

## مبدل آنالوگ به دیجیتال ( ANALOG TO DIGITAL CONVERTOR )

متداول ترین انواع ADC ها به قرار زیر است :

۱. مبدل ADC نوع شمارشی
۲. مبدل ADC از نوع تقریب های متوالی
۳. مبدل ADC با مقایسه موازی
۴. مبدل ADC دو شبیه

### مبدل نوع SUCCESSIVE-APPROXIMATION

مبدل آنالوگ به دیجیتال داخلی میکرو های AVR که ADC دارند از این نوع است به همین دلیل قصد داریم در مورد این نوع ADC مختصری توضیح دهیم . به جای شمارنده در این طرح از یک میکروکنترلر یا میکرو پروسسور استفاده می شود . با برنامه ای MSB یک شده و در یک DAC به آنالوگ تبدیل شده خروجی DAC در مقایسه گر با سیگنال آنالوگ مقایسه می شود و اگر خروجی DAC بزرگتر باشد MSB صفر شده و MSB بعدی 1 می شود و مقایسه می شود و اگر کوچکتر باشد MSB 1 باقی می ماند و MSB بعدی 1 می شود و این عمل به همین ترتیب ادامه پیدا می کند تا سیگنال آنالوگ خروجی DAC با سیگنال آنالوگ حاضر در پایه ADC برابر شود .

## مبدل آنالوگ به دیجیتال داخلی میکرو

خصوصیات مبدل آنالوگ به دیجیتال داخلی AVR به شرح زیر است :

• وضوح 10 بیتی

صحت مطلق  $\pm 2\text{LSB}$

زمان تبدیل (  $\text{CONVERSIONTIME}$  ) 65-260ms

وضوح 15 KSPS در بالاترین حد

کانالهای مولتی پلکس شده

مدهای تبدیل SINGLE و FREE

ولتاژ ورودی از 0V تا VCC

پرچم وقفه پایان تبدیل ADC

حذف کننده نویز ( NOISE CACELER )

ADC بسته به میکرو به چند کانال آنالوگ مالتی پلکس شده که به هر یک از پایه های پورت اجازه می دهد که به عنوان یک ورودی مبدل آنالوگ به دیجیتال عمل نماید. مبدل داخلی میکرو دارای وضوح 10 بیتی است و برای تبدیل با این وضوح نیاز به فرکانس کلاکی بین 50KHZ تا 200KHZ دارد و این کلاک را از تقسیم فرکانس کریستال تامین می کند. در صورت که نیاز به وضوح بالا (کمتر از 10 بیت) نیست می توان کلاکی بالاتر از 200KHZ به آن اعمال کرد. ADC دارای یک SAMPLE AND HOLD است که باعث می شود ولتاژ ورودی ADC در زمان تبدیل در سطح ثابت نگه داشته شود تا عملیات تبدیل با دقت بیشتری انجام شود.

ADC دارای دو منبع ولتاژ آنالوگ مجزا است. AVCC و AGND که AGND بایستی به زمین یا ولتاژ زمین آنالوگ متصل شود و AVCC نباید بیشتر از  $\pm 0.3V$  نسبت به VCC اختلاف داشته باشد. (ولتاژ مرجع) VOLTAGE REFERENCE (خارجی در صورت وجود باید به پایه AREF وصل شود که

این ولتاژ بایستی بین ولتاژ موجود بر روی پایه های  $AVCC$  -  $AGND$  باشد . در غیر این صورت به  $VCC$  وصل می شود .  $ADC$  مقدار آنالوگ ورودی را با تقریب متوالی به مقدار دیجیتال 10 بیتی تبدیل می کند . کمترین مقدار ، نشان دهنده مقدار آنالوگ موجود در پایه  $AGND$  و بالاترین مقدار نشان دهنده ولتاژ پایه  $AEF$  منهای یک  $LSB$  است .

به طور مثال اگر پایه به ولتاژ  $AREF=3.5V$  و  $AGND=0V$  وصل شده باشد ، مقدار دیجیتال شده 1023 نشان دهنده ولتاژ  $3.5V$  و مقدار 0 نشان دهنده ولتاژ  $0.0V$  بر روی پایه مبدل  $ADC$  انتخاب شده است .  $ADC$  دارای دو مد تبدیل  $SINGLE$  و  $FREE$  است .

