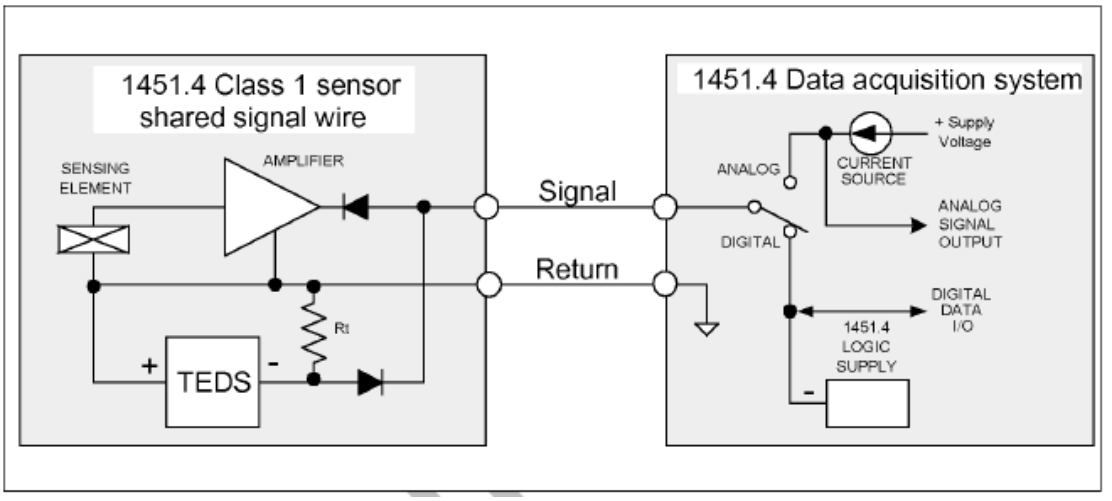
پژوهشگاه نیرو
Niroo Research Institute

NRI-ER-661-PCNPN40-99-L

طراحی و ساخت درایور دیجیتال سنسور هوشمند بر اساس استاندارد IEEE1451.4 بصورت نمونه بر روی مبدل پیزوالکتریک



مدیرپروژه : محمود تکابی
مجری پروژه : بابک امینی

DOI: 10.30503/nripress.2020.296


بسم الله الرحمن الرحيم

طراحی و ساخت درایور دیجیتال سنسور هوشمند بر اساس استاندارد IEEE1451.4 بصورت نمونه بر روی مبدل پیزوالکتریک

مدیر پروژه: محمود تکابی

همکاران پروژه : فرهاد متین فر

مجری پروژه: بایک امینی

انتشارات پژوهشگاه نیرو
اسفند ۱۳۹۹

کد گزارش: NRI-ER-661-PCNPN40-99-L

DOI: 10.30503/nripress.2020.296

ایران. وزارت نیرو. پژوهشگاه نیرو. گروه پژوهشی الکترونیک و ابزاردقیق

طراحی و ساخت درایور دیجیتال سنسور هوشمند بر اساس استاندارد IEEE1451.4 بصورت نمونه بر روی مبدل پیزوالکتریک/ محمود تکابی. کارفرما: پژوهشگاه نیرو، ۱۳۹۹.

۱۵۳ ص.: مصور، نمودار، جدول

۱. متین‌فر، فرهاد، همکار پروژه. ۲. امینی، بابک، مجری. ۳. محور/ محورهای اصلی پروژه: سنسورهای هوشمند ۴. محور/ محورهای فرعی پروژه: دیتاشیت الکترونیک، استاندارد IEEE1451.4

انتشارات پژوهشگاه نیرو

نام گروه پژوهشی الکترونیک و ابزاردقیق

کلیه حقوق قانونی این اثر متعلق به پژوهشگاه نیرو است.

نشانی: تهران،شهرک‌غرب،انتهای‌بلوارشهیددادمان،پژوهشگاه‌نیرو، کدپستی: ۱۴۶۸۶۱۳۱۱۳

تلفن: ۹-۸۸۰۷۹۴۰۱

نمابر: ۸۸۰۷۸۲۹۶

وبگاه: www.nri.ac.ir

پست الکترونیک انتشارات: publications@nri.ac.ir

فرم تعهدنامه اصالت اثر

اینجانب محمود تکابی متهد می‌شوم که مطالب مندرج در این گزارش حاصل کار پژوهشی اینجانب و همکاران به شرح زیر است و به دستاوردهای پژوهشی دیگران که در این پژوهش از آن استفاده شده است، مطابق مقررات ارجاع و در فهرست منابع و مآخذ ذکر گردیده است.

کلیه حقوق مادی و معنوی این اثر متعلق به پژوهشگاه نیرو است.

همکاران :

۱. فرهاد متین فر

پیشگفتار

از آنجا که طراحی و ساخت مدار مبدل سنسور پیزوالکتریک بعنوان یکی از چالشهای مطرح درساخت سنسورهای شتابسنج بوده‌است، لذا پژوهشگاه نیرو با اجرای چند پروژه موفق به رفع این چالش و ساخت نمونه‌های صنعتی این مبدل شده‌است. با ورود تکنولوژی دیجیتال به حوزه این سنسورها، انواع جدید آنها به این تکنولوژی مجهز شده‌اند. از آنجا که این سنسورها غالبا با خروجی آنالوگ کاربرد دارند اما با اضافه کردن یک بخش حافظه دیجیتال جانبی، تمامی اطلاعات مورد نیاز کاربر که در برگه‌های اطلاعاتی سنسورها اعلام می‌گردد در این حافظه ذخیره می‌شوند. با طراحی مناسب می‌توان بگونه‌ای سنسور را ساخت که قابلیت Plug&Play داشته باشد بطوریکه به محض اتصال تغذیه و یا درخواست از آنها به این اطلاعات دست یافت. در این راستا حتی اطلاعات کالیبراسیون هر سنسور را نیز می‌توان در این حافظه ثبت نمود و کاربر را بی نیاز از اطلاعات اضافی نموده و یا احتمال خطای کاربر را در استفاده از آنها صفر کرد. در واقع بدین ترتیب حتی سنسورهای آنالوگ نیز بگونه‌ای دیجیتال شده و می‌توانند در یک شبکه از سنسورهای دیجیتال و هوشمند کار کنند.

موسسه استاندارد IEEE مجموعه‌ای از استانداردها را برای سنسور و محرک هوشمند به نام IEEE1451 توسعه داده‌است. بر این اساس، علاوه بر تعاریف فوق، ترانسدیسر هوشمند باید دارای تابع‌های لازم در تولید و نمایش صحیح کمیت اندازه‌گیری و یا کنترل‌شونده باشد. این استاندارد فارغ از پروتکل‌های ارتباطی نوشته شده است. در این استاندارد ایده و کاربرد دیتا شیت الکترونیک برای هر سنسور بر حسب نیاز مطرح شده است. بعلاوه یک بخش این استاندارد IEEE1451.4 اختصاصاً برای استفاده از سنسورهای آنالوگ بوده که بصورت مد مخلوط سیگنال‌های آنالوگ و دیجیتال بکار می‌روند.

بر اساس استاندارد فوق سنسورها و مبدل‌های پیزوالکتریک نیز به فناوری دیجیتال مجهز شده‌اند بطوریکه سازندگان معتبر سنسورهای ارتعاشی، در ساخت سنسورهای جدید از آن استفاده کرده‌اند. این امر مزایای زیادی را به همراه دارد. در پروژه حاضر، اولاً هدف، پیاده‌سازی این استاندارد بر روی مبدل بار به ولتاژی است که قبلا در پژوهشگاه نیرو ساخته شده و بعلاوه دیگر مشخصه‌ها و ویژگی‌های این استاندارد معلوم خواهد شد.

گزارش حاظر گزارش نهایی پروژه "طراحی و ساخت درایور دیجیتال سنسور هوشمند بر اساس استاندارد IEEE1451.4 بصورت نمونه برای مبدل پیزوالکتریک" بوده که در واقع شامل مجموعه گزارشات مرحله اول و دوم پروژه است و توسط مهندسین محمود تکابی و فرهاد متین‌فر تهیه شده‌است. شایان ذکر است که نظارت این پروژه، با جناب آقای دکتر محمد اسدنژاد می‌باشد.

DOI: 10.30503/nripress.2020.296

فهرست مطالب

پیشگفتار III

مقدمه ۱

فصل اول: مطالعه و بررسی شبکه 1-WIRE و تجهیزات آن ۲

۱-۱- شبکه ارتباطی MicroLan یا 1-Wire ۳

۱-۲- پروتکل ارتباطی در شبکه MicroLan ۵

۱-۳- کیفیت سیگنال شبکه Microlan ۹

۱-۴- سخت‌افزار درایور Master یک شبکه تک سیمه یا 1-Wire ۱۰

معرفی کوتاهی از انواع چیپ اندازه‌گیری مجهز به باس تک سیمه ۱۲

۱-۵- کاربرد مستقیم پورت سریال بعنوان مرکز کنترل باس 1-Wire ۱۲

۱-۶- آشنایی با یک نمونه حافظه EEPROM منطبق با باس تک سیمه ۱۵

۱-۶-۱- حافظه RAM و دستورات آن ۱۷

۱-۶-۲- دستورات بخش ROM حافظه ۱۹

فصل دوم: مروری بر استاندارد IEEE1451.4 ..... ۲۰

مقدمه ۲۱

۲-۱- مروری بر استاندارد IEEE1451 ..... ۲۱

۲-۱-۱- ترانسدیوسر هوشمند ۲۲

۲-۱-۲- ترانسدیوسر هوشمند مبتنی بر اساس استاندارد IEEE1451 ..... ۲۳

۲-۱-۳- خانواده استاندارد IEEE1451 ..... ۲۴

۲-۲- ترانسدیوسر استاندارد IEEE1451.4 ..... ۲۶

۲-۲-۱- پایه و اساس ۲۶

۲-۲-۲- پیکربندی ترانسدیوسر بر پایه IEEE1451.4 ..... ۲۷

۲-۲-۲-۱- چندین ترانسدیوسر روی یک MMI مبتنی بر IEEE 1451.4 ..... ۲۸

۲-۲-۲-۲- چندین نود در یک ترانسدیوسر IEEE1451.4 ..... ۲۸

۲-۲-۲-۳-ارتباط با یک ترنسدیوسر IEEE 1451.4 ..... ۲۸

۲-۲-۲-۴- ارتباط با چندین ترانسدیوسر IEEE1451.4 روی یک MMI مبتنی بر IEEE1451.4 ..... ۲۸

۲-۳- دیتاشیت الکترونیکی ترانسدیوسر ۲۹

۲-۳-۱- TEDS پایه ۲۹

۲-۳-۱-۲- محتویات TEDS پایه ۲۹

۲-۳-۱-۲- جاگذاری و فرمت TEDS پایه ۳۰

۲-۳-۲- IEEE ، کاربر و TEDS سازنده ۳۰

DOI: 10.30503/nripress.2020.296

۳-۳-۲- فرمت داده و الگوها ۳۱

۴-۳-۲- نودها، آدرس‌ها، کدهای خانوادگی، URN، و CRC ۳۲

۵-۳-۲- ارسال داده ۳۲

۶-۳-۲- ساختار سیستم داده TEDS ۳۳

۴-۲- الگوها ۳۴

۱-۴-۲- مرورکلی ۳۴

۲-۴-۲- کشف وجود ترانسدیوسرها ۳۵

۳-۴-۲- شناسایی ترانسدیوسرها و نودهای آن ۳۶

۱-۳-۴-۲- شماره سازنده یا انتخاب کننده ویژه ۳۸

۴-۴-۲- گردآوری اطلاعات TEDS یک ترانسدیوسر ۳۹

۵-۴-۲- تجزیه کردن اطلاعات TEDS یک ترانسدیوسر ۴۰

۱-۵-۴-۲- انتخاب کننده وصفی ۴۱

۲-۵-۴-۲- شناسه الگو ۴۲

۵-۲- زبان توصیف الگو TDL ۴۲

۱-۵-۲- کلیات ۴۲

۱-۱-۵-۲- فرمانها ۴۲

۲-۱-۵-۲- نقشه بیتی ۴۳

۳-۱-۵-۲- TEDS پایه ۴۳

۴-۱-۵-۲- حساسیت به حروف بزرگ یا کوچک ۴۳

۵-۱-۵-۲- رشتههای کاراکتری ۴۳

۲-۵-۲- فرمانهای تعیین هویت یا شناسایی ۴۳

۱-۲-۵-۲- الگو و خاتمه الگو ۴۳

۲-۲-۵-۲- شناسه الگو ۴۴

۳-۲-۵-۲- الحاق الگو ۴۴

۴-۲-۵-۲- اعتبارسنجی الگو ۴۵

۵-۲-۵-۲- صحت سنجی فایل الگو ۴۵

۶-۲-۵-۲- شماره نسخه TDL ۴۵

۷-۲-۵-۲- UDID (شناسه توصیف یکتا) ۴۶

۸-۲-۵-۲- UGID (شناسه یکتای گروه) ۴۷

۳-۵-۲- فرمانهای کنترلی ۴۸

۱-۳-۵-۲- چکیده ۴۸

۲-۳-۵-۲- توضیحها (// comments) ۴۸

۳-۳-۵-۲- SelectCase و Case ۴۹

۴-۳-۵-۲- آرایه ساختار ۵۰

۵-۳-۵-۲- فاصله گذاری ۵۱

۶-۳-۵-۲- در یک ردیف قرار دادن ۵۱

DOI: 10.30503/nripress.2020.296

۴-۵-۲- فرمانهای ویژگی % (property command) ۵۲

۱-۴-۵-۲- ساختار فرمان (command stucture) ۵۲
۲-۴-۵-۲- لیست ویژگی (property list) ۵۳
۳-۴-۵-۲- توصیف (description) ۵۳
۴-۴-۵-۲- دسترسی سطح (Acess_level) ۵۳
۵-۴-۵-۲- نوع داده (Datatype) ۵۳
۶-۴-۵-۲- پارامتر فرمت ۵۹
۷-۴-۵-۲- واحدهای فیزیکی ۶۳
۸-۴-۵-۲- اختصاص دهی اختیاری ۶۵
۹-۴-۵-۲- عملکردهای گسترش یافته با استفاده زیرویژگی‌ها ۶۶
۱۰-۴-۵-۲- برخی ویژگی‌های خاص ۷۴
۱۱-۴-۵-۲- فرمت الگوی Embedded ۷۵
۱۲-۴-۵-۲- شکل آزاد ویژگی‌های TEDS ۷۶
۱۳-۴-۵-۲- رخدادهای چند باره از یک ویژگی ۷۷
۱۴-۴-۵-۲- اداره کردن ویژگی‌های غیر رسمی ۷۷

۶-۲- ویژگی‌های اینترفیس ترانسدیوسر در مود ترکیبی (MMI) ۷۸

۱-۶-۲- مقدمه ۷۸

۱-۱-۶-۲- تعریف MMI ۷۸
۲-۱-۶-۲- عملکرد MMI ۷۸
۲-۶-۲- مود آنالوگ ۸۱

۱-۲-۶-۲- کلاس ۱ ۸۱
۲-۲-۶-۲- کلاس ۲ ۸۲
۳-۶-۲- مود دیجیتال ۸۲

۱-۳-۶-۲- کلاس ۱ ۸۲
۲-۳-۶-۲- کلاس ۲ ۸۲
۴-۶-۲- تعریفهای خط ۸۳

۱-۴-۶-۲- کلاس ۱ ۸۳
۲-۴-۶-۲- کلاس ۲ ۸۳
۵-۶-۲- پرتکل ارسال داده دیجیتال MMI ۸۴

۱-۵-۶-۲- مراحل تراکنش ۸۴
۲-۵-۶-۲- ویژگی‌های الکتریکی ۸۸
۳-۵-۶-۲- منبع تغذیه لاجیکی IEEE 1451.4 ۹۰

فصل سوم: طراحی درایور و مبدل منطبق بر استاندارد IEEE1451.4 ۹۱

۱-۳- سازگارسازی شبکه 1-Wire برای استاندارد IEEE1451.4 ۹۲
۲-۳- طراحی مبدل و درایور استاندارد IEEE1451.4 ۹۳
۱-۲-۳- طراحی مبدل مجهز به بخش دیجیتال منطبق با استاندارد IEEE1451.4 ۹۳
۲-۲-۳- طراحی درایور شبکه منطبق با استاندارد IEEE1451.4 ۹۴
۳-۲-۳- لیست اقلام موردنیاز ۹۵

DOI: 10.30503/nripress.2020.296

فصل چهارم: طراحی و ساخت بخش دیجیتال مبدل و تست‌های اولیه ۹۷

۴-۱- مروری بر مشخصات سنسورهای شتابسنج و مبدل آنها ۹۷
۴-۲- طراحی بخش دیجیتال مبدل ۱۰۰
۴-۳- ساخت برد جدید مبدل ۱۰۲

فصل پنجم : طراحی و ساخت بخشهای کمکی جهت ارتباط مبدل با کامپیوتر و قرائت داده‌ی TEDS و ارسال و دریافت دستورات مختلف ۱۰۵

۵-۱- طراحی درایور ارتباطی مبدل با کامپیوتر ۱۰۶
۵-۲- ساخت برد درایور ارتباطی ۱۰۷
۵-۳- تست و اصلاح درایور در مد شبکه تک سیمه ۱۰۸
۵-۳-۱- راه‌اندازی درایور ۱۰۹
۵-۴- تست و اصلاح درایور و مبدل در استاندارد IEEE1451.4 ..... ۱۱۳
۵-۴-۱- راه‌اندازی و تست مبدل و درایور ۱۱۴
۵-۴-۲- تست نهایی عملکرد درایور و مبدل پروژه ۱۱۷

فصل ششم: پیاده‌سازی دیتاشیت الکترونیک منطبق بر الگوهای استانداردی ۱۱۸

۶-۱- مقدمه ۱۱۹
۶-۲- قابلیت های نرم‌افزار IEEE1451.4 TEDS editor از شرکت B&K ۱۲۰
۶-۲-۱- پیاده‌سازی الگوهای نمونه استانداردی روی مبدل پروژه با نرم‌افزار شرکت B&K ۱۲۲
۶-۳- قابلیت های نرم‌افزارDeweSoft TEDS editor از شرکت DeweSoft ۱۲۵
۶-۳-۱- پیاده‌سازی الگوهای نمونه استانداردی روی مبدل پروژه با نرم‌افزار DeweSoft ۱۲۷
۶-۴- پیاده‌سازی لیست نود مطابق استاندارد ۱۲۸
۶-۵- پیاده‌سازی الگوهای کالیبراسیون و کاربری ۱۳۰
۶-۶- بررسی تشخیص الگوی تولید شده توسط یک نرم‌افزار در محیط نرم افرار دیگر ۱۳۳
نتیجه گیری: ۱۳۶
مراجع: ۱۳۸

فهرست اشکال

شکل ۱-۱- ساختار یک شبکه 1-Wire ۳
شکل ۱-۲- مدار تغذیه پارازیتی ۴
شکل ۱-۳- سیگنال ریست و حضور ۶
شکل ۱-۴- سیگنال نوشتن یک و صفر ۷

