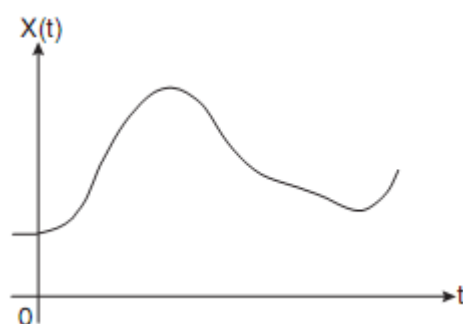
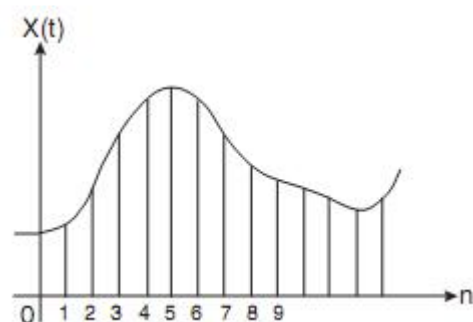


ضبط و پخش صدا با استفاده از AVR و MMC

قبل از هر چیز و قبل از اینکه سیگنال صحبت بتواند روی حافظه ذخیره شود؛ می بایست به سیگنال مجزا از زمان تبدیل شود (شکل 1). این کار بوسیله نمونه برداری پریودیک انجام می شود (شکل 2). فاصله بین دو نمونه Sampling Period و عکس آن فرکانس نمونه برداری نامیده می شود.

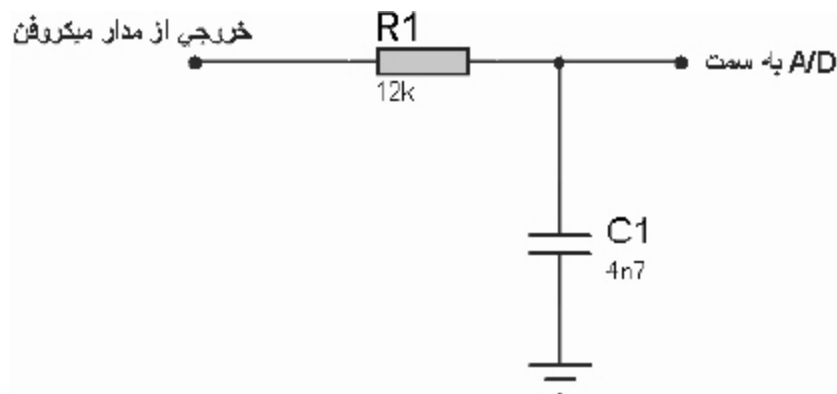


شکل 1

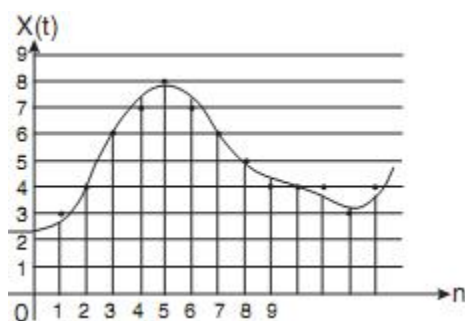


شکل 2

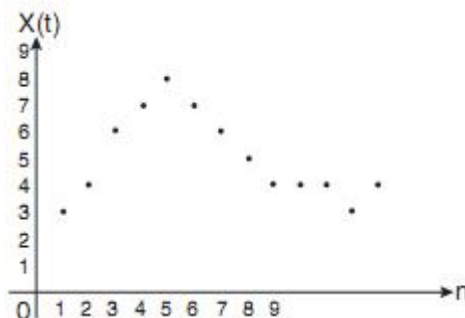
فرکانس نمونه برداری می بایست حداقل دو برابر فرکانس سیگنال باشد؛ در غیر اینصورت صدای خروجی همچون صدای روح خواهد شد. اطلاعات عمده فرکانس صحبت زیر 3000 هرتز است. بنابراین می توان برای محدودسازی باند سیگنال؛ در خروجی مدار میکروفرن از یک فیلتر پایین گذر استفاده نمود. البته در این پروژه برای دست یابی به بهترین نتیجه از قرار دادن فیلتر در خروجی میکروفرن صرف نظر شده است؛ اما شما می توانید در صورت تمایل فیلتری به شکل زیر را در خروجی قرار دهید. توجه داشته باشید که این کار موجب تضعیف سیگنال خروجی خواهد شد.



