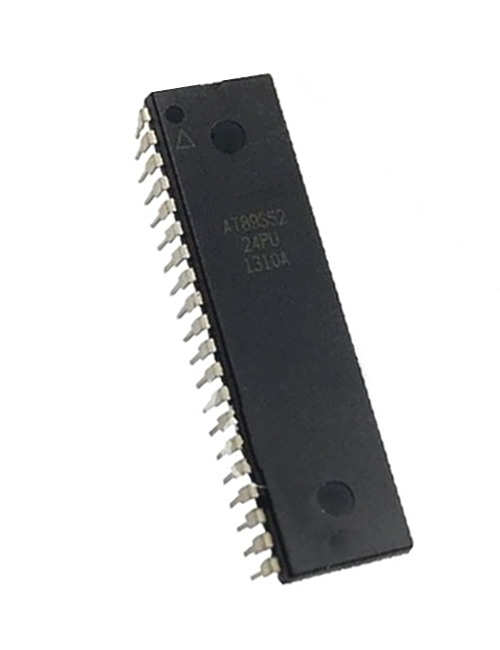
**AT89S52 microcontroller**

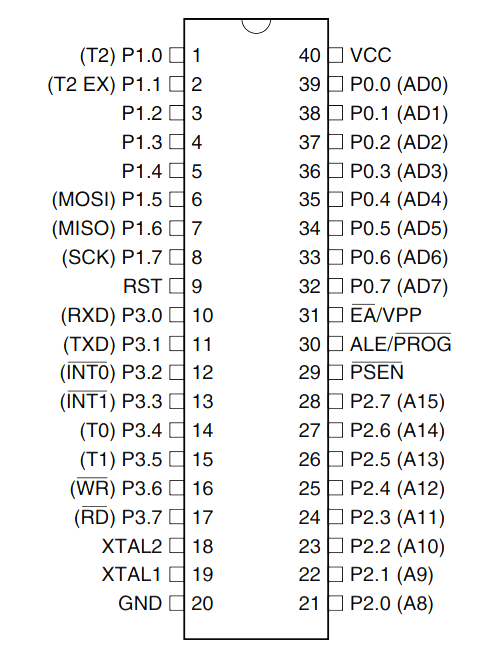
The **AT89S52**comes from the popular 8051 family of Atmel Microcontrollers. It is an 8-bit CMOS microcontroller with 8K as Flash memory and 256 bytes of RAM. Since it is similar to the trust worthy 8051 architecture these microcontrollers are as per industry standard. It has 32 I/O pins comprising of three 16-bit timers, external interrupts, full-duplex serial port, on-chip oscillator and clock circuitry.

The Microcontroller also has Operating mode, Idle Mode and Power down mode which makes it suitable for battery operated applications.



