



وب سایت جامع الکترونیک ، برق و کامپیوتر

www.ir-micro.com

جزوه آزمایشگاه میکروپروسسور Z80
(قسمت دوم)

شما هم می توانید مقالات ، آموزش ها و ... خود را برای ما ارسال کنید تا با نام خودتان در سایت قرار داده شود .

Hamed_2_0_0_5@Yahoo.com
Hamed@Ir-micro.com

www.ir-micro.com

مراجع فارسی
میکروکنترلرهای PIC



آزمایش نهم

فراخواندن زیر برنامه

مفاهیم کلی

اصولاً برای ممانعت از حکرار دستور است مشخصی در برنامه های اصلی، از زیر برنامه ها استفاده میشود، به این صورت که این دستورات که کار مشخص را انجام میدهند در زیر برنامه گنجانیده میشوند و سپس در برنامه اصلی، زیر برنامه هر چند باری که لازم باشد فراخوانده میشود (CALL میشود) پس لازم است که کاری که زیر برنامه انجام میدهد، تاثیر آن روی شباتها و ورودیها و خروجیهای آنرا بروشنی بدانیم .

در این آزمایش، تعدادی زیر برنامه به شما معرفی میشود، بطوریکه با داشتن آنها میتوانید برنامه شبیه سازی یک سیستم کنترل ترموستات ساده را با آلام بنویسید .
توسط بیت ۵ از سوئیچهای باب 80H میتوان هیتر را روشن و خاموش نمود، هنگامیکه این بیت "1" منطقی باشد هیتر روشن و وقتی "0" منطقی باشد هیتر خاموش میباشد. توجه داشته باشید که کلید کنترل هیتر باید در چنین مواقعی روشن و باب 80H برای کنترل هیتر برنامه ریزی شده باشد .

استفاده از زیر برنامه ها

زیر برنامه هایی که در زیر لیست شده اند، برای نوشتن برنامه مربوط به آزمایش ترموستات مورد استفاده قرار میگیرند، لازم است شباهت وضعیت ورودی و خروجی و همچنین کار آنها و تاثیر که روی شباتها دارند، دقت کنید، اگر شباتی در زیر برنامه استفاده میشود، محتوای اصلی آن در صورتیکه مهم باشد باید قبل از ورود به زیر برنامه در محلی ذخیره شود و یا با استفاده از دستورات PUSH، POP در زیر برنامه محتوای آن حفظ گردد .

۱- نام : MHINI

آدرس : 1D00H

عملکرد: برنامه ریزی باب 80H برای کنترل هیتر و موتور

ورودی : —

خروجی : —

شباتهای استفاده شده : —

۲- نام : ANALOG

آدرس : 1D20H

عملکرد: تبدیل ورودی آنالوگ به دیجیتال (بمورت‌آبیتی)

ورودی : —

خروجی: شبات A شامل معادل دیجیتال ورودی آنالوگ بین 00 تا FFH

میباشد.

شباتهای استفاده شده : AF

-۳-

نام : DISPA

آدرس : 1D50H

عملکرد: محتوای شبات A را بمورت‌یک‌جفت digit هگزادسیمال در طرف

راست نمایشگرها، ظاهر میکند.

ورودی: انباره A شامل عددی است که قرار است نمایش داده شود.

خروجی: —

شباتهای استفاده شده: —

توجه: این زیربرنامه در طی زمان ۳ میلی ثانیه اجرا میگردد.

نام : ALARM

-۴-

آدرس : 1D90H

عملکرد: بلندگوی بوق میزند و به مدت ۱ ثانیه دو digit سمت راست نمایشگرها

چشمک میزنند.

ورودی: —

خروجی: —

رجیسترهای استفاده شده: AF

توجه این زیربرنامه در طی زمان یک ثانیه اجرا میگردد.

قبل از پرداختن به برنامه اصلی، این زیربرنامه‌ها را بدقت امتحان کنید.

آدرس	کدهگز	دستورات
1D00.....	F5.....	MHINI:PUSH AF
1D01.....	3EFF.....	LD A:0FFH

```
1D03.....D382.....OUT (ACON)·A
1D05.....3E1F.....LD A·1FH
1D07.....D382.....OUT (ACON)·A
1D09.....F1.....POP AF
1D0A.....C9.....RET
1D20.....C5.....ANALOG:PUSH BC
1D21.....0680.....LD B·80H
1D23.....48.....LD C·B
1D24.....CD3A1D.....LOOP:CALL BITCHK
1D27.....2003.....JR NZ·NEXT
1D29.....79.....LD A·C
1D2A.....A8.....*OR B
1D2B.....4F.....LD C·A
1D2C.....CB08.....NEXT:RRC B
1D2E.....3805.....JR C·DONE
1D30.....79.....LD A·C
1D31.....B0.....OR B
1D32.....4F.....LD C·A
1D33.....18EF.....JR LOOP
1D35.....79.....DONE:LD A·C
1D36.....D381.....OUT (81H)·A
1D38.....C1.....POP BC
1D39.....C9.....RET
1D3A.....79.....BITCHK:LD A·C
1D3B.....D381.....OUT (81H)·A
1D3D.....DB80.....IN A·(80H)
1D3F.....E608.....AND 08H
1D41.....C9.....RET
```

1D50.....F5.....DISPA:PUSH AF
1D51.....E5.....PUSH HL
1D52.....D5.....PUSH DE
1D53.....210019.....LD HL·OUTBUF
1D56.....CD6D06.....CALL HEX7SG
1D59.....2602.....MPLE*R:LD H·02
1D5B.....DD210019.....LD Ix·OUTBUF
1D5F.....1E01.....LD E·01
1D61.....DD7E00.....DIG:LD A·(Ix+0)
1D64.....D301.....OUT (01)·A
1D66.....7B.....LD A·E
1D67.....F6C0.....OR 0COH
1D69.....D302.....OUT (02)·A
1D6B.....CD821D.....CALL DELAA
1D6E.....3EC0.....LD A·0COH
1D70.....D302.....OUT (02)·A
1D72.....DD23.....INC Ix
1D74.....CB03.....RLC E
1D76.....25.....DEC H
1D77.....C2611D.....JP NZ·DIG
1D7A.....3E00.....LD A·00
1D7C.....D301.....OUT (01)·A
1D7E.....D1.....POP DE
1D7F.....E1.....POP HL
1D80.....F1.....POP AF
1D81.....C9.....RET
1D82.....C5.....DELAA:PUSH BC

```

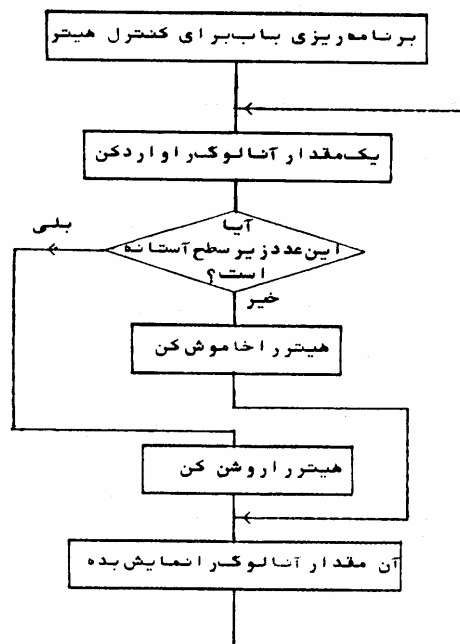
1D83.....018000.....LD BC·0080H
1D86.....0B.....DELAB:DEC BC
1D87.....78.....LD A·B
1D88.....B1.....OR C
1D89.....C2861D.....JP NZ·DELAB
1D8C.....C1.....POP BC
1D8D.....C9.....RET
1D90.....C5.....ALARM:PUSH BC
1D91.....E5.....PUSH HL
1D92.....0E06.....LD C·06H
1D94.....0640.....REPEAT:LD B·40H
1D96.....CD501D.....REP1:CALL DISPA
1D99.....05.....DEC B
1D9A.....C2961D.....JP NZ·REP1
1D9D.....C5.....PUSH BC
1D9E.....F5.....PUSH AF
1D9F.....216400.....LD HL·0064H
1DA2.....CDDE05.....CALL TONE1K
1DA5.....F1.....POP AF
1DA6.....C1.....POP BC
1DA7.....0D.....DEC C
1DA8.....C2941D.....JP NZ·REPEAT
1DAB.....E1.....POP HL
1DAC.....C1.....POP BC
1DAC.....C9.....RET

```

فلو چارت

یکتررموستات ساده به این صورت کار میکند که اگر درجه حرارت هیتر پایین تر از درجه حرارت سطح آستانه (threshold) باشد، هیتر روشن و اگر بالاتر باشد، هیتر خاموش میشود. اکثر ترموستاتها

دارای چند سطح آستانه می باشد، یعنی در درجه حرارت های مختلفی روشن و خاموش میشوند. ولی برای طراحی یک ترموستات ساده این موضوع در نظر گرفته نمیشود. به فلوجارت زیر توجه کنید :



برنامه اول

برنامه زیر بر مبنای فلوجارت بالا نوشته میشود. عدد سطح آستانه 80H در نظر گرفته شده که در خط سوم مورد استفاده قرار میگیرد :

دستورات	توضیحات
THERM:CALL MHINI	
LOOP:CALL ANALOG	
CP 80H	80 هگز سطح آستانه است
JP C·HON	
HOFF:PUSH AF	
LD A·00	
OUT (80H)·A	

```

POP AF
JP DIS
HON: PUSH AF
LD A, 20H
OUT (80H), A
POP AF
DIS: CALL DISPA    مقدار آنالوگ را نمایش میدهد
JP LOOP

```

تمرین:

الف) برنامه ترموستات ساده

- ۱- ابتدا برنامه اول را به صورت کد ماشین نوشته و آنرا از آدرس 1800H وارد حافظه کنید.
- ۲- مطمئن شوید که سوئیچهای مختلف، در موقعیتهای زیر هستند:

کلید کنترل موتور - خاموش

کلید کنترل هیتر - روشن

کلید کنترل ADC - روشن

کلید کنترل BAR-GRAPH - خاموش

سلکتور سه حالت - در وضعیت TERM.

باب 80H - همه سوئیچها در حالت "۱" منطقی

- ۳- حال برنامه اول را اجرا کنید.

روی نمایشگرها، دو digit، روشن شده و یک عدد گزرا نمایش میدهند، و در صورتیکه این عدد نمایش داده شده کوچکتر از 80H باشد، چراغ مربوط به هیتر روشن میباشد، و درجه حرارت هیتر همچنان در حال افزایش است و اوتیتر خود را به سطح آستانه برساند. هنگامیکه عدد روی نمایشگرها، به 80H رسید، چراغ هیتر خاموش میشود و در واقع هیتر خاموش شده و این نشان میدهد که برنامه بطور صحیح کار میکند، پس از اینکه مدتی از خاموش بودن هیتر گذشت و هیتر خنکتر از معادل 80H شد، مجدداً روشن میشود، و این اساس کار یک ترموستات ساده میباشد. هر چند که ولتاژی توسط سنسور حرارت اندازه گیری میشود، کاملاً بستگی به درجه حرارت واقعی هیتر دارد، ولی درجه حرارت هیتر بر حسب درجه سانتیگراد، فقط وقتی روی نمایشگرها، نمایش داده میشود که برنامه ای،

براین مبنا نوشته و اجرا شده باشد.

۴- میتوان برنامه را با یک سنسور داغ و یا یک سنسور سرد آغاز کرد، برای اینکار به ترتیب زیر عمل کنید: کلید [RESET] را برای توقف برنامه فشار دهید، اگر بیت پنجم از سوئیچ سبک 80H هنوز در حالت "۱" منطقی است پس هیتر روشن بوده است و اگر اینطور نبود، این بیت را در حالت "۱" قرار دهید تا هیتر روشن شود.

حال چهار دقیقه صبر کنید و سپس برنامه اول را اجرا کنید. چونکه در این مدت زمان هیتر داغ شده است، با اجرای برنامه، خاموش میشود و درجه حرارت آن بتدریج کاهش پیدا میکند تا اینکه به معادل 80H برسد، در این هنگام هیتر مجدداً روشن میشود. با این روش میتوان آلارمی را که در قسمت بعدی آزمایش میشود، امتحان نمود.

