The main window displays C code for AVR microcontroller configuration. The code includes register definitions and initialization for the Analog Comparator. The 'Configure' option is selected in the left-hand menu.

```
CodeVisionAVR - Usart_M16.prj - [D:\Documents and Settings\Administrator\My Documents\My Pictures\Codevision_SampleCode\Usart_M16.c]
File Edit Project Tools Settings Windows Help
Compile F9
Make Shift+F9
Stop Compilation
Information Ctrl+I
Notes Ctrl+N
Configure
FD Functions
Other Files

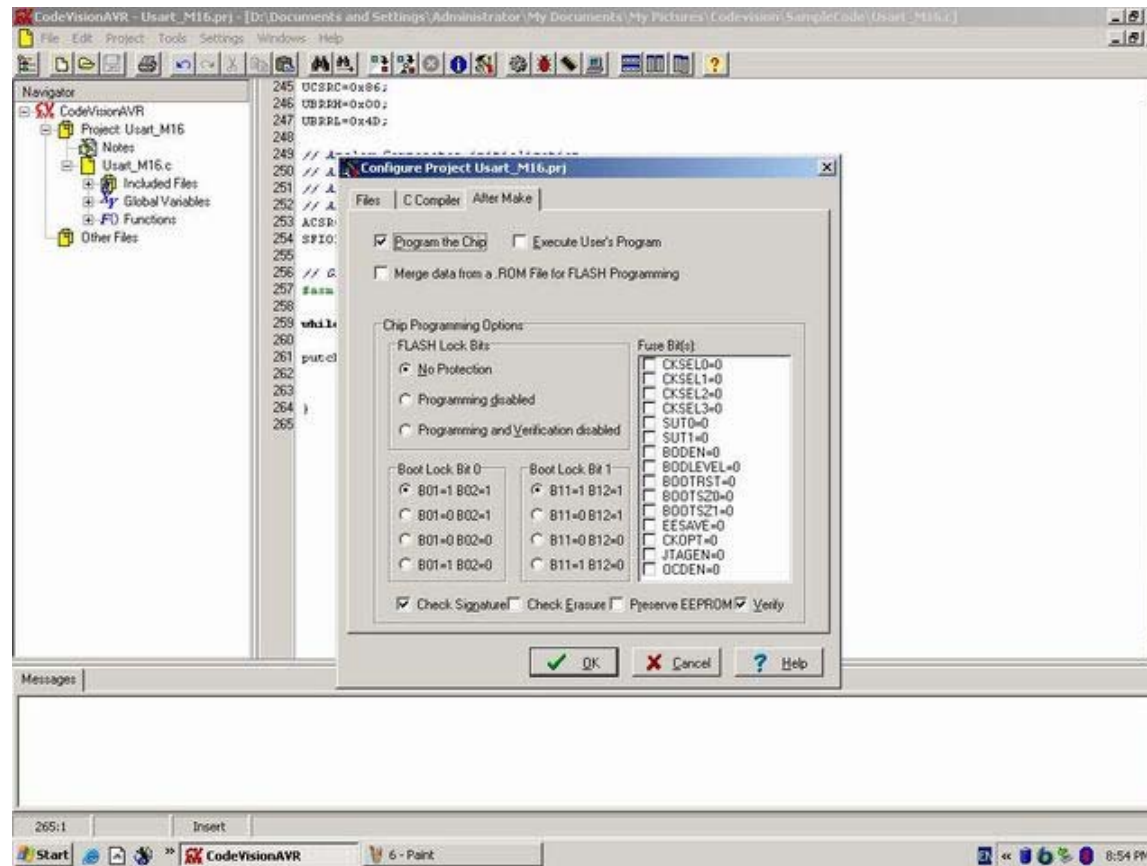
245 UCSPC=0x06;
246 UBRRH=0x00;
247 UBRRL=0x4D;
248
249 // Analog Comparater Initialization
250 // Analog Comparator: Off
251 // Analog Comparator Input Capture by Timer/Counter 1: Off
252 // Analog Comparator Output: Off
253 ACSP=0x80;
254 SPIOR=0x00;
255
256 // Global enable interrupts
257 #asm("sei")
258
259 while (1)
260 {
261 putchar('g');
262
263 };
264
265
```