**AT89S52 Pin Configuration**

|  |  |  |
| --- | --- | --- |
| **Pin Number** | **Pin Name** | **Description** |
| 1 | P1.0 (T2) | Timer/Counter or 0th GPIO pin of PORT 1 |
| 2 | P1.1 (T2.EX) | Timer/Counter/External Counter or 1st GPIO pin of PORT 1 |
| 3 | P1.2 | 2nd GPIO pin of PORT 1 |
| 4 | P1.3 | 3rd GPIO pin of PORT 1 |
| 5 | P1.4 | 4th GPIO pin of PORT 1 |
| 6 | P1.5 (MOSI) | MOSI for in System Programming or 5th GPIO pin of PORT 1 |
| 7 | P1.6 (MISO) | MISO for in System Programming or 6th GPIO pin of PORT 1 |
| 8 | P1.7 (SCK) | SCK for in System Programming or 7th GPIO pin of PORT 1 |
| 9 | RST | Making this pin high will reset the Microcontroller |
| 10 | P3.0 (RXD) | RXD Serial Input or 0th GPIO pin of PORT 3 |
| 11 | P3.1 (TXD) | TXD Serial Output or 1st GPIO pin of PORT 3 |
| 12 | P3.2 (INT0’) | External Interrupt 0 or 2nd GPIO pin of PORT 3 |
| 13 | P3.3 (INT1’) | External Interrupt 1 or 3rd GPIO pin of PORT 3 |
| 14 | P3.4 (T0) | Timer 0 or 4th GPIO pin of PORT 3 |
| 15 | P3.5 (T1) | Timer 1 or 5th GPIO pin of PORT 3 |
| 16 | P3.6 (WR’) | Memory Write or 6th GPIO pin of PORT 3 |
| 17 | P3.7 (RD’) | Memory Read or 7th GPIO pin of PORT 3 |
| 18 | XTAL2 | External Oscillator Output |
| 19 | XTAL1 | External Oscillator Input |
| 20 | GND | Ground pin of MCU |
| 21 | P2.0(A8) | 0th GPIO pin of PORT 2 |
| 22 | P2.1 (A9) | 1st GPIO pin of PORT 2 |
| 23 | P2.2 (A10) | 2nd GPIO pin of PORT 2 |
| 24 | P2.3 (A11) | 3rd GPIO pin of PORT 2 |
| 25 | P2.4 (A12) | 4th GPIO pin of PORT 2 |
| 26 | P2.5 (A13) | 5th GPIO pin of PORT 2 |
| 27 | P2.6 (A14) | 6th GPIO pin of PORT 2 |
| 28 | P2.7 (A15) | 7th GPIO pin of PORT 2 |
| 29 | PSEN’ | Program store Enable used to read external program memory |
| 30 | ALE / PROG’ | Address Latch Enable / Program Pulse Input |
| 31 | EA’ / VPP | External Access Enable / Programming enable Voltage |
| 32 | P0.7 (AD7) | Address / Data pin 7 or 7th GPIO pin of PORT 0 |
| 33 | P0.6 (AD6) | Address / Data pin 6 or 6th GPIO pin of PORT 0 |
| 34 | P0.5 (AD5) | Address / Data pin 5 or 5th GPIO pin of PORT 0 |
| 35 | P0.4 (AD4) | Address / Data pin 4 or 4th GPIO pin of PORT 0 |
| 36 | P0.3 (AD3) | Address / Data pin 3 or 3rd GPIO pin of PORT 0 |
| 37 | P0.2 (AD2) | Address / Data pin 2 or 2nd GPIO pin of PORT 0 |
| 38 | P0.1 (AD1) | Address / Data pin 1 or 1st GPIO pin of PORT 0 |
| 39 | P0.0 (AD0) | Address / Data pin 0 or 0th GPIO pin of PORT 0 |
| 40 | VCC | Positive pin of MCU (+5V) |



**Programming AT89S52 Microcontroller**

Atmel microcontroller can be programmed with different software's that is available in the market. Arduino, Keil uVision are the most used platforms to name a few.

In order to program the Atmel microcontroller we will need an IDE (Integrated Development Environment), where the programming takes place. A compiler, where our program gets converted into MCU readable form called HEX files. An IPE (Integrated Programming Environment), which is used to dump our hex file into our MCUs.

IDE: [Keil uVision IDE](https://www.keil.com/download/)

Programming Hardware: USB In-circuit programmer (USBASP)

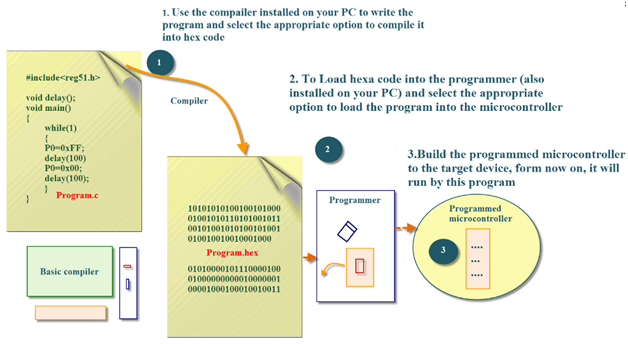
Programmer: [USBASP](https://www.robotshop.com/letsmakerobots/how-install-usbasp-drivers-windows-8)

To dump or upload our code into Atmel IC we need a programmer, the most commonly used programmer is the USBASP which has to be purchased separately.

**Embedded C is most popular programming** **language** in software field for developing electronic gadgets. Each processor used in electronic system is associated with embedded software.

Embedded C programming plays a key role in performing specific function by the processor. In day-to-day life we used many electronic devices such as mobile phone, washing machine, digital camera, etc. These all device working is based on microcontroller that are programmed by embedded C.

Let's see the block diagram representation of embedded system programming:



In embedded system programming C code is preferred over other language. Due to the following reasons:

* Easy to understand
* High Reliability
* Portability
* Scalability

**Software Specifications**

* Keil µVision IDE
* MC Programming Language: Embedded C

**Code for 8051 microcontroller**

#include<reg51.h>

#define SEGMENT P0

sbit switch1=P3^0;

sbit switch2=P3^1;

sbit digit1=P2^0;

sbit digit2=P2^1;

void delay (int);

int x=0,y,z;

unsigned char ch[]={0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x82,0xf8,0x80,0x98};

void delay (int d)

{

unsigned char i;

for(;d>0;d--)

{

for(i=250;i>0;i--);

for(i=248;i>0;i--);

}

}

void main()

{

switch1=1;

switch2=1;

digit1=1;

digit2=1;

while(1)

