

دستورات AVR

رجیسترها و آدرس های حافظه

دستور SET

توسط این دستور می توان یک بیت را یک کرد .

Set Bit/pin

Set Var .X

Bit می تواند یک بیت یا یک SFR مانند PORTB.1 باشد و VAR متغیری از نوع داده BYTE INTEGER WORD یا LONG است. X برای BYTE می تواند 0 تا 7، 0 تا 15 برای WORD و برای LONG می تواند 0 تا 31 باشد .

دستور TOGGLE

این دستور مقدار منطقی یک پایه با یک بیت را معکوس میکند .

TOGGLE pin /Bit

Pin می تواند یک SFR مانند PORTB.1 و یا یک بیت باشد .

دستور RESET

توسط این دستور می توان یک بیت را صفر کرد .

Reset Bit/pin

Reset Var .X

Bit می تواند یک بیت و یا یک SFR مانند PORTB.1 باشد و Var متغیری از نوع داده BYTE INTEGER WORD یا LONG است و X برای BYTE می تواند 0 تا 7، 0 تا 15 برای WORD و برای LONG می تواند 0 تا 31 باشد .

دستور BITWAIT

BITWAIT X,SET/ RESET

توسط این دستور اجرای این برنامه تا زمانی که بیت X (SET یا 0) توسط این دستور اجرای این برنامه تا زمانی که بیت X (SET یا 0) شود و در خط جاری متوقف می ماند. در صورت TRUE شدن شرایط، اجرای برنامه از خط بعد ادامه می یابد. X می تواند یک بیت رجیستر داخلی مانند PORTB.Y باشد که Y میتواند بین اعداد 0 تا 7 تغییر کند.

دستور CPEEK

var = CPEEK(address)

از این دستور برای برگرداندن بایتی که در آدرسی از حافظه کدی ذخیره شده است استفاده می کنیم. با این دستور می توانید به رجیسترهای داخلی نیز دسترسی پیدا کنید. لازم به تذکر است که با این دستور شما نمی توانید در حافظه داخلی بنویسید.

دستور CPEEKH

با این دستور می توان بایت ذخیره شده در صفحه (PAGE) بالای حافظه کدی (FLASH MEM) میکرو MEGA103 با دیگر میکروها که دارای 128K حافظه است را خواند.

VAR = CPEEKH(address)

Address آدرس حافظه و محتوای آدرس در متغیر یک بایت var قرار می گیرد. CPEEKH(0) محتوای اولین بایت حافظه بالای 64K را بر می گرداند.

دستور LOADADR

LOADADR var , reg

با این دستور می توانید آدرس یک متغیر را در یک جفت رجیستر ذخیره کنید. VAR متغیری است که آدرس آن در متغیرهای دو بایتی X و Y و ذخیره می شود و REG رجیسترهای X و Y هستند. این دستور جز دستورات اسمبلی است و برای کمک به برنامه نویسان اضافه شده است.

دستور OUT

توسط این دستور می توان یک بایت به یک پورت سخت افزاری یا آدرس حافظه داخلی /خارجی ارسال کرد .

OUT address, value

Value به آدرس address که می تواند بین 0H تا FFFFH باشد فرستاده می شود. دستور OUT می تواند در تمام مکانهای حافظه AVR بنویسد. توجه کنید که برای address یک WORD تعریف شود. در ضمن برای نوشتن در مکان حافظه خارجی (XRAM) باید در محیط BASCOM در منوی (OPTION-COMPILER-CHIP), گزینه (EXTERNAL ACCESS ENABLE) را فعال کنید .

دستور INP

توسط این دستور می توان یک بایت از پورت سخت افزاری یا آدرس حافظه داخلی /خارجی خواند .

Var = INP(address)

محتوای آدرس address که می تواند بین 0H تا FFFFH باشد خوانده شده و در متغیر VAR قرار می گیرد. دستور INP می تواند از تمام مکانهای حافظه AVR بخواند. در ضمن برای خواندن از مکان حافظه خارجی (XRAM) باید در محیط BASCOM و در منوی (OPTION-COMPILER-CHIP), گزینه (EXTERNAL ACCESS ENABLE) را فعال کنید .

دستور PEEK

این دستور محتوای یک رجیستر را بر می گرداند .

Var = PEEK (address)

Address آدرس رجیستر های R0 تا R31 است که بین 0 تا 31 می باشد. محتوای رجیستر در متغیر var جای می گیرد. دستور (PEEK فقط می تواند محتوای

رجیستر ها را بخواند ولی () INP از تمام مکان های حافظه توانایی خواندن را داراست .

دستور POKE

با این دستور می توانیم یک بایت داده را در یکی از رجیستر ها بنویسیم .

POKE address ,Value

مقدار متغیر یا ثابت یک بایتی value در آدرس address که بین 0 تا 31 برای رجیستر های R0-R31 است نوشته می شود .

دستور VARPTR

این دستور آدرس یک متغیر را در مکان حافظه برمی گرداند .

Var = VARPTR(var 2)

آدرس متغیر var2 در مکان حافظه بدست آمده و در متغیر VAR قرار می گیرد .

دستور العمل های حلقه و پرش

دستور GOTO و JMP

GOTO lable

JMP lable

با این دستورات می توان به برچسب lable پرش کرد. برچسب lable باید با علامت (colon) پایان یابد و می تواند تا 32 کاراکتر طول داشته باشد. به خاطر داشته باشید زمانی که از دو lable هم نام استفاده شود کامپایلر به شما هشدار (Warning) می دهد. دستور RETURN برای برگشت از برچسب وجود ندارد.

دستور العمل DO - LOOP

فرم کلی دستور DO...LOOP به صورت زیر می باشد.