Messages

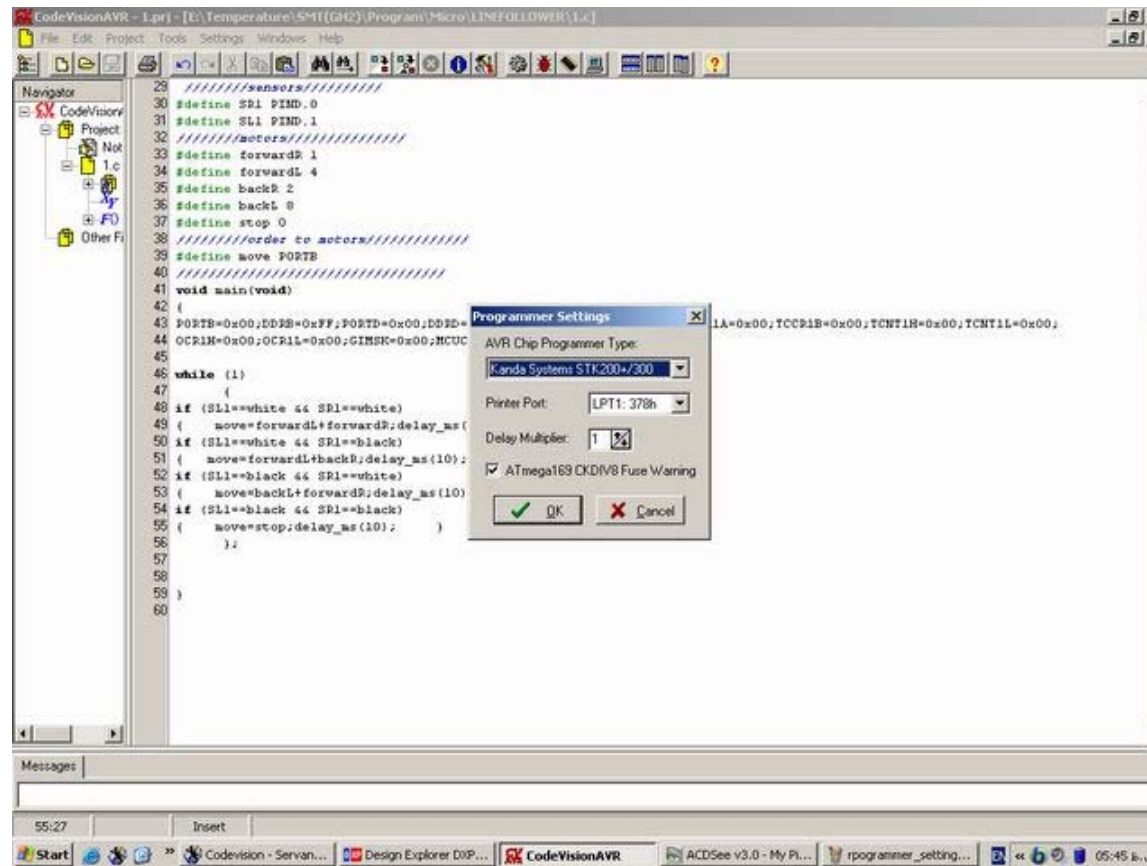
265:1 Insert

Start CodeVisionAVR S - Paint 8:54 PM

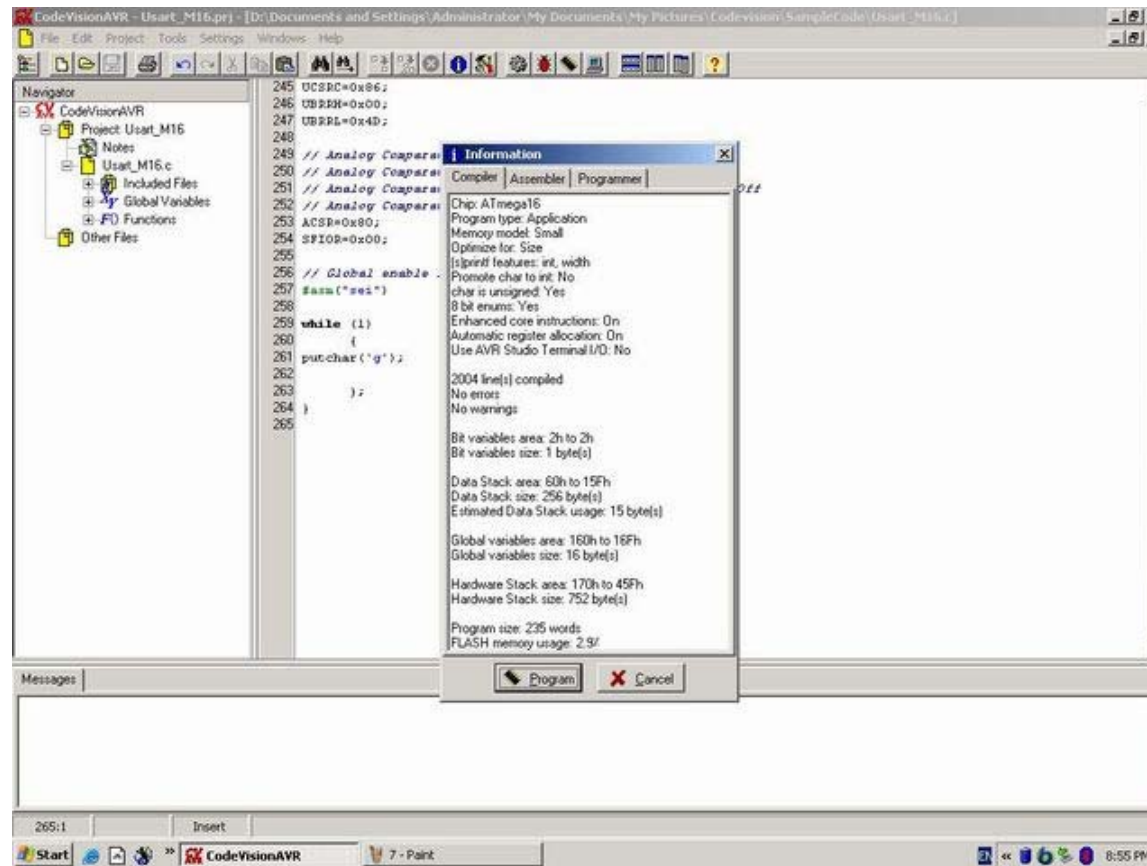
Configuration



Configuration



Compile program and program the chip





Programming with C

- Each program in codevision with wizard have tree part
 - A : Headers :
 - Some basic definition and function you need in your program
 - B: initialization part:
 - Some basic settings for futures of device usually compiler generate them .
 - C: main body :
 - Codes you generate .

Programming with codevision

- `#include <mega16.h>` ← Header for ATMEGA16 device
 - `// Declare your global variables here`
 - `void main(void)`
 - `{`
 - `// Declare your local variables here` → Define variables
 - `// Input/Output Ports initialization`
 - `// Port A initialization`
 - `// Func0=In Func1=In Func2=In Func3=In Func4=In Func5=In Func6=In Func7=In`
 - `// State0=T State1=T State2=T State3=T State4=T State5=T State6=T State7=T`
 - `PORTA=0x00;`