{

if(switch1==0)

{

x++;

delay(200);

}

else if(switch2==0)

{

x--;

delay(200);

}

y=x/10;

SEGMENT=ch[y];

digit1=0;

delay(10);

digit1=1;

z=x%10;

SEGMENT=ch[z];

digit2=0;

delay(10);

digit2=1;

}

}

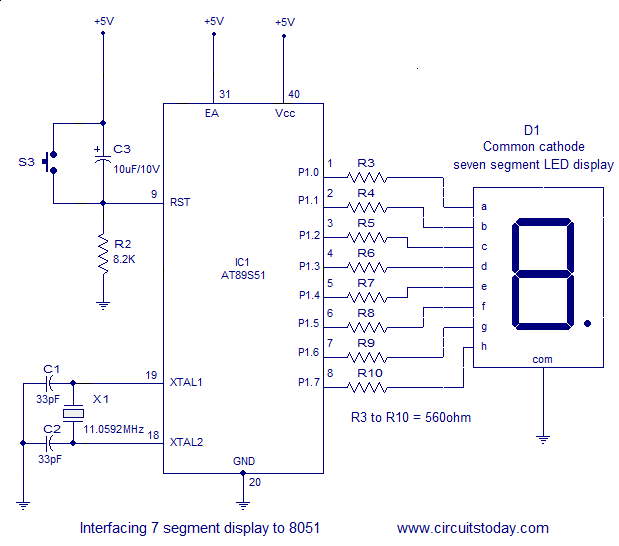
### Interfacing seven segment display to 8051.

### 

Interface a seven segment LED display to an 8051 microcontroller. 7 segment LED display is  very popular and it can display digits from 0 to 9 and quite a few characters like A, b, C, ., H, E, e, F, n, o,t,u,y, etc.

The circuit diagram shown above is of an AT89S51 microcontroller based 0 to 9 counter which has a 7 segment LED display interfaced to it in order to display the count.  This simple circuit illustrates two things. How to setup simple 0 to 9 up counter using 8051 and more importantly how to interface a seven segment LED display to  8051 in order to display a particular result. The common cathode seven segment display D1 is connected to the Port 1 of the microcontroller (AT89S51) as shown in the circuit diagram. R3 to R10 are current limiting resistors. S3 is the reset switch and R2,C3 forms a debouncing circuitry. C1, C2 and X1 are related to the clock circuit. The software part of the project has to do the following tasks.

* Form a 0 to 9 counter with a predetermined delay (around 1/2 second here).
* Convert the current count into digit drive pattern.
* Put the current digit drive pattern into a port for displaying.



**IR Sensor Interfacing With 8051**

Infrared is light that has a wavelength longer than visible red light. The ranges of infrared include near-infrared, mid-infrared, and far-infrared, spanning wavelengths from about 710 nanometers (near-infrared) to 100 micrometers (far infrared).

All objects emit light according to their temperature–this is called “black body radiation.” The hotter the object, the shorter wavelength of light it emits.

**IR Sensor**

**The infrared Obstacle Sensor Module** has a built-in **IR transmitter** and **IR receiver** that sends out IR energy and looks for reflected IR energy to detect the presence of any obstacle in front of the sensor module. The PCB of this electronic circuit has a potentiometer. That onboard potentiometer lets users adjust the detection range. The sensor has a very good and stable response even in ambient light or in complete darkness.

**Specifications**

1.Operating Voltage:**3.0V – 5.0V**

2.Detection range:**2cm – 30cm (Adjustable using potentiometer)**

3.Current Consumption: **at 3.3V : ~23 mA** ,**at 5.0V: ~43 mA**

4.Active output level: **Outputs Low logic level when an obstacle is detected**

5.Onboard Obstacle Detection LED indicator

Working Principle of IR Obstacle Sensor

An IR sensor consists of an IR LED and an IR Photodiode; together they are called as Photo–Coupler or Opto–Coupler. As said before, the Infrared Obstacle Sensor has a built-in IR transmitter and IR receiver. An **infrared Transmitter** is a light-emitting diode (LED) that emits infrared radiations. Hence, they are called IR LED. Even though an IR LED looks like a normal LED, the radiation emitted by it is invisible to the human eye.

Working Principle of IR Obstacle Sensor

An IR sensor consists of an IR LED and an IR Photodiode; together they are called as Photo–Coupler or Opto–Coupler. As said before, the Infrared Obstacle Sensor has a built-in IR transmitter and IR receiver. An **infrared Transmitter** is a light-emitting diode (LED) that emits infrared radiations. Hence, they are called IR LED. Even though an IR LED looks like a normal LED, the radiation emitted by it is invisible to the human eye.