حال به سوالات زیر پاسخ دهید:

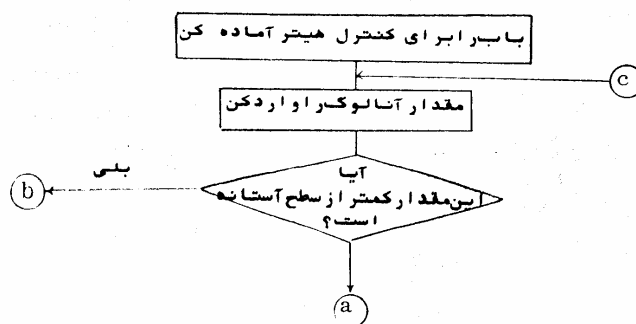
- ۱- بالاترین و پائین ترین درجه حرارت ضبط شده پس از راه اندازی سیستم کدام است؟ این ایده خوبی از واقعیت کار این نوع سیستم کنترلی را نشان میدهد.
- ۲- اگر سطح آستانه موثر شود چه اتفاقی می افتد؟ سعی کنید مقدار پائین تر و بالاتر را در برنامه امتحان کنید.

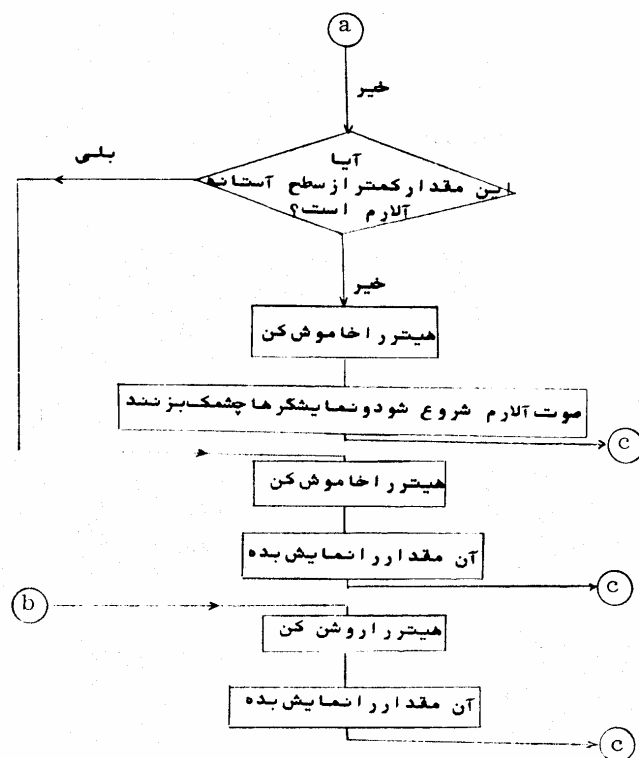
ب) ترموستات با آلارم

وقتی که یک میکرو کامپیوتر، پروسسی را کنترل میکند، میتواند چک کند که آیا این کار در یک محدوده صحیح انجام میگردد یا نه. برای مثال میتوان برنامه ترموستات ساده را توسعه دهیم تا برای وضعیتهای غیر نرمال آلارمی را تولید کند. البته این قسمت از برنامه را میتوان به تنهایی امتحان نمود.

فلو چارت

فلو چارت زیر برای ساختن ترموستات با آلارم طراحی شده است. در اینجا سه کار ممکن با دو سطح آستانه پیش بینی شده است که یکی برای کنترل نرمال کار و بقیه برای ایجاد آلارم میباشد:





کارهای زیر را با استفاده از سطح آستانه 80H برای کنترل آستانه، و 90H برای کنترل آلارم انجام دهید:

- 1- با استفاده از فلوچارت قبلی، برنامه‌ای مناسب بنویسید، و آن را از آدرس 1800H شروع کنید.
- 2- برنامه خود را چک کنید. بخصوص آدرسهای پرش و CALL را امتحان کرده و سپس در حافظه قرار دهید.

3- برنامه را اجرا کنید و ببینید که آیا هیتر را مانند برنامه قبلی کنترل میکند؟

4- حال کامپیوتر را RESET کنید و حدود چهار دقیقه هیتر را روشن بگذارید.

5- مجدداً برنامه را اجرا کنید اگر سنسور به اندازه کافی داغ شده باشد، موت آلارم شروع شده و نمایشگرها چشمک‌بزنند. در غیر این صورت، اگر درجه حرارت هیتر بالاتر از عدد معادل 90H می‌باشد، برنامه نوشته شده را دقیقاً چک کنید.

در حالیکه سنسور سرد میشود، درجه حرارت به تدریج به زیر معادل 90H میرسد و آلارم قطع میگردد و هنگامیکه الت حرارت به زیر معادل 80H رسید، هیتر روشن میشود.

تمرین

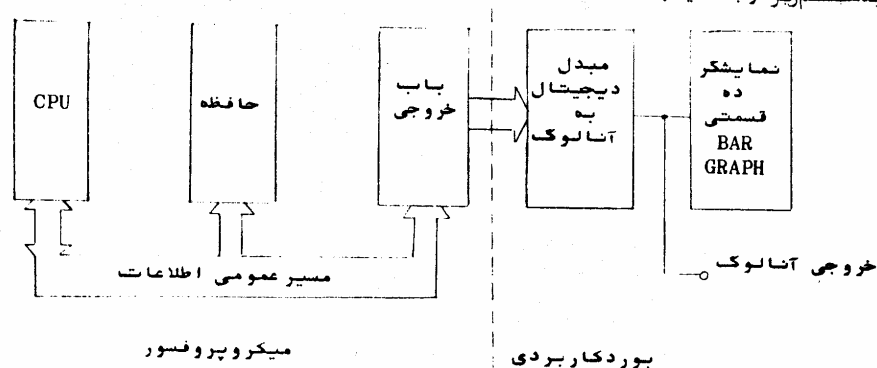
۱- برنامه ای بنویسید که شامل دو آلارم برای دو سطح آستانه مخصوص باشد مثلاً عدد 70H برای حدپائین درجه حرارت و عدد 80H برای کنترل نرمال کار و عدد 90H برای حدبالای درجه حرارت.

آزمایش دهم

تولید شکل موج

مفاهیم کلی

در این آزمایش، کار اصلی کامپیوتر، تولید معادل دیجیتال، ولتاژ آنالوگ مورد نظر، در هر لحظه میباشد. شکل موجهای متفاوتی را میتوان بر طبق اطلاعات ذخیره شده در حافظه کامپیوتر تولید نمود. به سیستم زیر توجه کنید:



اطلاعاتی که باید خارج شود در یک جدول حالت در حافظه کامپیوتر قرار دارد. در هر لحظه یک بایست آن از طریق باب خروجی خارج میشود. وبا استفاده از یک مبدل دیجیتال به آنالوگ به معادل آنالوگش تبدیل میشود. در این سیستم ترزیبی داده شده است که:

خروجی آنالوگ = اندازه عددباینری \times میلی ولت

برای مثال: عدد باینری 00000001 معادل ۰.۱ ولت

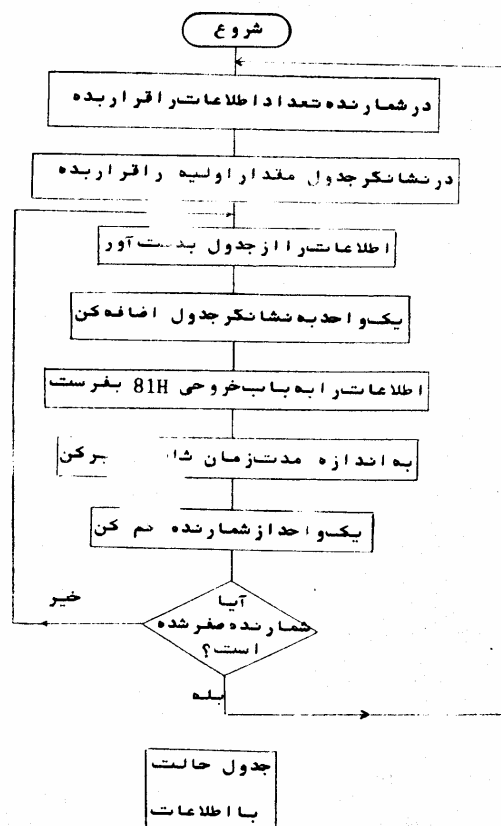
عددباینری 11111111 معادل ۲/۵۵ ولت میباشد.

وقتی که خروجی دیجیتال عوض میشود بلافاصله خروجی آنالوگ نیز جدید میشود. برای مثال از یک اندازه ولتاژ تا اندازه بعدی حدود ۱ میکروثانیه طول میکشد و به هر حال روی بارگراف این ولتاژ بقدری سریع عوض میشود که قابل مشاهده نمیشد، مگر اینکه برای هر ولتاژ جدید یک بار سوییچ بارگراف روشن و خاموش شود.

فلو چارت

میخواهیم برنامه ای بنویسیم که یک سری اطلاعات موجود در جدول را با فواصل زمانی ثابت به باب

خروجی بفرستد. در فلوچارت زیر یک شمارنده برای نگاه داشتن تعداد اطلاعاتی که باید به خروجی فرستاده شوند، در نظر گرفته شده است. و یک نشانگر جدول برای مشخص کردن آدرس اطلاعات بعدی در جدول استفاده میشود. اگر چه این فلوچارت با آنچه که در متن گفته شده یک تفاوت جزئی دارد ولی در عوض این امکان را به ما میدهد که بتوانیم برنامه ای برای تولید هر شکل موج بنویسیم.



میتوان برنامه زیر را برای فلوچارت بالا طراحی نمود. توجه کنید که آدرس شروع جدول حالت 1900H میباشد. ثابت B به عنوان شمارنده تعداد اطلاعاتی که باید به خروجی فرستاده شود، در نظر گرفته شده است و عدد 20H برای ۳۲ عدد اطلاعات در آن قرار گرفته است.

برنامه اول

دستورات کدهای آدرس

```

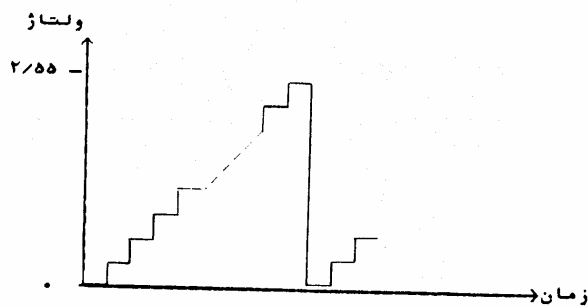
1800.....0620.....START:LD B,20H
1802.....210019.....LD HL,1900H
1805.....7E.....LOOP:LD A,(HL)
1806.....23.....INC HL
1807.....D381.....OUT (81H),A
1809.....110010.....DELAY:LD DE,1000H
180C.....1B.....DEL:DEC DE
180D.....7A.....LD A,D
180E.....B3.....OR E
180F.....C20C18.....JP NZ,DEL
1812.....05.....DEC B
1813.....C20518.....JP NZ,LOOP
1816.....C30018.....JP START

1900.....N1.....TABLE:DATA
1901.....N2.....DATA
-----
190F.....N32.....DATA

```

تمرین

۱- شکل موج دندانه‌اره پله‌ای زیر را در نظر بگیرید :

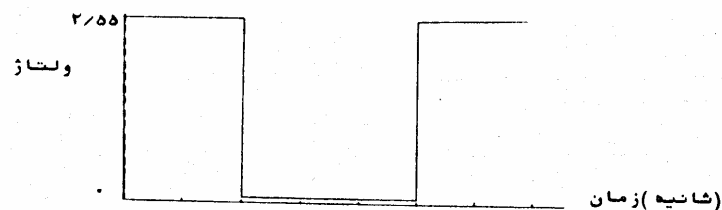


- اگر موج دندانه آره ای را با ۳۲ پله انتخاب کنیم ، اطلاعات جدول برابر 28*20*18*10*08*00 و ... خواهند شد. در این حالت همه ۳۲ عدد جدول را بنویسید.
- ۲- برنامه اول را از آدرس 1800H و اطلاعات را از آدرس 1900H وارد کنید.
- ۳- روی بوردمثلشن شوید که همه سوئیچها خاموش هستند به استثنای سوئیچ مربوط به بارگرافیکه باید روشن باشد.
- ۴- برنامه را اجرا کنید. روی بارگرافیکه اتفاقی می افتد؟ آیا همان اتفاقی می افتد که انتظار داشتید؟
- ۵- اثر تغییر زمان تاخیر (عدد موجود در زوج شبات DE) چه خواهد بود؟ سعی کنید برای زمانهای کوچک و زمانهای بزرگ آزمایش را تکرار کنید.
- ۶- اگر بخوانید بدون تغییر اطلاعات جدول، شکل موج به نصف مقدار ماکزیمم برسد (نصف ۲/۵۵ ولت) برنامه را چگونه اصلاح میکنید.
- ۷- چگونه میتوانید برنامه را تمحیح کنید بطوریکه بدون تغییر در جدول، شکل موج به جای حرکت از 0 تا ۲/۵۵ برعکس یعنی از بالای پاشین حرکت کند؟

تمرین

تمرینهای زیر همگی بر اساس یک برنامه مشابه ولی با جدول اطلاعات متفاوت میباشد.

