Counting type ADCs

Posted on February 28, 2008, by Ibrahim KAMAL, in General electronics, tagged

In this article, we will discuss a very common type of digital to analog converters called the counting type ADC. Based on our previous simple tutorial about DACs (digital to analog converters), this article presents a technique of building ADCs that you can use to learn and master the process of analog to digital conversion, but also use it in many of you projects, adding an incredible feature to basic microcontrollers like the 8051 that don't have integrated ADCs..

Note: You can find the schematic of a very simple 3-bit Flash ADC here.

The principle of operation

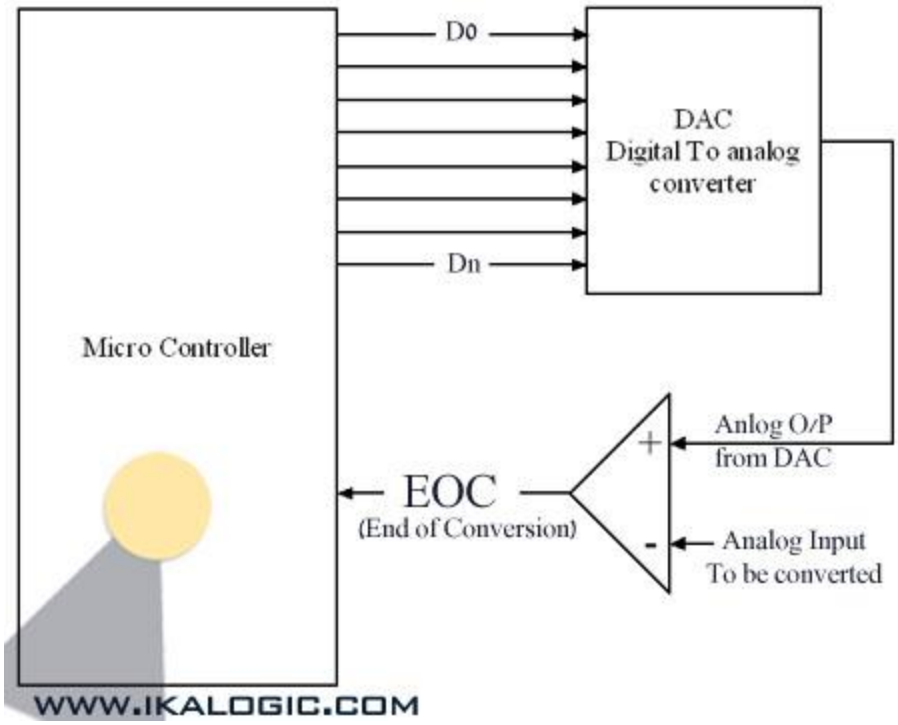


figure1.A

While there are many ways of building and implementing a counting type ADC, they all rely on the same basic idea. I am going to use a micro-controller by default to perform all the required logical operations, because in most cases, where an ADC is found, a micro-controller is also present. As you can see in ***figure 1.A***, the counting type ADC is based on Digital to analog converter (DAC). An R/2R DAC is very suitable for this task, and it can offer more precision by adding more branches to the R/2R network, consequently adding more bit depth to the converter, or you may call it "more resolution".

To understand how an analog input is converted to digital data, you have to think in the reverse direction, because that's what really happens, the microcontroller tries to mimic the analog input by producing the closest analog input through the DAC. More precisely, the micro controller generates a digital ramp (a signal increasing from 0 to max scale) on the input of the DAC, which in case of an 8-bit DAC with a 5V maximum output voltage, would be a

counting from 0 to 255, generating an increasing analog voltage (0v to 5V) at the output of the DAC, which is then fed to a comparator to be compared with analog input signal being converted.

