

"به نام خداوند بخشنده مهربان"

گزارش پروژه ی درس ابزار دقیق دستگاه محاسبه و تحلیل نوار قلب

تهیه کنندگان:

امین سرخی لله لو، محمد مهدی شجاعی فر،

سپهر قمری، درسا نظری،

سید محمد صالح میرزا طباطبایی

استاد:

دکتر شریفی

دانشگاه صنعتی امیرکبیر (دانشکده مهندسی برق)

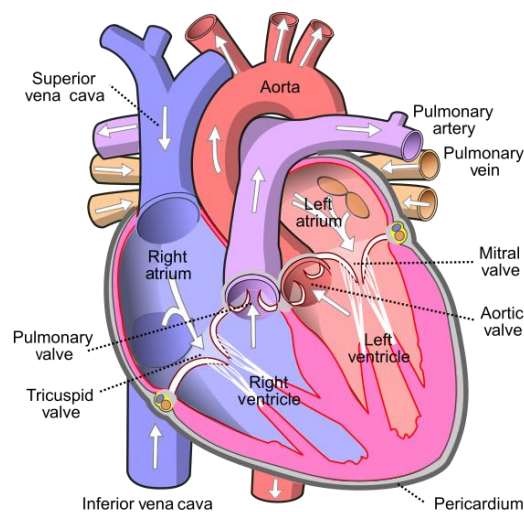
مرداد ۱۳۹۹

ECG مخفف واژه‌ی Electrocardiogram یا Electrocardiograph است. این لغت در بعضی کشورها EKG نامیده می‌شود. الکتروکاردیوگراف دستگاهی است که جریان‌های الکتریکی قلب را از طریق الکترودهایی که روی مناطق مختلف پوست بدن قرار داده می‌شوند، دریافت کرده و آن‌ها را به شکل یک نمودار ترسیم می‌کند. این نمودار الکتروکاردیوگرام نامیده می‌شود.

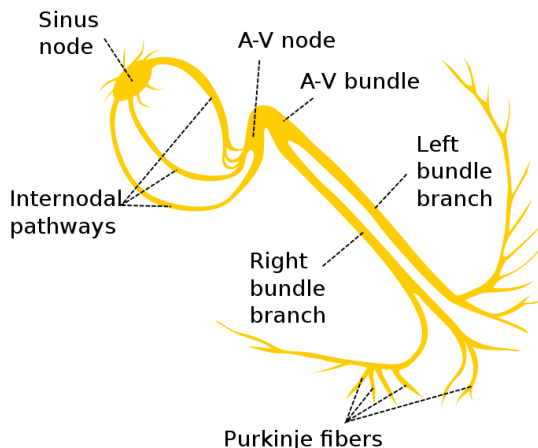
انقباض تمام ماهیچه‌های بدن در اثر یک تغییر الکتریکی به نام دپولاریزاسیون (depolarization) ایجاد می‌شود. اگر الکترودهایی را بر روی سطح پوست بچسبانیم، این جریانات قابل دریافت هستند.

قلب نیز یک ماهیچه است؛ پس از این قانون مستثنی نیست. جریانات الکتریکی قلب، به شرط شل بودن سایر ماهیچه‌های بدن، توسط دستگاه الکتروکاردیوگراف قابل دریافت و ثبت هستند.

ساختار درونی قلب از نظر بافت ها (دریچه ها - بطن ها - دهلیز ها)



ساختار درونی قلب از نظر بخش های الکتریکی



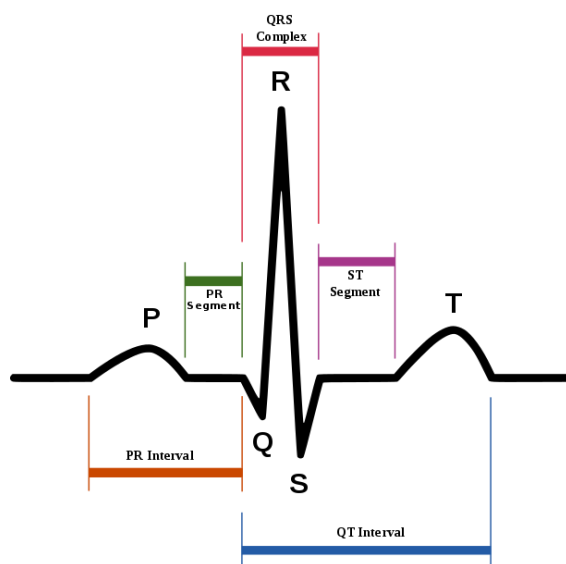
جرقه‌ی هر چرخه‌ی قلبی در نقطه‌ای از دهلیز راست قلب به نام گره سینوسی - دهلیزی (sinoatrial node/ SA node) زده می‌شود. جریان الکتریکی تولید شده، سبب دپولاریزاسیون سلول‌های قلب می‌گردد، دپولاریزاسیون نیز انقباض سلول‌ها را به دنبال دارد. جریان الکتریکی از طریق مسیرهای هدایتی در نقاط مختلف قلب توزیع می‌شوند. این مسیرها را در شکل زیر می‌بینید:

جریان الکتریکی پس از خروج از گره سینوسی - دهلیزی توسط مسیرهای بین گره‌ای (internodal pathways) در دو دهلیز راست و چپ توزیع می‌شوند. سپس جریان برای عبور از دهلیزها و رن به مناطق پایین‌تر (بطن‌ها) می‌بایست از ساختاری به نام گره دهلیزی - بطنی (atrioventricular node/ AV node) عبور کند.