DO

Statements

LOOP [UNTIL expression]

دستور العمل statements تا زمانی که expression دارای ارزش TRUE یا غیر صفر است تکرار خواهد شد بنابراین این نوع حلقه حداقل یک بار تکرار می شود. DO-LOOP به تنهایی یک حلقه بینهایت است که با EXIT DO می توان از درون حلقه خارج شد و اجرای برنامه در خط بعد از حلقه ادامه یابد.

دستور العمل FOR- NEXT

فرم کلی دستور FOR...NEXT به صورت زیر می باشد.

FOR var = start TO end [STEP value]

Statements

Next var

VAR بعنوان یک کانتر عمل می کند که START مقدار اولیه و END مقدار پایانی است و هر دو می توانند یک ثابت عددی یا متغیر عددی باشند. VALUE مقدار عددی STEP (قدمها) را نشان می دهد که می تواند مثبت یا منفی باشد. در صورت

حذف کردن STEP VALUE کامپایلر به صورت پیش فرض مقدار یک را در نظر می گیرد .

نکته :نوشتن نام متغیر var بعد از NEXT الزامی نیست .

دستور العمل WHILE- WEND

دستور العمل WHILE- WEND تشکیل یک حلقه تکرار می دهد که تکرار این حلقه زمانی ادامه می یابد که عبارت به کار برده شده نتیجه FALSE را حاصل کند و یا دارای مقدار صفر شود . دستور العمل WHILE به صورت ورود به حلقه با شرط می باشد یعنی قبل از ورود به حلقه شرط حلقه تست می شود و در صورت TRUE بودن کنترل اجرای برنامه به حلقه وارد می گردد . بنابراین حلقه WHILE ممکن است در حالت هایی اصلا اجرا نشود یعنی حتی یک بار هم مراحل حلقه طی نشود .

While condition

Statements

Wend

بخش statements تا وقتی که حاصل Condition صفر یا FALSE نشده است تکرار خواهد شد .

دستور العمل IF

در کلیه حالت های زیر عبارت statement می تواند یک دستور العمل ساده یا چند دستور العمل مرکب باشد .

حالت: 0

If Expression Then statement

دستور العمل statement زمانی اجرا می شود که عبارت expression دارای ارزش TRUE باشد.

حالت 1 :

If Expression Then

Statement1

Else

Statement2

End If

در صورتی که عبارت expression دارای ارزش TRUE باشد دستور العمل statement2 اجرا می شود .

If Expression 1 Then

Statement1

Elseif [Expression 2 Then]

Statement 2

Else

Statement3

End If

در صورتی که عبارت 1 expression دارای ارزش TRUE باشد دستور العمل statement 1 اجرا خواهد شد. در صورتی که عبارت expression1 دارای ارزش FALSE ولی عبارت اختیاری 2 expression دارای ارزش TRUE باشد دستور العمل statement2 اجرا خواهد شد و در غیر این صورت یعنی در حالتی که هر دو عبارت 2 expression و expression1 دارای ارزش FALSE باشند دستور العمل statement اجرا خواهد شد. همچنین با دستور IF می توان یک یا صفر بودن یک بیت از یک متغیر را امتحان کرد .

IF bit =1 THEN or IF bit = 0 THEN

یا

Dim Var As Byte , Idx As Byte

Idx = 1

If Var .idx = 1 Then

‘if bit 0 of var is 1 then

Set portb. 0

Else

دستور العمل CASE

کنترل اجرای دستورات یک برنامه دارای ترتیب بالا به پایین است ولی در صورت نیاز می توان توسط دستورالعمل های انشعاب یا پرش جهت کنترل اجرای دستورات یک برنامه را تغییر داد یکی از این دستورالعمل های انشعاب یا پرش جهت کنترل اجرای دستورات یک برنامه را تغییر داد . یکی از این دستورات SELECT- CASE است که می توان یکی از چندین دستور را با توجه به مقدار ورودی اجرا کرد .

Select Case Var

CASE test1 : statement1

[CASE test2 : statement2]

CASE ELSE: statement3

End Select

اگر متغیر var با مقدار test1 برابر باشد statement1 اجرا می شود و سپس اجرا برنامه بعد از End Select ادامه می یابد و نهایتاً اگر متغیر var با هیچکدام از مقادیر test1 و test2 برابر نباشد statement3 اجرا می شود و سپس اجرای برنامه بعد از End select ادامه می یابد . شما می توانید به صورتهای زیر نیز متغیر را امتحان کنید :

اگر متغیر مورد نظر بزرگتر از دو باشد .

CASE IS>2:

و یا می توان محدوده ای را برای امتحان کردن در نظر گرفت :

اگر متغیر مورد نظر بین 2 تا 5 باشد .

CASE2 TO 5:

دستور EXIT

با این دستور می توانید فقط از یک ساختار یا حلقه خارج شوید و ادامه برنامه را بعد از ساختار یا حلقه ادامه دهید .

Exit a FOR...NEXT DO ...LOOP WHILE ...WEND SUB
...END SUB or FUNCTION...END FUNCTION.

EXIT FOR

EXIT DO

EXIT WHILE

EXIT SUB

EXIT FUNCTION

دستور العمل ON VALUE

با این دستور با توجه به مقدار متغیر می توان به توابع یا برچسبهای مختلفی پرش کرد

ON var [GOTO] [GOSUB] label 1 [, label2]

Var متغیر مورد نظر برای امتحان شدن که می تواند یک SFR مانند PORTD باشد

و..... LABEL1, LABEL2 برچسبهایی هستند که با توجه به مقدار var به آنها

پرش می شود . توجه داشته باشید زمانی که $var = 0$ باشد به اولین برچسب پرش

می شود .

نکته : زمانی که GOSUB استفاده می نمایید باید با دستور RETURN از برنامه

برگردید .

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 حلویی 09125307794