DOI: 10.30503/nripress.2020.296

شکل ۱-۵- سیگنال قرائت داده ۷
شکل ۱-۶- ترتیب ارسال دستورات در یک محاوره[1] ۸
شکل۱-۷- فرایند خواندن اطلاعات ROM [1] ۹
شکل ۱-۸- درایور مستقیم پورت سریال به شبکه 1-Wire ۱۰
شکل ۱-۹- درایور پورت سریال به شبکه 1-Wire به کمک چیپ DS2480 ..... ۱۱
شکل ۱-۱۰- تولید سیگنال ریست و حضور ۱۳
شکل ۱-۱۱- تولید سیگنال نوشتن یک و صفر ۱۴
شکل ۱-۱۲- تولید سیگنال خواندن داده ۱۵
شکل ۱-۱۳- ساختار داده ROM ۱۶
شکل ۱-۱۴- بلوک دیاگرام داخلی DS2430 ..... ۱۶
شکل ۱-۱۵- الگوریتم تولید کدینگ CRC ۱۷
شکل ۱-۱۶- نقشه حافظه ۱۷
شکل ۲-۱- a) ترانسدیوسر هوشمند b) ترانسدیوسر هوشمند IEEE1451 ..... ۲۲
شکل ۲-۲- اطلاعات داده TEDS ۲۴
شکل ۲-۳- خانواده استاندارد IEEE1451 ..... ۲۵
شکل ۲-۴- ساختار اینترفیس مد مختلط و ترانسدیوسر ۲۷
شکل ۲-۵- URN و تولید CRC ۳۲
شکل ۲-۶- بلوک دیاگرام سیستم داده TEDS با IEEE 1451.4 ..... ۳۴
شکل ۲-۷- فلوچارت کشف ترانسدیوسرها ۳۶
شکل ۲-۸- فلوچارت شناسایی ترانسدیوسر(ها) و نودهایش ۳۸
شکل ۲-۹- فلوچارت گردآوری اطلاعات TEDS ترانسدیوسرها ۴۰
شکل ۲-۱۰- فلوچارتی تفکیک اطلاعات TEDS ترانسدیوسر ۴۱
شکل ۲-۱۱- فیلتر با گین و سیستم مالتی پلکسر ۷۱
شکل ۲-۱۲- پیکربندی نود فیلتر با گین و مالتی پلکسر ۷۲
شکل ۲-۱۳- MMI کلاس ۱ استاندارد IEEE 1451.4، سیم سیگنال مشترک ۷۹
شکل ۲-۱۴- MMI کلاس ۱ استاندارد IEEE 1451.4 ، سیم تغذیه مشترک ۸۰
شکل ۲-۱۵- MMI کلاس ۱ استاندارد IEEE 1451.4، سیم برگشت مشترک ۸۰
شکل ۲-۱۶- MMI کلاس ۲ استاندارد IEEE 1451.4 محرک(actuator) 4-20mA ۸۱
شکل ۲-۱۷- MMI کلاس ۲ استاندارد IEEE 1451.4، سنسور پل مقاومتی ۸۱
شکل ۲-۱۸- مراحل آماده سازی ۸۶
شکل ۲-۱۹- Time Slot نوشتن یک ۸۷
شکل ۲-۲۰- time slot نوشتن صفر ۸۸
شکل۲-۲۱- time slot خواندن ۸۸
شکل ۳-۱- کلاس ۱ اینترفیس MMI ۹۲

DOI: 10.30503/nripress.2020.296

شکل ۳-۲- سطوح سیگنال 1-Wire در مقابل کلاس ۱ استاندارد IEEE1451.4 ..... ۹۲
شکل ۳-۳- شماتیک مبدل دو کاناله پیزوالکتریک به همراه بخش TEDS ۹۴
شکل ۳-۴- شماتیک مبدل دو کاناله درایور کلاس ۱ استاندارد ۹۵
شکل ۴-۱- مدل الکتریکی سنسور شتاب سنج پیزوالکتریک ۹۹
شکل۴-۲- مدار معادل سنسور انتخابی و کابل آن ۱۰۰
شکل ۴-۳- کلاس ۱ اینترفیس MMI ۱۰۱
شکل ۴-۴- سطوح سیگنال 1-Wire در مقابل کلاس ۱ استاندارد IEEE1451.4 ..... ۱۰۱
شکل ۴-۵- شماتیک مبدل دو کاناله پیزوالکتریک به همراه بخش TEDS ۱۰۲
شکل ۴-۶- نقشه برد ساخته شده از مبدل (الف –پشت برد ب- روی برد) ۱۰۳
شکل ۴-۷- تصویر برد مونتاژه شده مبدل و کیس مربوطه بصورت باز شده ۱۰۳
شکل ۴-۸- تصویری از مبدل ساخته شده نهایی و پورتهای ارتباطی آن (مجهز به حافظه TEDS) ۱۰۴
شکل ۵-۱- شماتیک مبدل دو کاناله درایور کلاس ۱و ۲ استاندارد ۱۰۷
شکل ۵-۲- نقشه برد مدارچاپی درایور دو کاناله ۱۰۸
شکل ۵-۳- مونتاژ بخش دیجیتال برد کمکی منطبق با شبکه 1-Wire ۱۱۰
شکل ۵-۴- راه اندازی درایور نرم‌افزاری و شناسایی آداپتور شبکه تک سیمه ۱۱۰
شکل ۵-۵- شناسایی قطعات روی شبکه تک سیمه ۱۱۱
شکل ۵-۶- شناسایی قطعه دماسنج و توضیحات و اندازه دمای آشکار شده ۱۱۲
شکل ۵-۷- توضیحات مربوط به قطعه حافظه و نمایش داده‌های ذخیره شده در آن ۱۱۳
شکل ۵-۸- نقشه شماتیک بخش تبدیل سطح ولتاژ در نمونه اولیه(الف) و نمونه اصلاحی (ب) ۱۱۴
شکل ۵-۹- شماتیک مدار نهایی درایور ۱۱۵
شکل ۵-۱۰- نقشه ساخت برد از درایور نهایی پروژه ۱۱۵
شکل ۵-۱۱- تصویر برد درایور مونتاژی و چگونگی پیاده‌سازی در کیس ۱۱۶
شکل ۵-۱۲- تصویر نماهای مختلف و پورتهای داریور نهایی ۱۱۶
شکل ۵-۱۳- شناسایی قطعات و نودهای مبدل در مد دیجیتال ۱۱۷
شکل ۶-۱- لیست الگوهای پشتیانی شده توسط نرم‌افزار B&K ۱۲۱
شکل ۶-۲- نمایی از صفحه اصلی نرم‌افزار B&K ۱۲۲
شکل ۶-۳- وضعیت شناسایی مبدل بدون دیتاشیت با نرم‌افزار B&K ۱۲۲
شکل ۶-۴- منوی انتخاب قطعات حافظه ۱۲۳
شکل ۶-۵- منوی انتخای الگوهای استانداردی ۱۲۴
شکل ۶-۶- منوی اطلاعاتی از پارامترهای الگوی انتخابی ۱۲۴
شکل ۶-۷- شناسایی مبدل با الگوی پیاده‌سازی شده ۱۲۵
شکل ۶-۸- نمای کلی نرم‌افزار DewSoft ۱۲۶
شکل ۶-۹- انواع الگوهای قابل پشتیبانی با نرم‌افزار DewSoft ۱۲۶
شکل ۶-۱۰- شناسایی مبدل خام جهت برنامه‌ریزی الگوهای متفاوت ۱۲۷

DOI: 10.30503/nripress.2020.296

شکل ۶-۱۱- انتخاب یک الگوی نمونه و برنامه‌ریزی آن ۱۲۸
شکل ۶-۱۲- مجموعه نودهای آماده شده جهت پیاده‌سازی لیست نود ۱۲۹
شکل ۶-۱۳- پیاده‌سازی لیست نود برای یک ترانسدیوسر ۱۳۰
شکل ۶-۱۴- اطلاعات یک لیست نود ۱۳۰
شکل ۶-۱۵- مجموعه الگوهای انتخابی در نرم‌افزار DeweSoft ۱۳۱
شکل ۶-۱۶- نمایش و تنظیم پارامترهای الگوی انتخابی اول ۱۳۲
شکل ۶-۱۷- نمایش و تنظیم پارامترهای الگوی انتخابی دوم (الگوی کالیبراسیون) ۱۳۲
شکل ۶-۱۸- نمایش وضعیت الگوی کاربری ۱۳۳
شکل ۶-۱۹- تولید الگوی اول با نرم‌افزارDeweSoft و مشاهده آن با نرم‌افزار B&K ۱۳۴
شکل ۶-۲۰- تولید الگوی دوم با نرم‌افزارDeweSoft و مشاهده آن با نرم‌افزار B&K ۱۳۵
شکل ۶-۲۱- تولید الگو با نرم‌افزار B&K و مشاهده آن با نرم‌افزار DeweSoft ۱۳۶

DOI: 10.30503/nripress.2020.296

فهرست جداول

جدول ۲-۱- نقشه محتویات حافظه TEDS پایه ۲۹
جدول۲-۲- نقشه جزئیات حافظه TEDS پایه IEEE1451.4 ..... ۳۰
جدول ۲-۳- سازنده یا انتخاب کننده استفاده ویژه ۳۹
جدول ۲-۴- شمارههای نسخه ۴۶
جدول ۲-۵- سطوح دسترسی برای ویژگی‌های ترانسدیوسر ۵۳
جدول ۲-۶- انواع داده ۵۴
جدول ۲-۷- تعریف Bit_Digit ۵۶
جدول ۲-۸- نوع داده کاراکتری Chr5 ..... ۵۶
جدول ۲-۹- نشانهها و علایم فرمت بندی ۵۹
جدول ۲-۹- ادامه نشانهها و علایم فرمت بندی ۶۰
جدول ۲-۱۰- مثالهای فرمت ۶۰
جدول ۲-۱۱- پیشوندهای عددی ۶۱
جدول ۲-۱۲- مثالهای فرمت‌های پیشوندی و نسبی (با دقت بالا) ۶۲
دول ۲-۱۳- مثالهای فرمت‌های پیشوندی و نسبی (با دقت پایین) ۶۲
جدول ۲-۱۴- ساختار انواع داده واحدهای فیزیکی ۶۴
جدول ۲-۱۴- ساختار انواع داده واحدهای فیزیکی (ادامه) ۶۴
جدول ۲-۱۵- جدول توابع منطقی ۶۶
جدول ۲-۱۶- زیرویژگی‌ها ۶۸
جدول ۲-۱۶- زیرویژگی‌ها (ادامه) ۷۰
جدول ۲-۱۹- ویژگی‌های الکتریکی DC ۸۸
جدول ۲-۲۰- ویژگی‌های الکتریکی AC ۸۹
جدول ۳-۱- اقلام مورد نیاز برد مبدل ۹۵
جدول ۳-۲- اقلام مورد نیاز برد درایور ۹۶

DOI: 10.30503/nripress.2020.296

مقدمه

سه فصل اول گزارش مربوطه به مرحله اول پروژه می باشد. فصل اول گزارش، به آشنایی با شبکه 1-Wire اختصاص یافته است. از آنجاییکه این شبکه در استاندارد IEEE1451.4 استفاده می‌شود لذا این آشنایی به فهم بهتر مطالب فصل دوم کمک می‌کند. فصل دوم بصورت مفصل استاندارد مورد اشاره را در چند بخش شامل

۱. ترانسدیوسر استاندارد IEEE1451.4
۲. دیتاشیت الکترونیک
۳. الگوهای ثبت داده
۴. زبان الگو نویسی
۵. اینترفیس مد ترکیبی آنالوگ و دیجیتال ترانسدیوسر

تشریح می‌نماید. فصل سوم گزارش، به طراحی مبدل و درایور مربوطه که منطبق با استاندارد IEEE1451.4 هستند اختصاص یافته است. با توجه به مطالعات و بررسی‌های دو فصل اول طراحی‌های مورد نظر به تفصیل در فصل سوم ارائه شده است.

سه فصل بعدی به مرحله دومم پروژه اختصاص یافته است.. فصل چهارم گزارش، به طراحی و تست مبدل اختصاص یافته است. راه اندازی و تست بخش آنالوگ مانند پروژه‌های قبلی انجام پذیر بوده ولی تست بخش دیجیتال به تنهایی امکان پذیر نیست. لذا در فصل پنجم گزارش که اختصاص به طراحی و ساخت درایور ارتباطی دارد تستهای دیجیتال مبدل نیز تکمیل شده‌اند. در فصل پنجم گزارش پس از طراحی و ساخت درایور تستهای آن در مد شبک تک سیمه ارائه شده و در نهایت در مد IEEE1451.4 مجموعه مبدل و درایور تست و اصلاح نهایی شده است. در این فصل به کمک درایورهای نرم‌افزاری اشاره شده در متن گزارش، راه‌اندازی کامل یک شبکه مبتنی بر IEEE1451.4 آورده شده است. در فصل ششم گزارش به کمک نرم‌افزارهای دیگری که معرفی شده‌اند دیتاشیت الکترونیک[1] در قالب الگوهای استانداردی برای برنامه ریزی حافظه مبدل بکار رفته است.

[1] Transducer Electronic Data Sheet

DOI: 10.30503/nripress.2020.296

فصل اول: مطالعه و بررسی شبکه 1-Wire و تجهیزات آن

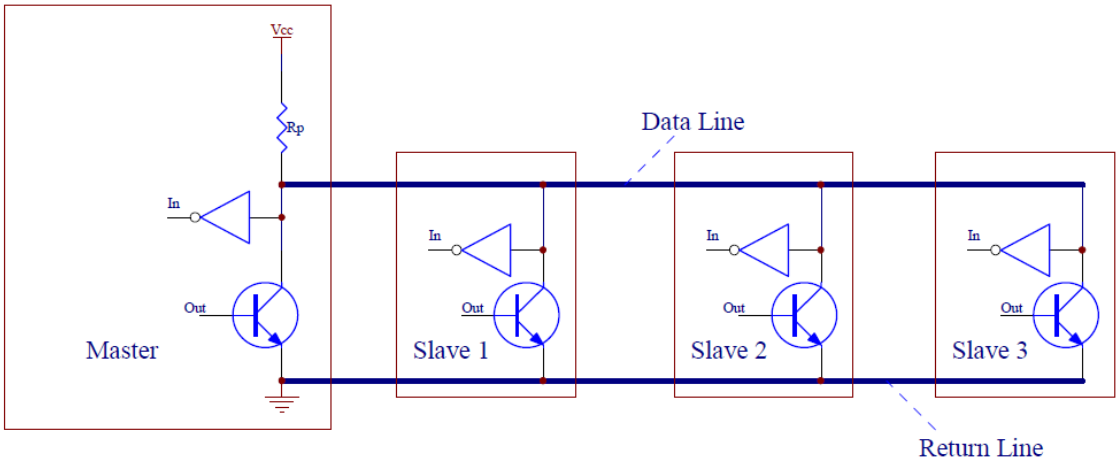
DOI: 10.30503/nripress.2020.296

۱-۱- شبکه ارتباطی MicroLan یا 1-Wire

این شبکه متشکل از سه جزء است

- مرکز کنترل اصلی بالادست یا Master
- مراکز ارتباطی پایین دست یا Slave
- یک زوج سیم ارتباطی بین مرکز اصلی و مراکز ارتباطی

که یک سیم بعنوان داده بوده و سیم دیگر، سیم زمین می‌باشد. در واقع پورت ارتباطی مرکز اصلی و یا مراکز ارتباطی بصورت کلکتور باز[۱] هستند که به همدیگر متصل شده و در مرکز اصلی از مسیر یک مقاومت به تغذیه مثبت وصل شده‌اند. شکل (۱-۱) ساختار این شبکه را نشان می‌دهد.



شکل ۱-۱- ساختار یک شبکه 1-Wire

با توجه به شکل (۱-۱) اولاً مرکز اصلی[۲] و یا مراکز فرعی[۳] می‌توانند فقط باس داده را صفر نمایند و رها کردن باس باعث بالا آمدن ولتاژ باس توسط مقاوت بالاکشنده[۴] Rp می‌شود. بعلاوه از اتصال سیمی آنها یک منطق اشتراک

---

[۱] Open Collector

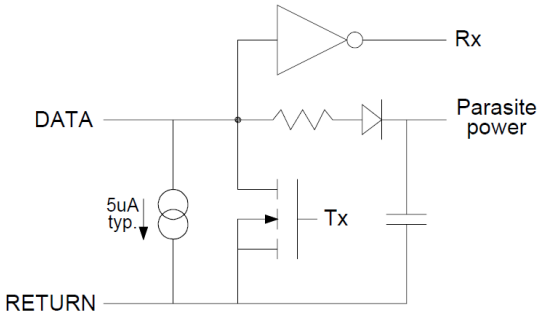
[۲] Master

[۳] Slave

[۴] Pullup

DOI: 10.30503/nripress.2020.296

لاجیکی and ساخته شده است. در این ساختار چند اتصالی[1] از تجهیزات، در هر زمان تنها یک قطعه باید ارتباط را در دست بگیرد و ارتباط دو طرفه و همزمان ممکن نیست. به منظور تفکیک مراکز پایین دست از همدیگر، به هر یک از آنها آدرسی یکتا اختصاص یافته است و بعلاوه تمامی ارتباطات از طریق مرکز اصلی صورت می‌گیرد. سطح ولتاژ ارتباطی همان سطح استاندارد TTL بوده و لذا سطح تغذیه اتصالی شبکه بین ۳.۸ تا ۵.۵ ولت است. در این شبکه دو سیمه، حتی تغذیه مراکز ارتباطی نیز می‌تواند از همان سیم ارتباطی صورت گیرد که به تغذیه پارازیتی معروف است. البته مصرف این مراکز بایستی خیلی اندک و یا تعدادشان محدود باشد و الا مجبور به استفاده از تغذیه مستقل خواهیم شد. شکل (۱-۲) شماتیک تغذیه پارازیتی از خط را نشان داده است.



شکل ۱-۲- مدار تغذیه پارازیتی

همانطور که از شکل (۱-۲) پیداست، یک مدار یکسوساز با خازن و مقاومت بدین منظور بکار می‌رود. زمانی که ولتاژ باس بالاست یک خازن ۸۰۰ میکروفارادی از طریق یک دیود و مقاومت سری شارژ می‌شود تا اینکه ولتاژ خازن به ولتاژ باس نزدیک می‌شود. به محض کم شدن ولتاژ باس، دیود در مسیر معکوس قرار گرفته و تغذیه تجهیز از شارژ خازن فراهم می‌شود بنابراین کم‌مصرف بودن تجهیز و کوتاه بودن زمان کاهش ولتاژ باس و همچنین داشتن مدت زمان کافی برای شارژ خازن از اهمیت برخودار بوده و در طراحی شبکه باید مد نظر باشد. بعلاوه افزایش تعداد مراکز، میزان جریان مصرفی باس را زیاد کرده و باعث افت ولتاژ باس می‌شود و ممکن است به حدی برسد که از حداقل سطح ولتاژ لاجیک یک کمتر شود لذا این محدودیت نیز همواره باید مد نظر باشد. البته نکات دیگری نیز وجود دارد که در ادامه به آنها اشاره می‌شود.