جریان الکتریکی در این نقطه مقداری توقف می‌کند و سپس وارد شاه‌راهی به نام شاخه هیس (bundle of His) می‌شود. در ادامه این شاه‌راه به دو مسیر به نام‌های شاخه‌های دسته‌ای راست و چپ (right and left bundle branches) تقسیم می‌شود که جریان را در بطن‌های

راست و چپ توزیع می‌کنند. مسیرها، نهایتاً به الیاف‌های بسیار باریکی به نام الیاف پورکینژ (Purkinje fibers) می‌رسند که این الیاف امواج الکتریکی را به سلول‌های میوکارد منتقل می‌کنند.

ساختار سیگنال ECG دریافت شده و موج های P QRS T بر روی سیگنال دریافت شده



سیکل قلبی همراه با الکتروکاردیوگرافی

مرحله ی اول:

در این مرحله بطن‌ها و دهلیزها در حال استراحت‌اند. خون تیره به‌وسیله بزرگ سیاهرگ‌های زیرین و زیرین (وریدهای اجوف فوقانی و تحتانی)، به دهلیز راست می‌ریزد. این خون به‌خاطر وزنش، از طریق دریچه‌های دهلیزی-بطنی - که به‌هنگام پایان موج T سیکل قلبی پیشین باز شده‌اند- وارد بطن می‌شود و آن‌ها را تا حدی پر می‌کند. اما برای اینکه خون دهلیزها به‌طور کامل وارد بطن شود، دهلیزها باید منقبض شوند. لازم است ذکر شود هر ماهیچه‌ای در قلب که بخواهد منقبض شود یا استراحت کند، ابتدا باید موج انقباض یا استراحتش در تمام نقاط آن ماهیچه منتشر شود. پس برای انقباض دهلیزها، ابتدا باید پیام انقباض در سراسر آن‌ها منتشر بشود. این کار توسط بافت گرهی دهلیز انجام می‌شود. در بین دو دهلیز این تنها دهلیز راست است که دارای بافت گرهی است، از سویی کانون زایش انقباضات قلب نیز که همان گره پیش‌آهنگ (SA node) می‌باشد در دیواره پشتی دهلیز راست و در زیر منفذ بزرگ سیاهرگ زیرین قرار دارد. پس برای انقباض ابتدا گره پیش‌آهنگ به صورت ریتم خودبه‌خودی تحریک می‌شود و این پیام انقباض را از طریق ۳ رشته گرهی دهلیز راست به گره دهلیزی-بطنی - که در حد فاصل بین دیواره دهلیزها و بطن و کمی متمایل به دهلیز راست قرار دارد - هدایت می‌کند. طی حرکت پیام از پیش‌آهنگ به دهلیزی-بطنی، میون‌های میوکارد قلب که در مسیر انتقال این پیام قرار دارند، منقبض شده و این انقباض از میونی به میون دیگر در دهلیز راست انتشار می‌یابد و نهایتاً از طریق میون‌های دهلیز راست به میون‌های دهلیز چپ نیز منتشر شده و کل دهلیزها را فرا می‌گیرد. البته این پیام نمی‌تواند از طریق میون‌های دهلیزها به میون‌های بطن منتقل شود، چون در دیواره بین بطن و دهلیزها بافت پیوندی رشته‌ای عایقی قرار دارد که باعث می‌شود انتقال پیام از دهلیزها به بطن تنها از طریق بافت گرهی که از وسط این عایق رد می‌شود صورت گیرد. اگر این بافت عایق نبود، دهلیزها و بطن هم‌زمان به هم منقبض می‌شدند و کارایی قلب بسیار پایین می‌آمد؛ چون در این حالت، پس از پمپاژ مقدار کمی خون به بطن، آن‌ها نیز همین مقدار کم را به سمت بدن و شش‌ها پمپ می‌کردند و خون کمی به آن‌ها می‌رسید. پس از این‌که این پیام به‌طور کامل

سراسر دهلیز را فراگرفت، در الکتروکاردیوگرام موج P ثبت می‌گردد. بلافاصله بعد از آن، مدت استراحت عمومی قلب، یعنی ۰.۴ ثانیه به اتمام می‌رسد.

- ورود بیش از ۷۵٪ خون از دهلیزها به بطن بر اثر وزن خون
- انتشار موج انقباض دهلیزها در سراسر دهلیزها ---- ایجاد موج P در الکتروکاردیوگرام
- سینی‌ها ---- بسته؛ تا خون وارد شده به بطن از طریق این دریچه‌ها وارد سرخرگ‌ها نشود.
- لختی‌ها ---- باز؛ تا خون دهلیزها وارد بطن شوند.