تعیین مقدار دیجیتالی که متناسب با نمونه آنالوگ در آن فرکانس است؛ تدریج نامیده می شود (شکل 3 و 4). مقادیر دیجیتالی محدود بوده و رزولوشن نامیده می شود. به عنوان مثال 255 برای رزولوشن 8 بیتی. بنابراین تدریج سیگنال آنالوگ موجب از دست رفتن اطلاعات سیگنال می شود. این خطای تدریج به طور معکوس متناسب با رزولوشن سیگنال دیجیتال است؛ همچنین به طور معکوس متناسب با رنج دینامیکی سیگنال است. می توان با تنظیم AGND و AREF به مقدار حداقل و حداکثر؛ رنج دینامیکی سیگنال را تعیین نمود. به شکلی دیگر تقویت کننده میکروفن می تواند برای پوشش رنج دینامیکی ADC تنظیم شود. هر دو روش فوق موجب کاهش خطای تدریج می شود. به علاوه روش دوم موجب افزایش سیگنال به نویز می شود؛ بنابراین ارجح است.



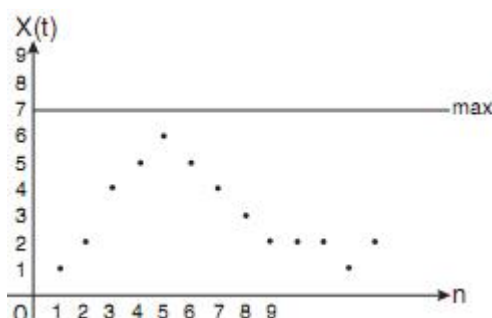
شکل 3: سیگنال تدریجی شده



شکل 4: سیگنال دیجیتال

در این مدار؛ سیگنال دارای مقادیر Min و Max است که هرگز از آن تجاوز نمی کند. بخش های پایین Min و بالای Max دارای هیچ گونه اطلاعاتی نیستند؛ بنابراین برای حفظ حافظه می توان آنها را

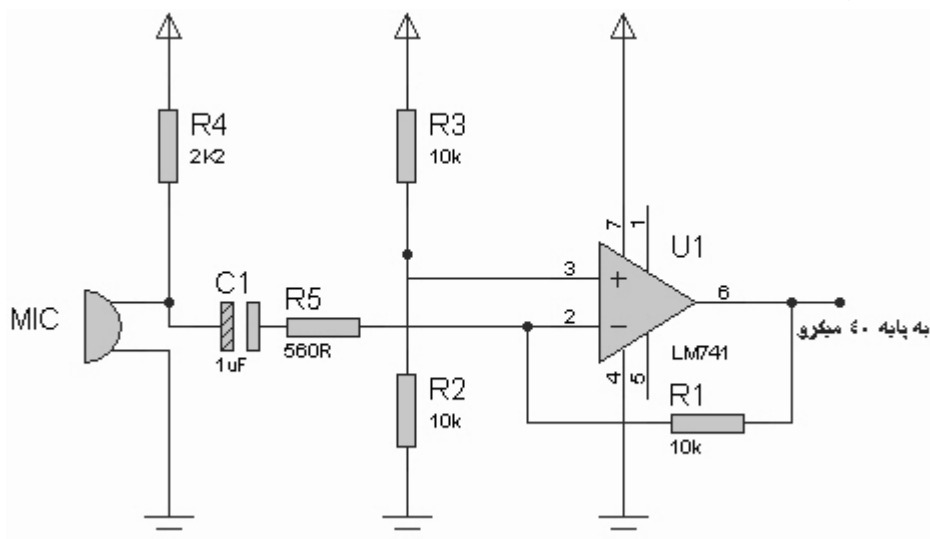
حذف کرد. این کار بوسیله شیفت دادن کل سیگنال به پایین و دور ریختن قسمت های بالای Max انجام می گیرد. (شکل 5)



در این پروژه سیگنال نهایی 8 بیتی است و این سیگنال می تواند بر روی MMC ذخیره شود.