۱- جدول اطلاعاتی برای تولید شکل موج زیر تشکیل دهید:



- ۲- برای تولید شکل موج سینوسی ، ۳۶ پله با فواصل ۱۰ درجه ای را باید ایجاد کرد. سعی کنید در برنامه خود چنین شکل موجی را تولید کنید.
- مقادیر اطلاعات شکل موج سینوسی را به آسانی میتوان ابتدادر مبنای دسیمال (ده تایی) محاسبه کرده و سپس آنرا به مبنای هگزادسیمال تبدیل کنید و بعد وارد حافظه کامپیوتر کنید. توجه کنید که شکل موج باید در اطراف یک مبدا، کاملاً متقارن باشد. برای محاسبه مقدار دسیمال هر عدد مورد نیاز میتوانید از فرمول زیر استفاده کنید:

$$\text{عدد دسیمال} = 128 + (127 \times \sin N)$$

N بر حسب درجه و از صفر تا ۳۵۰ میباشد.
باتفیییر اطلاعات جدول، و طول تاخیر بین هر حالت، میتوان شکل موجهای متفاوتی را تولید کرد.

آزمایش یازدهم

روندهائی با تاخیرهای متغیر

مفاهیم کلی

استفاده از روندهائی که دارای پله هائی با طول زمانی متفاوت هستند، کاربرد فراوانی دارد. یکی از این روندها، کنترل نوری ترافیک می باشد. در شکل ساده این نوع کنترل، دو مجموعه از نور، میتوانند، عبور کردن و یا ایستادن را مشخص کنند ولی در سیستمهای پیچیده، تعداد بیشتری کنترل را در دست میگیرند.

به جای فرستادن اطلاعات به خروجی برای ایجاد نور میتوان از صوت نیز استفاده نمود برای اینکار میتوان از یکی از زیر برنامه های مونیتور که ایجاد صوتی ساده میکند، کمک گرفت. در دل همه این روندها با تاخیرهای متفاوت، یک زیر برنامه تاخیر وجود دارد که دارای مینیمم زمانی معادل کوچکترین پریود زمانی است که باید اندازه گیری شود. بطور نمونه امیلی شانیه ،،، امیلی شانیه ،،، امیلی شانیه یا یک شانیه که این بستگی به کاربرد برنامه دارد. کل زمان تاخیری که تولید میشود بستگی به پارامترهائی دارد که در جدول قرار داده شده است.

برنامه اول

زیر برنامه زیر میتواند به عنوان اساس ایجاد تاخیرهای متفاوت، بکارگرفته شود. در حلقه اصلی برنامه ، از زوج ثبات DE به عنوان شمارنده استفاده شده است. برنامه از آدرس 18A0H شروع میشود.

آدرس	کدهگز	دستورات
18A0.....	11nn.....	VARDEL:LD DE,nn
18A3.....	1B.....	VLOOP:DEC DE
18A4.....	7A.....	LD A,D
18A5.....	B3.....	OR E
18A6.....	C2A318.....	JR NZ,VLOOP
18A9.....	0D.....	DEC C
18AA.....	C2A018.....	JP NZ,VARDEL
18AD.....	C9.....	RET

مقادیر محاسبه برای nn برای ایجاد تاخیرهای متفاوت به صورت زیر میباشد:

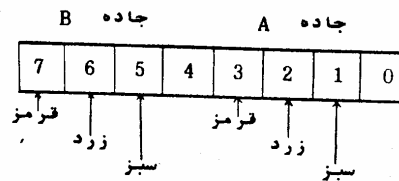
1 μs --- 004AH

10 ms --- 02D8H

100 ms --- 1D10H

500 ms --- 9150H

نورهای چراغیکبه بابخروجی 81H به ترتیب زیرمتمل شده اند :



بیت صفرو بیت چهار برای محافظت عابر پیاده در نظر گرفته شده اند یعنی فقط در مواقعیکه وسایط نقلیه در حال توقف هستند، عابر میتواند عبور کند، که بیت 0 برای کنترل عبور و مرور پیاده در درجاده A و بیت 4 برای جاده B در نظر گرفته شده است.

برنامه دوم

برنامه اصلی به صورت زیر میباشد :

```

START: LD B, n          تعداد حالات = n
      LD HL, 1A00H      شروع جدول حالت
      REP: LD A, (HL)
      INC HL
      OUT (81H), A
      LD C, (HL)
      INC HL
      CALL VARDEL
      DEC B              یک واحد از شمارنده کم میشود
      JP NZ, REP
      JP START
  
```

VARDEL: برنامه اول

جدول حالت در حافظه از Tدرس 1A00H شروع میشود. در روندهای بعدی از یک تاخیر اصلی 500ms استفاده کنید.

تمرین

الف) کنترل نوری ترافیک

- ۱- جدولی تشکیل دهید که برداشت اطلاعات از آن وضعیت‌های زیر را بوجود بیاورد:
- توجه داشته باشید که اطلاعات این جدول، مربوط به ایجاد تاخیر میباشد که هر حالت را دو بایت از جدول بوجود میآورند.

۱ شانه	قرمز
۱ شانه	قرمز
۴ شانه	قرمز و پیاده
۵ شانه	قرمز
۵ شانه	قرمز و پیاده
۵ شانه	قرمز
۵ شانه	قرمز و پیاده
۵ شانه	قرمز
۵ شانه	قرمز و پیاده
۱ شانه	قرمز
۱ شانه	قرمز و زرد
۸ شانه	سبز
۱ شانه	زرد

- ۲- برنامه دوم را بصورت کد ماشین نوشته و در آدرس 1800H حافظه وارد کنید.

- ۳- اطلاعات مربوط به جدول را نیز در حافظه وارد کنید.

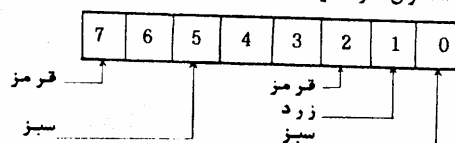
- ۴- همه سوشیچهای روی بورد را خاموش کرده و سپس برنامه را اجرا کنید.

ب) کنترل عبور و مرور بصورت دستی

کنترل عبور و مرور بصورت دستی شبیه بسیاری از روشهای کنترل نوری ترافیک میباشد به استثنای اینکه این نوع کنترل بطور دستی تریگر شده و بطور ممتد انجام نمیگیرد.

چراغهای روی بورد با تفاوضی جزئی نسبت به قبل به صورت زیر معرفی میشوند:

کنترل ترافیک کنترل عبور و مرور پیاده



پس از تریگر شدن، روند رویدادها به ترتیب زیر میباشد:

۵ شانه	ترافیک سبز و پیاده قرمز
۲ شانه	ترافیک زرد و پیاده قرمز
۱ شانه	ترافیک قرمز و پیاده قرمز
۶ شانه	ترافیک قرمز و پیاده سبز
۵ شانه	ترافیک زرد و پیاده سبز
۵ شانه	همه خاموش

ترافیک زرد و پیاده سبز	۵/۵ ثانیه
همه خاموش	۵/۵ ثانیه
ترافیک زرد و پیاده سبز	۵/۵ ثانیه
همه خاموش	۵/۵ ثانیه
ترافیک زرد و پیاده سبز	۵/۵ ثانیه
همه خاموش	۵/۵ ثانیه
ترافیک سبز و پیاده قرمز	۵/۵ ثانیه

آخرین حالت تا تریگر بعدی باقی خواهد ماند.

برای تریگر کردن میتوان از کلید USER روی صفحه کلید میکرو پروفسور استفاده کرد. کلید USER مستقیماً به بیت ۰ از باب 0 متصل است و هنگامیکه فشار داده شود برابر صفر منطقی و در حالیکه آزاد باشد یک منطقی را تولید میکند.

حال کارهای زیر را انجام دهید:

۱- جدول حالتی برای عبور و مرور دستی بنویسید، بطوریکه هر حالت خروجی به دنبال کد زمانی مربوطه اش در جدول قرار بگیرد.

۲- برنامه اول را طوری تصحیح کنید که منتظر فشار کلید USER بماند و پس از اینکه کلید زده شد، قسمت اصلی برنامه آغاز شود.

۳- برنامه را بصورت ماشین کد نوشته و از آدرس 1800H وارد حافظه کنید و همچنین جدول را از آدرس 1A00H شروع کنید.

۴- برنامه را اجرا کنید.

حال کلید USER را فشار دهید و چراغهای روی بورد را مشاهده کنید.

آزمایش دوازدهم

روندهای شرطی

مفاهیم کلی

روندهای شرطی کاربرد بسیار وسیعی در صنعت دارد. این نوع کنترل مفهومی از فیدبک را در خود دارد. نرم افزار کامپیوتر و جد اول حالت بسیار شبیه آنچه که در قبل بررسی شده میباشد. شاید آشناترین مثال این روند، ماشین لباسشویی اتوماتیک باشد. در این آزمایش، با استفاده از سوئیچها و چراغهای موجود روی بورد، یک ماشین لباسشویی را شبیه سازی میکنیم.

شبیه سازی ماشین لباسشویی

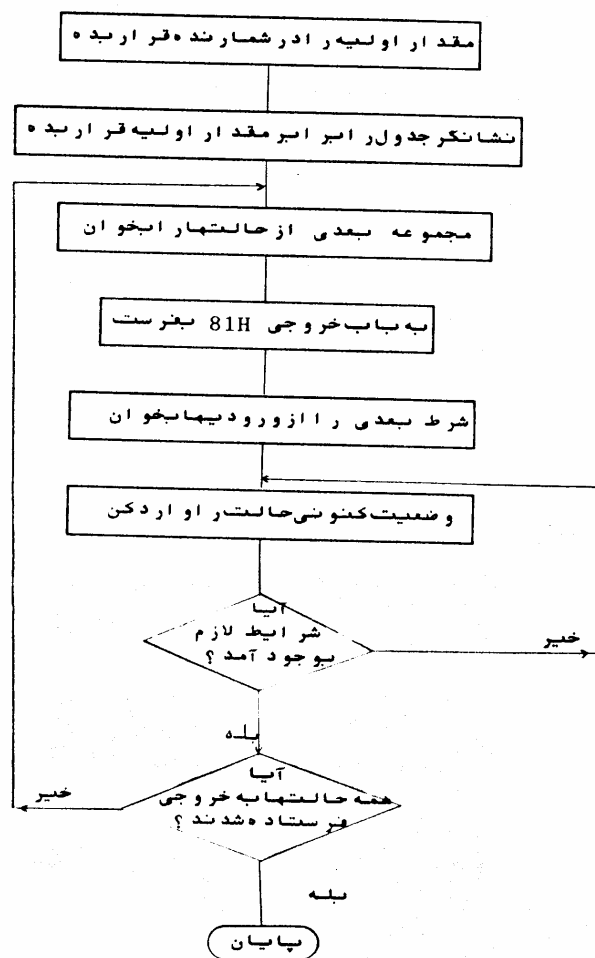
بابهای 81H و 80H به ترتیب به عنوان خروجی و ورودی مورد استفاده قرار میگیرند و هر دو دارای حالت فعال یک هستند.

باب 81H		باب 80H	
Bit	0	Bit	0
	در بسته		درجه حرارت صحیح
۱	شروع زمانگیری	۱	سطح پر
۲	پمپ روشن	۲	سطح خالی
۳	شیر آب سرد باز	۳	۵ دقیقه زمان کامل
۴	" گرم "	۴	" ۵ "
۵	هیتر روشن	۵	در بسته
۶	گردش تند استوانه	۶	استفاده نشده - صفر منطقی
۷	" آهسته "	۷	" "

ترتیب کار شستشو به ترتیب زیر میباشد. توجه داشته باشید که در تمام طول سیکل در به ماشین باید بسته باقی بماند.

