

به نام خدا

گزارش پروژه درس سیستم های هوشمند دیجیتال

Can bus

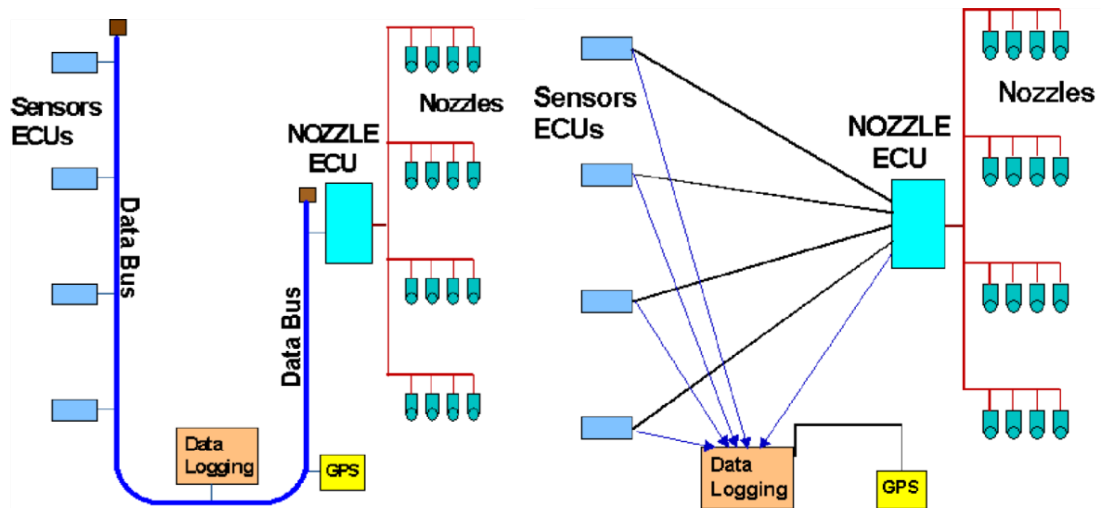
دکتر سید احمد معتمدی

حسین غلامی

مقدمه

شبکه کنترل کننده خود را توسعه داد و آن را در مجمع مهندسين Bosch در اوایل دهه 0891 شرکت

خودرو با نام Automotive Serial Controller Area Network معرفی کرد . بطوریکه اجزا کنترلی مثلاً "یک ماشین (چراغهاي خطر , کیسه هوا , چراغها , شیشه برقی و قفلهاي درب) همگی به یک کانال ارتباطی مشترك متصل می شوند. کارخانه هاي اتوماتیک دریافته بودند که در حالت عادي و بدون شبکه اگر سیم کشی یک قسمت دچار مشکل شود دورانداختن ماشین شاید ارزانترین راه حل باشد تا رفع عیب آن .



(مقایسه سیم کشی معمولی) دسترسی مستقیم کنترلر به سایر واحدها(و اتصال واحدها از طریق باس سریال به کنترلر

در اصل پروتکل can شرکت اینتل اولین تراشه ي CAN موسوم به تراشه ي 92528 را ساخت که بعدها در تراشه ي 92528 تکمیل شد. در همان زمان کارخانه ي نیمه هادي فیلیپس نیز اقدام به ساخت تراشه ي C20092 برای کنترلر CAN نمود . پس از آن شرکتهاي موتورلا و NEC نیز اقدام به ساخت تراشه هاي CAN نمودند.

در سال 0882 گروه CAN In Automation (CIA) تشکیل شد و در سال بعد استاندارد ISO11898 جهت تعریف CAN برای استفاده هاي صنعتی انتشار یافت.

در سال 0882 کارخانه ي بنز جهت ارتباطات الکتریکی داخل خودروهايش از شبکه ي CAN استفاده کرد که منجر به کاهش حدود یکصد کیلوگرم از وزن ماشین شد . این موضوع مورد استقبال سایر خودرو سازان نیز قرار گرفت. شرکت ولوو نیز با بهره گیری از این تکنیک در مدل خودروي سواري S80 خود توانست به میزان یک کیلومتر از طول دسته سیم این خودرو بکاهد.

