

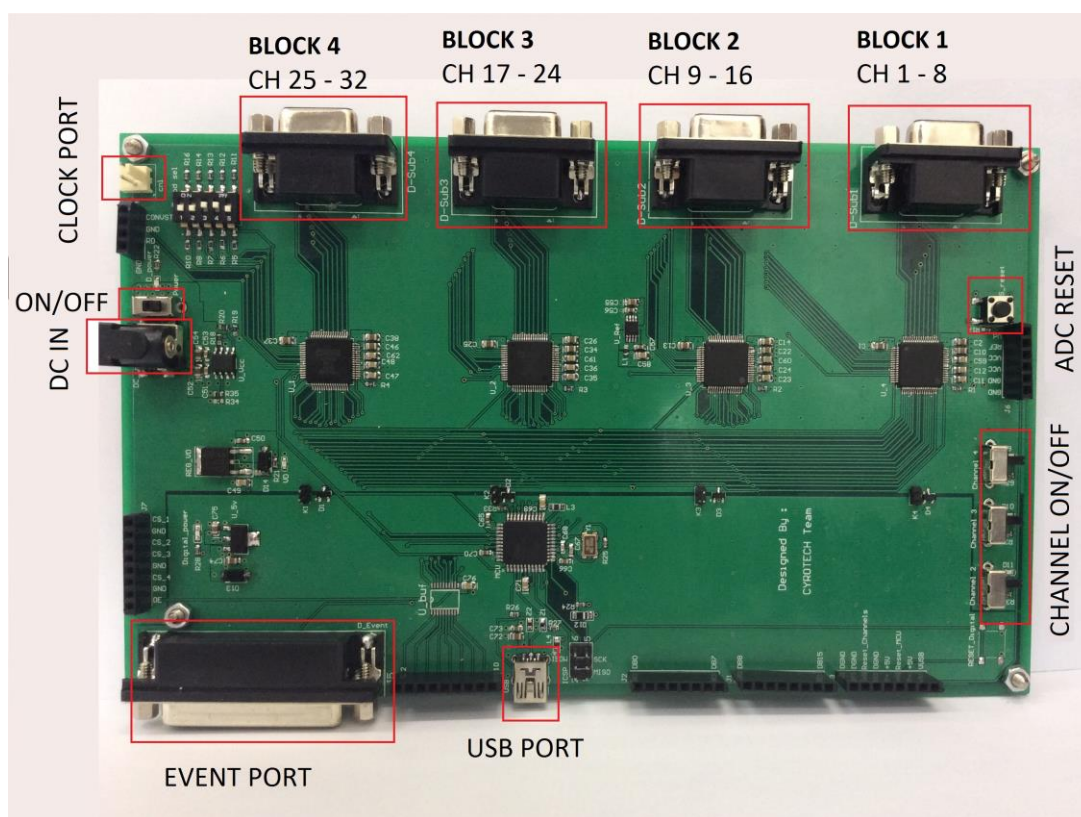
راه اندازی اولیه دستگاه نمونه برداری همزمان ۳۲ کاناله

فهرست مطالب

2 راه اندازی سخت افزار
3 راه اندازی نرم افزار
3 شروع به کار
3 دستورات رابط کاربر
5 نمونه برنامه دسترسی به داده ها به صورت real time