طبقه اول: میکروفرن و تقویت کننده آن

در این مدار تقویت کننده میکروفرن؛ یک تقویت کننده معکوس کننده ساده است که بهره آن توسط $G = R1/R5$ تعیین می شود. برای تغذیه میکروفرن از مقاومت R4 استفاده شده است و C1 نیز وظیفه بلوکه کردن ولتاژهای DC را برعهده دارد. R2 و R3 نیز وظیفه تنظیم آفست ورودی را برعهده دارند. می توانید بجای LM741 از LM324 استفاده کنید.



شکل 7: طبقه ورودی مدار

طبقه دوم: میکروکنترلر و MMC

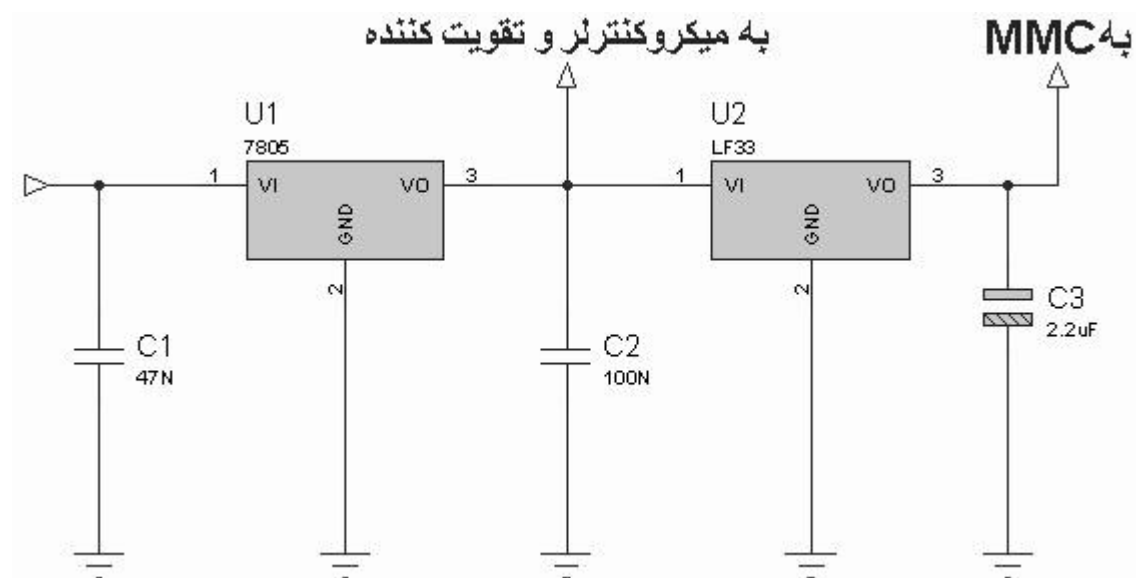
در این بخش یک میکروکنترلر ATMEGA32 قرار دارد که وظیفه دریافت سیگنال آنالوگ از مدار میکروفرن و تبدیل آن به سیگنال دیجیتال و سپس ذخیره نتیجه تبدیل؛ بر روی MMC را برعهده دارد. سیگنال آنالوگ به پایه ADC0 وارد می شود و پس از تبدیل بر روی MMC ذخیره می شود.

پروتکل ارتباطی میکروکنترلر با MMC بصورت SPI است که برای دستیابی به حداکثر سرعت نوشتن و خواندن از پروتکل SPI بصورت سخت افزاری استفاده شده است.

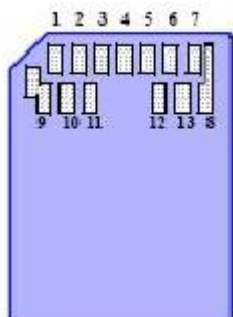
دو کلید فشاری در مدار وظیفه ضبط و پخش اطلاعات را برعهده دارند و دو دیود نوری نیز برای نشان دادن صحت عملکرد مدار بکار رفته اند. اما دلیل قرار گیری مقاومت روی پایه های MMC بدین جهت است که تغذیه MMC 3.3 ولت بوده و نمی تواند ولتاژ 5 ولت را بر روی پایه هایش تحمل کند و چون خروجی میکروکنترلر بصورت TLL است بنابراین مقاومت ها ی نشان داده شده وظیفه تقسیم ولتاژ را برعهده دارند که برای دستیابی به حداکثر اطمینان ؛ ولتاژ روی پایه های MMC ؛ 2 ولت در نظر گرفته شده است $5 = (2.2 / 2.2 + 3.3) \cdot$.