امروزه تعداد زیادی شبکه مخصوص خودرو طراحی و از طرف شرکت‌های مطرح دنیا معرفی شده اند . از جمله این شبکه ها می‌توان به گذرگاه‌های VAN2 ، CAN3 ، IDB 1349 ، TTP/C MOSTLIN اشاره کرد . کشور های مختلف استانداردها و نام های خاصی را برای کاربرد این سیستم ها به کار می برند که از جمله معروفترین آنها می توان به:

- استاندارد Can : ساخت آلمان که در خودرو های بنز ب . ام . دابل یو و فیات و ولوو به کار می رود.
- استاندارد VAN : ساخت کشور فرانسه و روی محصولات رنو و پژو این کشور مورد استفاده است.
- استاندارد J1850 : ساخت کشور آمریکا که بر روی محصولات تولید کرایسلر فورد و جی ام مورد استفاده می باشد.
- استاندارد Proprietary : تولید کشور ژاپن که بر روی محصولات این کشور مورد استفاده می باشد.
- استاندارد Abus : محصول آلمان و مورد استفاده بر روی محصولات گروه فولکس واگن است.

در میان استاندارد های بالا CAN از موفق ترین آن ها به حساب می آید و بیشتر خودرو سازان از این استاندارد بر روی محصولاتشان استفاده می کنند. در خودروهای تولیدی کشور ما ایران ، دو گذرگاه CAN و VAN بیشتر از بقیه مورد توجه هستند. مطمئناً نقش ارتباطات در یک ماشین می تواند بیانگر تفاوت بین مرگ و زندگی باشد. CAN یک شبکه پایدار در برابر شرایط سخت کاری است و امکان بروز خطا در آن بسیار پائین است.

معرفی شبکه ی CAN

CAN مخفف شبکه محلی کنترلر است . اساساً این شبکه برای محیط‌های پر نویز صنعتی طراحی شده است . CAN BUS

رابط دوسیمه تفاضلی است که روی یک جفت سیم به هم پیچیده شده محافظ دار (STP) یا به همراه سیم زمین اجرا می شود

و به این بدون محافظ (UTP) یا کابل تخت به همراه سیم زمین اجرا می شود و این سیمها CAN_L و CAN_H گفته می

شود . تعداد وسایل قابل اتصال 001 وسیله است . توپولوژی بصورت باس است که دو طرف آن ترمیناتور نیاز دارد . برای

هر گره از یک کانکتور نوع 8 Pin D Male 9 استفاده می شود . نحوه ی رمز گذاری بیئی برای گذرگاه دیفرانسیلی دو

سیمه به صورت بدون بازگشت به صفر (با یک بیت Stuffing) است .

استفاده از رمز گذاری بدون بازگشت به صفر ارسال پیغامهای ترکیبی را با حداقل تعداد انتقال و اطمینان بالا برای

اغتشاشهای خارجی ، تضمین می کند.

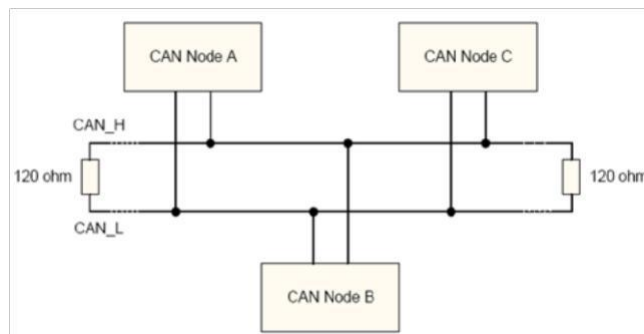
پروتکل CAN برای رسانیدن پیغام های کوتاه با طول حداکثر 9 بایت طراحی شده است و به طور معمول برای فرستادن سیگنال های راه انداز مثل قفل کردن کمر بند های ایمنی در مواقع ترمز های سنگین و اندازه گیری مقادیری مانند دما و فشار استفاده می شود .

