Interfacing ADC with 8051

Engr. Shahzada Fahim Jan

- We live in an analog world
- Temperature, humidity, pressure, are analog.
- We use transducers to convert physical quantity to electrical quantity such as voltage or current.
- For interfacing these sensors to microcontrollers we require to convert the analog output of these sensors to digital so that the controller can read it. Some microcontrollers have built-in Analog to Digital Convertor (ADC) so there is no need for external ADC. For microcontrollers that don’t have internal ADC external ADC is used.

- Well. 8051 doesn’t have an inbuilt ADC. So we have to use external ADC. There are many ADCs available.
- ADC 0804
- ADC 0808

- ADCs (analog-to-digital converters) are among the most widely used devices for data acquisition
- We need an analog-to-digital converter to translate the analog signals to digital numbers, so microcontroller can read them

ADC0804 interfacing with 8051

- The ADC0804 is a converter from analog to digital 8 bits.
- ADC0804 has only one analog input channel with digital output of eight bits that can be 256 values of different measures.
- The step size is adjusted by setting the reference voltage in pin9 the reference input voltage can be adjusted to allow encoding any smaller range for the entire 8-bit resolution analog voltage. When the voltage reference pin is not connected to the ADC0804, the reference voltage defaults to the operating voltage, ie, Vcc. The step size is 19.53mV 5V (5V / 256)

- ADC0804 also needs a clock to operate. The conversion time of the analog value to a digital value depends on the clock source. An external clock can be given at the Clock IN pin. ADC 0804 also has an inbuilt clock that can be used in absence of an external clock. A suitable RC circuit is connected between the Clock IN and Clock R pins to use the internal clock.

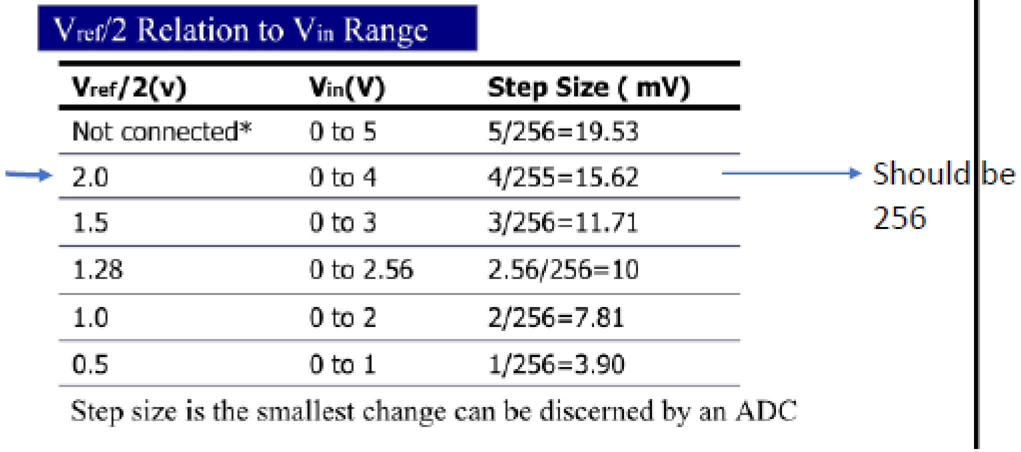
- ADC804 IC is an analog-to-digital converter
- It works with +5 volts and has a resolution of 8 bits
- Conversion time is another major factor in judging an ADC
- Conversion time is defined as the time it takes the ADC to convert the analog input to a digital (binary) number
- In ADC804 conversion time varies depending on the clocking signals applied to CLK R and CLK IN pins, but it cannot be faster than 110 µs

- CLK IN and CLK R
- CLK IN is an input pin connected to an external clock source
- To use the internal clock generator (also called self-clocking), CLK IN and CLK R pins are connected to a capacitor and a resistor, and the clock frequency is determined by

$$f = \frac{1}{1.1\,RC}$$

- Typical values are R = 10K ohms and C = 150 pF
- We get f = 606 kHz and the conversion time is 110 µs

- Vref/2
- It is used for the reference voltage If this pin is open (not connected), the analog input voltage is in the range of 0 to 5 volts (the same as the Vcc pin) If the analog input range needs to be 0 to 4 volts, Vref/2 is connected to 2 volts