مرحله دوم: انقباض دهلیزها

۱. انتشار موج انقباض بطن
 ۲. انتشار موج استراحت در دهلیزها
- انقباض دهلیزها ---- وقوع از P تا پایان R (انقباض، تغییری در الکتروکاردیوگرام ایجاد نمی‌کند، چون پیام الکتریکی منتقل نمی‌شود)
 - انتشار پیام انقباض بطن ---- ایجاد موج QRS در الکتروکاردیوگرام
 - انتشار پیام استراحت دهلیزها ---- زیر موج پیام قوی انتشار انقباض بطن گم می‌شود.
 - شنیدن صدای اول قلب.
 - سینی‌ها ---- بسته؛ به دلیل یک طرفه بودن دریچه‌های سینی و فشارخون موجود در سرخرگ‌ها
 - لختی‌ها ---- باز؛ تا خون دهلیزها وارد بطن شود.

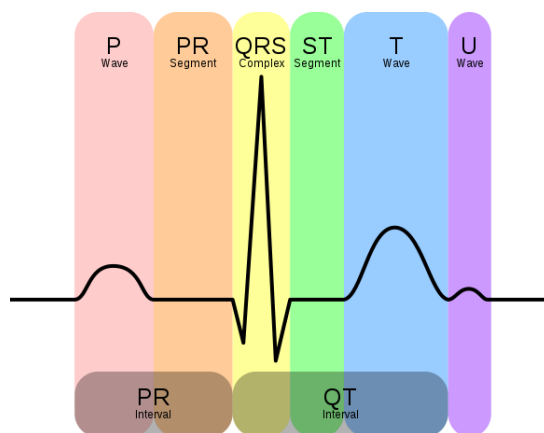
مرحله سوم: انقباض بطن‌ها

موارد روی داده در موج S :

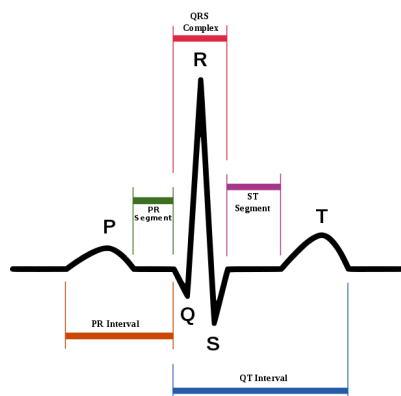
۱. استراحت دهلیزها و انقباض بطن
 ۲. بسته شدن دریچه‌های لختی بر اثر فشار خون ناشی از انقباض بطن
 ۳. باز شدن دریچه‌های سینی بر اثر فشار خون ناشی از انقباض بطن
- حداکثر انقباض بطن در ابتدای موج T انجام می‌شود.
 - زمان استراحت بطن ۰.۵ ثانیه و زمان استراحت دهلیزها ۰.۷ ثانیه است.
 - باز شدن دریچه صدایی ایجاد نمی‌کند.

- انقباض بطن به مدت ۰.۳ ثانیه --- فاصله S تا T استراحت دهلیزها
- شنیدن صدای دوم قلب
- سینی‌ها --- ابتدا باز (درهنگام انقباض بطن) و بعد بسته (پس از اتمام انقباض بطن)
- لختی‌ها --- ابتدا بسته (درهنگام انقباض بطن) و بعد باز (پس از اتمام انقباض بطن)

مدت زمان	شرح	نام
زیر ۸۰ میلی ثانیه	موج P نشان دهنده دپلاریزاسیون دهلیز است. دپلاریزاسیون دهلیزی از گره SA به سمت گره AV و از <u>دهلیز</u> راست به <u>دهلیز</u> چپ گسترش می یابد.	موج P
۱۲۰ تا ۲۰۰ میلی ثانیه	فاصله PR از ابتدای موج P تا ابتدای مجموعه QRS اندازه گیری می شود. این فاصله زمانی را نشان می دهد که پالس الکتریکی برای عبور از گره سینوس از طریق گره AV طول می کشد.	فاصله PR
۸۰ تا ۱۰۰ میلی ثانیه	مجموعه QRS نشان دهنده دپلاریزاسیون سریع بطن های راست و چپ است. بطن ها نسبت به دهلیز توده عضلانی بزرگی دارند، بنابراین مجتمع QRS معمولاً دامنه بسیار بزرگتری نسبت به موج P دارد.	مجموعه QRS
	قطعه ST مجموعه QRS و موج T را به هم متصل می کند. این دوره ای است که بطن ها دپولاریزه می شوند.	بخش ST
۱۶۰ میلی ثانیه	موج T بیانگر دفع مجدد بطن ها است. به طور کلی به طور مستقیم در تمام سربها به جز aVR و S _{V۱} است.	موج T
زیر ۴۰ میلی ثانیه	فاصله QT از ابتدای مجموعه QRS تا انتهای موج T اندازه گیری می شود. دامنه قابل قبول با ضربان قلب متفاوت است، بنابراین باید با تقسیم بر ریشه مربع فاصله RR به QTc اصلاح شود.	فاصله QT اصلاح شده (QTc)