تذکر: نیازی به قرار دادن مقاومت بر روی پایه DO نیست.

برای تغذیه MMC حتما می بایست از رگولاتور 3.3 ولت استفاده نمایید. شکل زیر این حالت را نشان می دهد.



شکل 8: تغذیه MMC و میکروکنترلر



بجای MMC می توانید از کارت SD نیز استفاده کنید ؛ هر دو نوع کارت از پروتکل SPI پشتیبانی می کنند؛ ضمن اینکه SD از سرعت بیشتری نیز برخوردار است. شکل مقابل ترتیب پایه های کارت حافظه MMC را نشان می دهد.

طبقه سوم: تبدیل سیگنال دیجیتال به آنالوگ بدون استفاده از چپ های مبدل D/A

در این مدار برای کاهش هزینه از یک روش ابتکاری برای تبدیل سیگنال دیجیتال به آنالوگ استفاده شده است. این روش R-2R Ladder نامیده می شود و با کم ترین هزینه و تنها با استفاده از 16 عدد مقاومت و یک خازن سرامیکی قابل اجرا می باشد. شما می توانید در صورت تمایل بجای استفاده از این روش از چپ های D/A 8 بیتی موجود در بازار همچون DAC0808 استفاده نمایید.

در استفاده از این مبدل DAC می بایست به نکات زیر توجه کنید:

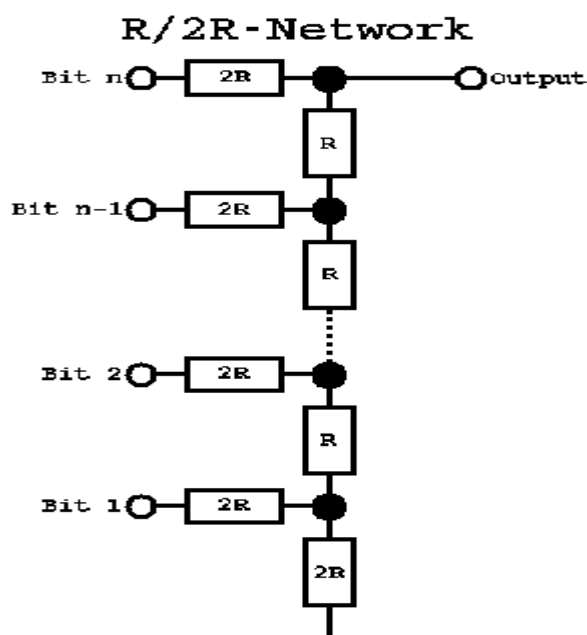
* چون این شبکه DAC؛ در خروجی جریان پایینی را ارائه می کند؛ بنابراین می بایست برای تقویت سیگنال از یک OPAMP استفاده شود.

* چون جریان خروجی از پایه های میکروکنترلر در حدود 20 میلی آمپر است بنابراین برای جلوگیری از تلفات می بایست مقدار مقاومت بیش از چند ده کیلو باشد.

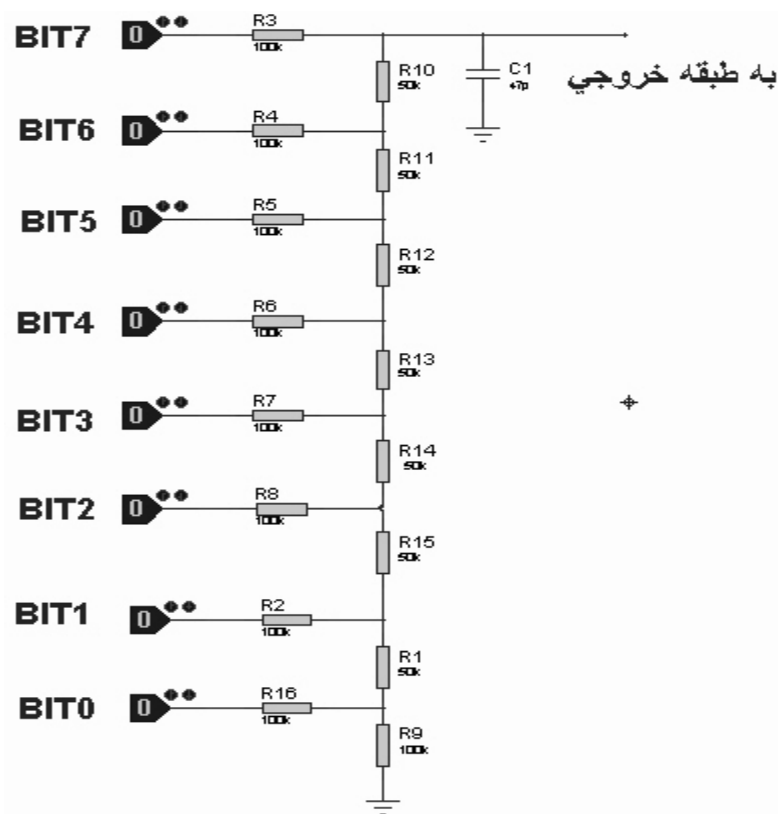
* برای دستیابی به حداکثر کیفیت در خروجی سعی کنید از مقاومت هایی با ضریب خطای 1 و یا 2% استفاده نمایید.