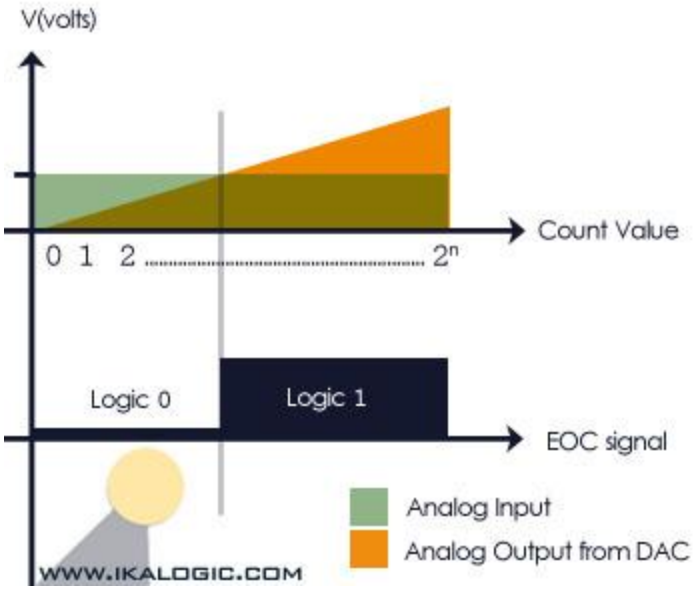


figure 1.B

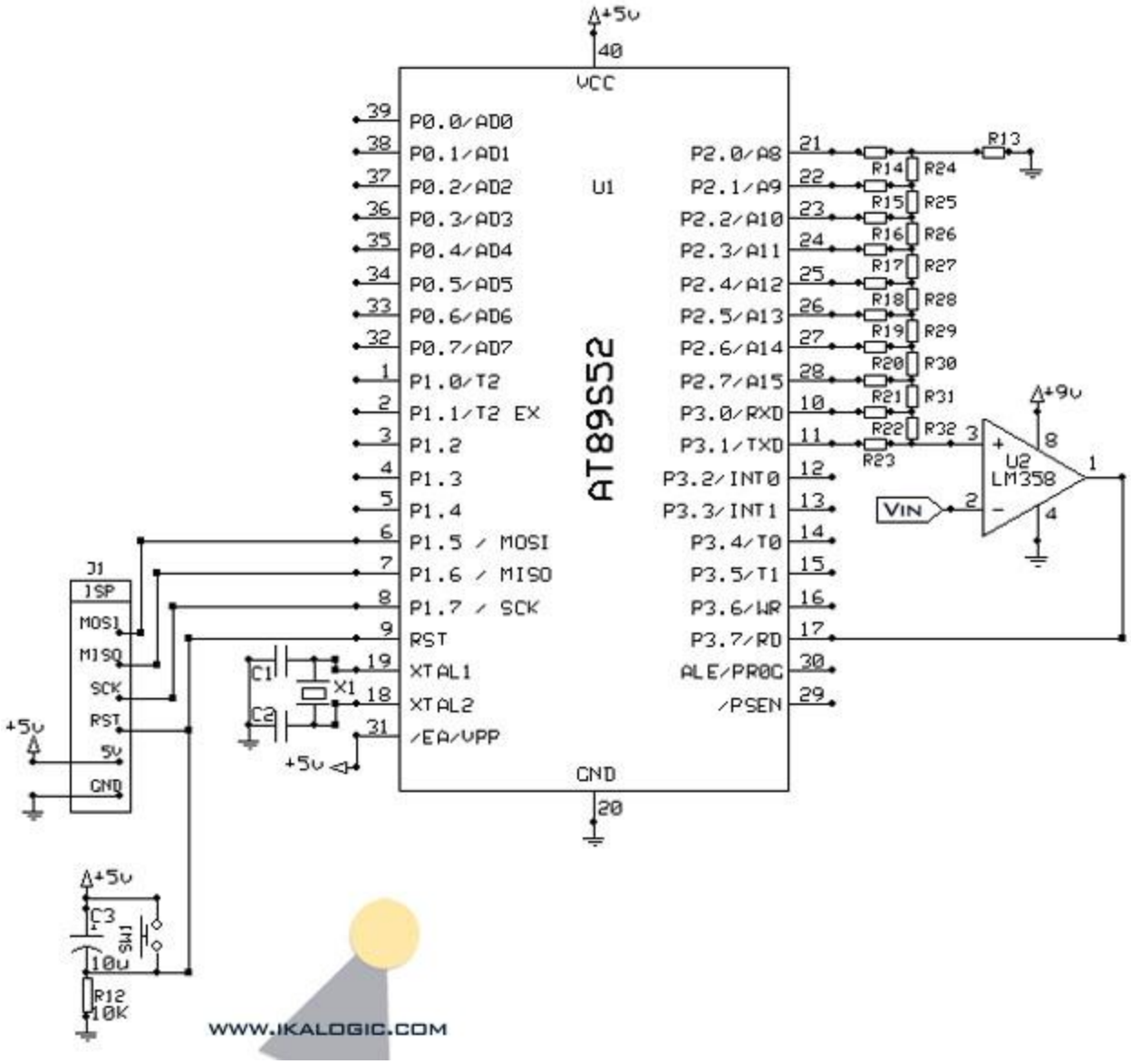
While the analog output voltage from the DAC is smaller than the measured input, the comparator will output a logic 0. Once the DAC output a voltage that is slightly bigger than the analog input, the comparator will output a logic 1 indicating the end of the conversion (refereed to as EOC). The job of the microcontroller is to monitor the output of the comparator, and record the value of the data that was sent to the DAC just before, or just after the comparator' output flipped form 0 to 1.