برای این باس، یک پروتکل ارتباطی نیز تعریف شده است که در آن تمامی ارتباطات از طریق مرکز اصلی و یا Master فراهم شده است. این سیستم ارتباطی با توجه به سادگی و قابلیت پرتابل بودن (ارتباط دو سیمه ساده) و همچنین عدم نیاز به تغذیه, برای اولین بار به منظور استفاده از برچسب شناسایی الکترونیک تجهیزات، توسط شرکت دالاس آمریکا معرفی شد. استفاده از یک حافظه کوچک الکترونیک در تجهیزات که با این باس کار می‌کنند، یک سیستم برچسب شناسایی را معرفی می‌نماید که از دیگر روشهای شناسایی مانند بار کد و برچسب مغناطیسی و یا رادیویی و ..... برتری‌های زیر را بهمراه دارد :

۱. افزایش حجم اطلاعات قابل دسترس
۲. قابلیت تغییر اطلاعات روی چیپ حافظه متناسب با اشیاء مختلف اتصال یافته
۳. کاهش هزینه ارتباط این چیپ با کامپیوتر و یا دستگاه برچسب خوان

بعلاوه این سیستم بصورت ویژه از سیستم بارکد به دلایل زیر ارجحتر است

[1] Multidrop

DOI: 10.30503/nripress.2020.296

۱. قابلیت خواندن کم هزینه بدون نیاز به تجهیزات اپتیکی گران
۲. دسترسی به اطلاعات بیشتر(چیپهای شناسایی ۱۰۰ برابر بارکد، می‌توانند اطلاعات در خود جای دهند)
۳. شناسایی هر تجهیز با آدرس یکتا
۴. آدرس یکتای شناسایی می‌تواند در یک شبکه بعنوان آدرس دسترسی به قطعه مورد نظر بکار رود
۵. محتوای چیپ‌های شناسایی می‌تواند برحسب مورد تغییر یابد
۶. تعداد دفعات تغییر محتوای چیپ تا یک میلیون بار امکان پذیر است
۷. امکان ایجاد یک ارتباط مطمئن در محیط‌های آلوده براحتی فراهم می‌شود
۸. وجود تجهیزات پرتابل دستی سبک و ارزان جهت ارتباط با چیپ شناسایی

همانطور که اشاره شد براحتی می‌توان از یک نقطه اتصال الکتریکی و یک رینگ برای سیگنال زمین، جهت ایجاد یک ارتباط پرتابل بر مبنای شبکه فوق ساخت. این اتصال که ظاهری مانند یک شستی دارد به شستی شناسایی (ibutton) معروف است. همانطور که اشاره شد بنیانگذار این تکنولوژی در دنیا، شرکت دالاس آمریکاست که با ساخت چیپ‌های مخصوص برای این منظور، پیشرفت‌های زیادی را بوجود آورده است. در ادامه پروتکل ارتباطی این شبکه را بصورت دقیقتر بررسی می‌کنیم.

۲-۱- پروتکل ارتباطی در شبکه MicroLan

این پروتکل برای اولین بار توسط شرکت دالاس آمریکا معرفی شده که به شبکه Microlan و یا 1-Wire معروف است. این شبکه از نوع Master/Slave و دارای ارتباط دوطرفه غیر همزمان بوده و بعلاوه بصورت سنکرونیزه بیتی می‌باشد. بدین مفهوم که تمامی ارتباطات در قطعات زمانی ثابت[۱] صورت گرفته و همواره هر ارتباط با Master آغاز می‌شود. هر تجهیز دارای یک آدرس ثابت ۶۴ بیتی است که به منظور آدرس دهی و سویچ ارتباط بکار می‌رود. با صفر کردن باس توسط مرکز اصلی (لبه یک به صفر) در واقع دیگر تجهیزات روی شبکه با آن سنکرون می‌شوند(البته با تایمینگ خودشان).
انواع سیگنالینگ روی این شبکه عبارتند از:

۱. سیگنال ریست و حضور
۲. سیگنال نوشتن صفر
۳. سیگنال نوشتن یک
۴. سیگنال خواندن داده

[۱] Time Slice

DOI: 10.30503/nripress.2020.296

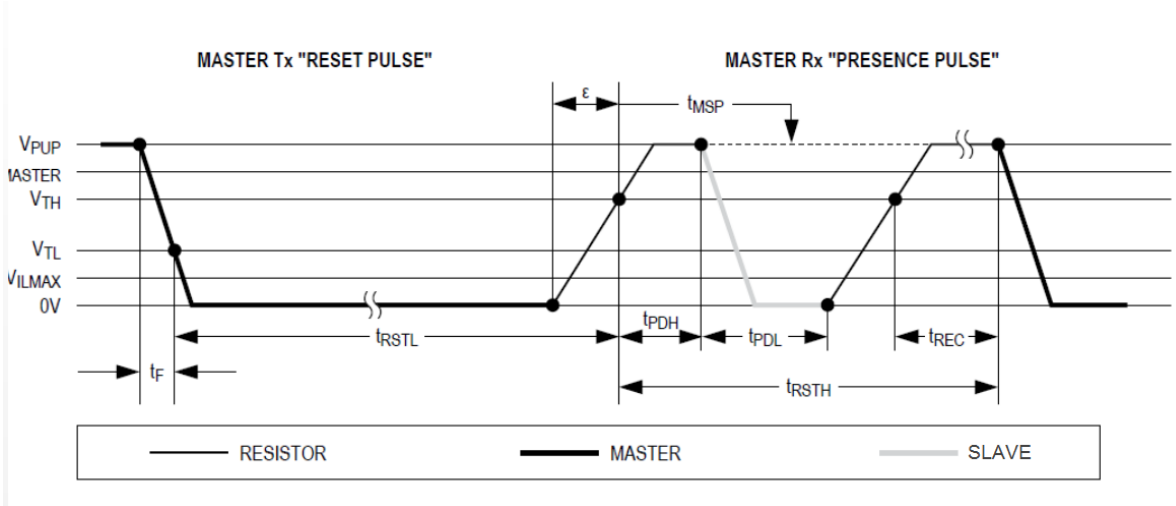
پالس ریست برای قراردادن تمامی تجهیزات روی شبکه به یک وضعیت مشخص آغازسازی[1] است. در واقع تمامی تجهیزات با دیدن این پالس، حضور خود را بوسیله پالس حضور اعلام می‌کنند.

سیگنال‌های نوشتن صفر و یک برای ارسال اطلاعات از مرکز به تجهیز بکار رفته و سیگنال خواندن داده به منظور ارسال اطلاعات از تجهیز به مرکز می‌باشد.

در این شبکه، قطعه زمانی هر بیت حدود ۶۰ میکر ثانیه می‌باشد که ۱۵ میکروثانیه اول آن اختصاص به مرکز اصلی داشته و باس داده را صفر میکند. همانطور که اشاره شد این لبه پایین رونده مبنای سنکرون شدن تجهیزات Slave با Master هستند. بعد از این مدت تا انتهای قطعه زمانی ۶۰ میکرو ثانیه, مدت زمانی است که باس داده یا توسط Master و یا توسط Slave تحت کنترل قرار می‌گیرند. زمانی که Master به تجهیز داده‌ای را می‌فرستد حالت نوشتن سیگنال بوده و وضعیت باس داده توسط مرکز کنترل شده و در این مدت تجهیزات اطلاعات را می‌خوانند. اما اگر اطلاعات از تجهیزات برای مرکز ارسال شود حالت خواندن یا قرائت داده بوده که کنترل وضعیت باس توسط تجهیز بوده و مرکز اطلاعات را می‌خواند.

طول پالس ریست ۸ برابر یک قطعه زمانی بیت پایه است که حدود ۴۸۰ میکروثانیه می‌شود. در این حالت Master باس را ۴۸۰ میکروثانیه صفر کرده و سپس رها می‌کند و منتظر پاسخ می‌ماند. حضور تجهیزات روی شبکه با پالس حضور (صفر کردن باس بمدت حداقل ۱۲۰ میکروثانیه) که مشابه پالس ریست است به مرکز اعلام می‌شود. شکل (۳-۱) پالس ریست و حضور را نشان داده است.

مطابق شکل (۱-۳) وضعیت باس در هر لحظه یا توسط مرکز و یا تجهیز و یا با مقاومت بالاکشنده کنترل می‌شود که در شکل با رنگهای متفاوت خطوط نشان داده شده است.



شکل ۱-۳- سیگنال ریست و حضور

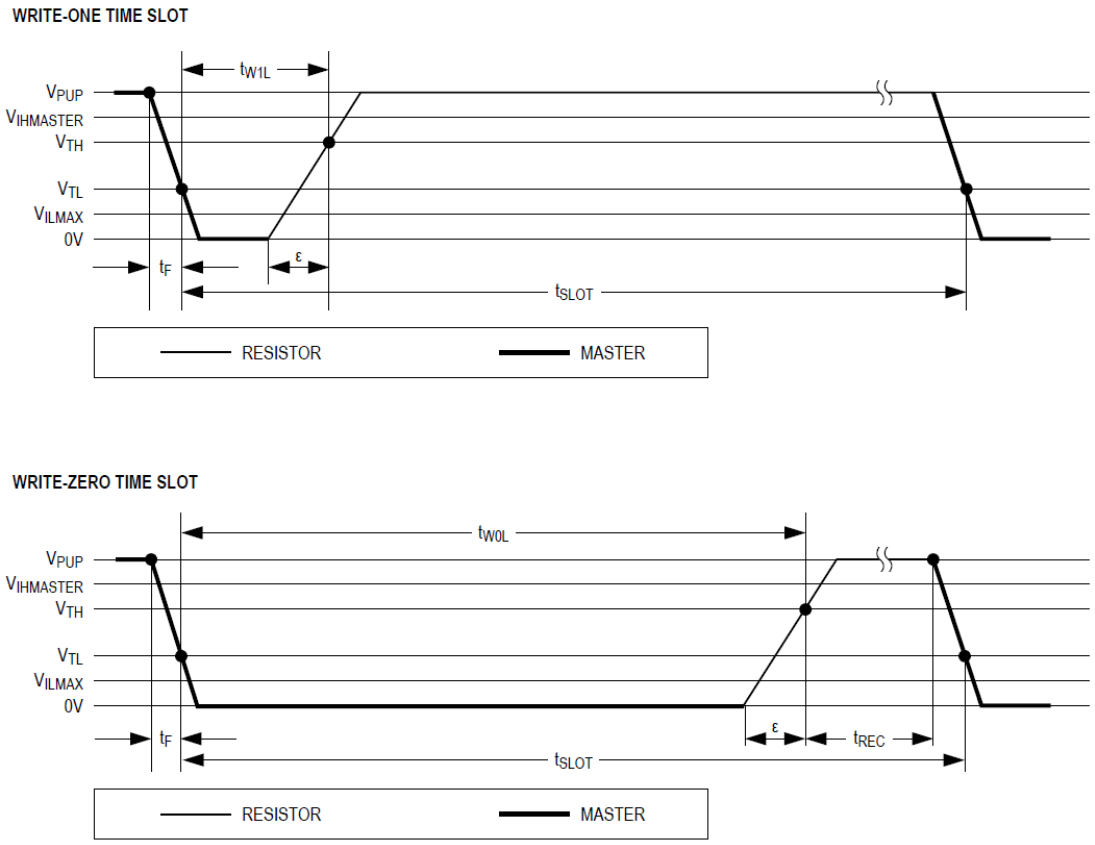
سیگنال نوشتن یک، (ارسال اطلاعات به تجهیز) با صفر کردن باس توسط Master به مدت ۱۵ میکروثانیه و سپس رها کردن باس (بالا آمدن باس از طریق مقاومت بالاکشنده) تا انتهای قطعه زمانی ۶۰ میکروثانیه صورت می‌گیرد.

سیگنال نوشتن صفر، با صفر کردن باس در کل قطعه زمانی ۶۰ میکروثانیه صورت می‌گیرد.

در شکل (۱-۴) این دو پالس نشان داده شده است.

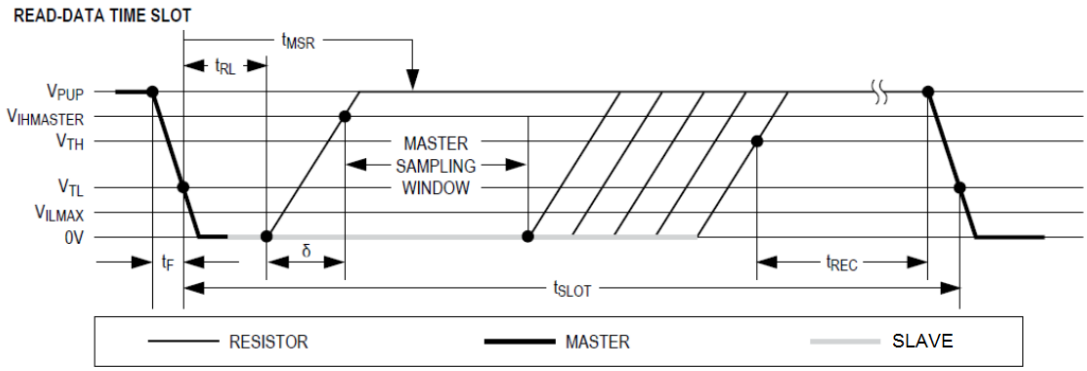
[1] Initialization

DOI: 10.30503/nripress.2020.296



شکل ۱-٤- سیگنال نوشتن یک و صفر

سیگنال خواندن به منظور ارسال اطلاعات از تجهیز به مرکز می‌باشد. همانطور که اشاره شد در این حالت مرکز ابتدا باس را به مدت ۱۵ میکروثانیه صفر کرده و سپس آنرا تا انتهای قطعه زمانی ۶۰ میکروثانیه رها می‌نماید. در مدت رهاسازی باس، وضعیت آن می‌تواند از طرف تجهیز تحت کنترل (صفر یا یک شدن) قرار گیرد و در این حین، مرکز این اطلاعات را می‌خواند. سیگنال قرائت داده در شکل (۱-۵) نشان داده شده‌است.



شکل ۱-۵- سیگنال قرائت داده

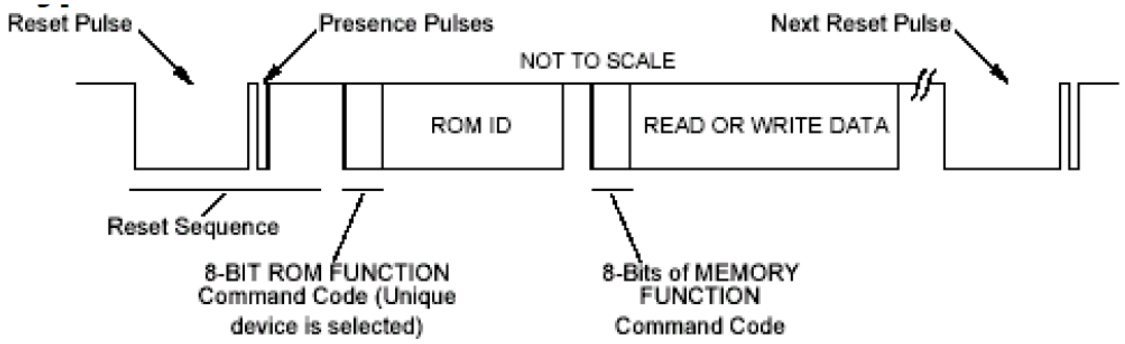
هر تجهیز Slave دارای یک آدرس ۶۴ بیتی است که در بخش حافظه Rom نگهداری می‌شود و به کمک دستورات مشخصی قابل دستیابی هستند. علاوه بر این اطلاعات، داده‌های دیگری نیز می‌توانند در بخش حافظه Ram که حجم آن تا ۶۴ کیلو بایت قابل افزایش است قرار گیرند. این بخش از حافظه نیز با دستورات مشخص دیگری قابل دسترس هستند. نوع داده و چگونگی دسترسی به بخش Ram در اختیار سازنده تجهیز بوده و یا در برخی موارد تحت کنترل کاربر قرار می‌گیرد.

DOI: 10.30503/nripress.2020.296

در یک محاوره، ترتیب دستورات ارسالی از مرکز به یک تجهیز شامل

۱. آغازسازی
۲. دستورات ROM
۳. دستورات RAM
۴. ارسال/دریافت داده

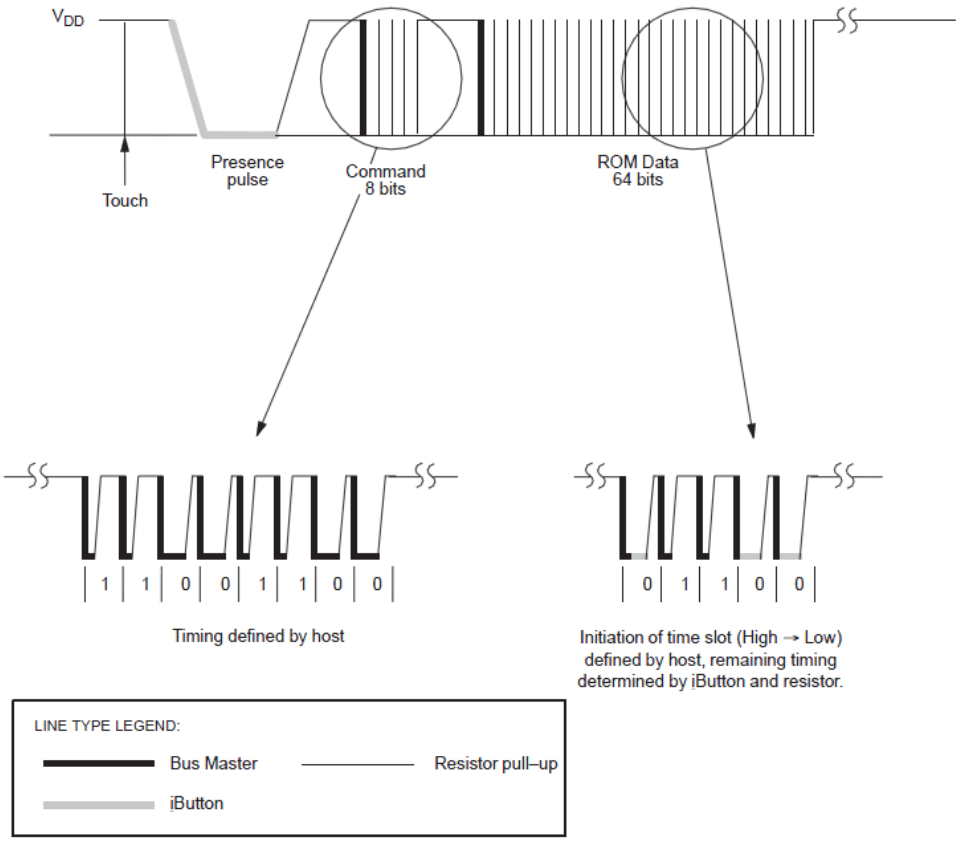
می‌باشد که همواره هر ارتباط با پالس ریست به عنوان آغازسازی شروع می‌شود سپس دستورات مربوط به ROM و پاسخ آنها را داریم و در آخر دستورات مربوط به RAM ارسال شده و داده‌های مربوطه دریافت و ارسال می‌شوند. اگر در حین مراحل فوق پالس ریست داشته باشیم نشان دهنده خاتمه ارتباط می‌باشد. شکل (۱-۶) این ترتیب را نشان می‌دهد



شکل ۱-٦- ترتیب ارسال دستورات در یک محاوره[1]

بصورت نمونه شکل (۱-۷) نحوه خواندن داده از Rom یک تجهیز Slave را بصورت تفصیلی نشان می‌دهد. توجه شود که تجهیزات ibotton در واقع همان تجهیزات Slave هستند.