این پروتکل هیچ گونه وقفه ای برای انتقال پیغام ندارد اما اولویت فرستادن پیغام را برای عدم برخورد خواهد داشت و معمولاً پیغام اورژانسی را در اولویت قرار می دهد.

سیستم های CAN بسیار سریع هستند و قابلیت انتقال حداکثر 8811 پیغام 9 بیتی و 09111 سیگنال راه انداز در ثانیه را دارا خواهند بود .

بالاترین نرخ ارسال داده در این پروتکل 1 Mbps و کمترین آن 10 Kbps می باشد . تمام ماژولهای استفاده شونده در این پروتکل می بایست حداقل نرخ ارسال 20 Kbps را پشتیبانی کنند . طول کابل به نوع ارسال استفاده شده بستگی دارد . به طور معمول همه ی قطعات استفاده شده در سیستم انتقال باید دارای نرخ انتقال یکسان باشند . حداکثر طول خط انتقال یک کیلومتر و حداقل آن 41 متر در نرخ 1Mbps است .

بطور کلی CAN در خودروها ، سیستمهای حمل و نقل، ماشین آلات الکترونیکی و پزشکی ، در ساختمان سازی و ... کاربرد دارد . یکی از ویژگی های CAN که باعث پذیرش گسترده ی آن شد ، امکان استفاده از محصولات تولید کنندگان مختلف در شبکه است . جایگاه CAN در هرم اتوماسیون در سطوح پایین است .



ترمیناتور مقاوتی در حدود 021 اهم است که در دو طرف باس بین CAN_L و CAN_H قرار میگیرد . نقش ترمیناتور تطبیق امپدانس و جذب سیگنالهاست . بدون وجود ترمیناتور وقتی سیگنال به دو سر سیم باز می رسد اکو شده و با دامنه ی معکوس روی سیم برگشت پیدا می کند. این سیگنال برگشتی شبیه نویز بوده و سیگنالهای داده را خراب می کند . بطور کلی ترمیناتور لازم است در ابتدا و انتهای کابل Trunk قرار گیرد.

حالات باس

دو نوع حالت در توصیف باس وجود دارد :

Dominant (حالت نهفته) و Recessive (حالت برجسته)

حالت نهفته هنگامی رخ می دهد که دو سیم CAN_L و CAN_H دارای پتانسیل یکسانی باشند و حالت برجسته زمانی رخ میدهد که دو سیم با یکدیگر اختلاف پتانسیل داشته باشند. باس CAN وقتی غیر فعال است به صورت نهفته باقی می ماند .

تشخیص حالات باس نهفته و برجسته از صفر و یک کار بسیار مهم و دشواریست . صفر و یک برای سیستم اعداد باینری مناسب است ولی آنها نمی توانند حالات باس را بازگو کنند . این دو مفهوم نهفته و برجسته باس یک مفهوم مهم و خاص در هنگام بحث دایری باس و میدان کنترل CAN خواهد بود .

کد گذاری منچستر و بدون بازگشت به صفر :

کدگذاری منچستر:

در این روش برای رمز کردن نیاز به نمایش انتقال بیت از 1 به 0 و از 0 به 1 می باشد . این کد گذاری برای ارتباطات

آسنکرون مناسب است به خاطر اینکه همیشه یک بیت برای سنکرون شدن وجود دارد. دریافت کننده با رسیدن لبه قادر به

تشخیص ابتدا و انتهای فریم است . عیب اولیه منچستر این است که برای رمز کردن نیاز به پهنای باند زیاد برای دیدن 2 بازه

ی زمانی دارد .