Figure 1.B may help you to imagine the relation between the three major signals: The analog input being measured, the Analog output from the DAC and the EOC signal.

The resolution – which is the smallest change in the input that can be detected – depend on the number of data line 'n'. That means that you can build your own ADC with any precision you may need. That becomes interesting sometimes when you need very low or very high precision ADCs.

The Electronic circuit

Any Experienced reader may have noticed that the hardware for such a device is very simple (with disregard to the microcontroller). Indeed, you only need to slightly change the design of the DAC explained in this early tutorial. The components on the left part of the schematic are standard in most of the projects, which are the capacitors (C_1 and C_2), the crystal oscillator (X_1), the reset switch (SW_1) with the debouncing capacitor (C_3) and resistor (R_{12}), and the connector (J_1) for ISP programming. The resistor R_{13} to R_{32} and LM358 comparator are responsible of converting Digital signals to Analog signals with 10 bits resolution, and adding to that a micro-controller, we get an ADC.



As mentioned before, the purpose of a micro-controller in this application is to generate the analog signals to be compared with the measured input voltage (Vin). The LM358 comparator will give a change of logic level (from 0 to 1) indicating that the simulated analog voltage reached the measured voltage, thus indicating the end of conversion.

The Software

The software to be loaded on a microcontroller to perform the task of converting analog signals to digital ones is very simple; The pins P3.0 and P3.1 are both set to Logic 0, and the value of port 2 (which is connected to the first 8 bits of the DAC) is gradually incremented. After each increment the pin P3.7 is checked. Whenever P3.7 is low (logic 0) that means that the generated Analog signal corresponds (with one bit accuracy) to the measured analog signal, and consequently corresponds to the last counted digital value in the micro-controller. In order to extend the resolution from 8 bits to 10 bits, the value of the pins P3.0 and P3.1 is increased each time Port 2 overflows. This way, the

precision of this converter is **5 / 1024 = 0.005V**, and you can freely increase the precision by increasing the number of bits.

Here is an example C code for the 8051 micro-controller to read the voltage from the above schematic:

```
While (1){
  done = 0;
  P3_0 = 0;
  P3_1 = 0;
  P3_7 = 1;     //set P3_7 as input
  P2 = 0;       //Start counting from 0
  delay(100);

  while (P2 < 255){
      P2++;
      delay(100);          //Slow down the process, to be compatible with
      if (P3_7 == 1){      //the response time of the Op-Amp.
        done = 1;
        break;
      }
  }
  if (done == 0){
  P3_0 = 1;
  P3_1 = 0;
  P3_7 = 1;
  P2 = 0;
  while (P2 < 255){
      P2++;
      delay(100);
      if (P3_7 == 1){
        done = 1;
        break;
      }
```

```
}
}

if (done == 0){
P3_0 = 0;
P3_1 = 1;
P3_7 = 1;
P2 = 0;
while (P2 < 255){
    P2++;
    delay(100);
    if (P3_7 == 1){
      done = 1;
      break;
    }
}
}

if (done == 0){
P3_0 = 1;
P3_1 = 1;
P3_7 = 1;
P2 = 0;
while (P2 < 255){
    P2++;
    delay(100);
    if (P3_7 == 1){
      done = 1;
      break;
    }
}
}
```

```
if (done == 1){
bit8 = P3_0;
bit9 = P3_1;
voltage = ((P2 + (bit8 * 256) + (bit9 * 512))*conversion_factor);
}
```

The variable ‘conversion_factor’ in the code represent a factor that relates the counted value to the corresponding voltage, and is easily obtained using some trials and errors. Noting that the relation between the counted value and the corresponding voltage is linear, it’s easy to find this factor based on only two readings.

Source: http://www.ikalogic.com/counting-type-adcs/