DOI: 10.30503/nripress.2020.296



شکل۱-۷- فرایند خواندن اطلاعات ROM [1]

همانطور که از شکل پیداست کنترل وضعیت باس در هر لحظه یا با مرکز اصلی و یا با تجهیز و یا از طرف مقاومت بالاکشنده می‌باشد و همواره شروع ارتباط با صفرکردن باس توسط مرکز اصلی می‌باشد. پالس ریست به منظور آغازسازی یک ارتباط بوده که اگر مرکز پالس حضور را دریافت کند بدنبال آن می‌تواند دستور قرائت از Rom را فرستاده و جواب آن را از طرف تجهیز دریافت نماید.

۳-۱- کیفیت سیگنال[۱] شبکه Microlan

کیفیت و صحت سیگنال شبکه، مهمترین پارامتر در کارکرد صحیح شبکه می‌باشد. در این بخش برخی از پارامترهای تاثیر گذار بصورت خلاصه اشاره می‌شوند و چون در این پروژه ارتقاء کیفیت سیگنال هدف نیست و راه‌اندازی آن به منظور تامین یک ارتباط پایدار کفایت می‌کند بنابراین بررسی جزئیات عوامل تاثیر گذار دیگر بر روی کیفیت سیگنال به خواننده واگذار میشود.

لبه تیز پایین رونده مرکز اصلی و یا تجهیزات می‌تواند باعث ایجاد موج رفت و برگشت در یک شبکه طویل شده و اختلال شبکه را بهمراه داشته باشد. یک راه حل، استفاده از امپدانس موجی معادل، در این شبکه است که بخاطر افزایش مصرف امکان پذیر نمی‌باشد. راه حل عملی دیگر، کاستن از شیب لبه سیگنال پایین رونده است. بعنوان نمونه

[۱] Signal Integrity

DOI: 10.30503/nripress.2020.296

برای شبکه‌ای بطول ۱۰۰ متر و بالاتر، نرخ کاهشی حدودی ۱ ولت بر میکروثانیه مناسب است. لذا در طراحی تجهیزات شبکه این نکته رعایت میشود.

رها کردن سیگنال باس توسط تجهیزات شبکه، باعث برگشت سطح ولتاژ باس از طریق مقاومت بالاکشنده به ولتاژ تغذیه می‌شود. اما این برگشت متاثر از خازن خط و خازن تجهیزات متصل به خط، با ثابت زمانی RC خواهد بود. بنابراین هرچه خط طولانی‌تر و یا تجهیزات آن بیشتر باشد سرعت برگشت به حالت نرمال کند شده و بدنبال آن سرعت انتقال داده در شبکه کم می‌شود بطوریکه اگر این زمان از زمان پایه بیت بیشتر شود باعث توقف ارتباطات شبکه می‌شود. با توجه به تغییرات شبکه (افزودن و کاستن تجهیزات) در حین استفاده، استفاده از مدار بالاکشنده اکتیو راه حل مناسبی به منظور افزایش شیب برگشت سطح ولتاژ باس به سطح منطقی یک (حدود ۲.۲ ولت) می‌باشد.

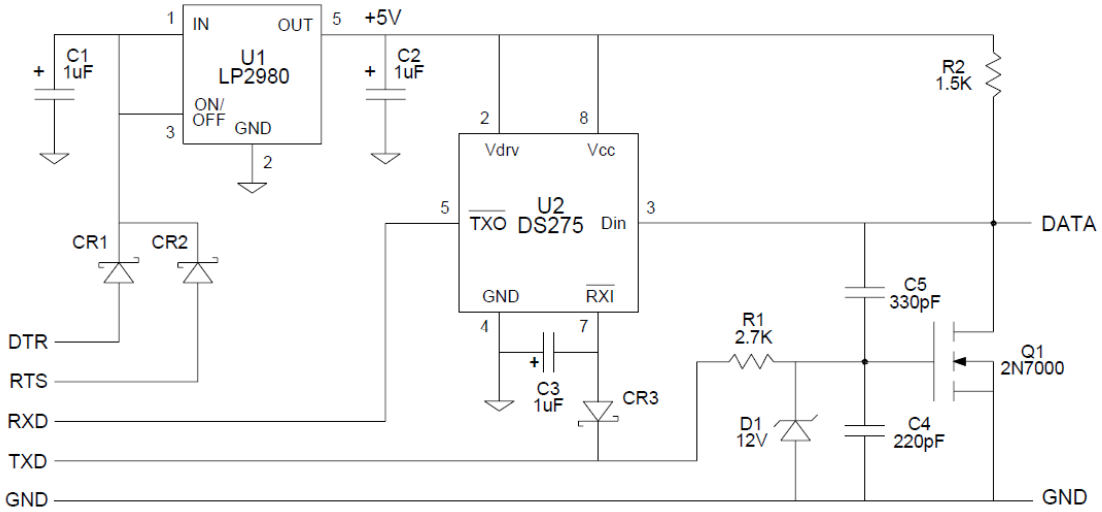
افزایش تعداد تجهیزات روی باس همواره باس را با مشکلات اشاره شده مواجه خواهد ساخت. یک راهکار دیگر کاهش مشکلات در شبکه با نودهای زیاد، شکستن شبکه به چند زیر شبکه کوچکتر و استفاده از سویچ یا هاب برای ارتباط با آنهاست. شرکت دالاس برای این منظور نیز چیپ‌هایی را معرفی کرده است.

بر اساس اشارات فوق، مشخصات کابل ارتباطی نیز می‌تواند در کیفیت سیگنال بسیار موثر باشد میزان خازن و مقاومت و سلف واحد طول کابل از دیگر پارامترهای تاثیر گذار بوده که بایستی در طراحی شبکه مد نظر باشند.

٤-١- سخت‌افزار درایور Master یک شبکه تک سیمه یا 1-Wire

معمولا از کامپیوتر به عنوان مرکز اصلی یا Master یک شبکه تک سیمه استفاده می‌شود و یا ممکن است از دستگاه‌های پرتابل دیجیتال برای این منظور استفاده شود.

می‌توان به کمک یک درایور مناسب، بصورت مستقیم از پورت سریال برای ایجاد یک شبکه تک سیمه استفاده نمود. بعبارت دیگر با مناسب سازی سطح سیگنال‌های پورت سریال از ±۱۲ ولت به سطح TTL می‌توان از آن یک شبکه تک سیمه ساخت. بعلاوه با استفاده از دیگر سیگنال‌های پورت سریال می‌توان تغذیه درایور را نیز تامین نمود و بنابراین یک درایور مستقل از تغذیه خارجی ساخت. در شکل (۱-۸) یک نمونه از آن نشان داده شده است.

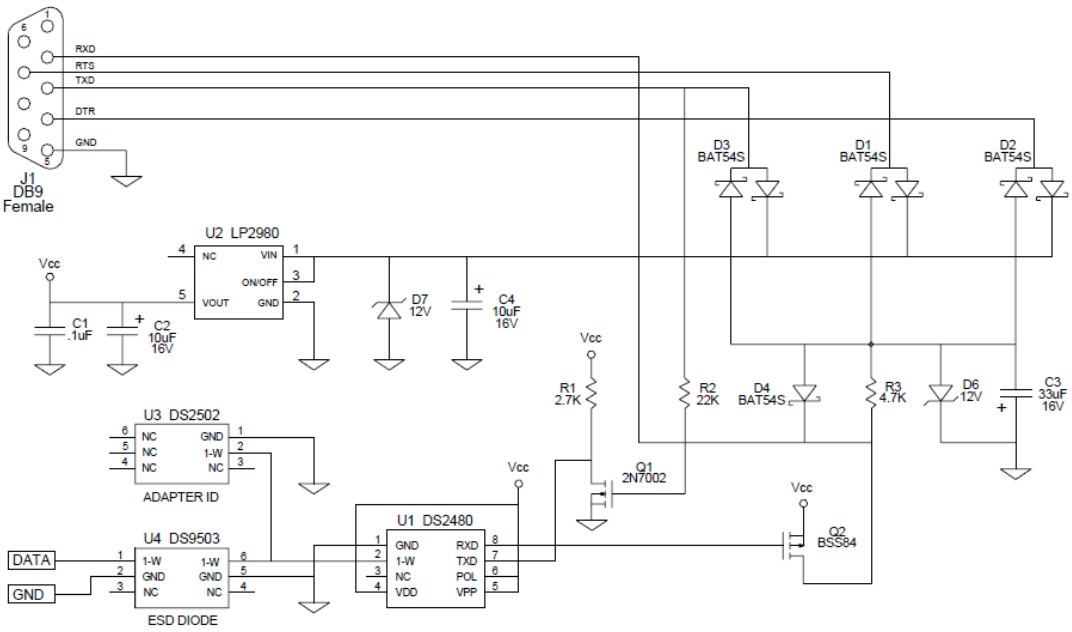


شکل ۱-۸- درایور مستقیم پورت سریال به شبکه 1-Wire

همانطور که از شکل (۱-۸) پیداست از دو پین آزاد پورت سریال RTS و DTR که دارای سطح ولتاژ ۱۲ ولت هستند استفاده شده و به کمک رگولاتور LP2980 ولتاژ تغذیه ۵ مورد نیاز درایور تهیه شده است. این ولتاژ ۵ ولت با مقاومت بالاکشنده ۱.۵ کیلواهم و ماسفت 2n7000 ، باس تک سیمه را ساخته است. گیت این ماسفت از طریق

DOI: 10.30503/nripress.2020.296

سیگنال TX پورت سریال کنترل شده و باس تک سیمه را می‌تواند صفر کند. اما دریافت سیگنال از باس تک سیمه به RX پورت سریال، به کمک یک چیپ مبدل ttl به سریال DS275 فراهم شده است. بعلاوه به کمک خازنهای C5 و C4 و مقاومت R1 لبه پایین رونده باس تک سیمه تا حدی قابل تنظیم است. مدار فوق از یک مقاومت بالاکشنده ثابت استفاده نموده و اکتیو نمی‌باشد. در طرح دیگری که در شکل (۱-۹) آمده است به کمک یک چیپ ویژه، این نقیصه رفع شده و بعلاوه کار با آن نیز ساده تر شده است.



شکل ۱-۹- درایور پورت سریال به شبکه 1-Wire به کمک چیپ DS2480

در این طرح از یک چیپ اختصاصی تبدیل پورت سریال به باس تک سیمه استفاده شده است که این چیپ نیز توسط شرکت دالاس معرفی شده و از قابلیت‌های خوبی برخوردار است. بطوریکه اکثر درایورها از آن استفاده می‌کنند. چیپ DS2480 می‌تواند با چند سرعت ثابت از پورت سریال کار کرده و تایمینگ باس تک سیمه را تولید نماید. در این حالت ارسال دستورات و داده بصورت بایتی می‌باشند لذا کار با آن ساده تر است. بعلاوه این چیپ دارای مدار بالاکشنده اکتیو و تنظیم کننده لبه پایین رونده می‌باشد. همچنین این درایور به کمک پین ۱۲ ولت خود می‌تواند حافظه‌های EPROM را نیز برنامه ریزی نماید. حافظه‌های EEPROM به چنین سطح ولتاژی نیاز نداشته و لذا می‌توان آنرا حذف نمود.

در طرح فوق نیز، از سیگنال‌های RTS و DTR و رگولاتور LP2980 برای تامین تغذیه ۵ ولت درایور استفاده شده است. و دو چیپ در خروجی باس تک سیمه دیده می‌شود که یکی DS2503 بمنظور حفاظت باس بوده و جیپ را در برابر ESD حفاظت می‌کند و دیگری یک چیپ شناسایی DS2502 است. این چیپ شناسایی می‌تواند از طرح حذف شود.

در طرح اول ایجاد سیگنالینگ مناسب روی باس تک سیمه باید با تغییر سرعت پورت سریال صورت گرفته و متناظر با هر بیت تک سیمه، یک بایت روی پورت سریال ارسال و یا دریافت می‌شود. لذا تفسیر دادهای ارسالی و دریافتی حجیم و پیچیدگی نرم افزار را بیشتر می‌نماید. اما در طرح دوم داده ارسالی و یا دریافتی از باس تک سیمه، بصورت بایتی بوده و تبدیل پروتکل و تنظیم سطح سیگنال و... با چیپ درایور DS2480 می‌باشد بدین لحاظ، این چیپ از پیچیدگی نرم‌افزاری کاسته و کار را راحتتر می‌نماید. و به عبارت دیگر پردازنده Master کمتر درگیر این درایور می‌شود.

DOI: 10.30503/nripress.2020.296

درایورهای دیگری نیز برای تبدیل پورت usb به باس تک سیمه وجود دارند که در آنها نیز از چیپ‌های آماده این کار استفاده می‌شوند. البته بررسی آنها در این گزارش لازم نمی‌باشد.

معرفی کوتاهی از انواع چیپ اندازه‌گیری مجهز به باس تک سیمه

شرکت آمریکایی دالاس پس از معرفی این شبکه تک سیمه، یک سری چیپ‌های درایور و جانبی برای ساخت و راه‌اندازی این شبکه معرفی کرده است که در فوق نمونه‌های از آن اشاره شد. بعلاوه چیپ‌های حافظه EPROM و EEPROM را نیز در ظرفیت‌های مختلف تا ۶۴ کیلوبیت ارائه داده است که می‌توانند با این باس کار کنند. این چیپها می‌توانند اطلاعات بسیاری علاوه بر شناسایی و معرفی تجهیزات به همراه داشته باشند. در راستای استفاده بیشتر از این شبکه تک سیمه، این شرکت یک مجموعه از چیپ‌های اندازه‌گیری را نیز معرفی نموده است که با این باس می‌توانند کار کنند.

بدین ترتیب می‌توان مجموعه و یا آرایه‌ای از این سنسورها را روی یک باس تک سیمه قرار داده و تنها با دو سیم به اطلاعات تمامی آنها دست یافت. در ادامه مواردی از این چیپ‌ها را معرفی می‌کنیم

۱. چیپ DS2423 که یک شمارنده می‌باشد
۲. چیپ DS2438 دارای دو مبدل ADC و یک سنسور دما است. از این چیپ به کمک چیپ‌های جانبی می‌توان سنسور رطوبت و فشار و یا سنسور میزان نور ساخت
۳. چیپ DS2450 دارای چهار مبدل ADC است که به کمک آن و تجهیزات جانبی می‌توان سنسورهای زیادی را ساخته و ارائه نمود مثلا به همراه چند سویچ مغناطیسی و مدار جانبی آن، می‌توان از آن بعنوان بادنما و تشخیص جهت باد استفاده کرد. البته کاربردهای دیگری می‌توان داشت که در این مقوله نمی‌گنجد.
۴. چیپ DS2760 یک چیپ برای نظارت بر باتری‌های لیتوم-یون است و دارای قابلیت‌های داخلی متعددی شامل مبدل ADC تا ۱۳ بیت دقت است که می‌توان از آن برای آشکارسازی ولتاژهای کم سنسورهای ترموکوپل استفاده نمود. بعلاوه وجود یک سنسور دمای داخلی، امکان جبرانسازی اتصال سرد را نیز ممکن ساخته است همچنین وجود حافظه داخلی ۳۲ بایتی با قابلیت قفل‌کردن می‌تواند در معرفی سنسور کمک نماید.

شایان ذکر است که با قطعات فوق می‌توان سنسورهای متعددی را ساخت که کمک بسیاری در پیاده سازی یک شبکه سنسوری منطبق با باس تک سیمه می‌نماید. برای اطلاعات بیشتر خواننده به مرجع [1] ارجاع داده می‌شود.

۵-۱- کاربرد مستقیم پورت سریال بعنوان مرکز کنترل باس 1-Wire

مرکز کنترل باس تک سیمه، در واقع همان تجهیز Master است که به کمک این باس با تجهیزات Slave ارتباط برقرار می‌نماید. در این بخش نکات سخت‌افزاری و نرم‌افزاری برای برقرای این ارتباط از مسیر پورت سریال با درایور ساده شکل (۱-۸) مطرح می‌گردد.

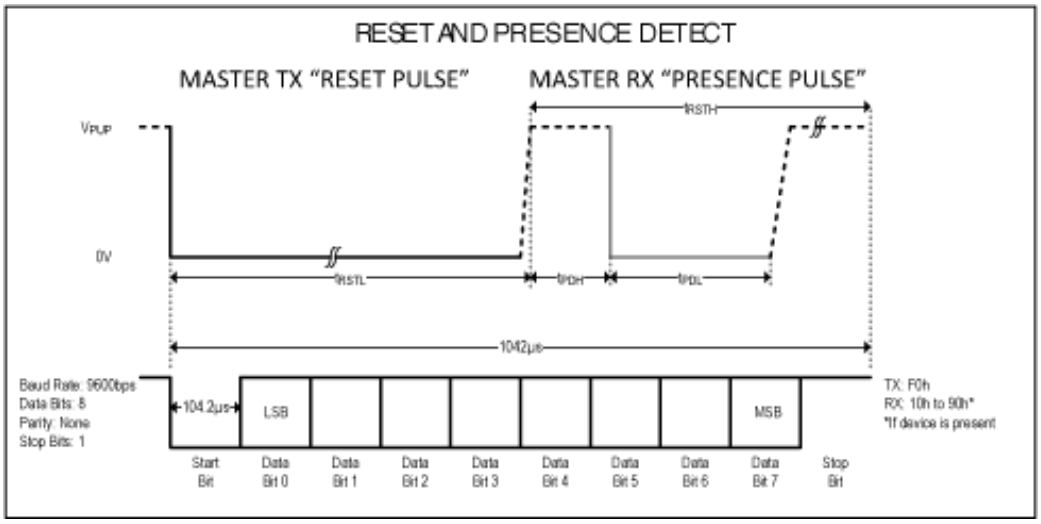
در این درایور برای ایجاد انواع سیگنالینگهای مورد نیاز زیر

۱. سیگنال ریست و حضور
۲. سیگنال نوشتن صفر

DOI: 10.30503/nripress.2020.296

۳. سیگنال نوشتن یک
۴. سیگنال خواندن داده

با تغییر سرعت پورت سریال امکان‌پذیر می‌شود. برای مثال برای ایجاد پالس ریست و حضور با پهنای پالس تعریف شده لازم است که از سرعت ۹۶۰۰ بیت بر ثانیه استفاده شود. بدین ترتیب با این سرعت پهنای هر بیت در حدود ۱۰۴ میکروثانیه بوده و با ۴ بیت اول پالس ریست ساخت شده و بیت‌های باقیمانده بایت می‌توانند پالس حضور را نیز آشکار کنند. در شکل (۱-۱۰) این موضوع نشان داده شده است.