مد  $SINGLE$  بایستی توسط کاربر پیکره بندی و کانال دلخواه برای نمونه برداری انتخاب شود . در مد  $FREE$  ،  $ADC$  با یک ثابت نمونه برداری ، رجیستر داده  $ADC$  را  $UPDATE$  می کند .

### تکنیک های کاهش نویز $ADC$

- مدارات دیجیتال در داخل و خارج میکرو ایجاد نویز کرده و ممکن است در صحت اندازه گیری  $ADC$  تاثیر بگذارند . اگر صحت و دقت تبدیل قابل قبول نیست با استفاده از تکنیک های زیر می توان نویز را کاهش داد .
۱. بخش آنالوگ چیپ و تمام قسمت های آنالوگ باید دارای زمین جداگانه باشند . این زمین ها با زمین دیجیتال به وسیله یک مسیر به هم متصل می شوند .
  ۲. مسیر های آنالوگ را می توانید کوتاه کنید و از تماس نداشتن مسیر های آنالوگ و زمین های آنالوگ اطمینان پیدا کنید و آنها را از مسیر های دیجیتال با فرکانس بالا دور کنید .
  ۳. پایه  $AVCC$  میکرو را به  $VCC$  با فیلتر پایین وصل می کنیم .
  ۴. از مدهای  $SLEEP$  و حالت  $ADC$  NOISE CANCELER برای کاهش نویز القاء شده توسط  $CPU$  استفاده کنید .

### پیکره بندی ADC در محیط BASCOM

مبدل آنالوگ به دیجیتال باید برای میکروهای که ADC دارند توسط این دستور پیکره بندی شود تا بتوان از آن استفاده نمود. مبدل داخلی میکرو دارای وضوح 10 بیتی است و برای تبدیل با این وضوح نیاز به کلاکی بین 50KHZ تا 200KHZ دارند و این کلاک را از تقسیم کریستال تامین می کند .

CONFIG ADC = SINGLE/ FREE, PRESCALER= AUTO,  
REFERENCE = OPTIONAL

**CONFIG ADC = SINGLE/ FREE** : برای تبدیل سیگنال آنالوگ خود به دیجیتال می توانید از دو مد SINGLE و FREE استفاده نمایید. زمانی که مد SINGLE را انتخاب می کنید باید از دستور (GETADC) استفاده کنید .

**PRESCALER** : این گزینه کلاک ADC را مشخص می کند. با قرار دادن **PRESCALER = AUTO** کامپایلر با توجه به فرکانس اسیلاتور ، بهترین کلاک را برای ADC تعیین می کند دیگر مقادیر معتبر 2,4,8,15,32,64 یا 128 می باشند .

**REFERENCE = OPTIONAL**

گزینه ای اختیاری برای ولتاژ مرجع ( VOLTAGE REFERENCE ) در بعضی از میکرو ها از جمله MEGA8 است. OPTIONAL می تواند گزینه های زیر باشد :

**OFF** : برای خاموش کردن ولتاژ مرجع داخلی ( **INTERNAL** )

( **REFERENCE TURNED OFF** ) و استفاده از ولتاژ موجود بر روی پایه

AREF به عنوان ولتاژ مرجع انتخاب می شود .

**AVCC** : زمانی که ولتاژ پایه AVCC به عنوان ولتاژ مرجع در نظر گرفته می

شود. در این حالت اتصال پایه های AVCC و AREF به این ترتیب است .

**INTERNAL**

زمانی که ولتاژ مرجع داخلی 2.56V با خازن خارجی بر روی پایه AREF

استفاده شود. انتخاب این گزینه برای میکرو هایی که ولتاژ مرجع داخلی ندارند ، هیچ تاثیری ندارد .

### دستور GETADC

با این دستور سیگنال آنالوگ وارد شده به کانالهای ( 0 تا 7 ) به مقدار دیجیتال تبدیل می شود و در متغیر var از نوع داده WORD قرار می گیرد .

Var = GETADC(channel)

Var نتیجه تبدیل و channel کانال مبدل آنالوگ به دیجیتال داخلی انتخاب شده است که می تواند بین 0 تا 7 باشد. این دستور فقط برای چیپهای AVR که دارای مبدل آنالوگ به دیجیتال می توانید به عنوان پورتهای ورودی و خروجی استفاده کنید ولی هنگامی که پورت به عنوان ADC پیکره بندی می شود دیگر نمی توانید از آن به عنوان I/O استفاده کنید .