قدم	عملکرد	شرایط تعیین کننده
۱ -	منتظر میماند در بسته و	در بسته است (سطح خالی است)
۲ -	شیر گرم باز میشود	سطح پر میشود
۳ -	هیتر روشن میشود	
۴ -	گردش استوانه آهسته میشود	۵ دقیقه زمان کامل
	تایمر شروع میکند	سطح خالی میشود
۵ -	پمپ روشن میشود	" پر "
۶ -	شیر سرد باز میشود	
۷ -	دور آهسته میشود	۵ دقیقه زمان کامل
	شروع کار تایمر	سطح خالی میشود
۸ -	پمپ روشن میشود	۲ دقیقه زمان کامل
۹ -	پمپ روشن است	(و سطح خالی است)
	دور تند میشود	
	شروع کار تایمر	همه ورودیها در حالت صفر هستند به
۱۰ -	در باز میشود	استثنای ورودی سطح خالی

فلو چارت



تمرین

- ۱- ابتدا مراحل ششگانه را بصورت یکسری از حالت‌های لاجیکی در جدول حافظه قرار دهید.
- توجه داشته باشید که اطلاعات را بطریقی در حافظه وارد کنید که اطلاعات کنترل و سایل، به عنوان اولین بایت هر حالت و پس از آن هر بایت شامل حالت‌های ورودی شرطی (شرایط تعیین کننده) قرار بگیرد. مقادیر باینری را بصورت هگزادسیمال بنویسید.
- ۲- برنامه اختصاصی بر حسب ماشین کد بر اساس فلو چارت بالا بنویسید. برنامه را از آدرس 1800H شروع کنید.
- ۳- برنامه و جدول مربوطه را در حافظه از آدرس شروع 1800H و 1A00H (به ترتیب) وارد کنید.
- ۴- همه سوئیچهای باب 80H را در وضعیت " 0 " منطقی قرار داده و برنامه را اجرا کنید.
- ۵- شرایط لازم ورودی را برای اولین حالت روی سوئیچهای باب 80H ایجاد کرده و سپس حالت خروجی تولید شده را امتحان کنید.
- در مرحله بعدی سوئیچهای باب ورودی 80H را در وضعیتهای دیگر ورودی که لازم باشد قرار دهید و خروجی ناشی از آن را امتحان کنید. هنگامیکه همه ورودیها با موفقیت داده شد، برنامه باید متوقف (HALT) شود.

آزمایش سیزدهم

اسکان کردن صفحه کلید

مفاهیم کلی

صفحه کلید میکروپروفسور دارای ۳۶ عدد کلید می باشد. هر چند شکل فیزیکی آن ۹۰ در ۴۰ میلی باشد ولی از نظر مداری کمی تفاوت دارد. چهار عدد کلید واقع در یک ستون دست چپ کلید، جداگانه با سیستم ارتباط دارند و ۳۲ کلید باقی مانده، تشکیل یک ماتریس 6×6 می دهند. هر کلید در نقطه تقاطع یک ستون عمودی و یک ردیف افقی قرار دارد. وقتی که کلیدی فشار داده میشود، یک ستون و یک ردیف مربوط به آن کلید به یکدیگر وصل میشوند. اگر هیچ کلیدی زده نشود، هیچ اتصال و وجود نداشته و خطوط ردیفها توسط مقاوماتهای به ولتاژ هولت مرتبط میشوند. وجود این مقاوماتها (Pull up resistors) ما را مطمئن میکند که هنگامیکه کلیدی زده نشده است خطوط مربوط به ردیفها حتما در حالت "۱" منطقی میباشند. در دیاگرام مداری، سرهای PC5-PC0، شش خط پائین باب 02 خروجی هستند و سرهای PA5-PA0 به عنوان ورودی استفاده میشوند، که این شش خط نیز، ۶ خط پائین باب شماره 00 می باشد.

در صفحه بعد شما مدار صفحه کلید و نمایشگرها را مشاهده میکنید.

نرم افزار اسکان صفحه کلید، هر کلید را بنوبت چک میکند، باین صورت که هر بار تنها به یکی از خطوط PC0 تا PC5 مفر و بقیه یک فرستاده میشود. برای شروع ابتدا خط PC0 مفر میشود. هر خط ورودی PA0 تا PA5 نیز دا چک میشود، که مفر بودن هر کدام از این خطوط دلالت بر فشردن شدن کلیدی (از ۳۲ کلید ذکر شده) واقع روی تقاطع این ردیف و ستونی که اخیرا مفر بوده است، دارد. پس از انجام شدن هر تست، به شماره ده یک واحد اضافه میشود که نشانه ای برای کلید فشردن شده در آن لحظه می باشد. در واقع شماره ده، در هر لحظه، شماره کلید زده شده را در خود حفظ کرده است. پس از آنکه به اولین ستون مفر فرستاده شد، ترتیبی داده شده است که این مفر منطقی به ستون بعدی منتقل شود و در واقع هر بار از PC0 تا PC5 حرکت کنند. پروسه مفر شدن ستونهای یکی پس از دیگری تست ورودیهای PA0 تا PA5 همچنین ادامه میابد، بطوریکه در هر لحظه ای که کلیدی فشرده شود، شماره کلید مشخص و با استفاده از آن کار مربوط به کلید انجام میپذیرد.

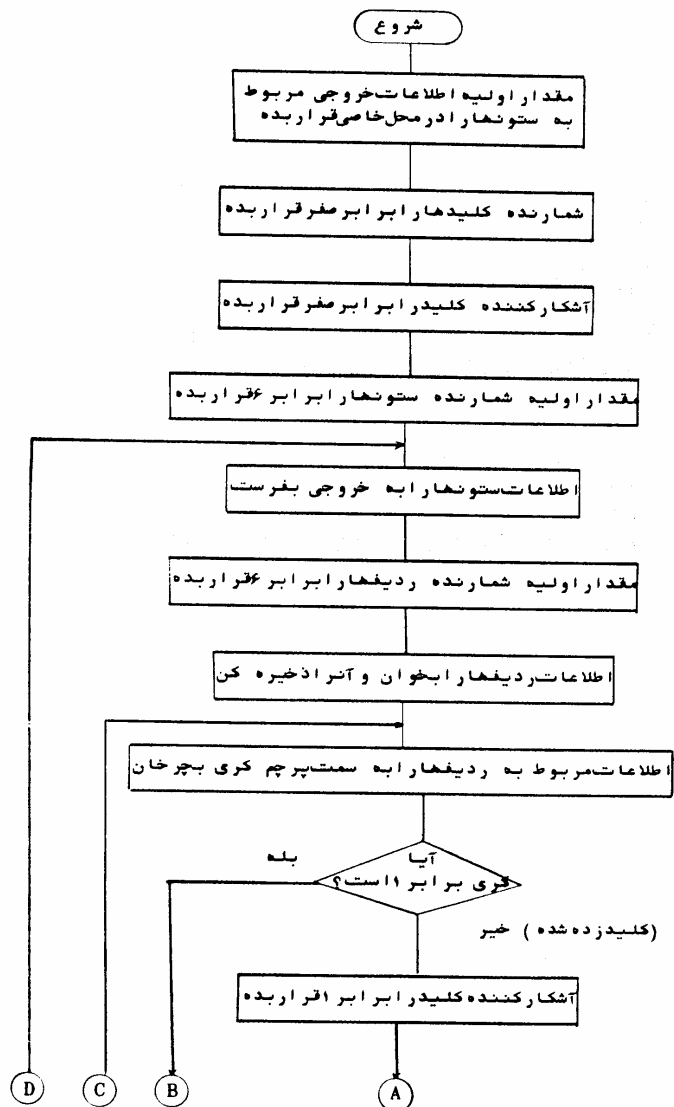
فلو چارت

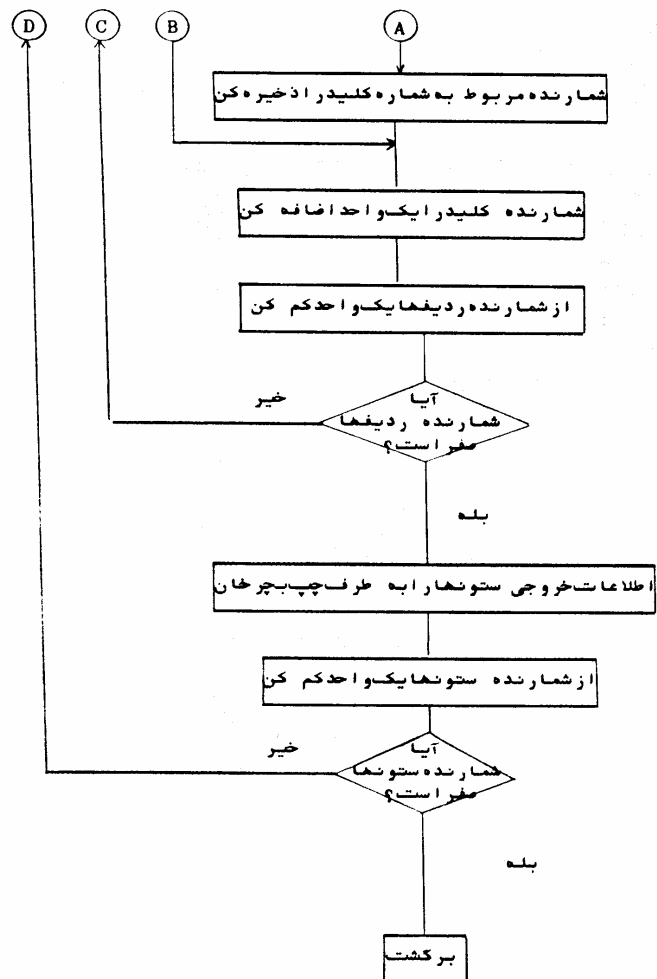
فلو چارت زیر روند اسکان صفحه کلید را نشان میدهد. سه عدد شماره ده، یکی برای شمارش ردیف و دیگری برای شمارش ستون و سومی برای شماره کلید استفاده شده است. برای مشخص کردن



اینکه کلیدی‌ها داده شده و یا نه و در واقع برای تشکیل آشکار کننده آزاد بودن و یا نبودن کلید می‌توان از یک پرچم و یا یک بیت از شباتی استفاده نمود.

همه برنامه به عنوان زیر برنامه ای که می‌تواند خود قسمتی از برنامه های دیگر باشد، در نظر گرفته می‌شود.





برنامه اول

شباتهای زیر باینمورت مورد استفاده قرار میگیرند :

شبات B - شمارنده ردیف

شبات C - شمارنده کلید

شبات D - اطلاعات ردیف

	شبات E - اطلاعات خروجی ستونها
	شبات H - شمارنده ستون
	شبات L - آشکار کننده کلید
	کد کلید زده شده - 1A00H آدرس حافظه
SCANKY:LD E*FEH	بایتی 11111110
LD C*0	شمارنده کلید = 0
LD L*0	هیچ کلیدی زده نشده است
LD H*6	تعداد ستونها = 6
KCOL:LD A*E	
OUT (02)*A	اطلاعات ستونها به خروجی فرستاده میشود
NOP	برای استفاده های بعدی
LD B*6	6 ردیف
IN A*(00)	اطلاعات ردیفها خوانده میشود
LD D*A	و در شبات ذخیره میشود
KROW:RR D	شبات D به سمت راست چرخیده و بیت صفر آن داخل کری قرار میگیرد
JP C*NOKEY	تست کردن کری
LD L*01	زده شدن کلید آشکار میشود
LD A*C	
LD (1A00H)*A	شماره کلید در حافظه ذخیره میشود
NOKEY:INC C	یک واحد به شمارنده کلید اضافه میشود
DEC B	یک واحد از شمارنده ردیف کم میشود
JP NZ*KROW	تست ردیف بعدی
RLC E	اطلاعات ستونها یک بیت به سمت چپ میچرخد
DEC H	یک واحد از شمارنده ستون کم میشود
JP NZ*KCOL	تست ستون بعدی
RET	برگشت

تمرین

۱- ابتدا برنامه اول را به صورت کد ماشین بنویسید و آن را در حافظه میکرو پرو سورو وارد کنید. چونکه این یک زیر برنامه میباشد آن را از آدرس شروع 1900H وارد کنید. در این صورت این امکان را دارید که برنامه اصلی خود را از آدرس 1800H شروع کنید.

۲- یک برنامه اصلی ساده ای لازم است که زیر برنامه SCANKY در آن فرا خوانده شود و با فشار داده شدن کلید، متوقف شود:

```
MAIN:CALL SCANKY
```

```
LD A+L
```

```
CP 01
```

```
JP NZ+MAIN
```

```
HALT
```

این برنامه را از آدرس 1800H در حافظه وارد کنید.

۳- قبل از اجرای برنامه، با استفاده از پروبالایی، میتوانید در حالت های مختلف، نقاط مختلفی از مدار صفحه کلید را مورد بررسی قرار دهید.