کدگذاری بدون بازگشت به صفر:

در این روش که پروتکل CAN از آن استفاده می کند ، برخلاف قبلی نیاز به نمایش هر بیت نیست و این سیگنال بازه ی زمانی 1 یا 0 ورودی را به قوت خود نگه می دارد . اگر یک فریم شامل رشته ای از 1 یا 0 باشد این سیگنال بازه های زمانی زیادی را به صورت ثابت نگه می دارد. عیب این روش این است که برای تشخیص شروع و انتهای هر بیت راه آسانی وجود ندارد و برای رفع کردن این مشکل نیاز به استفاده از کلاک هم فرکانس با فریم داریم تا بتواند آن رشته را کد گذاری کند .

سنکرون سازی

پروتکل CAN از روش انتقال اطلاعات سنکرون استفاده می کند . برای سیستم CAN این بدین معنیست که هر شاخه با استفاده از نرخ ساعت ثابت ارسال و دریافت می کند و همه ی نرخهای ساعت نیز بر مبنای نقطه مرجع خواهند بود . این روش ، انتقال اطلاعات را بسیار سودمند می سازد ولی سنکرون نگه داشتن کلاک ها در طول زمان های طولانی مشکل است . کلاک ها معمولاً سنکرون بودن خود را به خاطر رانش اسیلاتورها و تاخیرات انتشار و خطاهای مراحل از دست می دهند .

شاخه های CAN از دو روش برای سنکرون کردن کلاک هایشان استفاده می کنند سنکرون سازی سخت و دوباره سنکرونی سنکرون سازی سخت تنها در زمان انتقال پیغام و معمولاً در ابتدای فریم پیغام جدید رخ میدهد . قبل از اینکه فریم شروع شود حالت باس به صورت نهفته و غیر فعال است.

اولین بیت آغازگر فریم به صورت برجسته انتقال پیدا می کند . هر شاخه ساعت خود را با استفاده از انتقال ساخته شده توسط فریم سنکرون می کند . کلاک ها معمولاً قادر به سنکرون نگه داشتن خود در تمام طول فریم نیستند بنابراین باید دوباره سنکرون شوند . دوباره سنکرون سازی هر بار که حالت باس از نهفته به برجسته تغییر پیدا می کند رخ میدهد. اگر یک رشته از 1 یا 0 وجود داشته باشد آنگاه شاخه های CAN از انتقال ساخته شده توسط stuff bit برای دوباره سنکرون سازی کلاک هایشان استفاده می کنند .

کاربرد CAN در خودروها

به کار بردن سیستم کن باس در اتومبیل علاوه بر کاهش هزینه ها در ساختمان اتومبیل باعث افزایش انعطاف پذیری در افزودن سیستمها و تجهیزات پیشرفته و تسهیل در نصب آنها می شود . همچنین کار برای سازندگان ساده تر شده و باعث کاهش سیم کشی ها و حذف سیم کشی های زائد شده است .

از کاربرد های دیگر این شبکه در وسایل نقلیه می توان به موارد زیر اشاره کرد :
کن باس با سرعت بالا می تواند در مدیریت موتور و کنترل ترمز ، کنترلرهای شبکه ای برای تنظیم زمان موتور ، انتقال ، شاسی) با نرخ دیتا حدود 200 Kbit/s تا 1 Mbit/s کاربرد داشته باشد .

کن باس با سرعت کم می تواند در کنترل در و پنجره ها و اجزای شبکه ، دستگاههای الکترونیکی شاسی و دستگاه های الکترونیکی که راحتی خودرو را بیشتر می کند ، مانند کنترل نور ، تهویه هوا ، قفل مرکزی ، تنظیم صندلی و آینه کاربرد داشته باشد . کاربردهای بسیار دیگری نیز امکانپذیر است .

اصول تبادل داده ها در CAN

به طور کلی نحوه انتقال اطلاعات در شبکه های مختلف به سه صورت زیر است :

Point To Point : در این روش اطلاعات مشخص از ایستگاه فرستنده به ایستگاه گیرنده می رسد و در کل اطلاعات

مورد نظر دارای یک گیرنده مشخص است .