 - `DDRA=0x00;`
 - `// Port B initialization`
 - `// Func0=In Func1=In Func2=In Func3=In Func4=In Func5=In Func6=In Func7=In`
 - `// State0=T State1=T State2=T State3=T State4=T State5=T State6=T State7=T`
 - `PORTB=0x00;`
 - `DDRB=0x00;`
 - `// Port C initialization`
 - `// Func0=In Func1=In Func2=In Func3=In Func4=In Func5=In Func6=In Func7=In`
 - `// State0=T State1=T State2=T State3=T State4=T State5=T State6=T State7=T`
 - `PORTC=0x00;`
 - `DDRC=0x00;`
 - `// Port D initialization`
 - `// Func0=In Func1=In Func2=In Func3=In Func4=In Func5=In Func6=In Func7=In`
 - `// State0=T State1=T State2=T State3=T State4=T State5=T State6=T State7=T`
 - `PORTD=0x00;`
 - `DDRD=0x00;`
- initializations



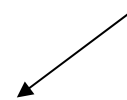
Programming with codevision

- // Timer/Counter 0 initialization
- // Clock source: System Clock
- // Clock value: Timer 0 Stopped
- // Mode: Normal top=FFh
- // OCO output: Disconnected
- TCCR0=0x00;
- TCNT0=0x00;
- OCR0=0x00;