برای این کار پروبال را در سوکت مخصوص خود قرار دهید و برنامه را در نقطه ای مطمئن متوقف کرده و خطوط صفحه کلید را توسط پروبال امتحان کنید.

اگر با استفاده از نقطه توقف (Break Point) برنامه متوقف شود، برنامه مونیتور کنترل صفحه کلید را بدست گرفته و منتظر زده شدن کلید میماند و در واقع همه سیگنال هایی که مانتظر آن هستیم ماسک شده و برای مقابله تست نمیباشد.

روش دیگر متوقف کردن برنامه، استفاده از دستور HALT است. بنابراین در برنامه دستور HALT را به جای NOP جایگزین کنید (با کد 76). با این صورت برنامه پس از دستور OUT متوقف میشود و آنگاه شما میتوانید خطوط صفحه کلید را امتحان کنید.

۴- حال قبل از اجرای برنامه بر شماره ۱۴ از PPI 8255 را پیدا کرده و پروبال روی آن قرار دهید. در این صورت هر دو نور LOW و HIGH روشن شده و نور PULSE چشمک میزند که این نشان دهنده این موضوع است که صفحه کلید توسط برنامه مونیتور در حال تست شدن میباشد.

۵- برنامه موجود در آدرس 1800H را اجرا کنید. نور HALT باید تقریباً بلافاصله روشن شود که این نشان میدهد که اولین قسمت برنامه اجرا شده و دستور HALT به وقوع پیوسته است. حال میتوانید خروجی های PC0 تا PC5 از PPI را توسط پروبالایی چک کنید.

دقت داشته باشید که شماره این سرهای ترتیب نمیباشد. اگر برنامه شما صحیح باشد شماره ۱۴ یعنی PC0 باید "0" منطقی بوده و بقیه سرها از PC1 تا PC5 باید یک باشند. نتیجه کار خود را در جدول زیر وارد کنید.

۶- در حالیکه اسکان صفحه کلید متوقف است، پروبر را روی پایه شماره ۴ از PPI قرار دهید. این خط PA0 از خطهای ورودی صفحه کلید است. اگر بالاترین کلید واقع در آخرین ستون دست راست صفحه کلیدی معنی کلید [3] زده شده باشد، سر شماره ۴ باید صفر باشد. حال سرهای شماره ۱، ۲، ۳ را امتحان کرده و وضعیت کلیدهای واقع در آخرین ستون دست راست را مشخص کنید.

۷- برای اجرای بقیه برنامه به صورت زیر اقدام کنید:

کلید [MONI] و سپس کلید [GO] برای ادامه حلقه برنامه بزنید. وقتی که بقیه برنامه اجرا میشود، نور HALT مجدداً روشن میشود. خروجیهای PC5-PC0 را با استفاده از پروب لاجیک امتحان کرده و وضعیت آنها را در جدول زیر وارد کنید.

آیا نتیجه همانطور که انتظار داشتید میباشد؟

۸- با فشار کلیدهای [MONI] و [GO] به ترتیب، پس از هر بار که HALT میشود، یکبار، و در واقع یک دور برنامه را اجرا کنید و جدول زیر را کامل کنید. تا وقتی که همه کارها تمام نشده است، هیچ کلیدی را فشار ندهید.

تعداد دفعاتی که برنامه را دور زده اید	PC5 (12)	PC4 (13)	PC3 (17)	PC2 (16)	PC1 (15)	PC0 (14)
---------------------------------------	-------------	-------------	-------------	-------------	-------------	-------------

۱
۲
۳
۴
۵
۶
۷
۸

اگر خروجیهای بدست آمده، با آنچه که در هر نقطه انتظار داشته اید فرق میکند، با دقت برنامه را چک کنید.

۹- میتوان هنگامیکه برنامه در حال اسکان کردن صفحه کلید است، چک کرد که آیا در این حالت کلید را آشکار میکند؟

دستور HALT را برداشته و همان NOP را در زیر برنامه قرار دهید. حال برنامه را اجرا کنید. کلید [GO] را خیلی سریع بزنید زیرا در غیر این صورت، سیستم زده شدن کلید [GO] را در نظر گرفته و در برنامه اصلی HALT میکند.

۱۰- یکی از کلیدهای [RESET] و یا [MONI] را برای برگرداندن کنترل به مونیتور فشار دهید و سپس محتوای آدرس 1A00H را امتحان کنید. در این صورت باید کلید زده شده در این آدرس قرار داشته باشد که میتواند یکی از شماره های 00 تا 23H باشد.

۱۱- دوباره برنامه را اجرا کنید و کلید دیگری را فشار دهید؛ سپس کد کلید واقع در آدرس 1A00H را چک کنید.

۱۲- کد کلید برای کلیدهای 9، A و STEP، TAPEWR و MOVE چیست؟

۱۳- در دیگرام مدار صفحه کلید، چهار کلید از ماتریس وجود دارند در حالیکه برنامه مکان آنها را تست میکند، کد کلید برای این چهار کلید کم شده چه میباشد؟

تمرین

در تمرینهای زیر، شما از زیر برنامه اسکان صفحه کلید استفاده کرده ولی مجبورید که برنامه اصلی را عوض کنید.

۱- در میکروپروفسور هنگام فشار هر کلید، صدای بوقی نیز تولید میشود. برنامه ای بنویسید که پس از فشار یک کلید بلافاصله بوقی را با فرکانس 1KHZ و در مدت زمانی ۱/۱ ثانیه ایجاد کند. میتوانید برای زیر برنامه بوق از زیر برنامه TONE1K موجود در برنامه مونیتور استفاده کنید.

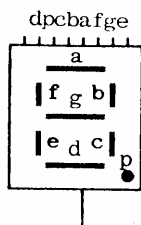
۲- برنامه قبلی را طوری اصلاح کنید که فقط وقتی کلید [STEP] زده میشود، چنین بوقی تولید شود.

آزمایش چهاردهم

مالتی پلکسینگ نمایشگرها

مفاهیم کلی

در کامپیوتر میکروپروسسور، نمایشگرها و صفحه کلید هر دو به آی سی PPI 8255 متصل هستند، که شامل سه بابابیتی مستقل از هم میباشد. هر پایه باباب C برای نمایشگرها و صفحه کلید استفاده میشود. البته صفحه کلید مستقیماً ولى نمایشگرها از طریق بافرهاى که برای ایجاد نور کافى روی LED ها، جریان بالائی را بوجود میاورند، به بابابها متصل میشود. نمایشگرها، شامل ۶ عدد هفت قسمتی است که هر کدام به صورت کاتد مشترک به یکدیگر مرتبط میشوند.



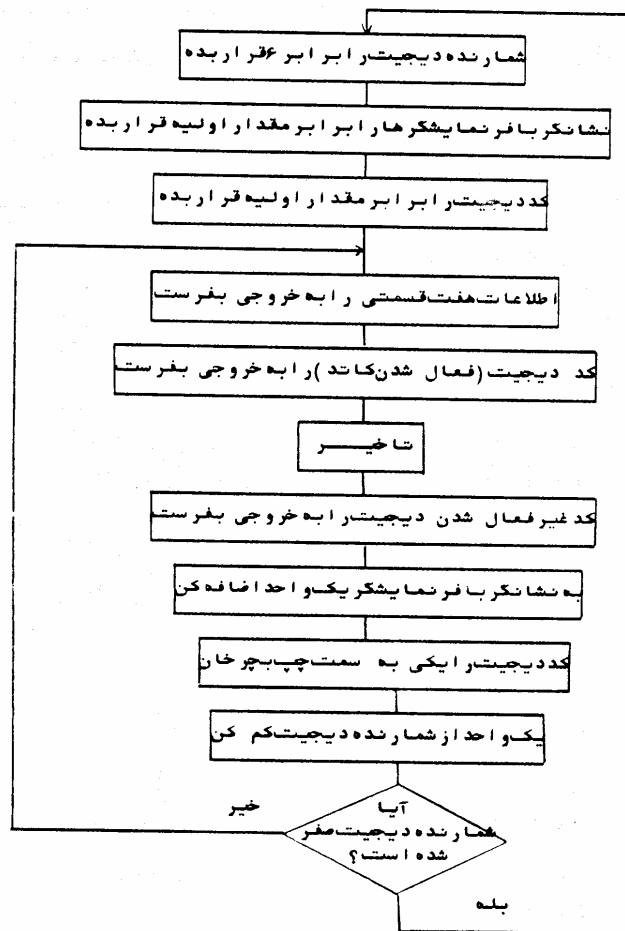
شکل ۱-۱۴) نمایشگرهای هفت قسمتی

هر قسمت با فرستادن "۱" منطقی به آن دو قرار دادن "۰" منطقی روی کاتد دیجیتال مربوطه اش روشن میشود. هر هفت خط همه هفت قسمتی ها به صورت موازی به باباب B از PPI متصل هستند. بنابراین هر اطلاعاتی که به باباب B فرستاده میشود به ۶ عدد هفت قسمتی رسیده و بسته به اینکه کاتد کدامیک فعال (۰) باشد، بر مبنای اطلاعات موجود در باباب B همان دیجیتال روشن میشود. با توجه به اینکه کاتدها از طریق معکوس کننده (INVERTER) به باباب C وصل میشوند، برای فعال کردن کاتد هر کدام، باید "یک" منطقی از طریق باباب فرستاده شود. نرم افزار مالٹی پلکسینگ شبیه آنچه در بالاتر توضیح داده شد، میباشد. در واقع، در هر لحظه فقط یک دیجیتال روشن میشود و برای هر هفت قسمتی، کدام مناسب، از طریق آندهای آنها (باباب B) فرستاده میشود و فقط کاتد همان دیجیتال فعال است. پس از زمان تاخیر کوتاه، کاتد غیر فعال میشود. پس هر دیجیتال به نوبت روشن میشود و در آن لحظه بقیه خاموش هستند. منتهی این کار بقدری سریع صورت میگیرد که همزمان بنظر می آید. چونکه روی هر کدام از ۶ عدد نمایشگر، ۶ حرف مختلف میخوانیم نمایش داده شود پس باید آنها بطریقی در حافظه کامپیوتر ذخیره شوند که این ناحیه از حافظه بافر نمایشگرها نامیده میشود. ویکی از زوج شباهت به عنوان نشانگر این بافر مورد استفاده قرار میگیرد. توجه کنید که منظور از دیجیتال همان یک هفت قسمتی میباشد ولی هرگاه صحبت از اطلاعات مربوط به آندها است

(p.g.f.e.d.c.b.a) کلمه هفت قسمتی و یا قسمتها و هرگاه منظور کاتد آنماست کلمه دیجیت بکار

میرود.

فلو چارت



در این فلو چارت فرض شده است که در بافر نمایشگر موجود در حافظه، اطلاعات مربوط به دیجیت

سمت راست نمایشگر ها در اولین آدرس حافظه و به ترتیب با اطلاعات دیجیت سمت چپ در آخرین آدرس

بافر قرار دارد.

برنامه اول

برنامه مالتی پلکسینگ نمایشگرها، از ثباتهای زیر استفاده میکند:

زوج ثبات BC - مقادیر متغیر تاخیر

ثبات E - کد دیجیت

ثبات H - شمارنده دیجیت

ثبات شاخصی دار Ix - نشانگر بافر نمایشگر

باب 01 - قسمتهای نمایشگر (آنها)

باب 02 - کاتددیجیت

آدرسهای حافظه 1A00H تا 1A05H - بافر نمایشگر

توجه کنید که بیتهای ۷ از باب خروجی C (PC6, PC7) در تمام طول برنامه باید "۱" منطقی بمانند،

زیرا این دو بیت به قسمتهای دیگری از کامپیوتر متصل میباشند.

MPLEX: LD H, 6

شمارنده دیجیت

LD Ix, 1A00H

نشانگر بافر نمایشگرها

LD E, 01

کد دیجیت سمت راستی

DIG: LD A, (Ix+0)

OUT (01), A

اطلاعات هفت قسمتی به خروجی فرستاده میشود

LD A, E

اطلاعات کاتددیجیتها در انباره قرار میگیرد

OR COH

بیت ۷ غیرفعال میشوند

OUT (02H), A

یکدییجیت روشن میشود

CALL DELAY

LD A, COH

غیرفعال شدن کاتددیجیتها

OUT (02), A

دیجیتها غیرفعال میشوند

INC Ix

به نشانگر بافر یک واحد اضافه میشود

RLC E

کد دیجیت به سمت چپ گردش میکند

DEC H

یک واحد از شمارنده دیجیت کم میکند

JP NZ, DIG

دیجیت بعدی

JP MPLEX

تکرار

DELAY: LD BC, 0000

DEL: DEC BC

LD A*B

OR C

JP NZ*DEL

RET

این برنامه تاخیر برای هر دیجیت ۵/۷ ثانیه تاخیر ایجاد میکند و این زمان با عث میشود که مالتی-پلکسینگ برنامه بطور خیلی آهسته دیده شود. پس از تست برنامه و اطمینان از صحیح بودن آن میتوان با کاهش زمان تاخیر، برنامه را سریعتر کرد.