Broad Cast : در این سیستم اطلاعات روی شبکه فرستاده می شود و هر ایستگاه بر حسب نیاز خودش به اطلاعات

منتشر شده از آنها استفاده می کند .

Multi Point : این روش با روش بالا کمی شبیه به هم می باشد با این تفاوت که در روش بالا شما اطلاعات را که از

یک ایستگاه دریافت و منتشر می کنید در سرتاسر شبکه پخش می شود ، اما در روش سوم اطلاعات به چند ایستگاه مشخص فرستاده می شود .

در اغلب پروتکل های باس روند ارسال اطلاعات بر اساس **Point To Point** یعنی با ارسال آدرس ایستگاه برای خود یک

گیرنده می باشد . هر ایستگاه **Slave** آدرس منحصر به فرد یا **ID** دارد . با ارسال اطلاعات از **Master** که حاوی یک

آدرس نیز هست ، فقط ایستگاه مورد نظر اطلاعات را دریافت می کند . به این ترتیب در اغلب سیستمهایی که به این روش عمل می کنند تنها یک گیرنده وجود دارد.

در پروتکل **CAN** ارسال اطلاعات بر اساس آدرس نبوده و از روش **Broad Cast** استفاده می شود . تمام ایستگاهها

اطلاعات را دریافت کرده و با توجه به نوع و محتوای آن در مورد پذیرش یا رد آن تصمیم گیری می کنند .

در یک سیستم کنترلی یک Server یا Master ورودی ها را می خواند و روی خروجی می نویسد و از دستگاهها ی دیگر اطلاعات را درخواست می کند . از طرف دیگر یک Slave یا Client اطلاعات را برای سیستم فراهم می کند و معمولاً وقتی با او صحبت می شود پاسخ می دهد .

در واقع بجای آدرس دهی ، محتوای پیام ارسالی (مثلاً دمای موتور و یا فرمان روشن شدن چراغ راهنما) به همراه اولویت آن ، توسط شناسه های اختصاصی در شبکه مشخص می شود . مثلاً اگر اطلاعات ارسال شده حاوی سرعت یا دمای موتور باشد همه ی ایستگاهها آن را دریافت کرده و با توجه به محتوای اطلاعات ، هر ایستگاهی که آن اطلاعات برایش دارای اهمیت باشد آن را پذیرش کرده و از آنها استفاده می کند ، بقیه ی ایستگاهها اطلاعات را رد می کنند. بدین ترتیب امکان داشتن چند گیرنده وجود دارد.

مراحل تبادل داده ها

ارسال یک پیغام از هر ایستگاه به باس و دریافت آن توسط ایستگاههای دیگر دارای چند مرحله است :

Make Ready : آماده سازی و ارسال دیتا و شناسه ها به چیپ CAN .

Send Message : بازسازی و ارسال پیغام توسط چیپ CAN به محض دریافت تخصیص باس

Receive Message : تمام ایستگاههای دیگر به عنوان گیرنده پیغام خواهند بود

Select : هر ایستگاهی که به درستی پیغام را دریافت کرده است بررسی می کند که آیا دیتای دریافتی مربوط به آن

ایستگاه است

Accept : در صورت دارای اهمیت بودن دیتا برای ایستگاه پردازش می شود .

مزایای این نوع تبادل داده

نوع تبادل داده استفاده شده در شبکه ی CAN دارای مزایای زیادی نسبت به دیگر شبکه های مبتنی بر باس است ، از جمله می توان به انعطاف پذیری زیاد سیستم و ساختار به دلیل آدرس دهی بر اساس محتوا و امکان اضافه کردن راحت ایستگاه ها به شبکه بدون اصلاحات سخت افزاری یا نرم افزاری اشاره کرد

پروژه انجام شده

پروژه ای که انجام شد اینگونه تعریف شد که بتوان اطلاعات را از یک میکرو به میکرو دیگر توسط can bus ارسال کرد.