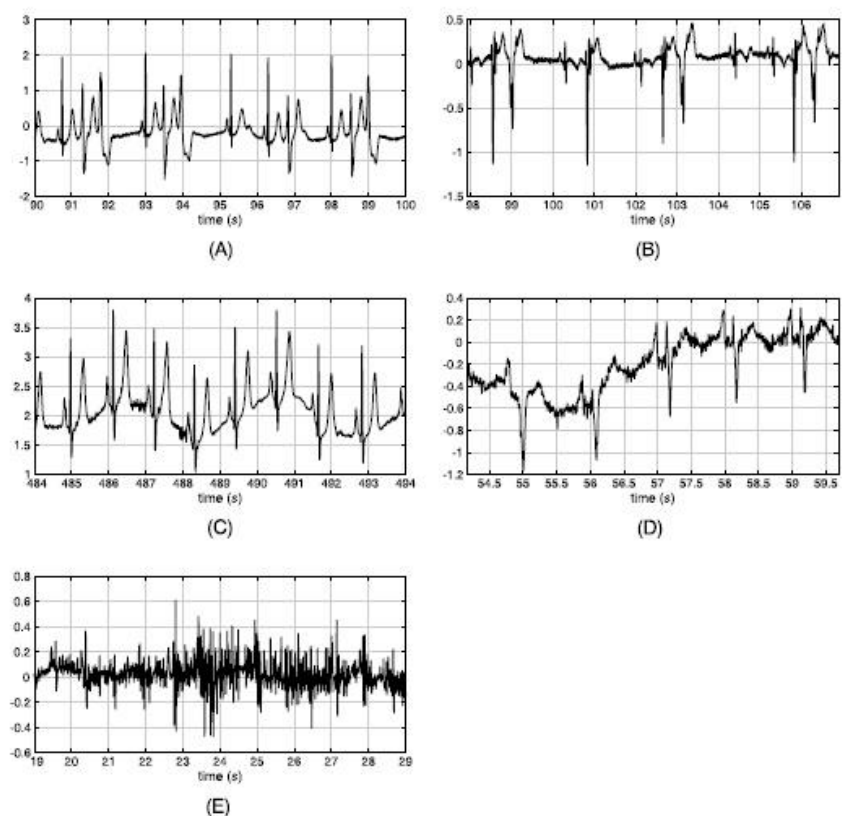


مجموعه ی QRS:



مجموعه QRS نشان دهنده دپلاریزاسیون سریع بطن های راست و چپ است. بطن ها نسبت به دهلیز توده عضلانی بزرگی دارند، بنابراین مجتمع QRS معمولاً دامنه بسیار بزرگتری نسبت به موج P دارد. (حدود زمانی ۸۰ تا ۱۰۰ میلی ثانیه)

برای تشخیص مجموعه ی QRS در سیگنال ECG خام دریافتی پس از فیلتر های دیجیتال اولیه؛ باید داده را بررسی کرد و زمان هایی که در سیگنال به عنوان مجموعه زمانهای QRS تشخیص داده می شوند را به خروجی داد. البته با وجود نویز در سیگنال ECG دریافتی، تشخیص مجموعه ی QRS نیز به مراتب سخت تر خواهد شد. در شکل زیر چند نمونه از نویز های موجود در سیگنال را مشاهده می کنید:



- (A) سیگنال ECG همراه با تغییر شکل و دگرگونی مجموعه ی QRS
- (B) سیگنال ECG همراه با تغییر شکل و دگرگونی مجموعه ی QRS به علاوه ی تغییر دامنه ی QRS
- (C) سیگنال ECG دارای دامنه ی زیاد موج T
- (D) سیگنال ECG دارای دامنه ی زیاد موج P
- (E) سیگنال ECG با نویز های شدید شبیه به مجموعه ی QRS که برای تشخیص سخت است.

در حالت کلی در همه ی روش های تشخیص مجموعه ی QRS مراحل مطابقی زیر طی خواهد شد البته فیلتر ها و روش ها در هر بلوک برای هر الگوریتم متفاوت خواهد بود و این تغییرات دقت، صحت، زمان محاسبه ی الگوریتم را تعیین خواهند نمود.