تمرین

الف) برنامه مالتی پلکسینگ

۱- ابتدا فرغ کنید می خواهید اعداد 1 2 3 4 5 6 به همین ترتیب روی نمایشگرها، نمایش داده شوند.

جدول زیر را برای هر عدد کامل کنید:

نمایشگر	قسمتها	d	p	c	b	a	f	g	e	کدهگز
1	b*c	0	0	1	1	0	0	0	0	30
2	a*b*d*e*g	1	0	0	1	1	0	1	1	9B
3										
4										
5										
6										

۲- برنامه مالتی پلکسینگ را به صورت کد ماشین بنویسید و آن را در آدرس شروع 1800H قرار دهید.

برنامه تاخیر را نیز در آدرس مناسب از حافظه قرار دهید.

۳- پس از وارد کردن برنامه در حافظه، کدهای هفت قسمتی را از آدرس 1A00H آغاز کنید.

کد را در آدرس 1A00H قرار دهید و بقیه را نیز به ترتیب وارد کنید.

۴- حال برنامه را اجرا کنید.

اگر برنامه صحیح کار میکند باید ابتدای آدرس دیجیت سمت راست عدد ۱ و بقیه تاریک و سپس در نمایشگر بعدی از همان سمت عدد ۲ و بقیه تاریک و

۵- در صورتیکه برنامه صحیح بوده است، تاخیر را کاهش دهید سعی کنید عدد تاخیر را در رنج

0001H تا 8000H انتخاب کنید.

۶- چهرنجی از مقادیر عدد تاخیر، یک نمایش قابل قبولی از خروجی مورد نظر را میدهد؟

۷- اگر این زمان را خیلی کوتاه کنید چه اتفاقی می افتد؟

۸- حال سعی کنید پیامهای دیگری مثل : OLS-L2 یا HELP US را نمایش دهید.

۹- اگر بخواهید که نمایشگرها از چپ به راست (برعکس قبل) حرکت کنند، برنامه را چگونه تغییر میدهید.

ب) استفاده از زیر برنامه نمایشگر مونیتور

فرستادن اطلاعات و پیامها به نمایشگر یکی از کارهای مهم در کامپیوتر است و معمولاً برنامه مونیتور شامل زیر برنامه ای برای نمایشگرها میباشد و استفاده کننده نیز میتواند در موارد لزوم از این زیر برنامه استفاده کند. در برنامه مونیتور، زیر برنامه مالتی پلکسینگ برای برنامه اسکان صفحه کلید ترکیب شده اند، به این مفهوم که در هر لحظه نمایشگرها اسکان شده و صفحه کلید نیز توسط ورودیها چک میشود. و شما میتونید صرف نظر از قسمتهایی که لازم ندارید از بخشهای مورد نیاز خود استفاده کنید.

نام : SCAN1

آدرس : 0624H

عملکرد: اسکان صفحه کلید و نمایشگرها، یک سیکل از راست به چپ. و زمان اجرا ۹/۹۷ ms

ورودی: نشانگر Ix شامل آدرس شروع بافر نمایشگر

خروجی: الف) اگر کلیدی زده نشده باشد پرچم کری = ۱

ب) اگر کلیدی زده شده باشد پرچم کری = 0 و انباره شامل کد کلید است.

شبهتهای استفاده شده : AF و AF' و BC' و DE'

توجه: الف) کد مربوط به هر هفت قسمتی را در آدرسی که Ix نشان میدهد ذخیره کنید.

ب) نشانگر Ix برای دیجیت سمت راست و نشانگر Ix+5 برای آخرین دیجیت واقع در سمت چپ میباشد.

برای اینکه فقط از قسمت مربوط به نمایشگرهای زیر برنامه استفاده کنید، آدرس شروع بافر

را در Ix قرار دهید و سپس زیر برنامه را صدا کنید. برای استفاده از فقط بخش اسکان صفحه کلید

در Ix آدرس 07A5 که باعث تاریک ماندن نمایشگرها میشود، را قرار دهید.

هر دو قسمت زیر برنامه را میتوان با انتخاب بافر نمایشگر مناسب، بطور همزمان مورد استفاده

قرار داد. مراحل زیر را انجام دهید.

۱- در برنامه اصلی MAIN زیر برنامه SCAN1 مرتباً تکرار میشود:

```
MAIN:LD Ix*1A00H
```

```
CALL SCAN1
```

```
JP MAIN
```

قبل از اجرای برنامه، کد حروف را در آدرسهای 1A00 تا 1A05 قرار دهید.

۲- در صورتیکه مدت زمان مشخصی، SCAN1 حروف مورد نظر را نمایش بدهد و سپس مدت زمان ثابتی نمایشگر تاریک بماند، به طرز جالبی، نمایشگرها چشمک میزنند.
با توجه باینکه اجرای زیر برنامه SCAN1 حدود ۱۰ ms طول میکشد، با ۵ مرتبه صدا کردن آن میتوان تاخیری برابر ۵۰ ثانیه را بدست آورد:

```
LD B*32H
```

```
HLFSEC:CALL SCAN1
```

```
DEC B
```

```
JP NZ*HLFSEC
```

۳- حال برنامه ای بنویسید که پیام 'HELLO' را روی ۵ عدد نمایشگر سمت راست، خاموش و روشن (چشمک زن) کند. (یک ثانیه پیام و یک ثانیه تاریک)

۴- میتوانید برنامه را طوری بنویسید که حروف مربوط به یک پیام طولانی به ترتیب سیب روی نمایشگرها حرکت کنند، اینکار را به آسانی میتوان با عوض کردن آدرس بافر نمایشگر انجام داد. تعداد حروف پیام که قرار است به نوبت روی نمایشگر حرکت کنند در شبات C قرار دارد.
به برنامه زیر توجه کنید:

```
MOVER:LD C*2AH
```

تعداد حروف در C قرار میگیرد

```
LD Ix*1A2AH
```

آدرس شروع بافر نمایشگر

```
MOVE:LD B*28H
```

```
MOVE1:CALL SCAN1
```

اسکان کردن برای ۴۰ ثانیه

```
DEC B
```

```
JP NZ*MOVE1
```

```
DEC Ix
```

انتقال نشانگر بافر

```
DEC C
```

یک واحد از شمارنده کم میشود

JP NZ MOVE

تست پایان پیغام

JP MOVER

تکرار پیغام

مقادیر مناسب در شبات IX و شبات C در رابطه با پیغام زیر قرار داده شده اند:

اطلاعات								آدرس
1A00	00	00	00	00	00	00	87	AE
1A08	8F	A7	00	AE	30	00	03	A3
1A10	87	AE	3F	8D	D0	87	A1	A7
1A18	00	B3	A3	A3	AD	00	AE	30
1A20	00	85	30	BD	00	85	85	8F
1A28	37	AE	00	00	00	00	00	00

برنامه قبل را به صورت کد ماشین نوشته و در آدرس 1800H وارد کنید و پیغام را نیز در آدرسهای

نشان داده شده وارد کنید. حال برنامه را اجرا کرده و پیغام را یادداشت کنید.

تمرین

۱- برنامه ای بنویسید که پیغام 'COFFEE or TEA' را نمایش دهد. حرکت از راست به طرف چپ

باشد.

۲- برنامه ای بنویسید که کلمه 'HELP' روی نمایشگرها چشمک بزند تا هنگامیکه کلید [STEP]

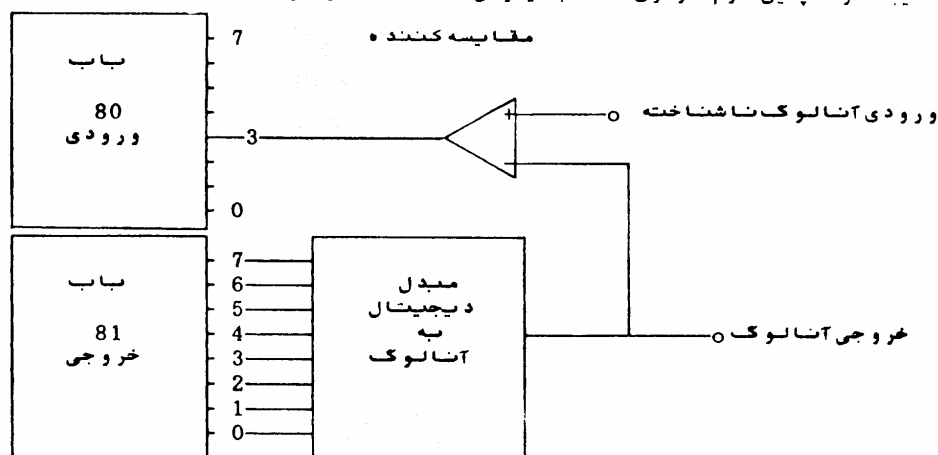
زده میشود.

آزمایش پانزدهم

مبدل آنالوگ به دیجیتال

مفاهیم کلی

هنگامیکه از کامپیوتر برای کنترل یک پروسه استفاده میشود، اغلب مواقع لازم است که ورودی خود را از دستگاههای آنالوگ بدست آورد. چونکه کامپیوتر فقط میتواند اطلاعات دیجیتال را پردازش کند، باید مدار رابطی ایجاد شود که بتواند اطلاعات را از فرم آنالوگ به دیجیتال تبدیل نماید. با استفاده از مدار مجتمع موجود روی برد که شامل DAC و مقایسه کننده میباشد و همچنین نرم افزاری مناسب میتوان اطلاعات آنالوگ را به دیجیتال تبدیل کرد:



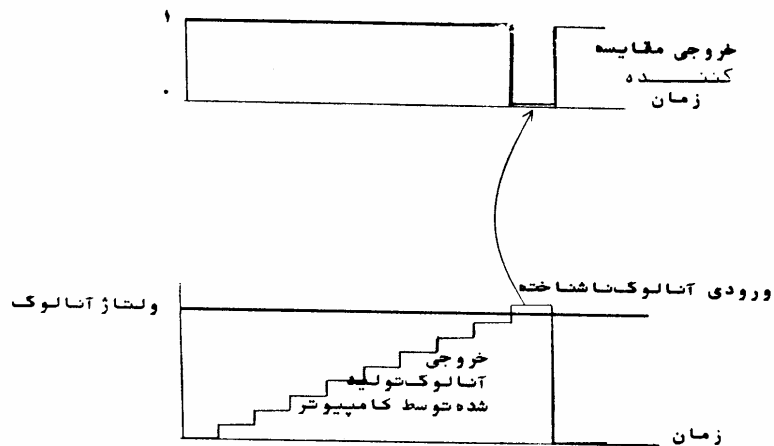
خروجی دیجیتال فرستاده شده به باب خروجی 81H به ولتاژ معادل آنالوگی تبدیل میشود. این ولتاژ با ولتاژ آنالوگ مجهول مقایسه میشود. اگر ورودی آنالوگ ناشناخته، بالاتر از ولتاژی باشد که توسط کامپیوتر تولید شده، خروجی مقایسه "۱" منطقی میشود و اگر پائین تر باشد "0" منطقی خواهد بود. کامپیوتر خروجی مقایسه کننده را از طریق بیت ۳ باب 80H خوانده و چک میکند.