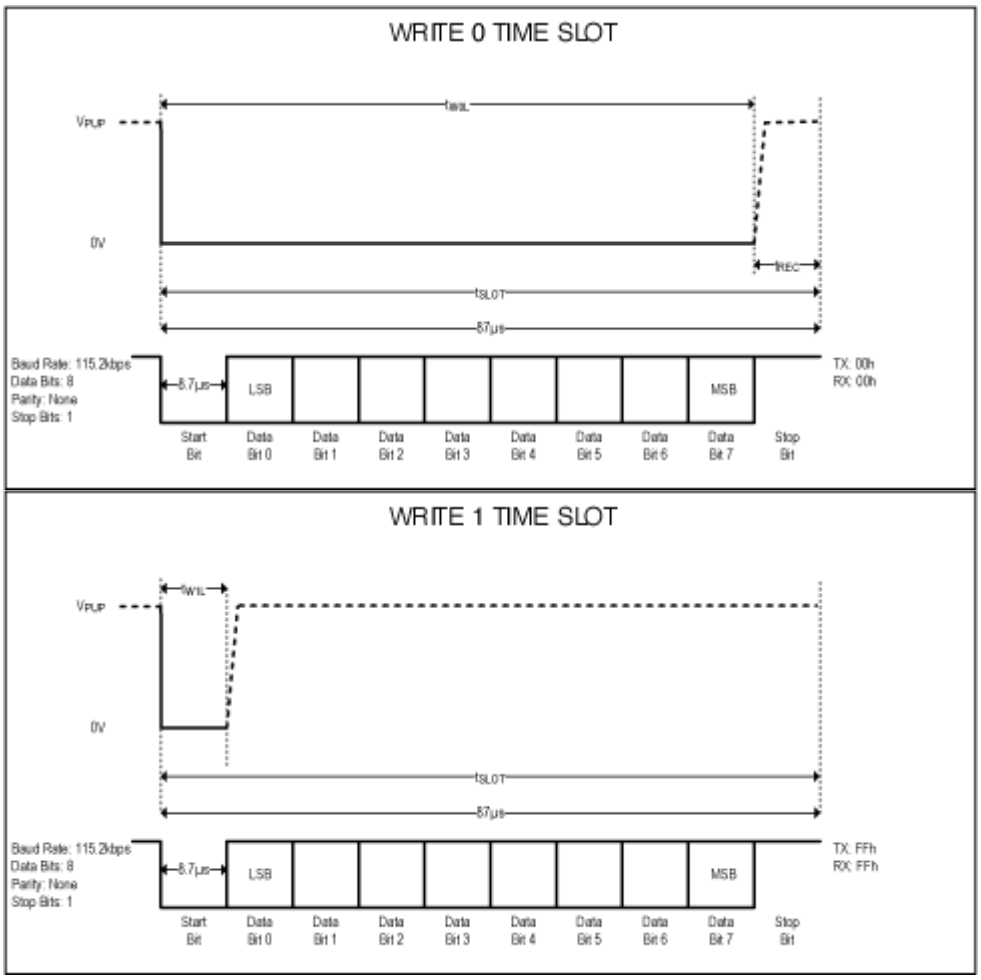


شکل ۱-۱۰- تولید سیگنال ریست و حضور

در واقع با ارسال F0h چهار بیت اول صفر و بقیه یک میشوند. بدین ترتیب پالس ریست با عرض ۵۲۰ میکروثانیه ساخته میشود. پاسخ دریافتی از پورت بستگی به وجود تجهیزات slave داشته و بسته به تعداد آنها عرض پالس حضور بیشتر می‌شود. اگر پاسخ دریافتی برابر مقدار ارسالی بود که هیچ تجهیزی روی باس نیست و اگر یک تجهیز باشد پاسخ E0h و در صورت وجو تعداد بیشتر می‌تواند تا 90h برسد.

برای ایجاد پالس‌های نوشتن صفر و یک و یا قرائت آنها چون عرض پالس آنها کمتر است از سرعت ۱۱۵۲۰۰ بیت بر ثانیه پورت سریال استفاده می‌کنیم. در این حالت پهنای هر بیت حدود ۸.۷ میکروثانیه خواهد بود که امکان ارسال و دریافت پالس‌های تا ۶۰ میکروثانیه را در طول یک بایت فراهم می‌سازد. در شکل (۱-۱۱) چگونگی تولید پالس‌های نوشتن صفر و یک را نشان داده است.

DOI: 10.30503/nripress.2020.296

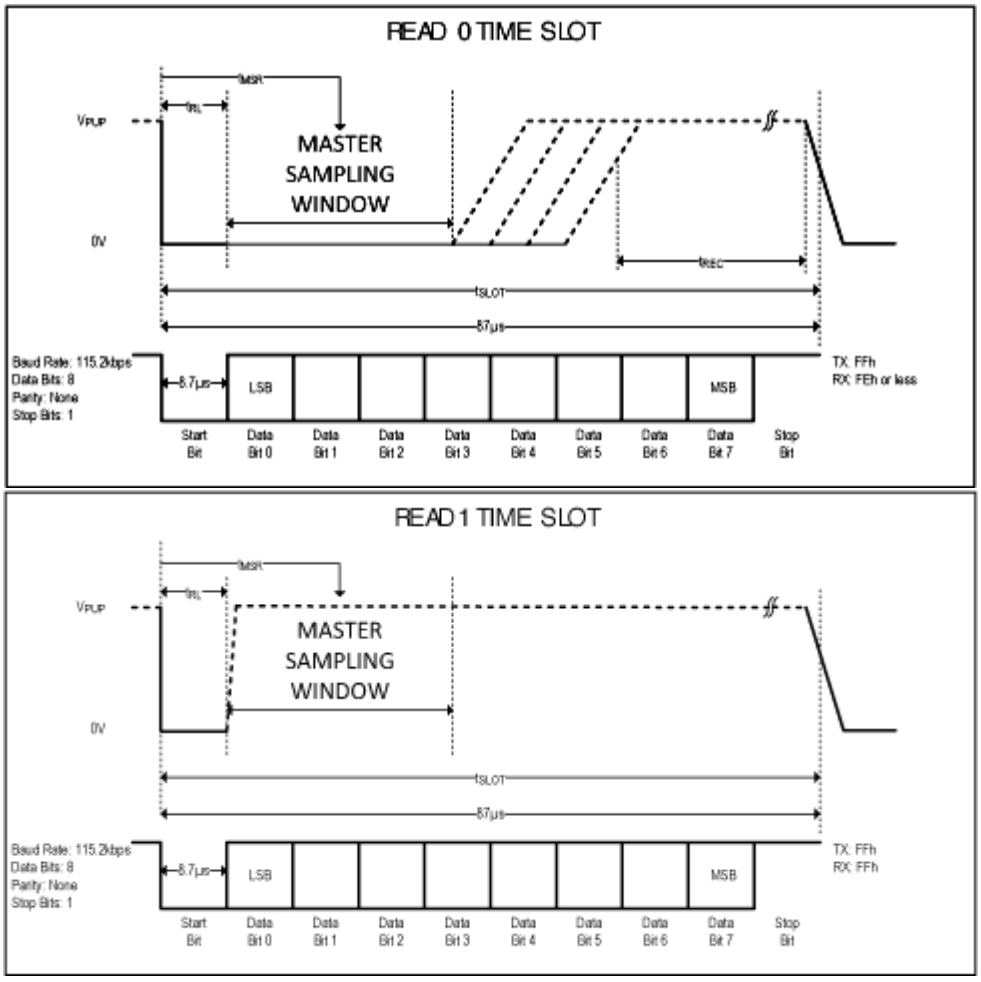


شکل ۱-۱۱- تولید سیگنال نوشتن یک و صفر

در حالت نوشتن یا ارسال داده هر آنچه که از پورت سریال فرستاده می‌شود همان دریافت خواهد شد و شکل پالس از طرف تجهیز تغییر نمی‌نماید. بطوریکه ارسال 00h باعث نوشتن صفر و دریافت پاسخی مشابه از پورت سریال است. و ارسال ffh نیز باعث نوشتن یک و دریافت پاسخ ffh روی پورت سریال است.

اما در پالس قرائت داده Master پالس صفری به عرض ۱۵ میکروثانیه ساخته و سپس باس را رها میکند. بعد از این زمان تجهیزات Slave می‌توانند کنترل وضعیت باس را در دست گرفته و داده مورد نظر خود را قرار دهند. بنابراین ارسال بایت ffh اگر با قرائت صفر همراه باشد پورت سریال مقادیری در رنج FEh تا 00h خواهند داشت اما اگر قرائت یک باشد همان مقدار FFh را دریافت خواهیم کرد. چون هیچ تغییری از طرف Slave روی سطح ولتاژ صورت نمی‌گیرد. شکل (۱-۱۲)

DOI: 10.30503/nripress.2020.296



شکل ۱-۱۲- تولید سیگنال خواندن داده

مشاهده می‌شود که به ازاء هر بیت ارسال و یا دریافت از باس تک سیمه، روی پورت سریال، یک بایت ارسال و یا دریافت می‌شود که برای یک ارتباط کامل، رشته‌هایی از بایت‌ها ارسال و یا دریافت می‌شوند. در این حالت لازم است برای داشتن زمان بازیافت[۱] (Trec) مناسب در شبکه، ارسال بایت‌ها با تاخیر مناسب صورت گرفته و پشت سرهم نباشد.

همانطور که اشاره شد در این مدل از درایور، پورت سریال بشدت درگیر شده و بدنبال آن نیز پردازنده‌ای که با این پورت کار می‌کند مشغول خواهد شد. در صورتی که اگر از چیپ DS2480 همانند درایور شکل (۱-۹) استفاده شود حجم کار پردازنده را کمتر کرده و با یک سرعت ثابت روی پورت سریال کار می‌کنیم و همچنین هر بایت سریال می‌تواند ۸ بیت داده را روی باس تک سیمه ارسال و یا دریافت نماید.(ارسال و دریافت کمتر)

۱-۶- آشنایی با یک نمونه حافظه EEPROM منطبق با باس تک سیمه

چیپ DS2430A یک حافظه EEPROM با ظرفیت ۲۵۶ بیت به همراه رجیسترهای جانبی است که با باس تک سیمه کار می‌کند. این چیپ دارای یک آدرس یکتای ۶۴ بیتی از نوع ROM بوده و می‌تواند از خط داده تغذیه شود.

[۱] Recovery Time

DOI: 10.30503/nripress.2020.296

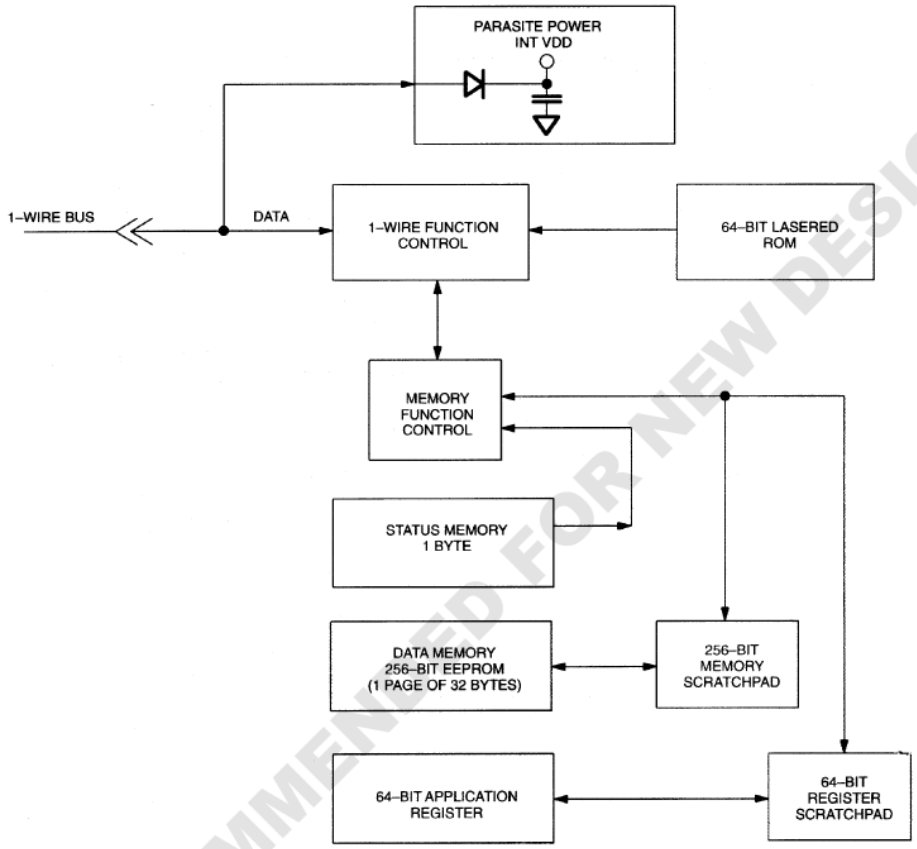
کل حافظه ۲۵۶ بیت بصورت یک بلوک ۳۲ بایتی قابل دسترس است. این چیپ به محض فعال شدن تغذیه حضور خود را با پالس حضور اعلام می‌نماید.

این حافظه کوچک امکان ذخیره اطلاعات مربوط به یک تجهیز را در یک فضای محدود فراهم می‌سازد. این چیپ دارای ۴۸ بیت آدرس یکتا و ۸ بیت CRC و ۸ بیت کد خانودگی (14h) است که در بخش حافظه ROM آن قرار گرفته است. در شکل (۱-۱۳) نقشه این بخش حافظه را نشان می‌دهد.

8-Bit CRC Code		48-Bit Serial Number		8-Bit Family Code (14H)	
MSB	LSB	MSB	LSB	MSB	LSB

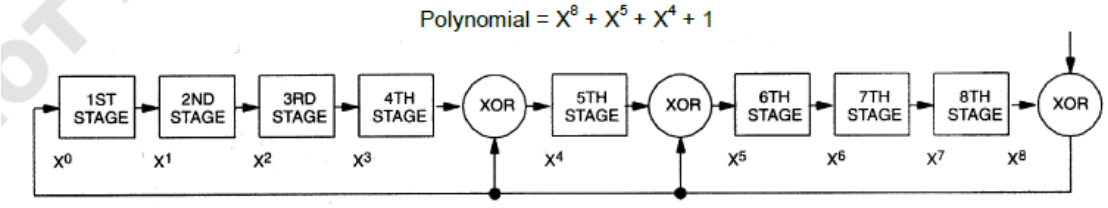
شکل ۱-۱۳- ساختار داده ROM

بعد از بخش ROM بخش EEPROM با ۲۵۶ بیت حافظه با قابلیت نوشتن و پاک کردن قرار دارد. بعلاوه در انتها، دارای ۶۴ بیت حافظه از نوع یکبار قابل نوشتن است. بلوک دیاگرام شکل (۱-۱۴) قسمت‌های مختلف داخلی چیپ را نشان می‌دهند.



شکل ۱-۱۴- بلوک دیاگرام داخلی DS2430

این چیپ با کد خانوادگی 14h شناخته میشود. به منظور تشخیص صحت اطلاعات دریافتی از کدینگ CRC استفاده می‌شود. روش کدینگ بکار رفته در شکل (۱-۱۵) نشان داده شده است



DOI: 10.30503/nripress.2020.296

شکل ۱-۱۵- الگوریتم تولید کدینگ CRC

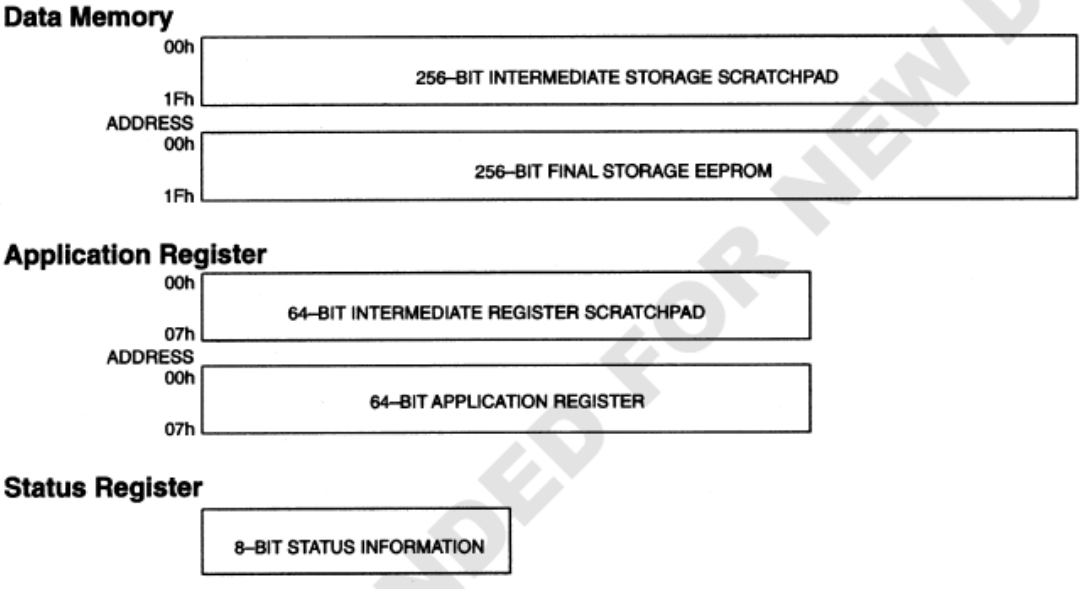
بدین منظور ابتدا شیفت رجیستر ریست می‌شود سپس بیت صفر کد خانوادگی شیفت داده میشود تا بیت هشتم آن نیز تمام شود بدنبال آن بیت‌های سریال شیفت می‌شوند بطوریکه بعد از آخرین بیت سریال یعنی بیت ۴۸ ، کد CRC ساخته شده است اگر بیت‌های CRC دریافتی در ادامه شیفت شوند بایستی در انتها همه بیت‌ها صفر شوند. در این صورت اطلاعات دریافتی صحیح بوده و در غیر اینصورت با خطا همراه بوده است. در شبکه تک سیمه همواره ارسال بایت‌ها با بیت LSB شروع شده و در آخر MSB ارسال می‌گردد.

۱-۶-۱- حافظه RAM و دستورات آن

این بخش حافظه به سه بخش تقسیم میشود

۱. بخش داده
۲. بخش رجیسترهای کاربردی
۳. بخش رجیستر وضعیت

بخش‌های داده و رجسترهای کاربردی دارای حافظه‌های موقت و یا چکنویس[۱] بوده و بعنوان بافر به هنگام نوشتن در حافظه بکار میروند. بخش داده دارای قابلیت خواندن و نوشتن به دفعات بوده اما بخش رجیسترهای کاربردی، با یکبار نوشتن امکان نوشتن مجدد را بصورت اتوماتیک از دست می‌دهند و تنها قابل خواندن هستند. زمانی که رجیسترهای کاربردی نوشته نشده باشند مقدار رجیستر وضعیت FFh می‌باشد و در صورت نوشتن در رجیسترهای کاربردی، مقدار آن به FCh تغییر می‌یابد(صفر شدن دو بیت اول). در شکل (۱-۱۶) نقشه حافظه داده این چیپ را نشان می‌دهد.