$V_{ref}/2$ Relation to V_{in} Range

$V_{ref}/2$(v)	V_{in}(V)	Step Size (mV)
Not connected*	0 to 5	5/256=19.53
2.0	0 to 4	4/255=15.62
1.5	0 to 3	3/256=11.71
1.28	0 to 2.56	2.56/256=10
1.0	0 to 2	2/256=7.81
0.5	0 to 1	1/256=3.90

Step size is the smallest change can be discerned by an ADC

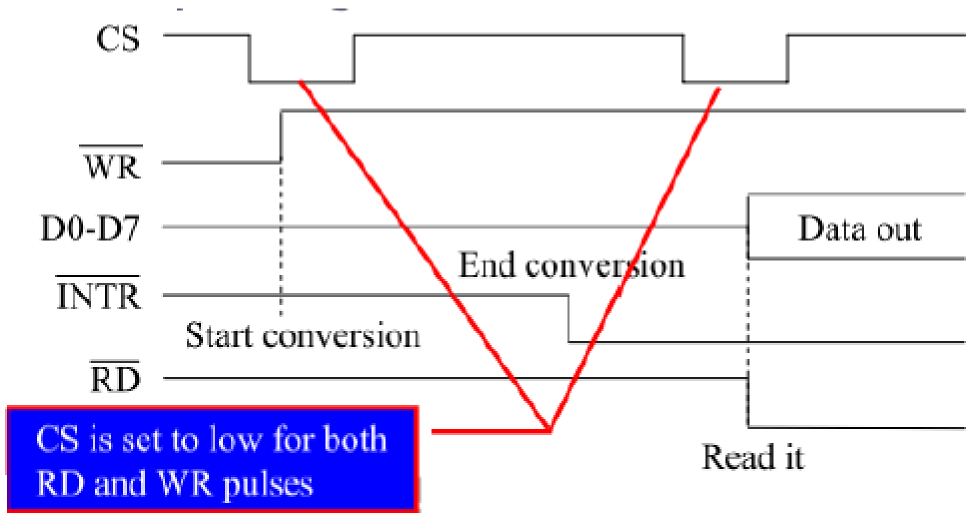
Should be 256

- D0-D7 The digital data output pins
- The converted data is accessed only when CS = 0 and RD is forced low To calculate the output voltage, use the
- following formula $D_{out} = \frac{V_{in}}{step\ size}$ 5V/255 = step size

Dout = digital data output (in decimal), Vin = analog voltage, and step size (resolution) is the smallest change

- Analog ground and digital ground
- Analog ground is connected to the ground of the analog Vin
- Digital ground is connected to the ground of the Vcc pin

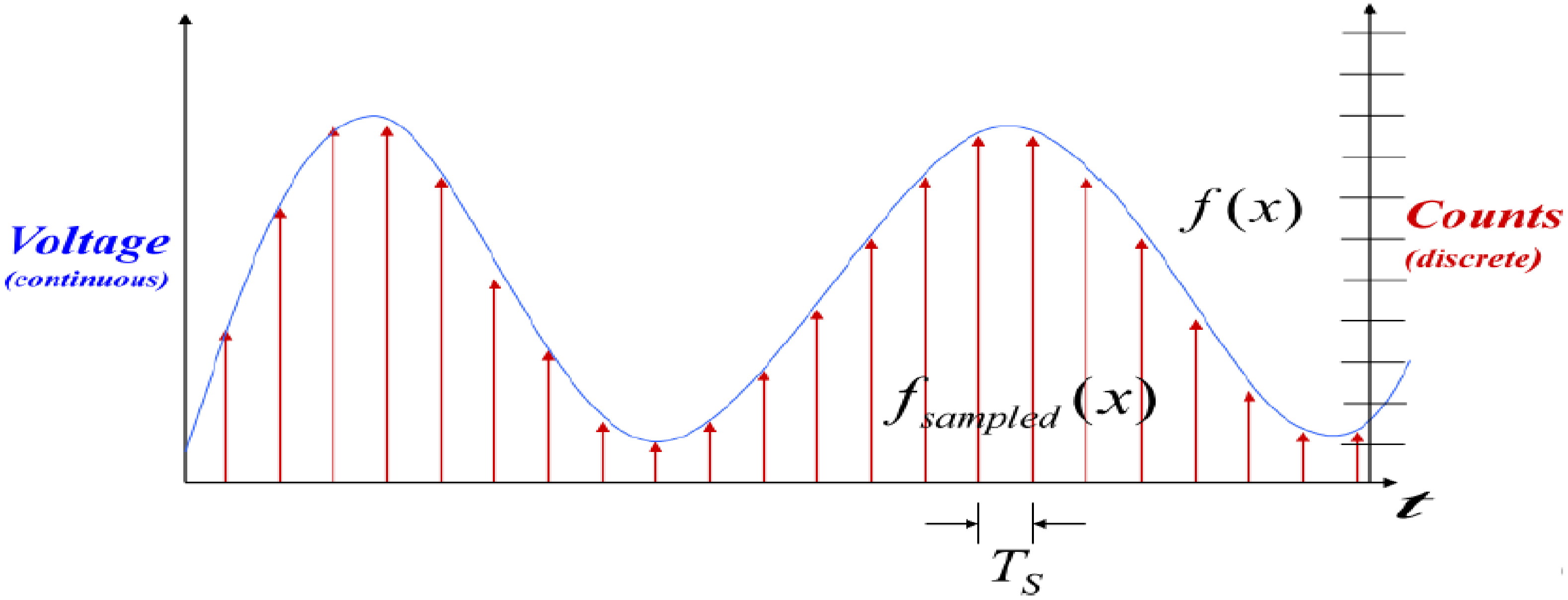
- The following steps must be followed
- for data conversion by the ADC804 chip
- Make CS = 0 and send a low-to-high pulse to pin WR to start conversion
- Keep monitoring the INTR pin
- If INTR is low, the conversion is finished
- If the INTR is high, keep polling until it goes low
- After the INTR has become low, we make CS = 0 and send a high-to-low pulse to the RD pin to get the data out of the ADC804