خروجی کامپیوتر خیلی پائین - "۱" منطقی به بیت ۳ از باب 80H

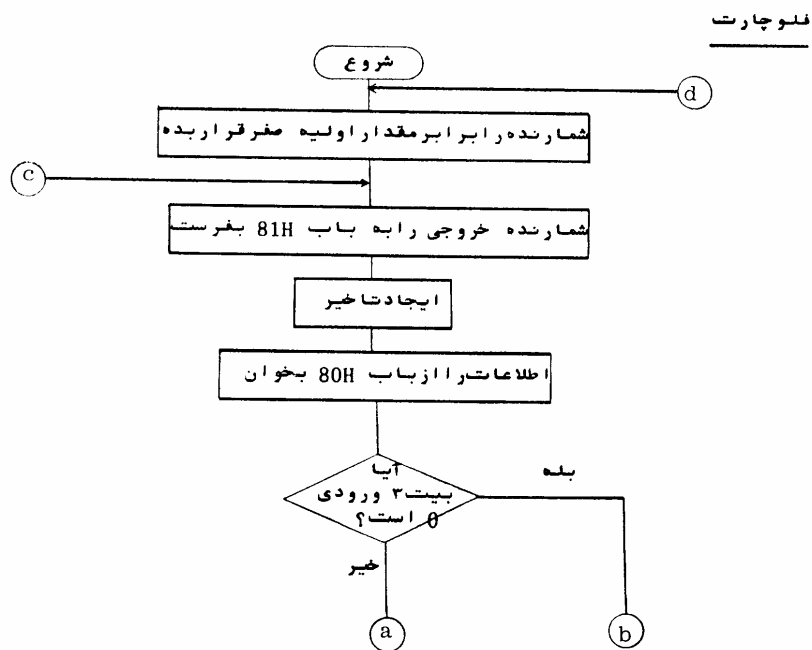
خروجی کامپیوتر خیلی بالا - "0" منطقی

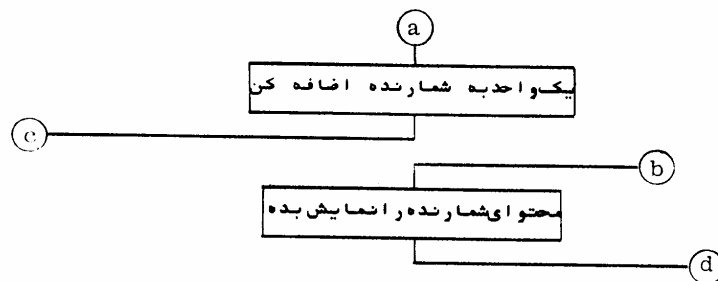
یک روش تبدیل A به D این است که خروجی دیجیتالی معادل 0 انتخاب کنیم و در هر مرحله یک واحد به آن اضافه کنیم و هر بار با خواندن بیت سه (خروجی مقایسه کننده)، تعیین شود که اندازه خروجی پائین تر و یا بالاتر از اندازه ورودی آنالوگ ناشناخته بوده است. بلافاصله پس

از اینکه کامپیوتر تشخیص داده و لحاظ خروجی بالا را از ولتاژ ورودی آنالوگ است، پرسش متوقف میشود.
این موضوع در شکل موجهای زیر نشان داده شده است:



این روش به عنوان متد 'پرش و مقایسه' نام برده میشود.





برنامه اول

برنامه زیر بر اساس فلوچارت بالا نوشته شده است. از شباهت C به عنوان شمارنده استفاده شده است. زیر برنامه نمایش همان زیر برنامه DISPA است که در آزمایش نهم معرفی شد. در این برنامه شما معادل دیجیتال ولتاژ آنالوگ ورودی را به صورت هگزادسیمال مشاهده میکنید.

ATOD:LD C*0	C شمارنده است
ATD:LD A*C	
OUT (81H),A	شمارنده خروجی
NOP	
NOP	
NOP	
IN A*(80H)	ورودی را از مقایسه کننده بدست میآورد
AND 08H	همه بیتها به جز بیت سوم صفر میشوند
JP Z*DIS	اگر تمام شده است، پرش انجام شود
INC C	یک واحد به شمارنده اضافه میشود
JP ATD	مقدار بعدی
DIS:LD A*C	
CALL DISPA	مقدار ولتاژ نمایش داده میشود
JP ATOD	

تمرین

- ۱- برنامه اول را به صورت ماشین کدنویسه و آدرس شروع آنرا 1800H در نظر بگیرید.
- ۲- برنامه را وارد حافظه میکروپروسور کنید.
- ۳- قبل از اجرای برنامه، مطمئن شوید که بوردر اویل کرده اید و همه سوییچهای آن به جز سوییچ

- ADC، در حالت خاموش هستند. روشن - ADC
- سلکتور آنالوگ ورودی را در حالت ۷ قرار دهید، در این صورت ورودی آنالوگ ولتاژ پتانسیومتر خواهد بود.
- ۴- حال برنامه را اجرا کنید.
- چراغهای باب 81H روشن شده و یا چشمک میزنند و روی دو عدد نمایشگر سمت راست، عدد هگزادسیمال نمایش داده میشود. توسط چرخاندن پتانسیومتر، ولتاژ ورودی آنالوگ را تغییر دهید و تغییر عدد هگز نمایش داده شده را بررسی کنید.
- ۵- رنج تغییرات این عدد هگز، در صورتیکه ولتاژ پتانسیومتر را از حداقل تا حداکثر مقدار تغییر دهید، چگونه است؟
- ۶- این برنامه خیلی سریع انجام میگیرد ولی شما نمیتوانید با کند کردن آن چگونگی اضافه شدن خروجی و رسیدن آن به ولتاژ ناشناخته را مشاهده کنید.
- به جای دستورات NOP یک دستور CALL DELAY برای ایجاد تاخیری حدود ۲ میلی ثانیه، قرار دهید.
- ۷- برنامه را مجدداً اجرا کنید و چراغهای باب 81H را نگاه کنید. وقتی آنتهای مقدار دیجیتال معادل ولتاژ آنالوگ ورودی برسند، صفر شده و دوباره همان کارها انجام میشود.
- توجه کنید که نمایشگرها، در تمام مدتی که برنامه به آهستگی کار میکند، خیلی کم نور هستند.
- ۸- میتوانید ولتاژ آنالوگ تولید شده توسط کامپیوتر را با روشن کردن سوئیچ BAR-G روی بارگراف مشاهده کنید.
- ولتاژ پتانسیومتر را تغییر دهید و نتیجه را روی بارگراف ببینید.
- ۹- اگر برنامه با سرعت بالا انجام شود، ولی ولتاژ ورودی آنالوگ بالاتر از ولتاژ تولید شده توسط کامپیوتر باشد، روی نمایشگرها چه اتفاقی میافتد؟
- ۱۰- اگر اجرای حلقه برنامه، مدت ۳ میکرو ثانیه طول بکشد، ماکزیمم مدت زمانی که لازم است تا تبدیل آنالوگ به دیجیتال انجام پذیرد، چقدر است؟
- ۱۱- اگر ورودی آنالوگ ناشناخته، مربوط به سنسور نور موجود روی برد باشد، رنج تغییرات ولتاژ دیجیتال معادل آن از بیشترین شدت نور تا تاریک کامل برای سنسور نور، چقدر است؟
- سلکتور در حالت L قرار دهید و این تغییرات را مشاهده کنید.
- ۱۲- اشکال اصلی، این نوع نرم افزار تبدیل A به D چیست؟

فهرست

آزمایش دوم

۲- الف)

آدرس	کدهگز	دستورات
1800.....	DB80.....	IN A*(80H)
1802.....	FE07.....	CP 07H
1804.....	CA0E18.....	JP Z*180EH
1807.....	3E55.....	LD A*55H
1809.....	D381.....	OUT (81H)*A
180B.....	C30018.....	JP 1800H
180E.....	3EAA.....	LD A*AAH
1810.....	D381.....	OUT (81H)*A
1812.....	C30018.....	JP 1800H

آزمایش چهارم

برنامه اول- ۴)

A15 - 0	A8 - 0	D7 - نور PULSE فعال
A14 - 0	A7 - 0	D6 - "
A13 - 0	A6 - نور PULSE فعال	D5 - "
A12 - نور PULSE فعال	A5 - "	D4 - "
A11 - "	A4 - "	D3 - "
A10 - 0	A3 - "	D2 - "
A9 - 0	A2 - "	D1 - "
	A1 - "	D0 - "
	A0 - "	

MREQ - PULSE فعال است

IORQ - "منطقی - بدون هیچ دستور ورودی خروجی

RD - PULSE فعال است

WR - "منطقی - بدون هیچ دستور نوشتن

آزمایش پنجم

تمرین صفحه ۳۹

آدرس	کدهگز	دستورات	۲-
------	-------	---------	----

```

1800.....3EFF.....INIT:LD A·FF
1802.....D382.....OUT (82H)·A
1804.....3E3F.....LD A·3F
1806.....D382.....OUT (82H)·A
1808.....DB80.....LOOP:IN A·(80H)
180A.....0F.....RRCA
180B.....0F.....RRCA
180C.....D380.....OUT (80H)·A
180E.....C30818.....JP LOOP

```

آزمایش ششم

برنامه دوم

-۶-

آدرس	کد هگز	دستورات
1800	DB80	START:IN A·(80H)
1802	E678	AND 78H
1804	FE50	CP 50H
1806	CA1018	JP Z·YES
1809	3E0F	LD A·0FH
180B	D381	OUT (81H)·A
180D	C30018	JP START
1810	3EF0	YES:LD A·F0H
1812	D381	OUT (81H)·A
1814	C30018	JP START

آزمایش هفتم

(۷

۱- برای ورودی استرابط برنامه به این صورت تصحیح میشود:

GETKEY:IN A·(80H)

AND 80H

GETKY:IN A*(80H)

AND 80H

سپس دستور AND 7FH را اضافه میکنیم

MAIN:IN A*(80H)

AND 7FH

AND A*H

آزمایش هشتم

تمرین صفحه ۵۲

آدرس	کدهگز	دستورات	۱- زیر برنامه MINI :
1800.....	3EFF.....	MINI:LD A*FFH	
1802.....	D382.....	OUT (82H)*A	
1804.....	3E3F.....	LD A*3F	
1806.....	D382.....	OUT (82H)*A	
1808.....	C9.....	RET	
1810.....	3E80.....	FWD:LD A*80H	-۲
1812.....	D380.....	OUT (80H)*A	
1814.....	C9.....	RET	
1820.....	3E40.....	REV:LD A*40H	-۳
1822.....	D380.....	OUT (80H)*A	
1824.....	C9.....	RET	
1830.....	3E00.....	STOP:LD A*00	-۴
1832.....	D380.....	OUT (80H)*A	
1834.....	C9.....	RET	
1840.....	010000.....	DELAY:LD BC*0000	-۵

```

1843.....0B.....DEL:DEC BC
1844.....78.....LD A·B
1845.....B1.....OR C
1846.....C24318.....JP NZ·DEL
1849.....C9.....RET

1850.....0603.....PROP:LD B·03
1852.....DB80.....LOOP1:IN A·(80H)
1854.....E610.....AND 10H
1856.....C25218.....JP NZ·LOOP1
1859.....DB80.....LOOP0:IN A·(80H)
185B.....E610.....AND 10H
185D.....CA5918.....JP Z·LOOP0
1860.....05.....DEC B
1861.....C25218.....JP NZ·LOOP1
1864.....C9.....RET

```

آز مایشی نعم

ب:

۱-

آدرس	کدهگز	دستورات
1800.....	CD001D.....	THERM2:CALL MHINI
1803.....	CD201D.....	LOOP:CALL ANALOG
1806.....	FE80.....	CP 80H
1808.....	DA2518.....	JP C·HON
180B.....	FE90.....	CP 90H
180D.....	DA1C18.....	JP C·HOFF
1810.....	F5.....	OVER:PUSH AF
1811.....	3E00.....	LD A·00

```

1813.....D380.....OUT (80H)·A
1815.....F1.....POP AF
1816.....CD901D.....CALL ALARM
1819.....C30318.....JP LOOP
181C.....F5.....HOFF:PUSH AF
181D.....3E00.....LD A·00
181F.....D380.....OUT (80H)·A
1821.....F1.....POP AF
1822.....C32B18.....JP DIS
1825.....F5.....HON:PUSH AF
1826.....3E20.....LD A·20H
1828.....D380.....OUT (80H)·A
182A.....F1.....POP AF
182B.....CD501D.....DIS:CALL DISPA
182F.....C30318.....JP LOOP

```

تمرین صفحه ۶۳

-۱

آدرس	کدهگز	دستورات
1800.....	CD001D.....	THERM3:CALL MHINI
1803.....	CD201D.....	LOOP:CALL ANALOG
1806.....	FE70.....	CP 70H
1808.....	DA3618.....	JP C·UNDER
180B.....	FE80.....	CP 80H
180D.....	DA2A18.....	JP C·HON
1810.....	FE90.....	CP 90H
1812.....	DA2118.....	JP C·HOFF
1815.....	F5.....	OVER:PUSH AF
1816.....	3E00.....	LD A·00

```

1818.....D380.....OUT (80H)·A
181A.....F1.....POP AF
181B.....CD901D.....ALAM:CALL ALARM
181E.....C30318.....JP LOOP
1821.....F5.....HOFF:PUSH AF
1822.....3E00.....LD A·00
1824.....D380.....OUT (80H)·A
1826.....F1.....POP AF
1827.....C33018.....JP DIS
182A.....F5.....HON:PUSH AF
182B.....3E20.....LD A·20H
182D.....D380.....OUT (80H)·A
182F.....F1.....POP AF
1830.....CD501D.....DIS:CALL DISPA
1833.....C30318.....JP LOOP
1836.....F5.....UNDER:PUSH AF
1837.....3E20.....LD A·20H
1839.....D380.....OUT (80H)·A
183B.....F1.....POP AF
183C.....C31B18.....JP ALAM

```

آزمایش دهم

تمرین صفحه ۶۸

```

1900.....FF
1901.....FF
1902.....00
1903.....00
1904.....00

```

-۱

حداقل پریود تاخیر باید ۱/۱ ثانیه باشد.