راه اندازی سخت افزار

- ۱- ابتدا دستگاه را با آداپتور ۱۲ ولت تغذیه کنید.
- ۲- سپس با استفاده از کابل USB Mini تعبیه شده، دستگاه را به کامپیوتر متصل کنید.
- ۳- سپس با استفاده از کابل های DB9 تعبیه شده، بلاک های ورودی آنالوگ دستگاه را به خروجی آمپلیفایر متصل کنید.
ترتیب کانال های هر بلاک، مطابق با دستگاه های آمپلیفایر می باشد. ترتیب بلاک های ورودی نیز در شکل زیر مشخص است.
- ۴- سپس دستگاه را با استفاده از کلید روشن و خاموش دستگاه، روشن کنید.
- ۵- **توجه:** برای استفاده کردن از تعداد کانال های ورودی کمتر از ۳۲ کانال، امکان تغییر تنظیمات به صورت نرم افزاری در نظر گرفته شده است؛ لذا همیشه کلید روشن و خاموش بلاک ها را در حالت روشن قرار دهید و از خاموش و روشن کردن آن بپرهیزید.
- ۶- جهت استفاده از دستگاه در حالت slave (حالتی که سیگنال کلاک توسط یک منبع خارجی به دستگاه وارد می شود)، سیگنال کلاک را به پورت کلاک دستگاه متصل کنید.
- ۷- **توجه:** اگر دستگاه روی حالت master باشد، خود دستگاه master می تواند سیگنال کلاک مورد نیاز دستگاه slave را تولید کند. به این منظور کافی است پورت کلاک هر دو دستگاه را توسط کابل تعبیه شده به هم متصل کنید.
- ۸- **توجه:** اتصال دو دستگاه در حالت master موجب آسیب هر دو دستگاه می شود. جدا از اتصال دو دستگاه در حالت master بپرهیزید.



راه اندازی نرم افزار

شروع به کار

- ابتدا نرم افزار MATLAB را اجرا کنید و Current Folder را به پوشه حاوی برنامه های DAQ.m و detect_port.m تغییر دهید.
- یک object از جنس کلاس DAQ با پارامتر های مورد نظر بسازید:

```
my_daq = DAQ([NOC],[sample_rate],[isEvent],[isSlave],[port]);
```

پارامتر هایی که برای ساختن این object عبارتند از:

NOC: تعداد کانال های مورد نیاز که عددی بین 1 و 32 است و مقدار پیش فرض آن 8 می باشد.

sample_rate: نرخ نمونه برداری بر حسب sample/second که می تواند یکی از حالت های 256 (مقدار پیش فرض) یا 512 یا 1024 باشد.

isEvent: یکی از دو مقدار 1 برای فعال کردن event و یا مقدار پیش فرض 0 برای غیر فعال کردن event. در حالتی که event فعال است، مقدار ورودی پورت event که توسط یک کابل DB25 به پورت پارالل کامپیوتر متصل است در هر نمونه برداری خوانده می شود.

isSlave: یکی از دو مقدار 1 برای حالت slave و یا مقدار پیش فرض 0 برای حالت master. در حالت slave سیگنال کلاک باید توسط منبع خارجی از طریق پورت کلاک به دستگاه داده شود. در حالت master سیگنال کلاک نمونه برداری داخلی دستگاه، به منظور هم زمان سازی روی پورت کلاک قرار می گیرد.

port: پورت اتصال DAQ به کامپیوتر برای مثال 'COM5'. اگر این ورودی مشخص نشود برنامه تلاش می کند به صورت خودکار پورت اتصال را تشخیص دهد.

- فرمان شروع نمونه برداری را صادر کنید:

```
my_daq.start();
```

حال مشاهده می کنید که دستگاه شروع به نمونه برداری کرده و داده های هر کانال روی صفحه قابل مشاهده است.

دستورات رابط کاربر

```
my_daq.start()
```

این دستور فرمان شروع نمونه برداری را صادر می کند. این دستور را فقط یک بار و هنگام شروع کار استفاده کنید. این دستور، پرچم isRunning را 1 می کند.

```
my_daq.stop()
```

این دستور نمونه برداری را متوقف کرده و پرچم isRunning را 0 می کند.

```
my_daq.resume()
```

این دستور نمونه برداری را از سر گرفته و پرچم `isRunning` را 1 می کند.

```
my_daq.record([seconds])
```

این دستور از لحظه فراخوانی به مدت `seconds` ثانیه داده های نمونه برداری شده را با فرمت `csv` و یا `txt` ذخیره می کند. برای اولین بار که این دستور فراخوانده شود، محل و نام ذخیره سازی پرسیده می شود. اگر آرگومان ورودی تابع حذف شود، عملیات ذخیره سازی تا فراخوانی دستور `stop_record` ادامه می یابد.