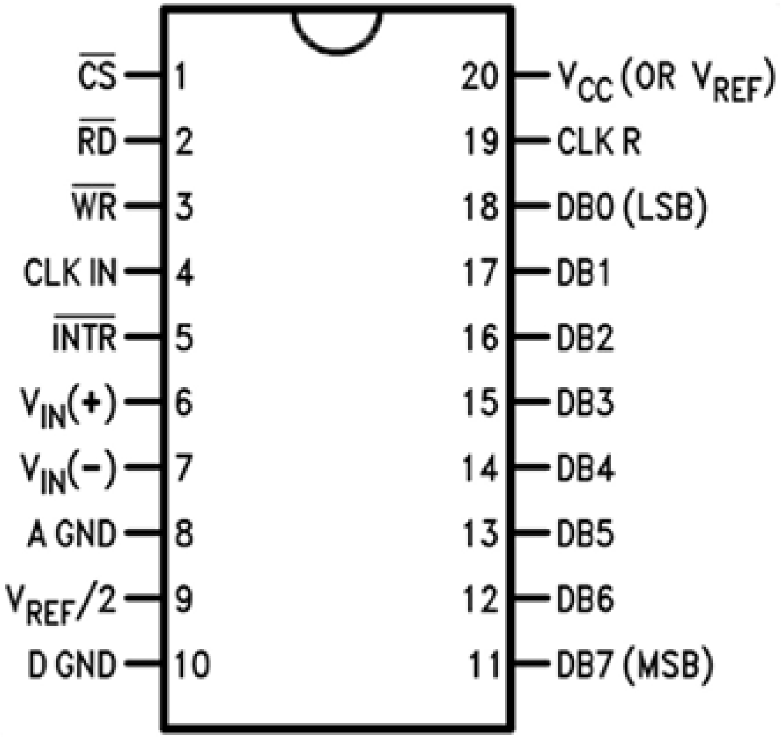
CS is set to low for both RD and WR pulses