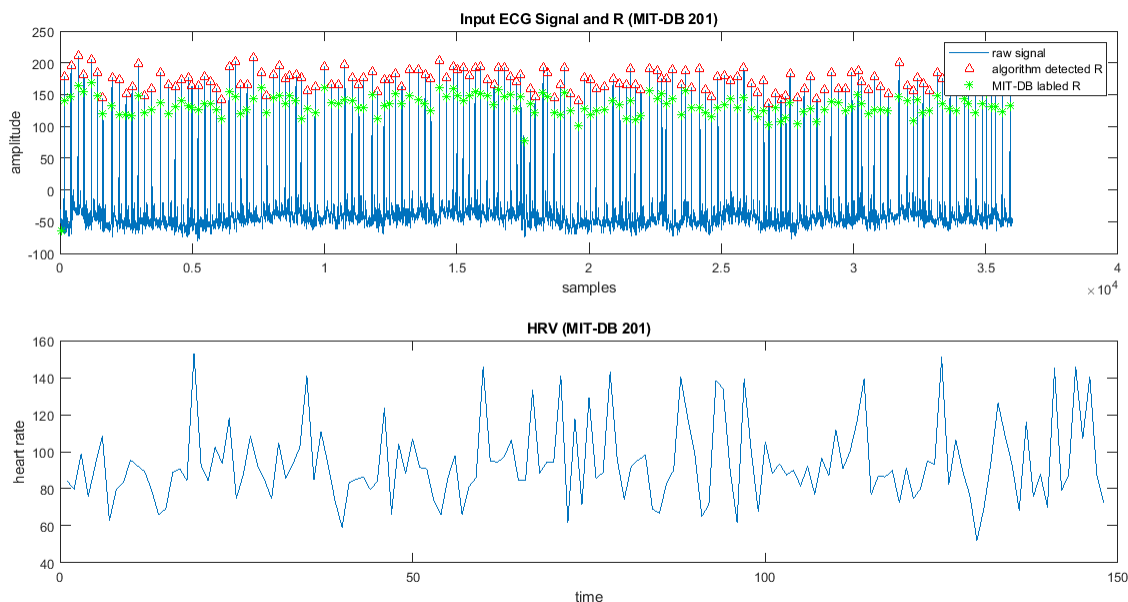
همانطور که مشاهده می شود در مرحله ی pre-processing از یک فیلتر خطی (اعم از مشتق اول یا دوم یا مراتب بالاتر – DWT موجک ها – تبدیل Hilbert – Moving average – باند پاس فیلتر دیجیتال) استفاده می شود.

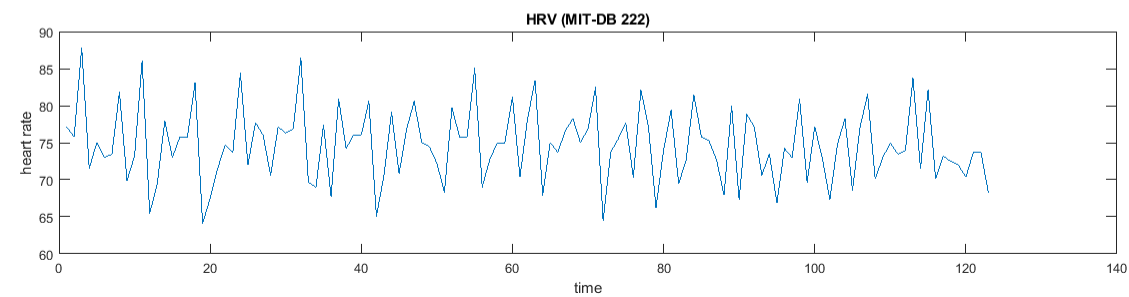
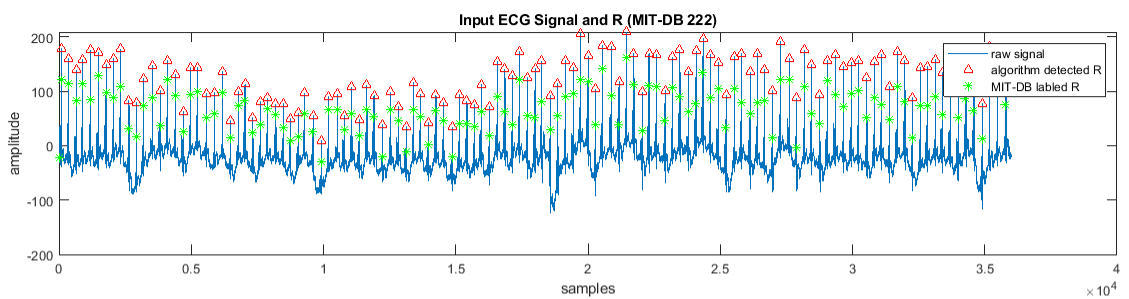
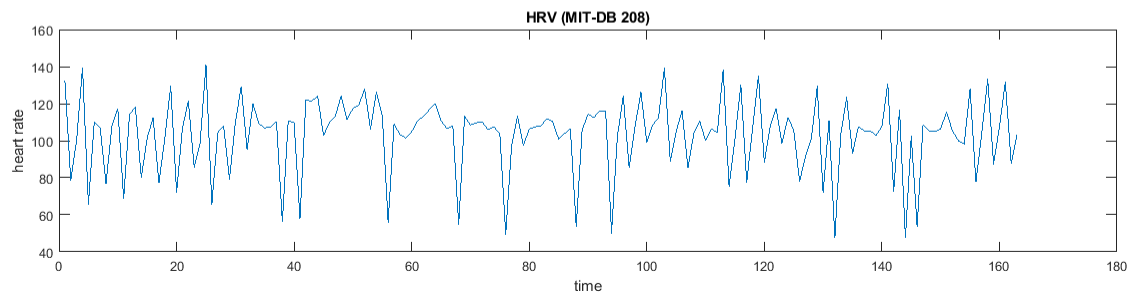
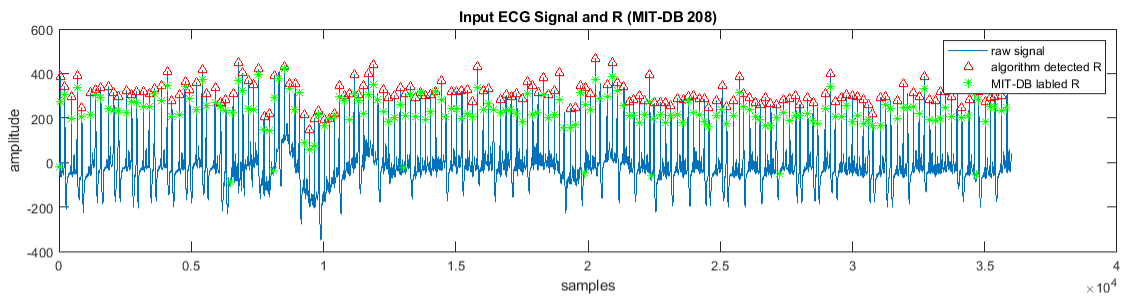
همچنین در این مرحله یک فیلتر غیر خطی (اعم از square function – محاسبه ی طول منحنی و ...) نیز موجود می باشد.