شکل ۱-۱۶- نقشه حافظه

دستورات دسترسی به حافظه عبارتند از

۱. نوشتن به چکنویس با کد 0Fh

[۱] Scratchpad

DOI: 10.30503/nripress.2020.296

بعد از ارسال این دستور مرکز باید یک بایت آدرس و سپس داده‌های نوشتنی را پشت سرهم ارسال نماید تا به پالس ریست برسد. هر بایت دریافتی، آدرس را بصورت اتوماتیک افزوده و اگر به 1Fh برسد آدرس به 00h برمی‌گردد.

۲. خواندن از چکنویس با کد AAh

این دستور به منظور تایید محتوای حافظه چکنویس قبل از نوشتن به حافظه بکار می‌رود. بعد از ارسال این دستور آدرس شروع و سپس داده‌ها پشت سرهم ارسال می‌شوندو انتهای کار با پالس ریست مشخص می‌شود.

۳. کپی چکنویس 55h

بعد از اطمینان از صحت اطلاعات چکنویس مرکز می‌تواند آنرا به حافظه کپی کند. مرکز بعد از ارسال این دستور یک کد اعتبار A5h باید ارسال نماید تا این کپی صورت گیرد. این دستور تمام حافظه چکنویس را کپی می‌کند.

۴. خواندن حافظه با کد F0h

این دستور برای خواندن کل و یا بخشی از حافظه بکار می‌رود. این دستور برای کپی کل اطلاعات از حافظه به چکنویس نیز بکار میرود. بدین منظور بعد از ارسال این دستور باید یک پالس ریست باشد. اگر بعد از دستور یک آدرس نیز ارسال شود بعد از کپی حافظه به چکنویس، داده‌ها از آدرس مورد نظر از چکنویس خوانده می‌شوند. افزایش اتوماتیک آدرس و بازگشت به آخر به اولین ادرس نیز همواره وجود دارد.

۵. نوشتن به رجیسترهای کاربردی با کد 99h

این دستور همانند نوشتن به چکنویس می‌باشد تنها تفاوت در نهایت این حافظه به رحیسترهای کاربردی نوشته می‌شود نوشتن تا زمانی که دستور کپی و قفل داده نشده می‌تواند اتفاق افتد. اما پس از کپی دیگر این امکان وجود ندارد

۶. خواندن رجیستر وضعیت با کد 66h

خواندن رجیستر وضعیت به منظور تشخیص برنامه ریزی شدن رجیسترهای کاربردی بوده و پس از ارسال این دستور یک کد اعتبار 00h باید ارسال شود تا بدنبال آن وضعیت این رجیستر دریافت شود

۷. خواندن از رجیسترهای کاربردی C3h

این دستور برای قرائت از چکنویس رجیستری و یا مستقیما رجیستر کاربردی است. تا زمانی که حافظه رجیستر قفل نشده از چکنویس خوانده می‌شود اما بعد از قفل کردن از خود رجیستر اصلی است. بعد از این دستور مرکز باید آدرس شروع را نیز بدهد. با هر خواندن آدرس داده‌ها بصورت اتومایک افزوده شده و پس از 07h به 00h برمی‌گردد.

۸. کپی و قفل کردن حافظه رجیسترهای کاربردی با کد 5Ah

بعد از تایید نهایی داده‌های موجود در چکنویس رجیستری می‌توان با دستور فوق این اطلاعات را به رجیسترها کپی و همزمان آنرا قفل نمود. بطوریکه بعد از آن دیگر امکان تغییر وجود ندارد. بعد از ارسال این دستور یک کد تایید A5h ارسال می‌گردد. اگر قبل از این کد، پالس ریست داده شود دستور لغو می‌شود.

دستورات فوق برای دسترسی به حافظه EEPROM می‌باشند اما برای حافظه ROM نیر دستورانی است که اشاره خواهد شد.

بطور کلی ترتیب دستورات ارسالی بصورت زیر است:

۱. آغاز سازی

۲. دستورات ROM

DOI: 10.30503/nripress.2020.296

۳. دستورات بخش EEPROM
۴. داده‌ها

همانطور که اشاره شد آغازسازی همواره با پالس ریست بوده و بدنبال آن اگر مرکز پالس حضوری دریافت کند می‌تواند دیگر دستورات را نیز بترتیب ارسال نماید. در ادامه دستورات بخش ROM اشاره می‌شود.

۲-۶-۱- دستورات بخش ROM حافظه

برای کار با حافظه ROM دستورات زیر بکار می‌رود:

۱. خواندن ROM با کد 33h

ابن دستور به منظور خواندن اطلاعات ROM بوده که شامل ۸ بیت کد خانوادگی چیپ، ۴۸ بیت سریال و ۸ بیت CRC است. این دستور تنها برای زمانی است که فقط یک تجهیز روی شبکه است. اگر بیشتر از یک تجهیز وجود داشته باشد، چون همگی می‌خواهند جواب دهند، لذا خطای عدم تطابق CRC پیش خواهد آمد.

۲. تطبیق ROM با کد 55h

این دستور به منظور صدا زدن یک slave خاص با آدرس داده شده می‌باشد و تمامی دستورات بعدی فقط با این آدرس می‌باشد. بنابراین بعد از این دستور ۶۴ بیت آدرس مورد نظر توسط مرکز ارسال می‌شود تمام آدرس‌های ناهمسان با این آدرس، دیگر هیچ عملی انجام نداده و تنها منتظر پالس ریست هستند.

۳. گذر از ROM با کد CCh

این دستور در زمانی که تنها یک Slave وجود دارد بکار رفته و بدون دادن آدرس می‌توان مستقیما به اطلاعات حافظه دست یافت. اگر بیش از یک تجهیز روی شبکه باشند تصادم اطلاعات رخ داده و ارتباط با خطا روبرو شده و عدم تطابق CRC را خواهیم داشت.

۴. جستجوی ROM با کد F0h

زمانی که برای اولین بار شبکه فعال می‌شود مرکز تعداد و آدرس تجهیزات روی باس را نمی‌داند این دستور با یک آلگوریتم تکراری می‌تواند آدرس تمام تجهیزات روی شبکه را تشخیص دهد. یک پروسه سه مرحله‌ای و تکراری با خواندن بیت آدرس و معکوس آن از تجهیزات و مقایسه با مقدار مطلوب می‌تواند در انتهای یک جستجوی ۶۴ بیتی مرحله اول، یک آدرس موجود را آشکار نماید. برای یافتن دیگر آدرس‌ها باید الگوریتم ادامه یابد. که برای آگاهی بیشتر خواننده باید به پیوست مرجع [4] مراجعه نماید

DOI: 10.30503/nripress.2020.296

فصل دوم: مروری بر استاندارد IEEE1451.4

DOI: 10.30503/nripress.2020.296

مقدمه

فصل دوم گزارش، به مطالعه و بررسی استاندارد IEEE1451.4 اختصاص یافته است. اما به منظور آشنایی با خانواده استانداردهای IEEE1451 در بخش اول این فصل، بصورت خلاصه این استاندارد و مفاهیم و کاربرد بخش‌های آن مطرح، بررسی و ارائه شده است. در ادامه فصل، بخش چهارم این استاندارد یعنی IEEE1451.4 که ویژه این پروژه است به تفصیل در چند بخش آمده است. بطوریکه عناوین اصلی زیر را می پوشاند:

۱. ترانسدیوسر منطبق با استاندارد IEEE1451.4
۲. دیتاشیت الکترونیک
۳. الگوهای کاربردی دیتاشیت
۴. زبان برنامه نویسی الگو
۵. اینترفیس مد مخلوط MMI

شایان ذکر است که با توجه به مطالب ارائه شده در فصل اول خواننده می‌تواند از مطالب بخش اینترفیس مد مخلوط سریعتر گذر کرده و تنها موارد اختلاف با شبکه 1-Wire را مد نظر و مطالعه قرار دهد.

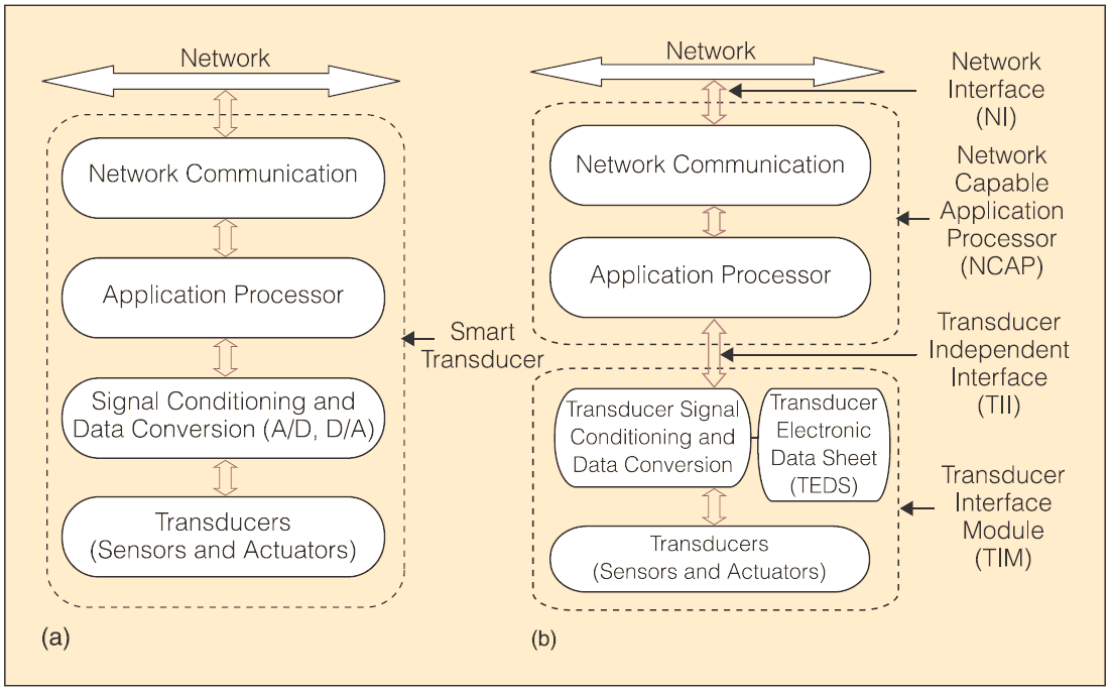
۱-۲- مروری بر استاندارد IEEE1451

DOI: 10.30503/nripress.2020.296

موسسه استاندارد IEEE مجموعه‌ای از استانداردها را برای سنسور و محرک هوشمند به نام IEEE1451 توسعه داده است. بر این اساس ترانسدیوسر هوشمند باید دارای تابع‌های لازم در تولید و نمایش صحیح کمیت اندازه گیری و یا کنترل شونده باشد. این توابع یکپارچه سازی ترانسدیوسر را در یک شبکه کاربردی آسان می‌نماید.

۱-۱-۲- ترانسدیوسر هوشمند

امروزه تکنولوژی دیجیتال تمام عرصه‌های زندگی را در نوردیده است. سیستم‌های کنترل صنعتی هم در بخش پردازش و هم در بخش اندازه‌گیری و درایو و هم در بخش ارتباطات سیگنال از این تکنولوژی تاثیر پذیرفته است و سیستم‌های آنالوگ قدیمی به سیستم‌های دیجیتال تبدیل شده‌اند.
ترانسدیوسر در واقع یک مبدل انرژی از نوعی به نوع دیگر است. یک سنسور در واقع ترانسدیوسری است که یک سیگنال الکتریکی متناسب با یک پارامتر فیزیکی یا شیمیایی و بیولوژیکی تولید مینماید و از طرف دیگر یک محرک نیز ترانسدیوسری است که متناسب با یک سیگنال الکتریکی یک عمل فیزیکی انجام می‌دهد. یک ترانسدیوسر هوشمند مجموعه‌ای از سنسور و محرک آنالوگ و یا دیجیتال به همراه بخش‌های پردازشی و ارتباطی است. یک ترانسدیوسر شامل سخت‌افزار و نرم‌افزارهای لازم جهت مناسب‌سازی سیگنال ، به همراه پردازش و ارتباط آن به دنیای خارج است. این امر به کمک میکروکنترلرها و نرم‌افراهای مربوطه، قابلیت‌های پردازشی و ارتباطی و بعلاوه قابلیت‌های کالیبراسیون و یا عیب‌یابی را برای یک سنسور و یا محرک به همراه دارند.
بر اساس آنچه که اشاره شده مدل یک ترانسدیوسر هوشمند در شکل (a)(۱-۲) نشان داده شده است.



شکل ۲-۱- a) ترانسدیوسر هوشمند b) ترانسدیوسر هوشمند IEEE1451

مطابق شکل، این مدل دارای چهار بخش است

۱. بخش ترانسدیوسر
۲. بخش مناسب سازی سیگنال و تبدیل داده آنالوگ به دیجیتال و یا برعکس
۳. بخش پردازش

DOI: 10.30503/nripress.2020.296

۴. بخش ارتباط با شبکه

سیگنال آنالوگ خروجی از یک سنسور پس از تقویت و مناسب‌سازی به کمک A/D به یک فرمت دیجیتال تبدیل می‌شود. پس از آن سیگنال دیجیتال شده سنسور، براحتی به کمک پردازنده و یک آلگوریتم نرم‌افزاری مورد پردازش قرار می‌گیرد. نتیجه پردازش خروجی و اندازه گیری به کمک رابط شبکه به سیستم مانیتورینگ و یا کنترل مرکزی ارسال می‌گردد. ارتباط با شبکه، به کمک یک زبان ارتباطی (پروتکل ارتباطی) صورت می‌پذیرد. ارتباط دو طرفه بوده و می‌تواند از لایه بالا به محرک نیز دستور ارسال شود.

۲-۱-۲- ترانسدیوسر هوشمند مبتنی بر اساس استاندارد IEEE1451

مطابق استاندارد IEEE1451 ترانسدیسر هوشمند باید دارای تابع‌های لازم در تولید و نمایش صحیح کمیت اندازه گیری و یا کنترل شونده باشد. این توابع یکپارچه سازی ترانسدیوسر را در یک شبکه کاربردی آسان می‌نماید. این بدان مفهوم است که یک ترانسدیوسر هوشمند IEEE1451 قابلیت‌هایی نظیر خود شناسایی ، خود توصیفی، کالیبراسیون و عیب‌یابی اتوماتیک به همراه اطلاعات زمانی و مکانی و پردازشی و اعلام هشدار و فرمت داده استاندارد و پروتکل ارتباطی را با خود به همراه خواهد داشت.

شکل (b)(۱-۲) ساختار ترانسدیوسر هوشمند را بر اساس IEEE1451 نشان می‌دهد.

تفاوت این ساختار با شکل (a) در اضافه شدن برگه اطلاعات الکترونیکی[۱] TEDS و مجزا سازی سیستم به دو بخش اصلی شامل قسمت پردازشی با قابلیت شبکه NCAP و بخش اینترفیس ترانسدیوسر[۲] TI و بخش TII[3] بوده که ارتباط بین دو بخش فوق را مستقل از TI شامل می‌شود.

بخش پردازشی و ارتباط با شبکه[٤] NCAP شامل یک نود شبکه بوده که کارهای پردازشی و توابع ارتباطی شبکه را انجام میدهد در حالی که بخش TIM مناسب‌سازی و تبدیل داده را برای مجموعه‌ای از سنسورها و محرکها تا تعداد ۲۵۶ را پشتیبانی می‌کند. این قابلیت برای محیط‌هایی که با آرایه‌ای از سنسورها مانند MEMS و یا مجموعه زیادی از سنسور و محرک مواجهه هستند مناسب است.

TII که بخش ارتباطی مستقل از ترانسدیوسر می‌باشد شامل پروتکل‌ها و محیط‌های مختلف ارتباطی برای ارتباط TIM با NCAP است. این اینترفیس شامل مجموعه‌ای از عملیات همانند خواندن و نوشتن پیامها و یا پاسخها و غیره می‌باشد.

اینترفیس با شبکه NI ، پروتکل‌های ارتباطی NCAP را با شبکه بالاسری فراهم می‌سازد.

علاوه بر بخش‌های ارتباطی، این استاندارد شامل بخش‌های دیگر مهمی همچون بخش استانداردسازی TEDS و فرمت آن است. TEDS همانند یک کارت شناسایی برای یک ترانسدیوسر عمل می‌کند. بطوریکه اطلاعات شناسایی سازنده و رنج اندازه‌گیری ،دقت و داده‌های کالیبراسیون را همانند آنچه که در دیتاشیت سازنده معرفی می‌شود به همراه

---

[۱] Transducer Electronic Data Sheet

[۲] Transducer Interface

[3] Transducer Indepenent Interface

[٤] Network Capable Application Processor

DOI: 10.30503/nripress.2020.296

خود داراست. این اطلاعات در حافظه الکترونیکی ترانسیوسر ذخیره می‌شوند بخشی از این اطلاعات قابل تغییر بوده و بخش دیگر آن تنها قابل خواندن است.

در IEEE1451.0 چهار نوع TEDS اجباری بوده و بقیه موارد اختیاری است. این موارد اجباری عبارتند از:

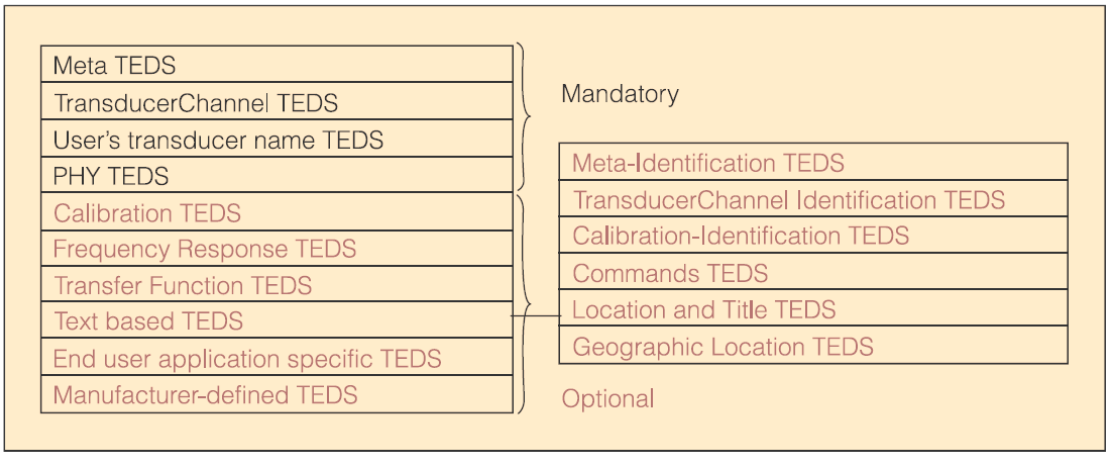
فوق داده Meta Teds

اطلاعات مربوط به کانال

اطلاعات پارامتر فیزیکی

نام داده‌های TEDS کاربری

اما موارد اختیاری شامل داده‌های کالیبراسیون، داده‌های پاسخ فرکانسی، داده‌های تابع تبدیلی، داده‌های متنی کاربری و داده‌های سازنده است شکل(۲-۲).