Representing an analog signal digitally

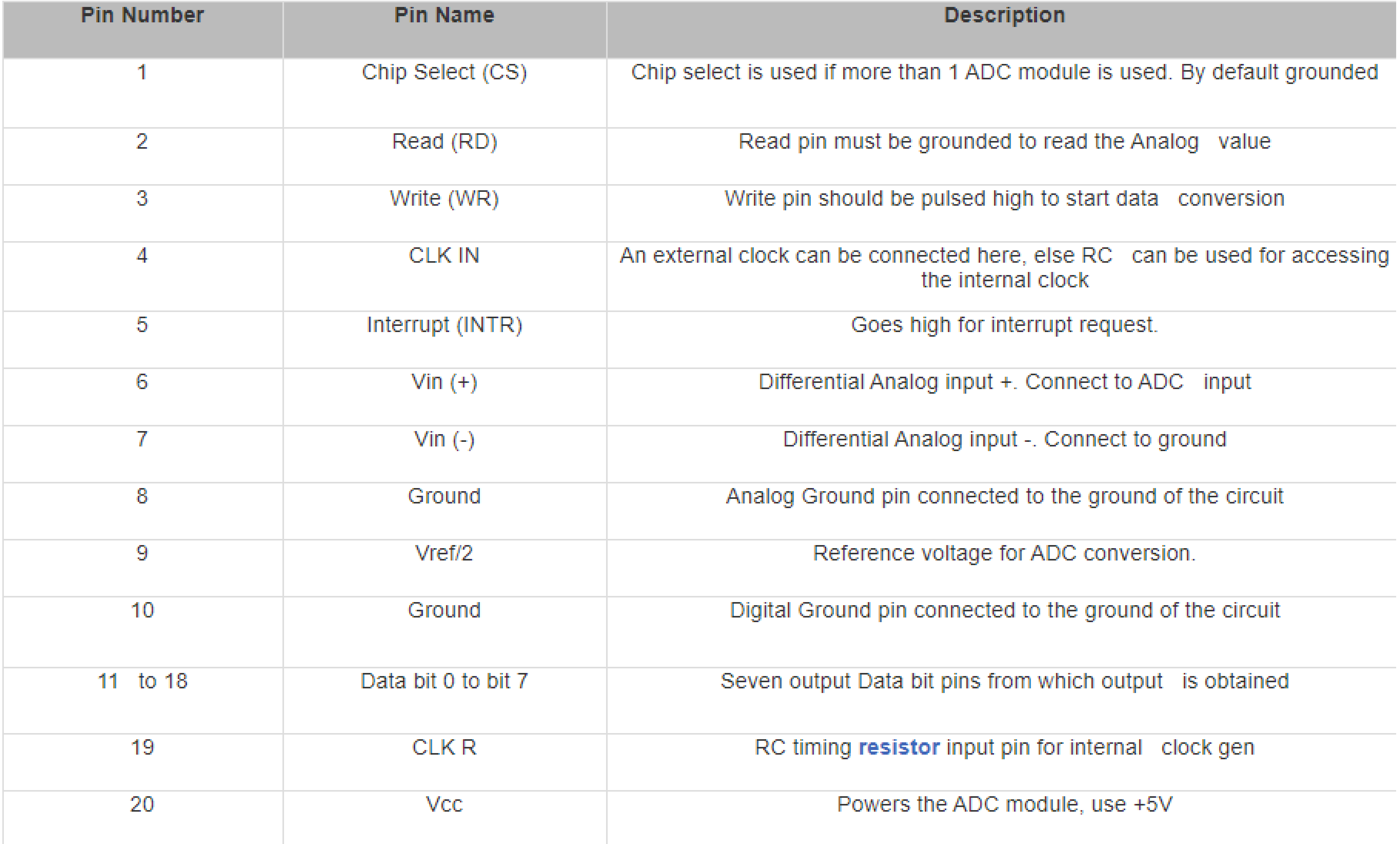
- How do we represent an analog signal (e.g. continuous voltage)?
  - As a time series of discrete values
    → On MCU: read ADC data register (counts) periodically (T_s)