۲- اطلاعات مربوط به تولید شکل موج سینوسی :

آدرس	اطلاعات	آدرس	اطلاعات
1900.....80	صفر درجه <---	1912.....80	۱۸. درجه <---
1901.....96		1913.....6A	
1902.....AB		1914.....55	
1903.....C0		1915.....41	
1904.....D2		1916.....2E	
1905.....E1		1917.....1F	
1906.....EE		1918.....12	
1907.....F7		1919.....09	
1908.....FD		191A.....03	
1909.....FF	۹. درجه <---	191B.....01	۲۷. درجه <---
190A.....FD		191C.....03	
190B.....F7		191D.....09	
190C.....EE		191E.....12	
190D.....E1		191F.....1F	
190E.....D2		1920.....2E	
190F.....C0		1921.....41	
1910.....AB		1922.....55	
1911.....96			

آزمایش یازدهم

الف) کنترل نوری ترافیک

زمان	7 6 5 4 3 2 1 0	کد اطلاعات	کد زمان
1	1 0 0 0 1 1 0 0	8C	02
1	1 0 0 0 0 0 1 0	82	02
4	1 0 0 1 0 0 1 0	92	08
0.5	1 0 0 0 0 0 1 0	82	01
0.5	1 0 0 1 0 0 1 0	92	01
0.5	1 0 0 0 0 0 1 0	82	01
0.5	1 0 0 1 0 0 1 0	92	01
0.5	1 0 0 0 0 0 1 0	82	01
0.5	1 0 0 1 0 0 1 0	92	01
1	1 0 0 0 0 1 0 0	84	02
1	1 1 0 0 1 0 0 0	C8	02
1	0 0 1 0 1 0 0 0	28	02
4	0 0 1 0 1 0 0 1	29	08
0.5	0 0 1 0 1 0 0 0	28	01
0.5	0 0 1 0 1 0 0 1	29	01
0.5	0 0 1 0 1 0 0 0	28	01
0.5	0 0 1 0 1 0 0 1	29	01
0.5	0 0 1 0 1 0 0 0	28	01
0.5	0 0 1 0 1 0 0 1	29	01
1	0 1 0 0 1 0 0 0	48	02

کد اطلاعات و کد زمان در حافظه از آدرس 1A00H به دنبال هم قرار میگیرند.

۲- برنامه دوم باینرمورت نوشته میشود:

آدرس	کد هگز	دستورات
1800.....	0614.....	START: LD B·14H
1802.....	21001A.....	LD HL·1A00H
1805.....	7E.....	REP: LD A·(HL)
1806.....	23.....	INC HL
1807.....	D381.....	OUT (81H)·A
1809.....	4E.....	LD C·(HL)
180A.....	23.....	INC HL
180B.....	CDA018.....	CALL VARDEL
180E.....	05.....	DEC B
180F.....	C20518.....	JP NZ·REP
1812.....	C30018.....	JP START

18A0.....115091.....VARDEL:LD DE*9150H

18A3.....1B.....VLOOP:DEC DE

18A4.....7A.....LD A*D

18A5.....B3.....OR E

18A6.....C2A318.....JP NZ*VLOOP

18A9.....0D.....DEC C

18AA.....C2A018.....JP NZ*VARDEL

18AD.....C9.....RET

ب) کنترل عبور و مرور بصورت دستی

زمان	7 6 5 4 3 2 1 0	کد اطلاعات	کد زمانی
5	1 0 0 0 0 0 1 0	82	0A
2	1 0 0 0 0 1 0 0	84	04
1	1 0 0 0 1 0 0 0	88	02
6	0 0 1 0 1 0 0 0	28	0C
0.5	0 0 1 0 0 1 0 0	24	01
0.5	0 0 0 0 0 0 0 0	00	01
0.5	0 0 1 0 0 1 0 0	24	01
0.5	0 0 0 0 0 0 0 0	00	01
0.5	0 0 1 0 0 1 0 0	24	01
0.5	0 0 0 0 0 0 0 0	00	01
0.5	0 0 1 0 0 1 0 0	24	01
0.5	0 0 0 0 0 0 0 0	00	01
0.5	1 0 0 0 0 0 1 0	82	01

آزمایش دوازدهم

تمرین صفحه ۷۶

قدم	عملکرد	شرایط تعیین کننده
	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
1	0 0 0 0 0 0 0 0	0 0 1 0 0 1 0 0
2	0 0 0 1 0 0 0 1	0 0 1 0 0 0 1 0
3	0 0 1 0 0 0 0 1	0 0 1 0 0 0 1 1
4	1 0 0 0 0 0 1 1	0 0 1 1 0 0 1 0
5	0 0 0 0 0 1 0 1	0 0 1 0 0 1 0 0
6	0 0 0 0 1 0 0 1	0 0 1 0 0 0 1 0
7	1 0 0 0 0 0 1 1	0 0 1 0 1 0 1 0
8	0 0 0 0 0 1 0 1	0 0 1 0 0 1 0 0
9	0 1 0 0 0 1 1 1	0 0 1 0 1 1 0 0
10	0 0 0 0 0 0 0 0	0 0 0 0 0 1 0 0

که جدول آن بصورت زیر خواهد بود:

1A00.....00.....24
 1A02.....11.....22
 1A04.....21.....23
 1A06.....83.....32
 1A08.....05.....24
 1A0A.....09.....22
 1A0C.....83.....2A
 1A0E.....05.....24
 1A10.....47.....2C
 1A12.....00.....04

-۲

آدرس	کدهگز	دستورات
1800.....060A.....	START:LD B·0AH	
1802.....21001A.....	LD HL·1A00H	
1805.....7E.....	REP:LD A·(HL)	
1806.....D381.....	OUT (81H)·A	
1808.....23.....	INC HL	
1809.....4E.....	LD C·(HL)	
180A.....23.....	INC HL	
180B.....DB80.....	LOOP:IN A·(80H)	
180D.....B9.....	CP C	
180E.....C20B18.....	JP NZ·LOOP	
1811.....05.....	DEC B	
1812.....C20518.....	JP NZ·REP	
1815.....76.....	HALT	

آزمایش سیزدهم

تمرین صفحه ۸۱

-۸

تعداد دفعاتی که برنامه را دور زده اید

	PC5 (12)	PC4 (13)	PC3 (17)	PC2 (16)	PC1 (15)	PC0 (14)
1	1	1	1	1	1	0
2	1	1	1	1	0	1
3	1	1	1	0	1	1
4	1	1	0	1	1	1
5	1	0	1	1	1	1
6	0	1	1	1	1	1
7	1	1	1	1	1	0
8	1	1	1	1	0	1

تمرین صفحه ۸۴

دستورات کدهنگز آدرس

1800.....CD0019.....BLEEP:CALL SCANKY

1803.....7D.....LD A,L

1804.....FE01.....CP 01H

1806.....C20018.....JP NZ,BLEEP

1809.....216400.....LD HL,0064H

180C.....CDDE05.....CALL TONE1K

180F.....C30018.....JP BLEEP

-۱

دستورات کدهنگز آدرس

1800.....CD0019.....STEP:CALL SCANKY

1803.....7D.....LD A,L

1804.....FE01.....CP 01H

1806.....C20018.....JP NZ,STEP

1809.....3A001A.....LD A,(1A00H)

180C.....FE10.....CP 10H

180E.....C20018.....JP NZ,STEP

1811.....216400.....LD HL,0064

1814.....CDDE05.....CALL TONE1K

1817.....C30018.....JP STEP

-۲

آزمایش چهاردهم

(الف)

نمایشگر	قسمتها	d	p	c	b	a	f	g	e	کدهنگز
1	b.c	0	0	1	1	0	0	0	0	30
2	a.b.d.e.g	1	0	0	1	1	0	1	1	9B
3	a.b.c.d.g	1	0	1	1	1	0	0	1	B9
4	b.c.f.g	0	0	1	1	0	1	1	0	36
5	a.c.d.f.g	1	0	1	0	1	1	1	0	AE
6	a.c.d.e.f.g	1	0	1	0	1	1	1	1	AF

-۸

2.....9B	S.....AE
L.....85	U.....B5
-.....02	P.....1F
5.....AE	L.....85
L.....85	E.....8F
O.....BD	H.....37

تمرین صفحه ۹۱

LD C*17H

۱- برنامه مانند قسمت ۴- بنویشته میشود به استثنای اینکه

LD Ix*1A17H

1A00	00	00	00	00	00	00	1B
1A08	3F	8F	87	00	03	A3	00
1A10	8F	8F	0F	0F	BD	8D	00
1A18	00	00	00	00	00		

-۲

آدرس	کدهنگز	دستورات
1800.....	DD21001A....	FHELP:LD Ix*1A00H
1804.....	0632.....	LD B*32H
1806.....	CD2406....	HLFSEC:CALL SCAN1
1809.....	DA1118.....	JP C*NOKY
180C.....	FE10.....	CP 10H
180E.....	CA2A18.....	JP Z*STOP
1811.....	05.....	NOKY:DEC B
1812.....	C20618.....	JP NZ*HLFSEC
1815.....	DD21A507....	BLANK:LD Ix*07A5H
1819.....	0632.....	LD B*32H
181B.....	CD2406....	BLANK1:CALL SCAN1

```
181E.....DA2618.....JP C•NOKEY
1821.....FE10.....CP 10H
1823.....CA2D18.....JP Z•STOP
1826.....05.....NOKEY:DEC B
1827.....C21B18.....JP NZ•BLANK1
182A.....C30018.....STOP:HALT
```

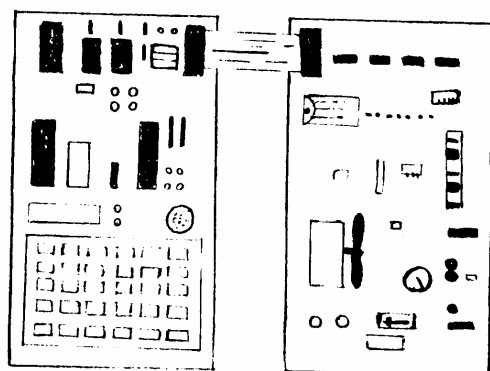
ضمیمه ۲

بورد کار بردی AB-1W

جزو هر آهنمای حاضر در مورد چگونگی کار با بورد کار بردی AB-1 نوشته شده است ، در صورتیکه بورد شما از نوع AB-1W باشد ، باز هم این جزو هر آهنمای شما خواهد بود .

بورد AB-1W در واقع همان AB-1 است با این تفاوت که این بورد دارای ترانزیستور PIO و سوکت ترانزیستور CTC بوده و از طریق کانکتور J2 به خطوط Z-80 وصل میشود .

اهمیت AB-1W از آن جهت است که در واقع بایهای قابل برنامهریزی را شامل میباشد و بر روی آن میکرو کامپیوتر هاشی که PIO و CTC دارند قابل استفاده است .



بورد AB-1W کاملاً با میکرو کامپیوتر MPF-1W منطبق بوده و علاوه بر آن به کلیه میکرو کامپیوتر هاشی که AB-1 به آنها متصل میشود قابل اتصال است .

محل کانکتور J1 این امکان را میدهد که در صورتیکه بخواهید AB-1W را از طریق میکرو کامپیوتر MPF-1B متصل کنید کانکتور مناسبی در این محل لحیم

کرده و سپس از طریق کابل مستقیماً آنرا به MPF-1B مرتبط کنید . بنابراین AB-1W به هر دو نوع میکرو کامپیوتر اعم از آنهایی که تنها خطوط Z-80 را در اختیار استفاده میکنند قرار داده اند و چه به آنهایی که خطوط خروجی ترانزیستورهای قابل برنامهریزی PIO و CTC دارند ، قابل وصل میباشد .

گرد آوری و تنظیم :

حامد مظاهری

hamed@ir-micro.com