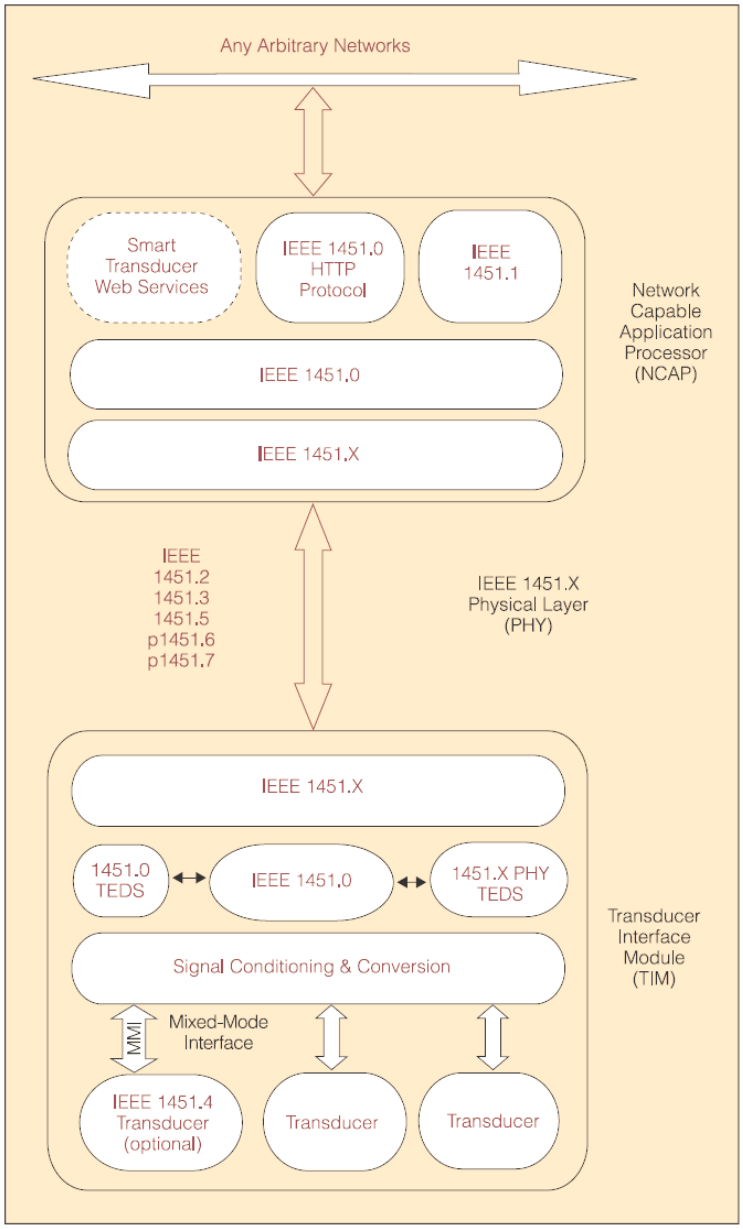


شکل ۲-۲- اطلاعات داده TEDS

۳-۱-۲- خانواده استاندارد IEEE1451

ساختار این استاندارد یک خانواده از استانداردهای ارتباط ترانسدیوسری را که باز هستند بوده و بعلاوه مشترک و اینترفیس‌های مستقل از شبکه را برای ارتباط ترانسدیسر با میکروپروسسور و سیستم اندازه گیری و شبکه کنترلی را تعریف می‌نماید. این استاندارد یک مجموعه از پروتکل‌ها را برای کاربردهای کنترلی و مانیتور کردن بصورت سیمی و یا بدون سیم معرفی می‌نماید. شکل (۲-۳) ساختار این استاندارد را نشان می‌دهد.

DOI: 10.30503/nripress.2020.296



شکل ۲-۳- خانواده استاندارد IEEE1451

بخش IEEE1451.0 مجموعه‌ای از دستورات مشترک را برای دسترسی سنسورها و محرکها را که در ساختارهای مختلف فیزیکی همانند نقطه به نقطه و ساختارهای بیسیمی و یا چنداتصالی Multi drop را برای انجام نیازهای مختلف کاربردی تعریف می‌نماید.

سه راه برای دسترسی به سنسورها و محرکهای داخل TIM از طریق شبکه است که عبارتند از:

- IEEE1451.1
- IEEE1451.0 و پروتکل فوق متنی HTTP
- ترانسدیوسر هوشمند تحت سرویس وب

لایه فیزیکی ارتباط بین NCAP و TIM شامل:

- ارتباط نقطه به نقطه مطابق IEEE1451.2
- ارتباط توزیع شده چند اتصالی مطابق IEEE1451.3

DOI: 10.30503/nripress.2020.296

- ارتباطات بی سیم مطابق با IEEE1451.5 شامل (وای فای و بلوتوث و زیگبی)
- ارتباط با باس CanOpen مطابق با IEEE1451.6
- شناسایی به روش فرکانس رادیویی یا RFID مطابق با IEEE1451.7

بصورت کلی ارتباط بین ترانسدیوسر و بخش مناسب‌سازی و یا تبدیل داده در این استاندارد مطرح نمی‌شود مگر اینکه در بخش IEEE1451.4 که مربوط به سنسورهای آنالوگ بوده و اینترفیس مد مختلط و سطح پایین برای ترانسدیوسر مطرح می‌گردد. این بخش از استاندارد دیتاشیت الکترونیک TEDS را برای ترانسدیوسرهای آنالوگ معرفی می‌نماید. بدین ترتیب امکان هوشمندسازی برای ترانسدیوسرهای آنالوگ نیز فراهم می‌شود.

در پروژه حاضر به کمک بخش چهارم یعنی IEEE14151.4 سعی در هوشمندسازی ترانسدیوسر پیزوالکتریک طراحی شده در پژوهشگاه نیرو هستیم. در آخرین پروژه انجام یافته بر روی این ترانسدیوسر تمامی استانداردهای صنعتی را برای این محصول اخذ شده و در این پروژه به دنبال افزودن یک بخش دیجیتال به تراسدیوسر به منظور پیاده سازی دیتاشیت الکترونیک در داخل این مبدل هستیم. در حافظه بخش دیجتیال مبدل مطابق با استاندارد IEEE1451.4 اطلاعات لازم پایه‌ای و یا اختیاری ثبت و ذخیره می‌شوند تا بدین ترتیب قابلیت‌های نظیر خودشناسایی[1] برای مبدل فراهم شده و مبدل بصورت اتوماتیک می‌تواند مجموعه‌ای از اطلاعات شامل رنج داده ورودی ، واحد سیگنال ، شماره مشخصه سازنده مبدل ، شماره سریال مبدل ، اطلاعات کالیبراسیون یا پاسخ فرکانسی و... در اختیار بخش Ncap قرار دهد تا همانطور که این استاندارد انتظار دارد اطلاعات ورودی و یا خروجی بصورت دقیق و درست و با دقت مناسب برای کاربردهای مختلف استفاده شده و یا به نمایش گذاشته شوند.

مطابق با بخش چهارم این استاندارد یعنی IEEE1451.4، شبکه ارتباطی ترانسدیوسرها از نوع سیمی و منطبق با استاندارد باس 1_wire است. بدین خاطر در فصل قبل به معرفی این باس پرداختیم تا مطالعه و بررسی این استاندارد در بخش‌های بعدی راحتتر شده و قابل فهمتر شود.

۲-۲- ترانسدیوسر استاندارد IEEE1451.4

این استاندارد برای اینترفیس یک ترانسدیوسر هوشمند برای سنسورها و ترانسدیوسرها و پرتکلهای ارتباطی ترکیبی و برای فرمت‌های دیتاشیت الکترونیکی است. مرجع اصلی و عمده این بخش از گزارش تا انتهای فصل همانا متن استاندارد IEEE1451.4 است که در مرجع [5] اشاره شده است.

۱-۲-۲- پایه و اساس

استاندارد IEEE1451.4-2004 موارد TEDS[2] و MMI[3] را تعریف میکند که در شکل (۲-۴) به آن اشاره شده است.

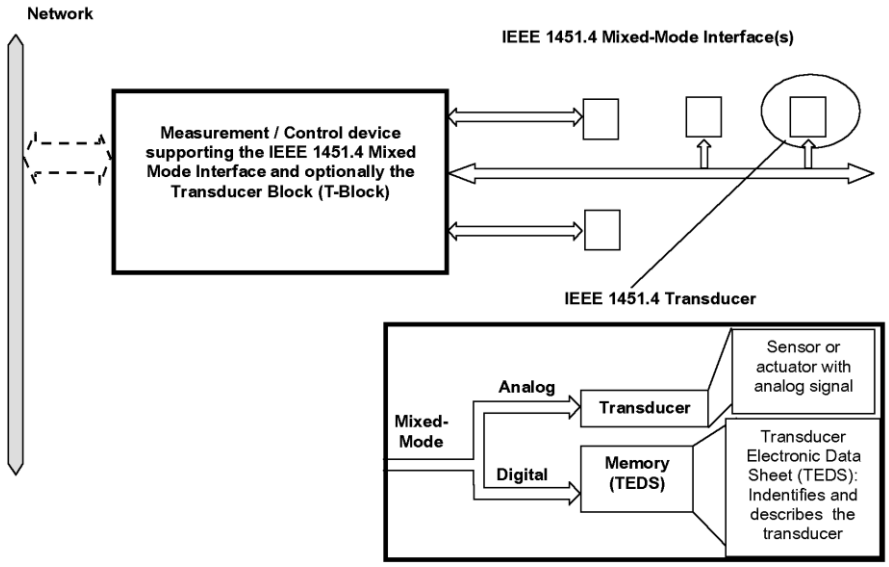
---

[1] Plug&Play

[2] Transducer Electronic Datasheet

[3] Mixed Mode Interface

DOI: 10.30503/nripress.2020.296



شکل ۲-٤- ساختار اینترفیس مد مختلط و ترانسدیوسر

ابزارهای کنترل و اندازه‌گیری باید شامل یک NCAP[1] مطابق با استاندارد ieee1451.1 باشند و به شکل مستقیم امکان دسترسی به شبکه را برای ترانسدیوسر فراهم کنند.

یک ترانسدیوسر منطبق بر IEEE1451.4 ممکن است به عنوان حسگر یا کنترل ویژگی‌های فیزیکی متعددی مورد استفاده قرار بگیرد. هر ویژگی که حس و کنترل می‌شود باید با یک نود[2] مرتبط باشد. اگر یک ترانسدیوسر IEEE1451 شامل بیش از یک نود باشد، یکی از نودها باید شامل یک بلوک حافظه برای نگه داشتن لیست نودها باشد. لیست نودها شامل شماره شناسایی نودهای دیگر است. اگر در ترانسدیوسر مذکور فقط یک نود وجود داشته باشد، دیگر لیست نود وجود ندارد. هر ترانسدیوسر IEEE 1451.4 به یک MMI با استاندارد IEEE1451.4 وصل می‌شود. هر MMI مذکور میتواند به چندین ترانسدیوسر IEEE1451.4 که قابلیت رفتن به مد غیرفعال را دارند وصل شده باشد.

برای برقراری ارتباط با نودهای داخل یک ترانسدیوسر با سیمهای مشترک، ترانسدیوسر IEEE1451.4 به مد دیجیتال سوییچ می‌کند. در کلاس ۱ این استاندارد این امر با معکوس کردن پلاریته منبع تغذیه بدست می‌آید.

مادامیکه در مد دیجیتال هستیم، خطوط مشترک برای ارتباط دیجیتال استفاده می‌شوند (غیر فعال بودن بخش آنالوگ) . در کلاس ۲، که خطوط مجزا برای ارتباط IEEE1451.4 استفاده می‌شوند. هر دو ارتباط می‌تواند به شکل همزمان استفاده شود.

چندین ترانسدیوسر IEEE1451.4 با همراه سوییچ روی نود ممکن است در یک پیکربندی چنداتصالی[3] با بیشینه یک ترانسدیوسر در مد فعال (فعال بودن بخش آنالوگ) و بقیه در مد غیرفعال متصل شوند.

۲-۲-۲- پیکربندی ترانسدیوسر بر پایه IEEE1451.4

[1] Network Capable Application Processor

[2] Node

[3] Multidrop

DOI: 10.30503/nripress.2020.296

TEDS باید شامل اطلاعاتی درباره پیکربندی نود در ترانسدیوسرIEEE1451.4 و امکان استفاده در یک پیکربندی چند اتصالی باشد.

۲-۲-۲-۱- چندین ترانسدیوسر روی یک MMI مبتنی بر IEEE 1451.4

یک اینترفیس IEEE1451.4 فقط می‌تواند یک سیگنال آنالوگ را پشتیبانی کند. اگر بیش از یک ترانسدیوسر از این اینترفیس استفاده کند یک لیست نود برای تشخیص و سویچ آنها لازم است .

۲-۲-۲-۲- چندین نود در یک ترانسدیوسر IEEE1451.4

اگر یک ترانسدیوسر شامل بیش از یک نود باشد، باید یک نود حافظه‌دار با یک قالب وجود داشته باشد که لیست نودهای داخل ترانسدیوسر را نگه دارد. در این لیست نود بجز خودش، شماره شناسه بقیه نودها را در خود دارد بطوریکه تعداد نودهای در لیست برابر تعداد کل نودها منهای یک است.

۲-۲-۲-۳-ارتباط با یک ترنسدیوسر IEEE 1451.4

پیکربندی ترانسدیوسر IEEE1451.4 با نودهای تخصیص داده شده به آن انجام می‌شود. زبان توصیفی[۱] اطلاعات لازم برای تغییر پیکربندی یک ترانسدیوسر را در خود دارد.
تغییر پیکربندی یک تعداد از ویژگی‌ها را مجاز می‌داند:
تست توسط خود سیستم۲
ریست کردن ترانسدیوسر
مالتی پلکس بین ترانسدیوسرها در یک ترانسدیوسر IEEE1451.4
شیفت گین در یک ترانسدیوسر IEEE1451.4
تنظیم آفست
تغییر تنظیمهای فیلتر در یک ترانسدیوسر IEEE1451.4

۲-۲-۲-٤- ارتباط با چندین ترانسدیوسر IEEE1451.4 روی یک MMI مبتنی بر IEEE1451.4

[۱] Template Description Language

[۲] Self Test

DOI: 10.30503/nripress.2020.296

فعال کردن یا از کار انداختن تابع آنالوگ ترانسدیوسرهای مختلف IEEE1451.4 اجازه می‌دهد تا چندین ترانسدیوسر به یک اینترفیس IEEE1451.4 متصل شوند. لیست نود، امکان مالتی پلکس کردن نودهای هر ترانسدیوسر IEEE1451.4 را فراهم می‌سازد.

۳-۲- دیتاشیت الکترونیکی ترانسدیوسر

TEDS از داده مرتبط با یک شیء یا ترانسدیوسر IEEE1451.4 تشکیل یافته و در حافظه ذخیره شده است. فرمت‌های ارسال و انبارش آنها در استاندارد IEEE 1451.4 وجود دارد.

۱-۳-۲- TEDS پایه

به عنوان یک حداقل، TEDS در یک ترانسدیوسر IEEE1451.4 باید شامل یک TEDS پایه[۱] باشد که به شکل منحصر به فردی ترانسدیوسر را مشخص می‌کند.

۲-۳-۱-۱- محتویات TEDS پایه

TEDS پایه ۶۴ بیت است که شامل شناسه کارخانه (۱۴ بیت) ، شماره مدل (۱۵ بیت) ، حرف نسخه (کد کاراکتر ۵ بیتی) ، عدد نسخه (۶ بیت) ، و شماره سریال وسیله (۲۴ بیت) میشود. این داده‌ها باید بر اساس فرمتی که در جدول(۲-۱) شرح داده شده است، ساماندهی می‌شود. TEDS پایه باید در یک حافظه غیرفرار[۲] قرار داده شود. از بایت checksum، که تضمین کننده صحت داده TEDS پایه است استفاده می‌شود.
فقط ۶۴ بیت، یکتایی (منحصر به فرد بودن) را تعیین می‌کند. و الا شماره مدلها و شماره سریال‌ها برای سازنده‌های مختلف می‌توانند یکی باشند، و یک سازنده میتواند شماره سریال یکسانی را برای شماره مدلهای مختلف دارا باشد. جدول (۲-۲) نقشه حافظه TEDS پایه را بصورت جزئی نشان میدهد.

جدول ۲-۱- نقشه محتویات حافظه TEDS پایه

Manufacturer ID	Model number	Version letter	Version number	Serial number
14 bits (17–16381) [a]	15 bits (0–32767)	5 bits Char (A–Z, data type Chr5)	6 bits (0–63)	24 bits (0–16777215)

خواندن و نوشتن محتویات TEDS پایه و مابقی داده‌های TEDS همواره بر اساس ابتدا کم ارزشترین بیت[۳] است.

---

[۱] Basic TEDS

[۲] non volatile

[۳] LSB first

DOI: 10.30503/nripress.2020.296

۲-۳-۱-۲- جاگذاری و فرمت TEDS پایه

در حافظه‌های تجاری موجود، یک سری رجیسترهای کاربردی (برای نمونه به EEPROM از نوع 1-wire با شماره DS2430A مراجعه شود[3]) وجود دارد که یک بار قابل برنامه ریزی (OTP[1]) هستند. TEDS پایه باید در رجیسترهای کاربردی قرار گرفته تا از دستکاری یا خرابی تصادفی داده در امان باشد. فرمت اطلاعات TEDS پایه باید مطابق جدول (۲-۲) باشد. (به شماره‌ها در پرانتزها بعد از هر آیتم دقت شود به عنوان نمونه شناسه سازنده (13:0) نشان دهنده یک شمارنده ۱۴ بیتی از صفر است). در هر یک از نودها، محتویات رجیستر وضعیت[۲] نشان دهنده وضعیت قفل بودن یا نبودن رجیسترهای کاربردی را نشان می‌دهند.