ADC0804



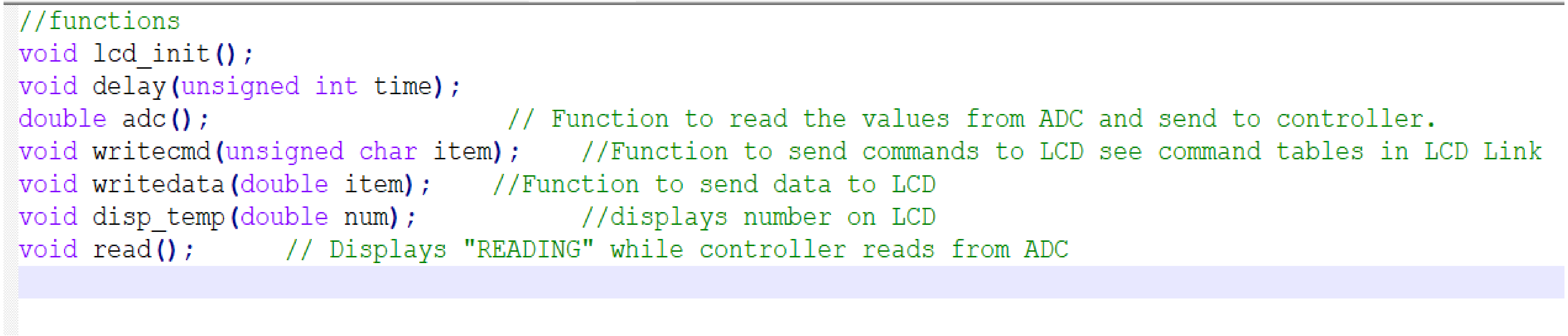
$\overline{CS}$ 1
$\overline{RD}$ 2
$\overline{WR}$ 3
CLK IN 4
$\overline{INTR}$ 5
$V_{IN}(+)$ 6
$V_{IN}(-)$ 7
A GND 8
$V_{REF}/2$ 9
D GND 10
20 V_{CC} (OR V_{REF})
19 CLK R
18 DB0 (LSB)
17 DB1
16 DB2
15 DB3
14 DB4
13 DB5
12 DB6
11 DB7 (MSB)



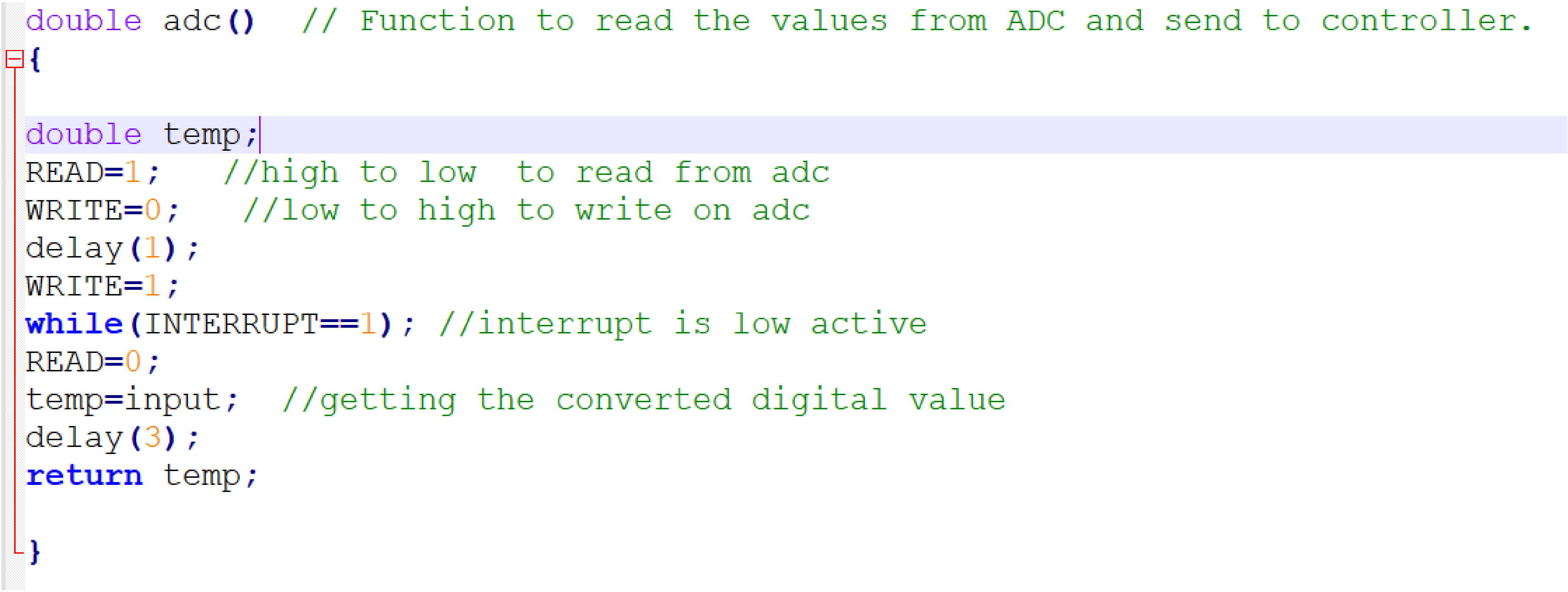
Pin Number	Pin Name	Description
1	Chip Select (CS)	Chip select is used if more than 1 ADC module is used. By default grounded
2	Read (RD)	Read pin must be grounded to read the Analog value
3	Write (WR)	Write pin should be pulsed high to start data conversion
4	CLK IN	An external clock can be connected here, else RC can be used for accessing the internal clock
5	Interrupt (INTR)	Goes high for interrupt request.
6	Vin (+)	Differential Analog input +. Connect to ADC input
7	Vin (-)	Differential Analog input -. Connect to ground
8	Ground	Analog Ground pin connected to the ground of the circuit
9	Vref/2	Reference voltage for ADC conversion.
10	Ground	Digital Ground pin connected to the ground of the circuit
11 to 18	Data bit 0 to bit 7	Seven output Data bit pins from which output is obtained
19	CLK R	RC timing **resistor** input pin for internal clock gen
20	Vcc	Powers the ADC module, use +5V