در نهایت یک مرحله ی تصمیم گیری موجود هست که در این مرحله بر اساس سیگنال فیلتر شده قبلی زمان های هر موج یعنی Q R S در هر فاصله ی زمانی تعیین خواهند شد.

برای فیلتر کردن سیگنال در این پروژه از فیلتر های ناچ، باند پس بین ۰.۵ تا ۴۰ هرتز، فیلتر smoothing و نرمالایز و همچنین همپل فیلتر استفاده شده است.

همچنین برای پیاده سازی الگوریتم دریافت ضربان قلب پس از مطالعه ی چندین مقاله از الگوریتم مقاله **Efficient Real-Time R and QRS Detection Method Using a Pair of Derivative Filters and Max Filter for Portable ECG Device** جهت یافتن تغییرات ضربان قلب یا همان HRV بهره برده شده است. در زیر می توانید نتایج الگوریتم را بر روی تعدادی از دیتاست های دیتابیس MIT-DB موجود در وبگاه فیزیو نت مشاهده بفرمایید:





ارتباط آردوینو و رسیپی پای از طریق سریال



VS.



ابتدا ۱۰۸۰ داده واقعی را در یک آرایه قرار می دهیم. این داده ها سه ثانیه سیگنال واقعی می باشند. برای این کار از یک حلقه for استفاده می کنیم که در داخل این حلقه یک حلقه دیگر وجود دارد. حلقه درونی برای تبدیل دیتا خوانده شده به عدد دیجیتال می باشد تا بتوانیم آن را به یک D^2A بدهیم و سیگنال دیجیتال را به آنالوگ تبدیل کنیم و آن را به ورودی فیلتر ها دهیم.

برای تبدیل کردن به عدد باینری از ساده ترین روش استفاده می کنیم، به عنوان مثال عدد ۱۴ را تقسیم بر دو می کنیم که باقی مانده آن صفر و خاج قسمت ۷ می شود، صفر بیت با کمترین ارزش می باشد، سپس ۷ را تقسیم بر ۲ می کنیم که باقی مانده آن یک میشود، پس بیت دوم برابر با یک می باشد و به همین صورت ادامه می دهیم.

پایه های ۲ تا ۹ آردوینو را به عنوان خروجی در نظر می گیریم که عدد دیجیتال صفر یا یک می باشد. با توجه به اینکه کم ارزش ترین بیت به پین ۹ و با ارزش ترین بیت به پین دو داده می شود از یک حلقه for برعکس استفاده می کنیم و با توجه به اینکه فرکانس نمونه برداری ۳۶۰ هرتز می باشد، برای اینکه داده ها واقعی باشند تاخیر ۲۷۷ میلی ثانیه قرار می دهیم.

این کد آردوینو با یک D^2A تبدیل به سیگنال آنالوگ می شود و به تعدادی فیلتر داده می شود. خروجی این فیلتر در بازه ۱۶- تا ۴ ولت می باشد. برای اینکه بتوانیم این سیگنال را با یک آردوینوی دیگر بخوانیم، باید آن را به بازه ۳.۳ تا ۰ ولت مپ کنیم. برای این کار یک زیرو اسپن طراحی می کنیم که با توجه به فرمول های جزوه

$$R_F = 330k, R_i = 200k, R_{os} = 120k, R_{comp} = 86.72k$$

تنها نکته ای که وجود دارد این است که عددی که خوانده شده از ۱۰۲۴ تا ۰ می باشد که باید به بازه ۳.۳ ولت مپ شود و همچنین با توجه به اینکه یک زیرو اسپن قرار داده ایم، با استفاده از آن فرمول $V_{in} = 6.06 V_{out} - 16$ تبدیل به همان عدد خروجی فیلتر می کنیم و داده ها را به رسیپی پای انتقال می دهیم.