### دستورات START و STOP

توسط دستور START ADC , شروع به نمونه برداری از سیگنال آنالوگ کرده و توسط STOP ADC تغذیه را از ADC بایستی نوشته شود .  
مثال .

\$Regfile = "M32def.dat"

config Adc = single , prescaler = Auto 'Now Give power To The Chip

stop Adc

start Adc ' Wite Stop Adc , You Can Remove The Power From The Chip

Dim W As Word , Channel As Byte

Channel 1 = 0

Do

W = Getadc (Channel ) 'Now Read A / D Value from Channel 0 to 7

Print "Channel " : Channel: "Value " : W

Incr Channel

If Channel > 7 Then Channel = 0

Loop

END

### کار با وقفه ADC

زمانی که کار نمونه برداری ADC از سیگنال آنالوگ به پایان رسید ADC پرچم اتمام تبدیل خود به نام ADC را یک می کند. با فعال کردن وقفه سراسری با دستور ENABLE INTERRUPTS و فعال کردن وقفه ADC با دستور ENABLE ADC می توان به زیر برنامه وقفه ADC یا ISR مربوطه است. می توان برای کاهش نویز سیستم میکرو را در زمان نمونه برداری در مد IDLE یا ADC NOISE REDUCTION قرار داد و سپس میکرو خود کار با بالا رفتن پرچم وقفه با بالا رفتن پرچم وقفه اتمام تبدیل ADC از این مد بیدار شده و مقدار دیجیتال شده را در متغیر نوع WORD قرار می دهد.

### تغذیه :

قسمت تغذیه به منظور فراهم کردن ولتاژ پنج ولت DC می باشد که جهت راه اندازی IC میکروکنترلر و کلیه مدارات سازگار با TTL به کار می رود. ورودی قسمت تغذیه می تواند یک سیگنال AC یا DC باشد که الزاماً به اندازه چند ولت از +5V بیشتر است. این ورودی توسط یک سوئیچ ON-OFF قطع و وصل می شود. در حالتی که سوئیچ روشن است و ورودی وارد یک پل دیود شارژ می شود این خازن به علت ظرفیت بالای خود باعث می شود ولتاژی تقریباً DC به دست می آید. که دارای ریبیل است. برای ایجاد یک ولتاژ +5V کاملاً DC از یک IC تنظیم کننده ولتاژ 7805 استفاده می کنیم که ورودی دارای اعوجاج را به یک ولتاژ کاملاً مستقیم پنج ولت تبدیل می کند در خروجی تنظیم کننده ولتاژ یک خازن 10mf قرار داده شده که به اندازه 5 ولت شارژ خواهد شد که نوسانات خروجی ناشی از تغییرات بار را کاهش می دهد. به منظور نشان دادن روشن بودن مدار تغذیه از یک LCD استفاده شده که توسط یک مقاومت در حدود 330 اهم به خروجی تنظیم کننده ولتاژ وصل شده است.

### دستورات کار با LCD دستور CLS

این دستور تمام صفحه نمایش LCD چه قسمت متنی و چه گرافیکی را پاک می کند

### دستور CLSGRAPH

این دستور فقط قسمت گرافیکی را پاک می کند .

### دستور CLS TEXT

این دستور فقط قسمت متنی را پاک می کند .

### دستور LCD

این دستور برای نوشتن متن بر روی LCD استفاده می شود . این دستور همانند دستور LCD برای LCD های ماتریسی عادی عمل می کنند .

### دستور PSET X , Y , COLOR

این دستور یک PIXEL را در مختصات ( X,Y ) به ازای COLOR=0 خاموش و به ازای COLOR=1 روشن می کند . X از 0-239 و Y از 0-127 می تواند تغییر کند .

### دستور LOCATE ROW , COLUMN

این دستور مکان نما را در مکان سطر ( ROW ) و ستون ( COLUMN ) مشخص شده قرار می دهد . ROW می تواند از 1 تا 16 تغییر کند . تغییرات COLUMN بستگی به انتخاب MODE دارد که می تواند از 1 تا 40 تغییر کند .