How to use the ADC0804?

- Make Cs(chip select) low(0) to activate the IC. Here we are directly connecting that CS into Ground instead of connecting into 8051.
- Make WR(write) pin low (0).
- Make RD(read) pin high (1).
- Make WR(write) pin high (1) after some delay small delay//This low to high impulse at WR pin starts your conversion.
- Now check the INTR(interrupt) pin if it is high(1) conversion is running if it is low(0) the conversion is over.
- Make RD=low(0) .



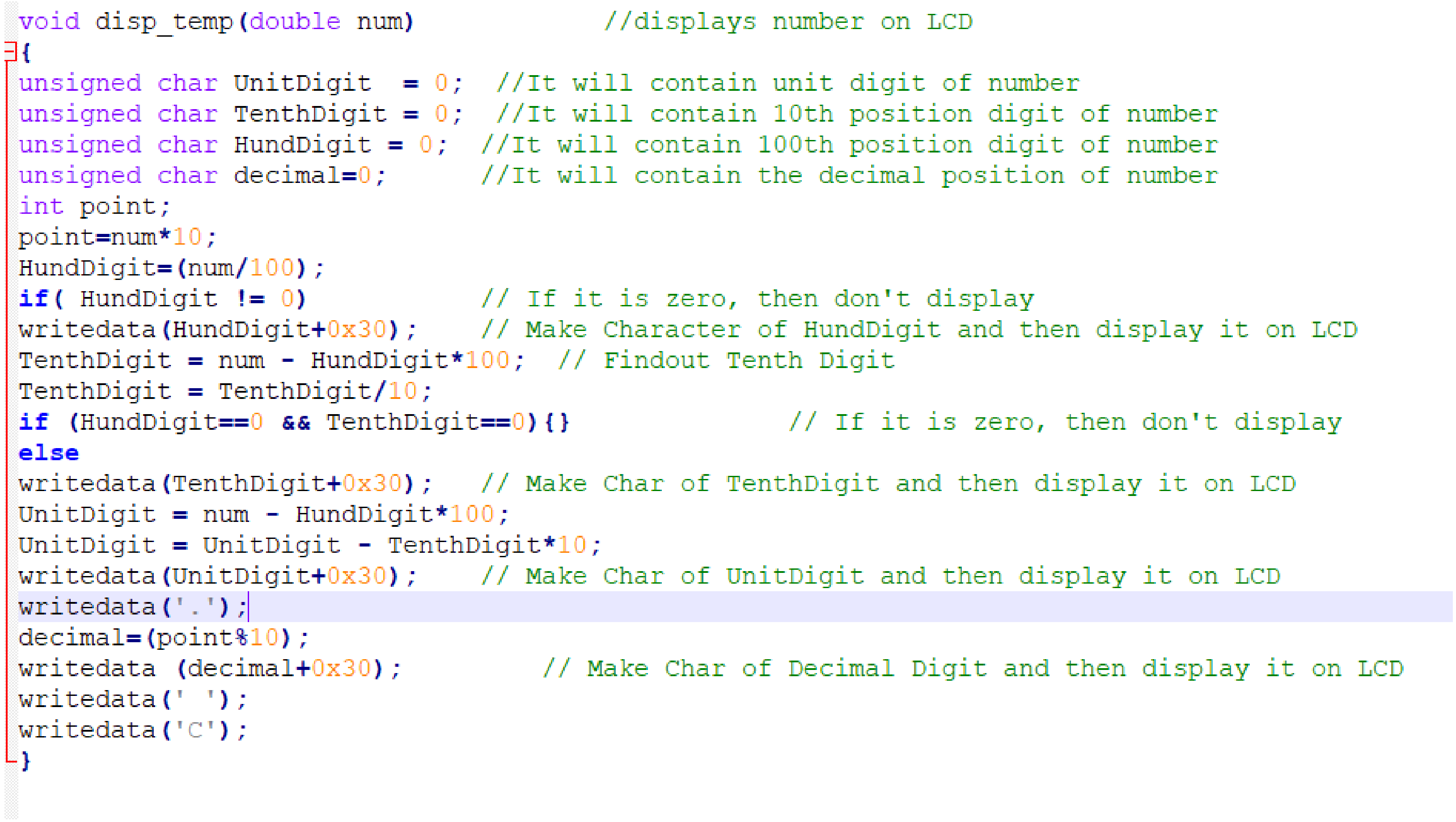
```
//functions
void lcd_init();
void delay(unsigned int time);
double adc();                    // Function to read the values from ADC and send to controller.
void writecmd(unsigned char item);    //Function to send commands to LCD see command tables in LCD Link
void writedata(double item);    //Function to send data to LCD
void disp_temp(double num);           //displays number on LCD
void read();      // Displays "READING" while controller reads from ADC
```