جواب پرسش اول: در برد آردوینو پین RX_0 و پین TX_1 می باشد و در رسیپی پای پین TX_8 و پین RX_{10} می باشد.

جواب پرسش دوم: در برد رسیپی پای واقعی از ASM^0 استفاده می نماییم.

ارسال اطلاعات و نمایش آنها در شبکه لوکال:

ارسال / اطلاعات:

برای انجام این بخش از روشی متفاوت با روش تدریس شده توسط استاد استفاده کرده ایم که علت استفاده از این روش و توضیحات آن در ادامه آمده است.

در ابتدا یک صفحه [index.html](#) تعریف شده است که در قسمت نمایش اطلاعات به توضیح کدهای آن میپردازیم

در ادامه صفحه `live_signal` تعریف شده است که خروجی این صفحه مقدار `output` را به عنوان مقدار پارامتر `final_signal` نمایش میدهد. همچنین تابع `gen` این وظیفه را بر عهده دارد که دیتا ها را ایجاد و با فرکانس مختلفی خروجی دهد که شیوه کار آن در بخش قبل توضیح داده شده است.

اما مهم ترین تابع این کد، تابع `run_signal` می باشد. اگر به بالای کد دقت کرده باشید ما یک متغیر به نام `output` و یک `threading.Lock()` تعریف کرده ایم. در واقع با استفاده از متغیر `lock` از ارسال ناقص اطلاعات جلوگیری خواهیم کرد. اگر در این تابع نگاه کنید یک حلقه تا بی نهایت در حال اجرا می باشد. در این حلقه خروجی تابع `gen` در متغیر `temp` ذخیره می شود. اما برای اپدیت کردن متغیر خروجی ما یعنی `output` از ترد ها استفاده میکنیم. به این صورت که تا زمانی که کدهای داخل اسکوپ `with lock` به طور کامل انجام نشده باشند، همه متغیرها (که در این جا فقط `output` منظور است) همان مقدار قبلی را دارا خواهند بود و به محض اینکه این اسکوپ تمام شوند، تمام متغیرها اپدیت می شوند و هیچ گاه متغیر خروجی ما یعنی `output` ناقص دریافت نخواهد شد و دیگر نیازی نیست از `try catch` که توسط استاد تدریس شده است استفاده کنیم و همیشه یک داده درست را می توانیم ارسال کنیم.

در `main` نیز ما یک ترد ساخته و به آن تابع `run_signal` را ورودی می دهیم. همچنین پراپرتی `daemon` را فعال می کنیم تا این تابع همیشه در پس زمینه در حال اجرا باشد.

نمایش اطلاعات:

کد این بخش به صورت زیر می باشد:

برای نمایش داده های `real-time` به صورت محلی از `html` و `javascript` و `css` کمک میگیریم. ابتدا کتابخانه ها `plotly.js` و `jquery` را اضافه میکنیم. سپس برای زیباسازی در تگ `style` رنگ پس زمینه و نوشته ها را به صورت `rgb` از `palette` موجود در اینترنت دریافت کردیم و به صورت وسط چین آنرا تعیین نمودیم. حال برای نمایش سیگنال `ECG` جدولی تهیه شد که عدد دریافتی در آن به صورت `real-time` نمایش داده میشود و برای نمایش بهتر آن با استفاده از کتابخانه `plotly` نمودار `real-time` اعداد را نمایش دادیم.

در ابتدا لازم است دیتا های ارسال شده توسط رزبری را دریافت کنیم. پس کافی است با فرکانس تعیین شده داده های موجود در صفحه `live_signal` را بخوانیم. پس از خواندن اطلاعات `jsonify` شده باید عدد مورد نظرمان را جدا کنیم که این عدد با پارامتر `final_signal` مشخص می شود. در آخر نیز این عدد را در متغیر `signal` ذخیره می کنیم تا در صفحه پلات شود. همچنین به عدد خوانده شده ای دی `live-signal` را نسبت می دهیم تا در هر جایی قابل نمایش باشد.

پرسش

- در جواب سوال پرسیده شده در دستورکار نیز بیان شود که ما از ایدی ۰.۰.۰.۰ استفاده میکنیم تا هر ایپی متصل به آن شبکه بتواند اطلاعات خروجی در رزبری را مشاهده کند.

