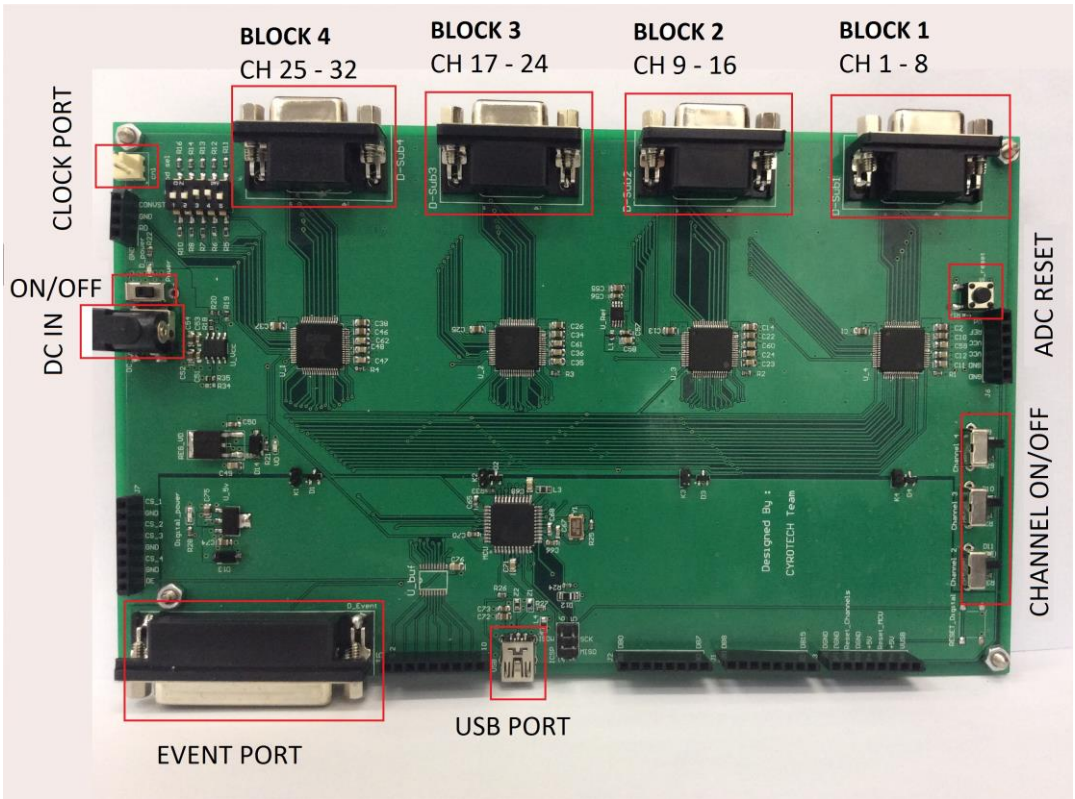
راه اندازی اولیه دستگاه نمونه برداری همزمان ۳۲ کاناله

فهرست مطالب



راه اندازی سخت‌افزار

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 کانال‌های هر بلاک، مطابق با دستگاه‌های آمپلیفایر می باشد. ترتیب بلاک های ورودی نیز در شکل مشخص است.

۴. سپس دستگاه را با استفاده از کلید روشن و خاموش دستگاه، روشن کنید.

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.

۵. 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.

توجه: اگر دستگاه روی حالت master باشد، خود دستگاه master می‌تواند 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.

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 نرم‌افزار MATLAB را اجرا کنید و Current Folder را به پوشه حاوی برنامه‌های 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]);
```

پارامتر هایی که برای ساختن این object عبارتند از:

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

sample_rate: نرخ نمونه برداری بر حسب sample/second که می‌تواند یکی از حالت‌های 256 (مقدار پیش فرض)، 512 و 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 هم‌زمان سازی روی پورت کلاک قرار می‌گیرد.

- فرمان شروع نمونه برداری را صادر کنید:

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

حال مشاهده می‌کنید که دستگاه شروع به نمونه برداری کرده و داده‌های 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 ۰ می کند.

```
my_daq.resume()
```

این دستور نمونه برداری را از سر گرفته و پرچم isRunning را ۱ می کند.

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

این دستور از لحظه فراخوانی به مدت seconds ثانیه داده‌های 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 می‌یابد.

هر سطر فایل ذخیره شده مربوط به یک نمونه است و داده‌های هر کانال توسط کاما از داده کانال بعد جدا می شود. برای مثال در حالت ۳۲ کاناله با event فعال، داده‌ها به شکل زیر ذخیره می شوند.

```
Channel1,Channel2,Channel3, … ,Channel32,Event
```

داده‌های هر کانال به صورت یک عدد float بین 5- تا 5+ و داده event به صورت یک عدد integer بین 0 تا 255 ذخیره می‌شود.

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، داده‌های نمونه برداری شده در بافر FIFO خروجی انباشته می‌شوند. این دستور به تعداد samples از داده‌های بافر را در قالب یک ماتریس بر می گرداند و این داده‌ها را از بافر خروجی حذف می‌کند. آرایش داده‌ها در ماتریس خروجی مشابه آرایش داده‌ها در فایل csv ذخیره شده است:

```
Channel1 | Channel2 | … | Event
```

اگر آرگومان ورودی حذف شود، این تابع تمام داده‌های بافر خروجی را تحویل می دهد. همچنین اگر هیچ داده‌ای در بافر خروجی نباشد، تابع یک ماتریس خالی بر می گراند.

```
my_daq.available()
```

این تابع، تعداد نمونه‌های 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
```

در ای برنامه ابتدا یک object از کلاس DAQ ساخته شده که تعداد کانال‌های تعریف شده آن ۳۲، و نرخ نمونه برداری ۱۰۲۴ نمونه بر ثانیه است. سپس نمونه برداری توسط daq1.start شروع شده. حال مادامی که نمونه برداری در جریان است daq1.isRunning مقدار 1 دارد و حلقه while برقرار است. در این حلقه ابتدا صبر کرده‌ایم تا ۵۱۲ 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 می رسد.