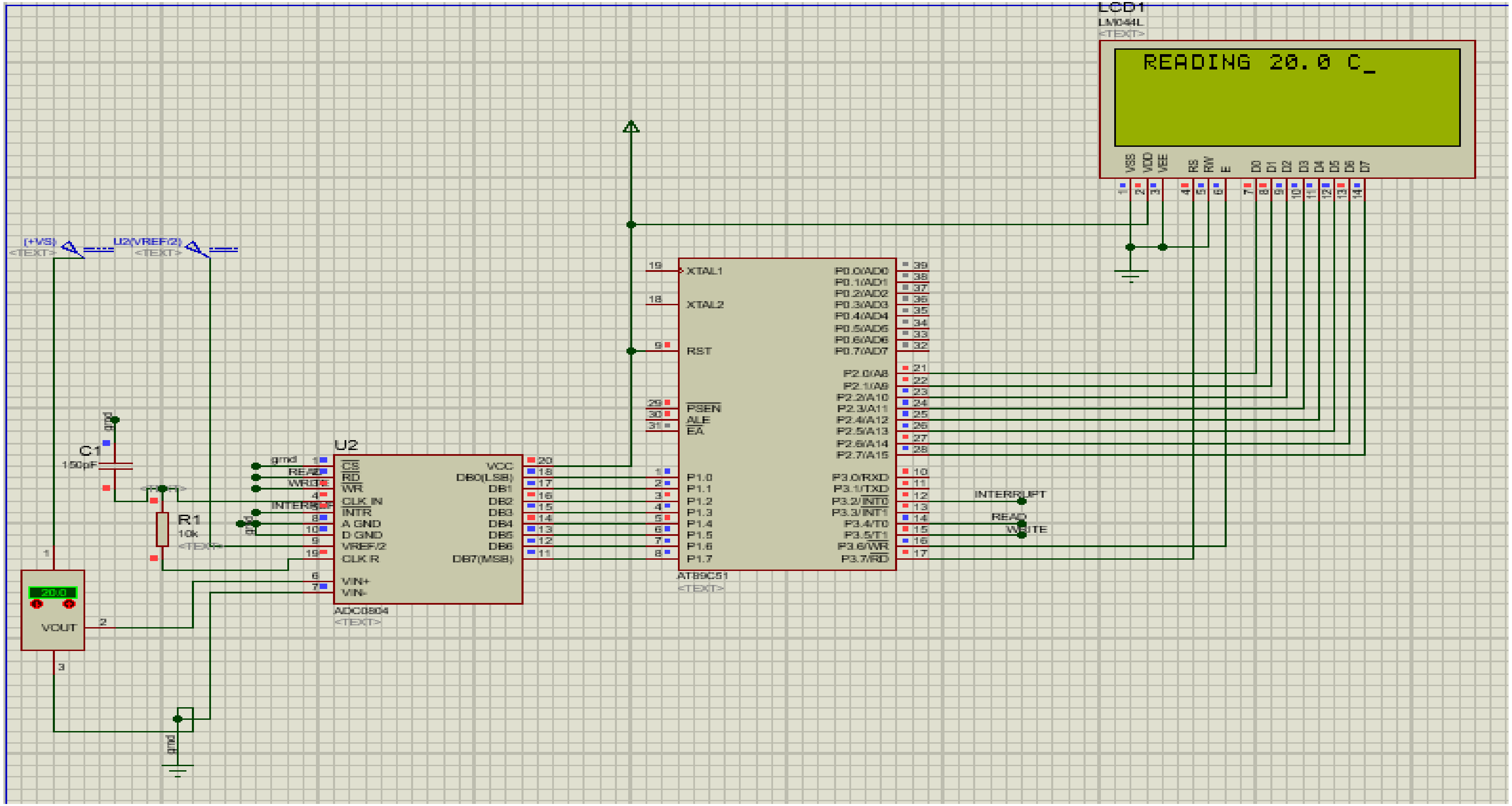
```
double adc()  // Function to read the values from ADC and send to controller.
{

double temp;
READ=1;   //high to low  to read from adc
WRITE=0;   //low to high to write on adc
delay(1);
WRITE=1;
while(INTERRUPT==1); //interrupt is low active
READ=0;
temp=input;  //getting the converted digital value
delay(3);
return temp;

}
```



```c
void disp_temp(double num)            //displays number on LCD
{
unsigned char UnitDigit  = 0;  //It will contain unit digit of number
unsigned char TenthDigit = 0;  //It will contain 10th position digit of number
unsigned char HundDigit = 0;  //It will contain 100th position digit of number
unsigned char decimal=0;      //It will contain the decimal position of number
int point;
point=num*10;
HundDigit=(num/100);
if( HundDigit != 0)           // If it is zero, then don't display
writedata(HundDigit+0x30);    // Make Character of HundDigit and then display it on LCD
TenthDigit = num - HundDigit*100;  // Findout Tenth Digit
TenthDigit = TenthDigit/10;
if (HundDigit==0 && TenthDigit==0){}              // If it is zero, then don't display
else
writedata(TenthDigit+0x30);   // Make Char of TenthDigit and then display it on LCD
UnitDigit = num - HundDigit*100;
UnitDigit = UnitDigit - TenthDigit*10;
writedata(UnitDigit+0x30);    // Make Char of UnitDigit and then display it on LCD
writedata('.');
decimal=(point%10);
writedata (decimal+0x30);         // Make Char of Decimal Digit and then display it on LCD
writedata(' ');
writedata('C');
}
```