شرح پروژه :

از دو میکرو کنترلر arm stm32f103c8t6 استفاده شد یکی به عنوان فرستنده و دیگری به عنوان گیرنده ، برای ارسال اطلاعات و پیاده سازی باس ، میبایست از یک آی سی کمکی استفاده کرد. که از mcp2550 استفاده شد.

ابتدا پیاده سازی در سمت فرستنده شرح داده میشود :

در سمت فرستنده 3 بخش به نام mailbox وجود دارد که پیام های ارسالی را در آن قرار داده و توسط شبکه ارسال میشود

این کار توسط تابع :

```
HAL_CAN_AddTxMessage(&hcan,&pck,data,&pt_mail_box)
```

انجام میشود که بخش سخت افزاری و هدر بسته ساخته شده ، payload و بخشی از ram باید به آن اختصاص یابد.

هدر بسته به این شکل تعریف میشود:

```
CAN_TxHeaderTypeDef pck;  
  
pck.DLC=4;  
  
pck.StdId=0x234;  
  
pck.IDE=CAN_ID_STD;  
  
pck.RTR=CAN_RTR_DATA;
```

dcl: تعداد بایت انتقالی است

StdId : شناسه 11 بیتی است

Ide: مشخص میکند که از پروتکل 11 بیتی استفاده میشود یا 29 بیتی

Rtr: مشخص میکند که بسته ریموت است یا اطلاعات

در بخش بعدی بسته ها اطلاعات ارسالی را مشخص میکنیم:

که با

```
unsigned char data[4] = {1,0,0,0};
```

و در هر بار ارسال

```
data[0]++;
```

```
data[1]--;
```

روند فوق انجام میشود

با اتمام ارسال اینتراپت ارسال فعال شده و تابع HAL_CAN_TxMailbox0CompleteCallback فراخوانی میشود

در این تابع فرمان تغییر وضعیت led روی میکرو را مینویسیم تا با دیدن میکرو ، متوجه ارسال و دریافت بسته بشویم :

```
void HAL_CAN_TxMailbox0CompleteCallback(CAN_HandleTypeDef *hcan)
```

```
{
```

```
    HAL_GPIO_TogglePin(GPIOC, GPIO_PIN_13);
```

```
    HAL_Delay(50);
```

```
}
```

پیاده سازی در سمت گیرنده :

در سمت گیرنده داستان کمی پیچیده تر است چرا که از هم بخش پیاده سازی حافظه های fifo است و هم بخش فیلتر کردن شناسه های ورودی به گره می باشد.

همانطور که در بخش فرستنده سه mailbox وجود داشت در سمت گیرنده نیز میتوان از 3 حافظه fifo به صورت نرم افزاری استفاده کرد، این حافظه ماهیت سخت افزاری نداشته و روی رم قرار دارد و اطلاعات را توسط بخش can رو رم میریزد و اختصاص این فضا به صورت نرم افزاری انجام میشود.

به طور کلی پیش شروع دریافت میبایست حافظه را اختصاص داد و فیلتر ها تنظیم کرد

ابتدا به آماده سازی حافظه و فعال کردن اینتراپ میپردازیم :

```
HAL_CAN_ActivateNotification(&hcan,CAN_IT_RX_FIFO0_MSG_PENDING);
```

با صدا کردن این تابع اینتراپت حافظه fifo0 را فعال میکنیم

```
HAL_CAN_GetRxMessage(&hcan,CAN_RX_FIFO0,&pck,data);
```

و بسته را در بخش pck و data تحویل گرفته و داخل حافظه fifo میریزیم
البته پیش از اینکه این اتفاق انجام شود فیلتر به صورت سخت افزاری روی این تابع پیاده میشود و عملکرد فوق
باید به صورت زیر تنظیم شود.

```
CAN_FilterTypeDef filter;
```

```
filter.FilterIdHigh=0xffff;
```

```
filter.FilterIdLow=0x0000;
```

```
filter.FilterFIFOAssignment=CAN_FILTER_FIFO0;
```

```
filter.FilterScale=CAN_FILTERSCALE_16BIT;
```

```
filter.FilterMode=CAN_FILTERMODE_IDMASK;
```

```
filter.FilterActivation=CAN_FILTER_DISABLE;
```

```
HAL_CAN_ConfigFilter(&hcan,&filter);
```