- // Timer/Counter 1 initialization
- // Clock source: System Clock
- // Clock value: Timer 1 Stopped
- // Mode: Normal top=FFFFh
- // OC1A output: Discon.
- // OC1B output: Discon.
- // Noise Canceler: Off
- // Input Capture on Falling Edge
- TCCR1A=0x00;
- TCCR1B=0x00;
- TCNT1H=0x00;
- TCNT1L=0x00;
- OCR1AH=0x00;
- OCR1AL=0x00;
- OCR1BH=0x00;
- OCR1BL=0x00;

- // Timer/Counter 2 initialization
- // Clock source: System Clock
- // Clock value: Timer 2 Stopped
- // Mode: Normal top=FFh
- // OC2 output: Disconnected
- ASSR=0x00;
- TCCR2=0x00;
- TCNT2=0x00;
- OCR2=0x00;

initializations





Programming with codevision

- // External Interrupt(s) initialization
- // INT0: Off
- // INT1: Off
- // INT2: Off
- GICR|=0x00;
- MCUCR=0x00;
- MCUCSR=0x00;

- // Timer(s)/Counter(s) Interrupt(s) initialization
- TIMSK=0x00;

- // Analog Comparator initialization
- // Analog Comparator: Off
- // Analog Comparator Input Capture by Timer/Counter 1: Off
- // Analog Comparator Output: Off
- ACSR=0x80;
- SFIOR=0x00;

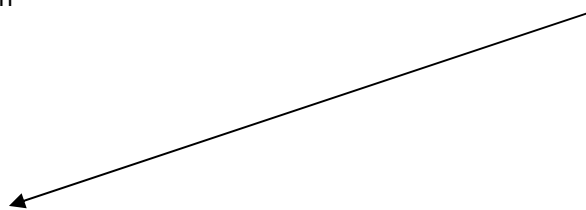
- **while (1)**
- {
- // Place your code here

- };

initializations

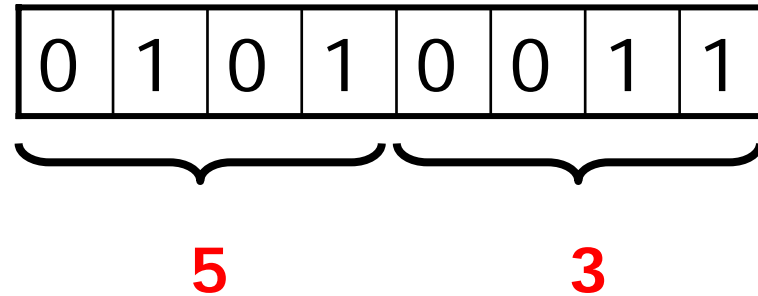


Your code is hear

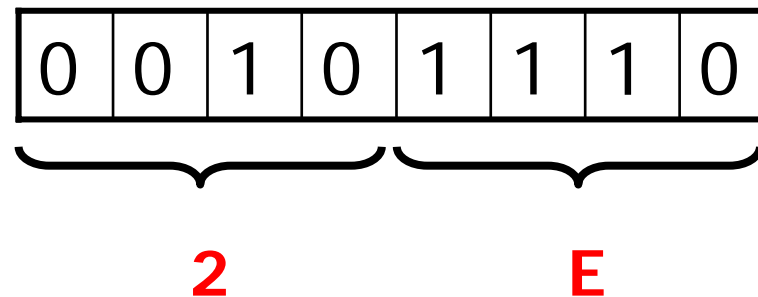


Codevision instructions

- **PORT:**
 - Output a value to parallel ports .
 - EX. `PORT=0x53;`



- EX. `PORT=0x2E;`





Codevision instructions

- `PORTA.1=1; // set bit 1 , hi .`
- `PORTA.3=0 ; // set bit 3 , low`
- `DDR`
 - `DDRA=0xff; // all 8 bit of port a set as output`
 - `DDRB=0x3f; //6 bit of 8 bit set as output and 2 of them set as input`
- `PIN`
 - `Int A ;`
 - `A=PINB;`
- `A=PINA.0; // access 1 bit from a port`



Codevision instructions

- Define a variable :
 - Int a;
 - Char ch;
 - Char * str;
 - Float f;



Codevision instructions

■ If – else

If (condition) then

{

Expression 1

Expression 2

...

}

Else

{

Expression 1

Expression 2

...

}



Codevision instructions

- For statement

```
For(i=0;i<100;i++)
```

```
{
```

```
    expression
```

```
}
```

```
For(i=100;i>0;i--)
```

```
{
```

```
    expression
```

```
}
```