شکل کلی این مبدل بصورت زیر است.

مقاومت 2R دو برابر مقاومت R در نظر گرفته شود. در این پروژه از مقاومت 100KΩ و 50KΩ استفاده شده است.



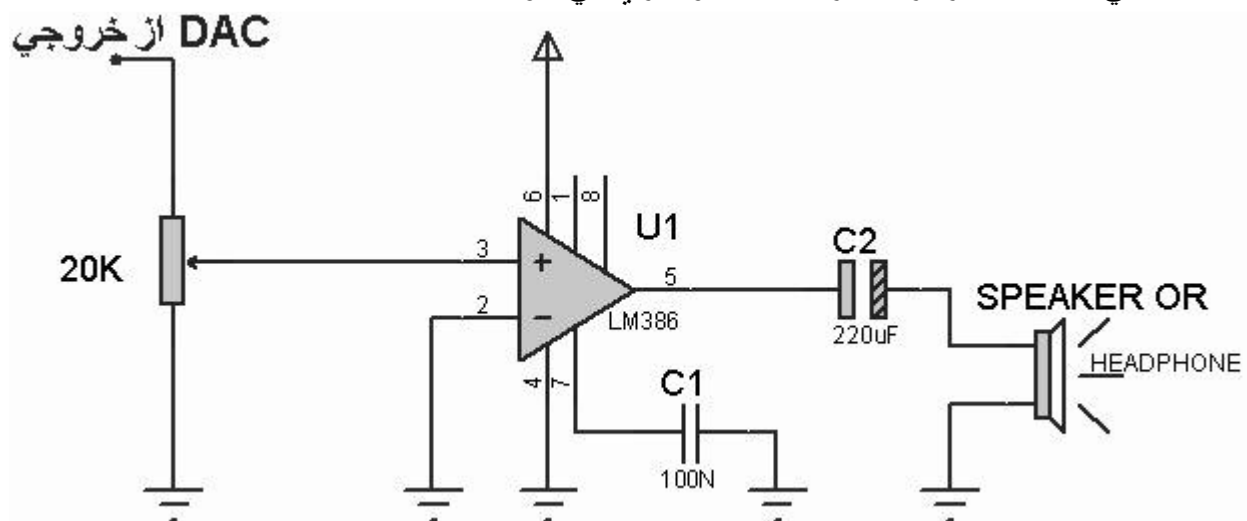
شکل 10: نمای کلی یک مبدل DAC مقاومتی



شکل 11: مبدل DAC استفاده شده در این مدار

طبقه چهارم: تقویت سیگنال خروجی از DAC برای استفاده از هدفون یا بلندگو

در این پروژه برای تقویت سیگنال خروجی از DAC ؛ از تقویت کننده صوتی LM386 استفاده شده است. شکل زیر این طبقه را نشان می دهد. بدلیل سادگی این طبقه ؛ از ارائه جزئیات آن خودداری می شود.