هر کدام از این بخش ها برای فیلتر کردن بخشی از آدرس بکار میرود
و نوع عملکرد آن در هنگام فعال سازی اینتراپت در بخش pck خود را نشان میدهد به این شکل که

```
CAN_RxHeaderTypeDef pck;
```

```
pck.DLC=4;
```

```
pck.StdId=0x234;
```

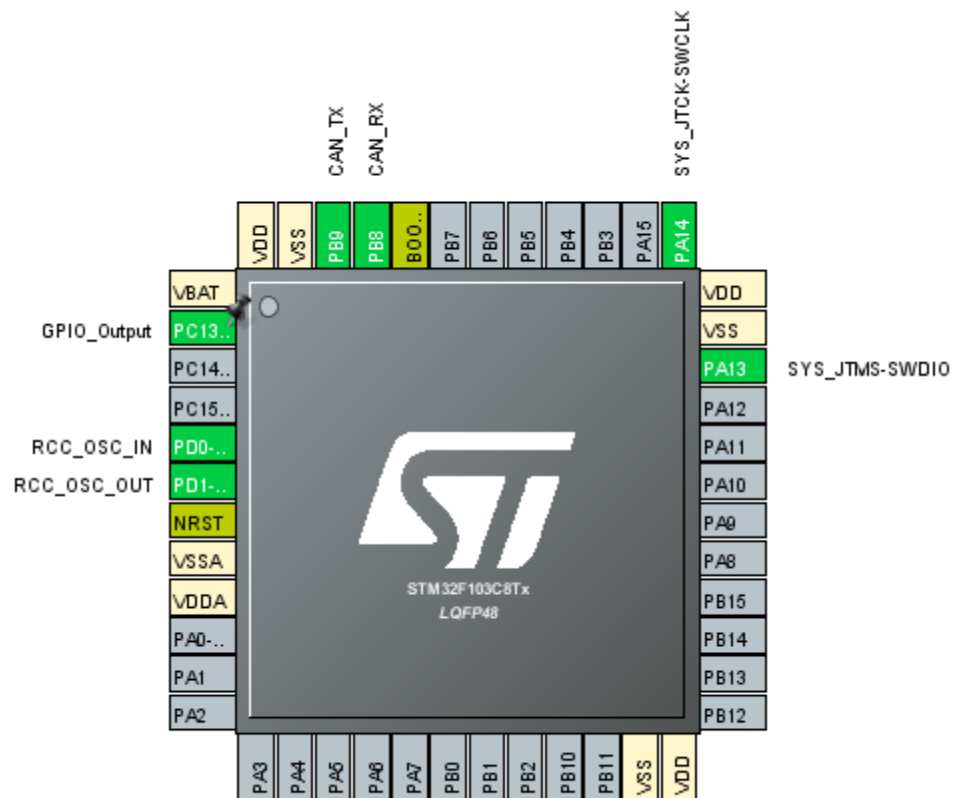
```
pck.IDE=CAN_ID_STD;
```

```
pck.RTR=CAN_RTR_DATA;
```

```
pck.FilterMatchIndex=0xff;
```

در هنگام طراحی endnode هدری با نام FilterMatchIndex وجود دارد. که میتواند فیلتر را حین پیکربندی
پروژه از طریق pck به درگاه ورودی can متصل کند. به این شکل اطلاعات در سمت گیرنده دریافت میشوند.

تصویر configutation میکرو ها در سمت گیرنده و فرستنده:



نحوه اتصال mcp2561 به میکرو و تشکیل باس