LCD1
LM044L
<TEXT>
READING 20.0 C_
VSS
VDD
VEE
RS
RW
E
D0
D1
D2
D3
D4
D5
D6
D7
(+VS)
<TEXT>
U2(VREF/2)
<TEXT>
XTAL1
XTAL2
RST
PSEN
ALE
EA
P0.0/AD0
P0.1/AD1
P0.2/AD2
P0.3/AD3
P0.4/AD4
P0.5/AD5
P0.6/AD6
P0.7/AD7
P2.0/A8
P2.1/A9
P2.2/A10
P2.3/A11
P2.4/A12
P2.5/A13
P2.6/A14
P2.7/A15
P1.0
P1.1
P1.2
P1.3
P1.4
P1.5
P1.6
P1.7
P3.0/RXD
P3.1/TXD
P3.2/INT0
P3.3/INT1
P3.4/T0
P3.5/T1
P3.6/WR
P3.7/RD
AT89C51
<TEXT>
INTERRUPT
READ
WRITE
U2
CS
RD
WR
CLK IN
INTR
A GND
D GND
VREF/2
CLK R
VIN+
VIN-
VCC
DB0(LSB)
DB1
DB2
DB3
DB4
DB5
DB6
DB7(MSB)
ADC0804
<TEXT>
gnd
READ
WRITE
INTERRUPT
C1
150pF
R1
10k
<TEXT>
20.0
VOUT
gnd

TASK

- From temperature sensor (LM35) read temperature and convert it into Digital value by using ADC0804 and display the value on the LCD. In LCD at first line write your registration Number and on the second line display the value of the temperature sensor attached with ADC0804.