با استفاده از تابع `Plotly.plot` به رسم نمودار خطی میپردازیم سپس برای گرفتن داده بعدی و به روز رسانی نمودار برای هر ۳ میلی ثانیه از `setinterval` استفاده میکنیم که در آن تابعی تعریف میشود که از دستور مرسوم `Plotly.extendTraces` برای بروزرسانی نمودار استفاده میشود و اگر بخواهیم نمودار را به صورت گذرا ببینیم باید از تابع `Plotly.relayout` برای تغییر محور `X` بهره ببریم و برای نمایش تعداد داده ای که میخواهیم نمایش دهیم از یک کانتر استفاده می کنیم.

در نهایت نمایش اطلاعات به صورت زیر است:

```
<html>
<head>
  <title>Local Network</title>
  <script src="https://cdn.plot.ly/plotly-1.37.1.min.js"></script>
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
  <style>
    body {
      background-color: rgb(0, 106, 113);
      text-align: center;
      color: rgb(211, 222, 50);
      font-family: Arial, Helvetica, sans-serif;
    }
  </style>
</head>
<body>

  <h1>Streaming!</h1>
  <table style="width:20%; border:.5px solid black;margin-left:auto;margin-right:auto;">
    <tr>
      <th>Heart ECG Signal</th>
    </tr>
    <tr>
      <td><p id = 'live-signal' style="text-align: center">NaN</p></td>
    </tr>
  </table>
  <p></p>
  <div class=wrapper>
    <div id="chart"></div>
    <script>
      var signal = 0
      setInterval(function(){
        fetch('{{ url_for('live_signal') }}').then(function(x) {
          return x.json();
        }).then(function(data) {
          document.getElementById('live-signal').innerHTML = data['final_signal'];
          signal = data['final_signal']
        });
      }, 3);

      Plotly.plot('chart', [{
        y:[[signal]],
        type:'line'
      }]);
      var cnt=0;

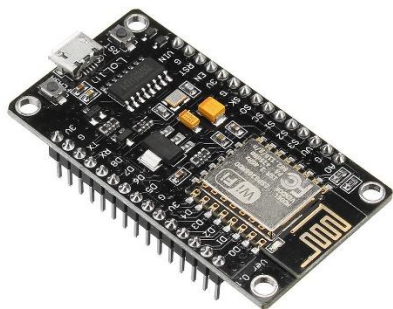
      var cnt=0;

      setInterval(function(){
        Plotly.extendTraces('chart', {y:[[signal]]}, [0]);
        cnt++;
        if(cnt>20000){
          Plotly.relayout('chart', {xaxis:{range:[cnt-20000, cnt]}});
        }
      }, 3);

    </script>
  </div>
</body>
</html>
```

قسمت بردن بر روی برد حقیقی Nodemcu و شبکه ی خارجی و thingsboard

دریافت اطلاعات ورودی:



از آنجایی که در این پروژه از سنسور واقعی استفاده نشده است ، بنابراین داده های واقعی وجود ندارند ، بنابراین از همان داده های مصنوعی استفاده می شود ، با این تفاوت که در این جا در ابتدا خروجی فیلتر های آنالوگ از طریق پورت به رزبری فرستاده شد. سپس این داده ها که از جنس byte هستند ، decode شده و خروجی str گرفته می شود و در نهایت داده های str شده داخل یک فایل به نام ans.txt ذخیره می گردند.

برای انتقال این داده ها به محیط آردوینو به دلایل ذکر شده در بالا ، داده های ذخیره شده در ans به صورت دستی در یک آرایه float با نام Data ذخیره می گردند.

پردازش بر روی داده های ورودی :

جهت ارسال مناسب داده ها از editor آردوینو و سخت افزار nodemcu استفاده شده است .

با توجه به توضیحات استاد در کلاس مجازی جهت دریافت اطلاعات هر سنسور از library مخصوص آن در محیط آردوینو استفاده میگردد اما از آنجایی که کتابخانه مربوط به سنسور ecg یافت نشد و همچنین جهت اطمینان بیشتر از خوانایی کد ، کتابخانه مربوط به سنسور DHT در editor اضافه شده است ، با این تفاوت که در تابع getAndSendTemperatureAndHumidityData همه توابع مربوط به دریافت دما و رطوبت حذف شده اند .

جهت آماده سازی داده داخل تابع getAndSendTemperatureAndHumidityData یک حلقه for نوشته شده است. در مرحله اول محتویات آرایه Data در یک متغیر به نام h ریخته می شود . سپس این متغیر همراه با زیباسازی هایی در سریال چاپ می گردد. جهت ارسال متغیر payload که شامل اعداد str شده به همراه زیباسازی هستند ، در آرایه کاراکتری attributes ذخیره شده و در نهایت ارسال می گردند. سایر کدهای نوشته در این قسمت جهت برقراری ارتباط esp با محیط رزبری است که همه از طریق اضافه کردن کتابخانه nodemcu در محیط کدنویسی اضافه می گردند.

دریافت اطلاعات :

جهت دریافت اطلاعات از سرویس thingsboard استفاده شده است. راه اندازی های مورد نیاز انجام شد و در نهایت سرویس run شد . در نهایت با توجه به آموزش ها نمودار متناسب با داده های ارسال شده رسم شد.



موفق باشید...