شکل 11: خروجی بلندگو

در صورتی که برای تقویت صدای خروجی از LM386 استفاده می کنید؛ به نکات زیر توجه داشته باشید:

وجود مدار فوق؛ در هنگام ضبط صدا موجب تولید نویز شده و این نویز از خط تغذیه وارد مدار شده و همراه با صدای ورودی از میکروفن بر روی MMC ذخیره خواهد و نتیجه اینکه صدای خروجی نویز دار می شود. برای جلوگیری از این حالت می توانید در هنگام ضبط صدا تغذیه LM386 را بوسیله فرمان به یکی از پایه های آزاد میکرو قطع نموده و در هنگام پخش صدا مجدداً؛ وصل نمایید. (در این پروژه این حالت پیش بینی نشده است)

تشریح برنامه:

در ADC؛ اولین سیکل تبدیل نیاز به زمان بیشتری نسبت به بقیه تبدیل ها دارد؛ بطوری که اگر فرکانس کلاک میکروکنترلر 8MHz در نظر گرفته شود؛ تبدیل اول بجای 448 سیکل؛ 832 سیکل زمان خواهد برد. در اینجا برای دستیابی به حداکثر سرعت تبدیل؛ فرکانس ADC روی 250KHz تنظیم شده است. بنابراین هر تبدیل نیاز به 448 سیکل؛ معادل 56us زمان نیاز خواهد داشت. طبق آزمایش های عملی انجام گرفته این مدار می تواند در یک ثانیه حدود 15 سکتور از MMC را پر نماید که در نتیجه می توان گفت که فرکانس نمونه برداری مدار در حدود 8KHz است. می توان با افزایش فرکانس کلاک به 16MHz؛ تعداد نمونه برداری را تا 23 نیز افزایش داد که بدین ترتیب می توان به فرکانس نمونه برداری 12KHz دست یافت.

مقادیر تبدیل شده ADC رنجی بین AGND و AVCC دارند که در اینجا 0 و 5 ولت است. البته باید به این نکته توجه داشت که سیگنال خروجی از میکروفن در رنج 2.43 الی 3.5 ولت محدود شده است. بنابراین نتیجه تبدیل ADC می بایست از مقدار معینی که نشان دهنده Min ولتاژ ورودی است؛ کم شود. برای 2.43 ولت این مقدار 498 می باشد. بخش هایی از اطلاعات که مقادیری بیش از 3.5 ولت دارند با برداشتن دو msb از بین می روند؛ این کار بطور اتوماتیک در هنگام نوشتن اطلاعات بر روی mmc انجام می شود.

در ابتدای برنامه و پس از مقدار دهی اولیه؛ وجود MMC و صحت عملکرد آن چک می شود. در صورت وجود کارت در اسلات؛ LED سبز به مدت یک ثانیه روشن شده و دوباره خاموش می شود. در غیر این صورت LED قرمز به مدت یک ثانیه روشن خواهد شد.

پس از آن برنامه در یک حلقه قرار می گیرد که در صورت زده شدن کلید REC از حلقه خارج شده و شروع به ضبط اطلاعات می کند.

به منظور جلوگیری از اتلاف زمان کلید STOP به وقفه خارجی 0 متصل گردیده است. هنگامی که این کلید فشرده شود ضبط متوقف گردیده و برنامه منتظر زده شدن مجدد کلید REC می ماند اما نه برای ضبط مجدد بلکه برای پخش اطلاعات ضبط شده! به عبارتی کلید REC هم وظیفه ضبط اطلاعات و هم وظیفه پخش آنرا بر عهده دارد.

در زیر برنامه پخش؛ برای همزمانی اطلاعات ضبط شده از یک توقف 117 میکروثانیه ای استفاده شده است. اطلاعات خوانده از حافظه به تناوب وارد PORTC می شود تا تبدیل به سیگنال آنالوگ شود. می توان برای پخش اطلاعات از روش PWM نیز استفاده نمود اما در آزمایشهای انجام شده توسط نویسنده در این حالت نویز خروجی زیاد و؛ صدای نامفهوم مشاهده گردید.