### دستور CURSOR ON/OFF BLINK/NOBLINK

برای قسمت هایی متنی استفاده می شود . مکان نما می تواند در حالت های ON یا OFF و چشمک زدن ( BLINK ) یا چشمک نزدن ( NOBLINK ) باشد .

### دستور COLOR , Line(X0,Y0)-(X1,Y1)

با این دستور از PIXEL اول با مختصات (X0,Y0) به PIXEL دوم با مختصات (X1,Y1) خطی با رنگ COLOR کشیده می شود. COLOR=0 خط را پاک کرده و به ازای COLOR = 255 خطی با رنگ سیاه رسم خواهد شد .

### دستور CIRCLE (X0,Y0) , RADIUS , COLOR

این دستور دایره ای به مختصات مرکزیت (X0,Y0) و شعاع RADIUS و رنگ COLOR رسم خواهد کرد. COLOR=0 دایره را پاک کرده و به ازای COLOR=255 دایره با رنگ سیاه رسم خواهد شد .

### دستور SHOWPIC X,Y , LABLE

برای نمایش عکسی که در منوی TOOLS و قسمت GRAPHIC ONVERTER ذخیره کرده اید استفاده می شود. X مکان قرار گیری افقی و Y مکان قرار گیری عمودی عکس را نشان می دهد. LABLE نام برجسبی است که اطلاعات عکس مورد نظر در آن قرار دارد .

### برجسب "FILE.BGF" \$BGF

اشاره به فایل BGF و یا همان عکس مورد نظر که با فورمت BGF و با نام دلخواه FILE در کنار برنامه اصلی ذخیره شده است , دارد .

### دستورات مربوط به ارتباط SPI

#### دستور SPIINIT

توسط این دستور پایه های به کار برده شده در ارتباط SPI, INITIAL می شوند . این دستور بعد از پیکره بندی SPI بایستی برای قرار گیری پایه های استفاده شده در جهت مناسب نوشته شود .



### دستور SPIIN

SPIIN var,bytes

توسط این دستور به تعداد bytes از باس SPI بایت دریافت می شود و در متغیر var قرار می گیرد .

### دستور SPIOUT

SPIOUT var , bytes

با این دستور به تعداد bytes داده var به باس SPI ارسال خواهد شد .

### دستور SPIMOVE

var=SPIMOVE(byte)

متغیر یا ثابت byte به باس SPI ارسال شده و همزمان داده دریافت شده از باس SPI در متغیر var جای می گیرد .

### دستور I2C SEND

این دستور داده را به خط یا باس I2C (2-WIRF) می فرستد .

I2csend Slave,Var

I2csend Slave ,Var , Bytes

### SLAVE

عدد ثابت یا متغیر نوع WORD, INTEGER , BYTE که حاوی آدرس Slave است .

### VAR

داده ای قصد ارسال به خط یا باس I2C را داریم که می تواند یک WORD,

INTEGER, BYTE یا عدد ثابت باشد .

### BYTES

مشخص کننده تعداد بایت دلخواه برای ارسال به خط یا باس سریال I2C است .

## دستورات I2CSTART, I2CSTOP, I2CRBYTE, I2CWBYTE

I2cstart

I2cstop

I2crbyte Var, Ack /Nack

### I2CSTART

که باعث ایجاد شروع (START CONDITION) در پرتکل ارتباطی I2C می شود .

### I2CSTOP

که باعث ایجاد پایان (STOP CONDITION) در پرتکل ارتباطی I2C می شود

### I2CRBYTE

یک بایت از خط یا باس I2C می گیرد . VAR. متغیری است که داده را از خط یا باس I2C می گیرد و ACK را برای دریافت بیش از یک بایت و NACK را زمانی که آخرین بایت را می خوانیم ، ایجاد می کنیم .

### I2CWBYTE

یک بایت به خط یا باس I2C می فرستد . VAL. متغیر یا ثابتی است که به خط یا باس I2C ارسال می شود .

این نوشته ها صرفا متعلق به وبلاگ ساعت خوش نمی باشد و ممکن است از سایت یا وبلاگ دیگری گرفته شده باشد.

در صورت نیاز به راهنمایی در مورد آموزش سایت یا آموزش خصوصی یا ساخت پروژه صنعتی یا دانشجویی با شماره زیر تماس

بگیرید

سعید حلوائی 09125307794