هر سطر فایل ذخیره شده مربوط به یک نمونه است و داده های هر کانال توسط کاما از داده کانال بعد جدا می شود. برای مثال در حالت 32 کاناله با `event` فعال، داده ها به شکل زیر ذخیره می شوند.

`Channel11,Channel12,Channel13, ... ,Channel32,Event`

داده های هر کانال به صورت یک عدد `float` بین -5 تا 5 و داده `event` به صورت یک عدد `integer` بین 0 تا 255 ذخیره می شود.

توجه: فایل ساخته شده توسط برنامه تنها پس از فراخوانی دستور `close` قابل استفاده خواهد بود.

```
my_daq.stop_record()
```

این دستور، عملیات ذخیره سازی را متوقف می کند.

```
my_daq.get_data([samples])
```

از لحظه شروع به نمونه برداری، داده های نمونه برداری شده در بافر `FIFO` خروجی انباشته می شوند. این دستور به تعداد `samples` از داده های بافر را در قالب یک ماتریس بر می گرداند و این داده ها را از بافر خروجی حذف می کند. آرایش داده ها در ماتریس خروجی مشابه آرایش داده ها در فایل `csv` ذخیره شده است:

`Channel11 | Channel12 | ... | Event`

اگر آرگومان ورودی حذف شود، این تابع تمام داده های بافر خروجی را تحویل می دهد. همچنین اگر هیچ داده ای در بافر خروجی نباشد، تابع یک ماتریس خالی بر می گرداند.

```
my_daq.available()
```

این تابع، تعداد نمونه های آماده خواندن در بافر خروجی را بر می گرداند.

```
my_daq.isRunning()
```

این پرچم هنگام نمونه برداری دستگاه 1، و در سایر مواقع 0 است. دستور های start و resume این پرچم را 1، و دستور های stop و close پرچم را 0 می کنند. هم چنین اگر در حین کار دستگاه خاموش و یا اتصال usb قطع گردید. این پرچم 0 می شود.

```
my_daq.cnt
```

این متغیر، تعداد نمونه‌هایی که از اول کار دستگاه نمونه برداری شده را نشان می‌دهد.

```
my_daq.close()
```

این دستور موجب اتمام کار دستگاه می شود. پس از استفاده از این دستور، برای استفاده مجدد از دستگاه، لازم است توسط کلید روشن و خاموش، دستگاه را خاموش و روشن کنید. این دستور هم چنین باعث می‌شود پرچم isRunning مقدار 0 شود.

توجه: بستن نمودار داده‌ها نیز خود به خود باعث فراخوانی این دستور و توقف کار دستگاه می‌شود.

نمونه برنامه دسترسی به داده‌ها به صورت **real time**

```
clear
close all
figure
daq1 = DAQ(32, 1024, 0, 0);
daq1.start;
while(daq1.isRunning)
    if daq1.available >= 512
        data = daq1.get_data(512);
        data_fft = fft(data(:,1));
        plot(20*log(abs(data_fft(1:256)))));
    end
end
end
```

در این برنامه ابتدا یک object از کلاس DAQ ساخته شده که تعداد کانال‌های تعریف شده آن 32، و نرخ نمونه برداری 1024 نمونه بر ثانیه است. سپس نمونه برداری توسط daq1.start شروع شده. حال مادامی که نمونه برداری در جریان است daq1.isRunning مقدار یک دارد و حلقه while برقرار است. در این حلقه ابتدا صبر کرده‌ایم تا 512 داده آماده شود:

```
if daq1.available >= 512
```

سپس همین تعداد داده را از بافر خروجی خوانده و از آن fft گرفته و نمایش دادیم. اگر در حین این برنامه به هر دلیل دستگاه خاموش و یا اتصال آن قطع گردد و یا پنجره نمودار ها بسته شود، پرچم isRunning مقدار 0 شده و برنامه از حلقه خارج شده و به اتمام می‌رسد.