برنامه:

```
$regfile = "m32def.dat"
$crystal = 8000000
$swstack = 64
$hwstack = 128
$framesize = 128
Config Portc = Output : Portc = 0
Config Pind.1 = Input : Portd.1 = 1
Rec_key Alias Pind.1
Config Porta.1 = Output : Porta.1 = 0
Led_rec Alias Porta.1
Config Porta.2 = Output : Porta.2 = 0
Led_play Alias Porta.2
Config Int0 = Falling
On Int0 Request
Enable Interrupts
Enable Int0
'*****

Waitms 1
$include "Config_MMC.bas"
$external _mmc
Gbdriveerror = Driveinit()
If Gbdriveerror <> 0 Then
    Led_rec = 1 : Wait 1 : Led_rec = 0
Else
    Led_play = 1 : Wait 1 : Led_play = 0
End If
```

```
Gbdriveerror = Drivecheck()
```

```
'*****
```

```
Config Adc = Single , Prescaler = 32 , Reference = Avcc
```

```
'*****
```

```
Declare Sub No_key
```

```
Dim Abuffer(512) As Byte
```

```
Dim Wsrampointer As Word
```

```
Dim Lsectornumber As Long
```

```
Dim Cop_lsec As Integer
```

```
Lsectornumber = 100
```

```
Dim I As Integer
```

```
For I = 1 To 512
```

```
    Abuffer(i) = 0
```

```
Next
```

```
I = 1
```

```
'*****WRITING PROCEDURE*****
```

```
Do
```

```
    If Rec_key = 0 Then
```

```
        Call No_key
```

```
        Led_rec = 1
```

```
        Start Adc
```

```
        Goto Main
```

```
    End If
```

```
Loop
```

```
Main:
```

```
    Wsrampointer = Varptr(abuffer(1))
```

```
Do
```

```
    For I = 1 To 512
```

```
        Abuffer(i) = Getadc(0)
```

```

        Abuffer(i)=Abuffer(i)-498
    Next
    Gbdriveerror = Drivewritesector(wsrampointer , Lsectornumber)
    Lsectornumber = Lsectornumber + 1
    I = 1
Loop
Request:
    Stop Adc
    Cop_lsec = Lsectornumber
    Led_rec = 0
    Lsectornumber = Lsectornumber - 1

'*****READING PROCEDURE*****
From_one:
Do
    If Rec_key = 0 Then
        Call No_key
        Goto Start_play
    End If
Loop

Start_play:
    Led_play = 1 : Led_rec = 0
    Lsectornumber = 100
    Wsrampointer = Varptr(abuffer(1))
    Gbdriveerror = Drivereadsector(wsrampointer , Lsectornumber)
Do
    I = 1
    For I = 1 To 512
        Portc = Abuffer(i)
        Waitus 117
    Next
    Lsectornumber = Lsectornumber + 1

```

```

Gbdriveerror = Drivereadsector(wsramppointer , Lsectornumber)
Loop Until Cop_lsec = Lsectornumber
'*****

Portc = 0
Led_play = 1 : Led_rec = 1
Goto From_one
Return

End
'*****

Sub No_key
Do
    Loop until Pind.1=1
    Waitms 5
End Sub

```

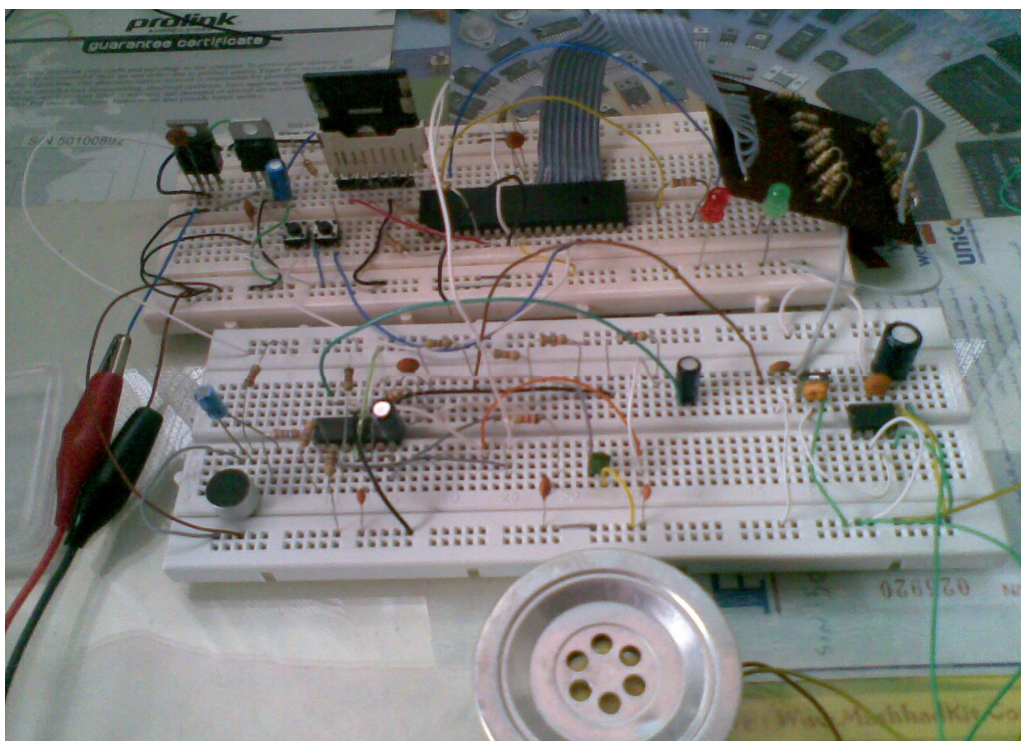
نکات پایانی:

1) این پروژه تنها جنبه آموزشی دارد بنابراین سعی شده تا حد امکان کدهای برنامه کاهش یابد تا درک عملکرد آن راحت تر باشد. به همین دلیل مدار از جنبه های گوناگونی شاید عیب هایی داشته باشد از جمله اینکه پس از ریست شدن ؛ امکان ضبط از ادامه وجود ندارد و اطلاعات جدید روی اطلاعات قبلی نوشته خواهد شد و یا اینکه ؛ هنگامی که کلید STOP زده شود ؛ دیگر امکان ضبط وجود نخواهد داشت مگر اینکه یکبار عمل ریست انجام شود. اصلاح عملکرد مدار و به عبارتی شخصی سازی آن به خواننده محترم واگذار می شود.

2) شما می توانید بسته به نیاز از هر نوع کارت MMC و یا SD با هر میزان حافظه ای استفاده کنید. می توان با استفاده از یک کارت 1GB ؛ حدودا 34 ساعت اطلاعات را ضبط و پخش نمود.

3) چون در این پروژه از System File استفاده نشده است. بنابراین اطلاعات ذخیره شده روی کارت توسط کامپیوتر قابل شناسایی نخواهد بود و در صورتی که کارت به کامپیوتر متصل شود ؛ کامپیوتر برای فرمت کردن کارت از شما سوال خواهد کرد. می توانید با هر فرمتی کارت را فرمت کنید اما توجه داشته باشید که حتی پس از فرمت کردن کارت ؛ باز هم اطلاعات نوشته شده توسط مدار قابل پخش خواهد بود و تنها راه از بین بردن اطلاعات نوشته شده ؛ نوشتن اطلاعات جدید روی آن سکتور ها می باشد.

4) برای دستیابی به فرکانس نمونه برداری 8KHz حتما از کریستال خارجی استفاده کنید. استفاده از نوسانساز داخلی بدلیل تیرانس حدود 3% ؛ توصیه نمی شود.



رعایت حقوق معنوی:

مراجع و منابع:

ایده اولیه و بخش هایی از توضیحات این پروژه از App Note 335 از سایت اتمل برگرفته شده است
www.atmel.com/dyn/resources/prod_documents/doc1456.pdf

مبدل DAC مقاومتی

www.avr-asm-tutorial.net/avr_en/AVR_DAC.html

دیتاشیت کارت حافظه MMC

<http://ec.transcendusa.com>

دیتاشیت چیپ های LF33CV ؛ LM324 ؛ LM386 و ATMEGA32

www.datasheet4u.com

مجله الکتور – مارس 2008 – صفحه 26 - Data Logger “deLuxe”

www.elektor.com

کتاب AVR & Codevision نوشته محمد امین مصلاهی-انتشارات سروش دانش-چاپ دوم 1386

انحصار طلبی علمی؛ در دنیای امروز جایگاهی ندارد
بیایید تا دانسته های خود را هر چند اندک، با دیگران به اشتراک بگذاریم.

فرهنگ و تمدن بی همتای ایرانیان به سبب حمله وحشیانه مغولان نابود شد
و
امروز نوبت من و دوست که با انتشار دانسته های خود ایران را به آن جایگاه بی همتای گذشته
بازگردانیم.

Fuba unlimited intelligent



Building A Sound recorder with AVR & MMC

2009-05-08

Rev:1.0

<http://electroway.blogfa.com>

Fuba_etech@operamail.com