۲-۳-۲- IEEE ، کاربر و TEDS سازنده

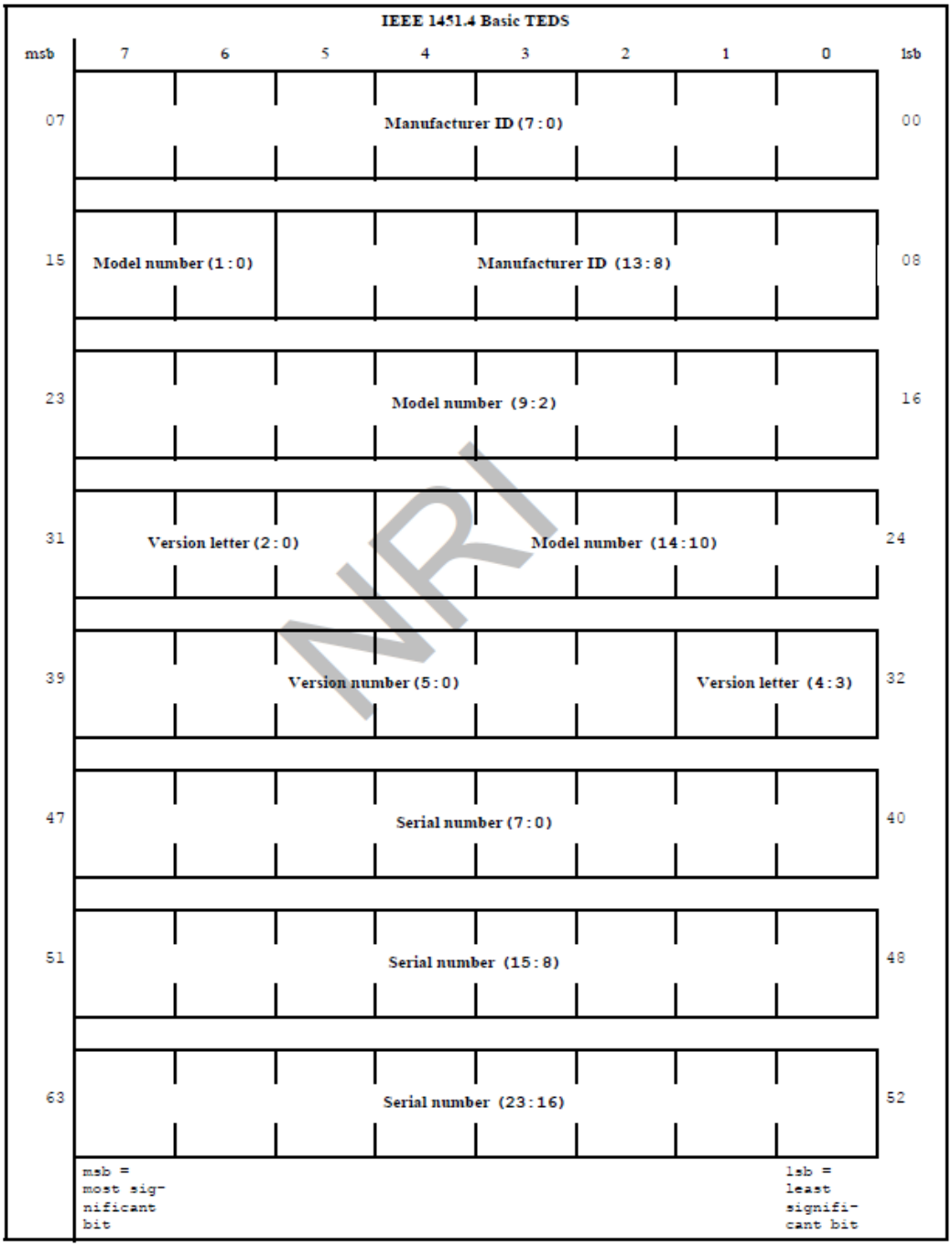
داده TEDS باید در حافظه غیر فرار باشد. اگر TEDS بیشتر از یک نود حافظه را اشغال کند، باید یک لیست نود را در یکی از نودها مطابق با فرمت مشخصی که بعدا به آن (برای مشخص کردن اندازه دقیق داده TEDS) اشاره خواهد شد، داشته باشد. TEDS باید شامل داده تعریف شده توسط الگوی IEEE باشد که در پیوست استاندارد آمده، یا یک الگوی منتشر شده توسط سازنده وسیله شامل TEDS، یا یک الگویی که بوسیله کاربر تعیین می‌شود باشد.

جدول۲-۲- نقشه جزئیات حافظه TEDS پایه IEEE1451.4

[1] One Time Programmable

[۲] Status Register

DOI: 10.30503/nripress.2020.296



۳-۳-۲- فرمت داده و الگوها

فرمت‌های داده TEDS مبتنی بر IEEE1451.4 باید براساس الگویی که سپس‌تر شرح داده خواهد شد، تعریف می‌شوند. یک الگو یک تعریف مستند جاگذاری و اهمیت هر تکه‌ای از داده ذخیره شده در حافظه TEDS است. الگو نباید بهمراه داده‌های TEDS باشد، اما داده‌های TEDS مشخص می‌کند کدام الگو برای تفسیر داده‌های TEDS ارجاع داده می‌شود. الگوها باید برای کد برنامه که برای نوشتن و خواندن داده‌های TEDS استفاده می‌شوند، قابل دسترس باشند.

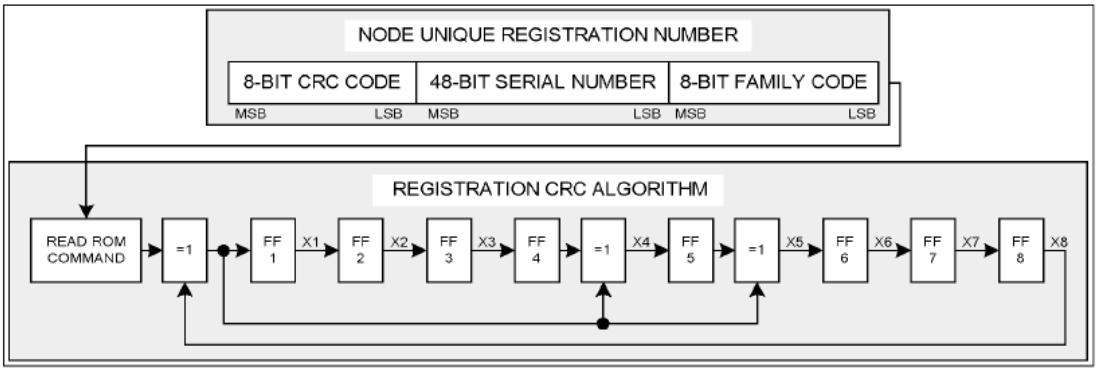
DOI: 10.30503/nripress.2020.296

TDL[1] یا زبان توصیفی الگو، زبانی برای نوشتن الگوها است. تعریف کد (T-block) لازم برای تفسیر داده‌های TEDS، در بین استفاده از الگو در بخش‌های بعدی خواهد آمد.

۴-۳-۲- نودها، آدرس‌ها، کدهای خانوادگی[۲]، URN[3]، و CRC[4]

داده‌های ذخیره شده TEDS در یک یا چند حافظه قابل آدرس دهی قرار داشته که نود نامیده می شوند و به شکل مستقیم برای پردازنده یا کنترلر در دسترس نمی باشند. هر پردازنده یا کنترلر یا هر وسیله‌ای که شامل داده‌های TEDS در حافظه خود است و میتواند این داده‌ها را به پردازنده، کنترلر و یا برنامه‌ای انتقال دهد باید به عنوان نود در نظر گرفته شود. نودها ممکن است همچنین دارای توابع دیگری به جز حافظه باشد.
هر نود شامل یک شماره ثبت یکتا (URN) در حافظه غیر فرار بوده که برای کنترل دسترسی حافظه TEDS استفاده می‌شود.
URN پایه و اساس ارتباط دیجیتال نودهای قابل آدرس‌دهی است که در یک معماری چند اتصالی رخ می‌دهد.

بخشی از URN، یک کد خانوادگی است که برای شناسایی تابع‌ال آن وسیله و دستورهای ارتباطی مشخص لازم برای آن نود استفاده می‌شود. کدهای خانواده و فایل دستورات توصیفی برای نود وسایل، به شکل تجاری موجود است. در پیوست E استاندارد این کدهای خانوادگی آمده است.
ارسال URN نیز مانند دیگر اطلاعات با بیت ارزش کم شروع می‌شود. اولین ۸ بیت از ۶۴ بیت URN همان کد خانوادگی و سپس ۴۸ بیت سریال آن به همراه یک کد تایید ۸ بیتی CRC است که به منظور صحه‌گذاری اطلاعات دریافتی بکار می‌رود. این کد بوسیله شیفت رجیستر و تابع XOR همانند شکل (۲-۵) تولید می‌شود. در ابتدا شیفت رجیستر پاک شده و با ارسال اطلاعات کد خانوادگی و ۴۸ بیت سریال یک کد CRC مطابق این روال تولید خواهد شد که اگر با کد CRC دریافتی یکسان باشد نشان از صحت اطلاعات دریافتی است.



شکل ۲-۵- URN و تولید CRC

۵-۳-۲- ارسال داده

[1] Template Description Language
[۲] Family code
[3] Unique Registeration Number
[4] Cyclic Redundancy Check

DOI: 10.30503/nripress.2020.296

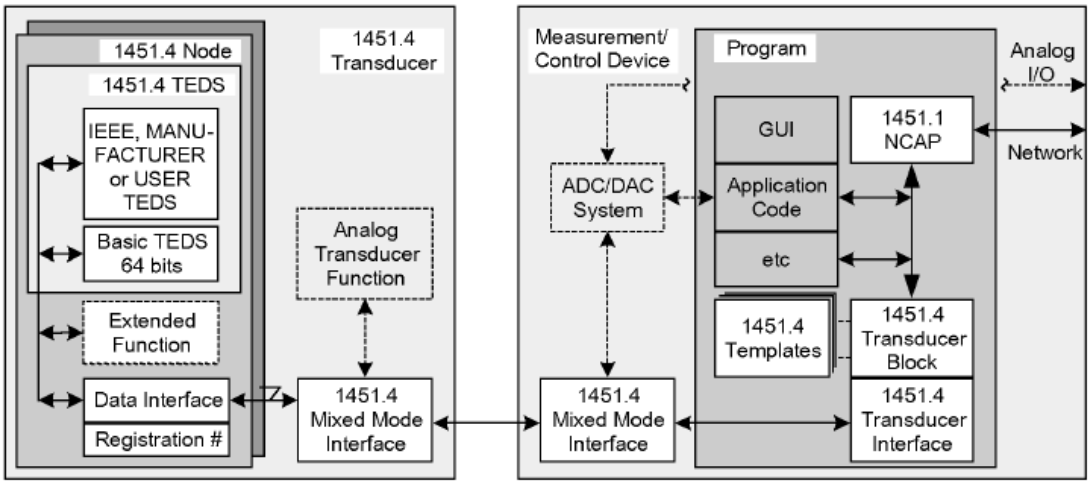
ارسال داده به و از حافظه TEDS ، یا دیگر توابع که درون نودها وجود دارند، باید مطابق پرتکل باس دو سیمه سریال باشد که در بخش‌های بعدی اشاره خواهد شد، این باس از نوع چنداتصالی و قابل آدرس دهی است. شایان ذکر است که این باس، مشابه باس 1-Wire فصل قبل، با اندکی تفاوت می‌باشد.
یک نمونه از نود حافظه EEPROM از نوع 1-wire با ظرفیت ۲۵۶ بیت قطعه DS2430A است[3]. راهنما برای پیاده سازی پرتکل در مستند "1-wire Master Device Configuration" [2] آمده است.

۶-۳-۲- ساختار سیستم داده TEDS

بلوک دیاگرام سیستم داده TEDS در شکل ۲-۶ یک سلسله از توابع لازم برای جادادن داده‌های TEDS در بین حافظه فیزیکی، و مجوز دسترسی داده، از طریق یک پرتکل ارسال داده را نشان می‌دهد. دقت کنید که این شکل همه اجزا یک سیستم کامل پیاده سازی شده TEDS منطبق با IEEE1451.4، که در آن TEDS درون یک نود از ترانسدیوسر قرار گرفته و شامل MMI است را نشان می‌دهد. یک وسیله کنترل/اندازه‌گیری داده‌های TEDS را به کمک اینترفیس ترانسدیوسر 1451.4 و بلوک ترانسدیوسر و یک الگوی کتابخانه‌ای که منطبق بر استاندارد IEEE1451.4 است شناسایی می‌کند.
زیرمجموعه این پیاده‌سازی تحت این استاندارد که به یک پیاده‌سازی کمینه کاهش داده می‌شود، باید به شکلی که TEDS از نوع IEEE1451.4 در بلوک ترانسدیوسر به همراه یک تک الگو، که موضوع یک ابزار دقیقی که شامل یک ترانسدیوسر مجتمع است شناخته شود.
همه بلوک‌های اشاره شده در شکل(۲-۶) در این استاندارد نمی‌باشند. در زیر خلاصه‌ای از این موارد اشاره شده است.
در یک ترانسدیوسر IEEE 1451.4، یک ترانسدیوسر آنالوگ اینترفیس MMI را به اشتراک می‌گذارد.
داده‌های TEDS از طریق اینترفیس داده از نود 1451.4 وارد یا خارج می‌شوند این ارتباط از نوع 1-Wire بوده و هر تراکنش بوسیله URN مربوط به آن نود صورت می‌گیرد.
یک نود ممکن است شامل توابع دیگری همچون سویچ، شمارنده، اندازه‌گیری دما و غیره باشد که بوسیله توابع گسترش‌یافته[1] ارایه می‌شوند، دسترسی به این موارد به کمک اطلاعات ساختاری وسیله که شامل کد خانوادگی آن است صورت می‌پذیرد.
در داخل وسایل اندازه گیری/ کنترل، داده‌های آنالوگ از بلوک برنامه کاربردی از طریق یک سیستم آنالوگ وارد یا خارج می‌شوند، که MMI از نوع 1451.4 را به اشتراک می‌گذارند. همه توابع IEEE1451.4 بخشی از کد برنامه کلی است، که ممکن است در بردارنده کد برنامه کاربردی برای تابع ویژه ابزار دقیق، یک واسط کاربر گرافیکی (GUI) و نمایش، اینترفیس شبکه، و غیره باشد. اینترفیس ترانسدیوسر 1451.4 شامل یک ساختار فرمان سطح پایین لازم برای انجام پرتکل ارتباطی که بوسیله نودهای TEDS نیاز است، می‌باشد. نمونه‌های راهنما برای پیاده سازی این پرتکل در دیتاشیت DSA2430A 256-bit 1-wire EEPROM[3] و "1-wire Master Device Configuration" [2]یافت می‌شوند.

[1] Extended Function

DOI: 10.30503/nripress.2020.296



شکل ۲-٦- بلوک دیاگرام سیستم داده TEDS با IEEE 1451.4

٤-۲- الگوها[۱]

این بخش استفاده از ساختار الگو برای خواندن یا نوشتن نودها و استخراج داده‌های TEDS را شرح می‌دهد.
یک وسیله اندازه‌گیری/کنترل باید به یک باس وصل شود که یک یا چندین ترانسدیوسر IEEE1451.4 به آن وصل شده‌اند. هر ترانسدیوسر IEEE1451.4 ممکن است یک یا چند نود در داخل خود داشته باشند. هر نود از ترانسدیوسر IEEE1451.4 با وسیله اندازه‌گیری/کنترل از طریق اینترفیس MMI قابل دسترس است.

۱-۴-۲- مرورکلی

یک الگو ساختار حافظه داده TEDS را شرح می‌دهد که شامل اطلاعات درباره شناسه ترانسدیوسر و داده‌های ویژه ترانسدیوسر است. این ساختار به زبان TDL نوشته شده و در T-block جای داده می‌شوند. الگوهای لازم برای T-block بصورت یک یا چند فایل توصیفی متنی ۸ بیتی ASCII با پسوند tdl. معرفی می‌شوند. T-block استاندارد IEEE1451.4 ، یک اینترفیس بین ترانسدیوسرها و نرم‌افزار کاربردی در یک وسیله منطبق بر IEEE1451 را فراهم می‌سازد و شامل روشهایی برای استخراج و رمزگذاری داده TEDS منطبق با IEEE1451.4 است که از الگوها استفاده می‌کند. T-block همچنین به ترانسدیوسرهای IEEE 1451.4 و اینترفیسها اجازه می‌دهدکه پیکربندی و مدیریت شوند.
ساختار الگو با هدف اصلی برای استفاده از حافظه‌های کم حجم به یک سبک موثر طراحی شده است. بنابراین، الگوها ساختار داده حافظه را معرفی می‌کند. این بدان معنی است که یک بخش ویژه حافظه ممکن است شامل انواع مختلف اطلاعات وابسته به الگوی استفاده شده و در یک ترانسدیوسر خاص باشد.

---

[۱] Templates

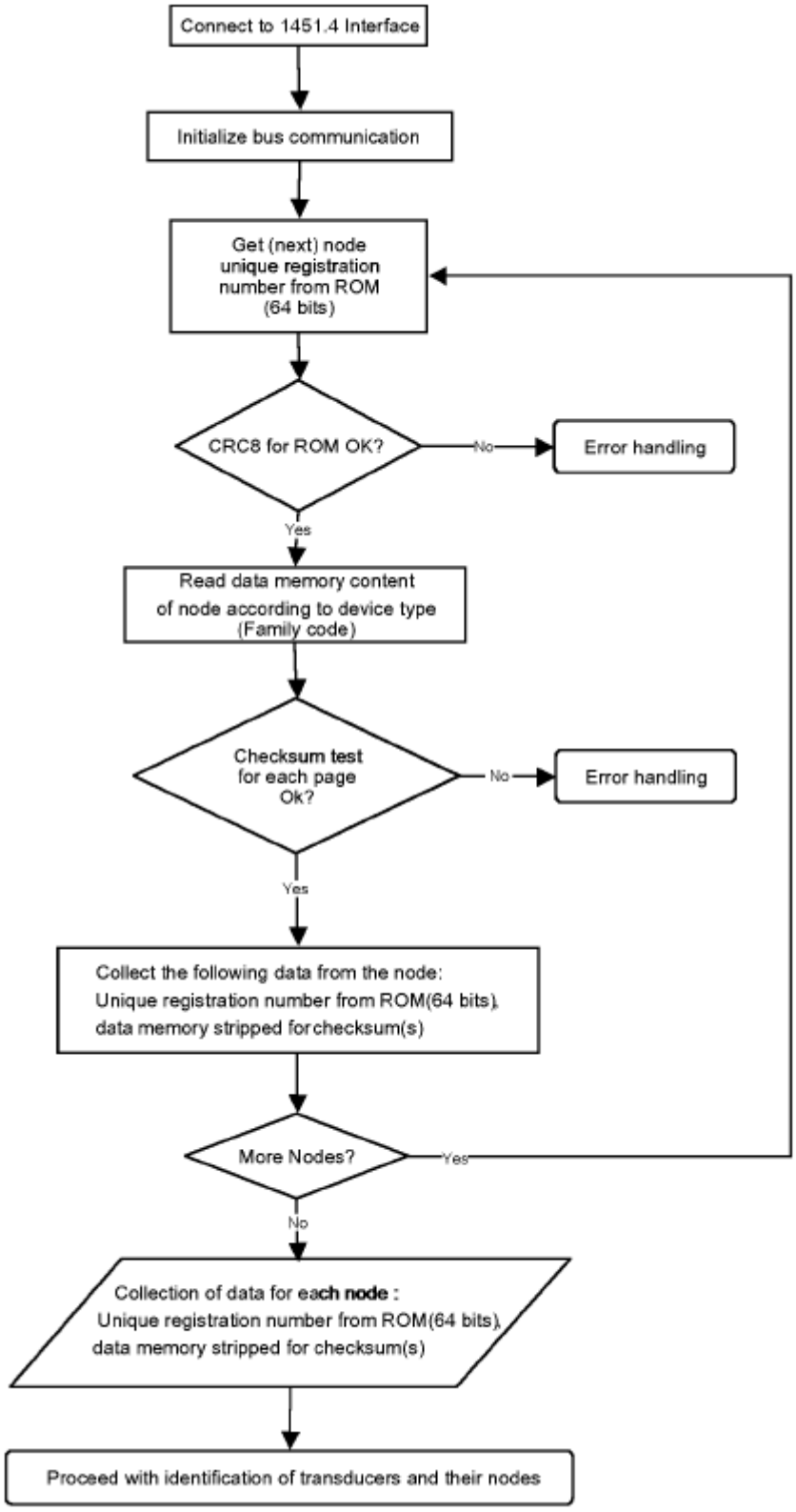
DOI: 10.30503/nripress.2020.296

TEDS پایه، شناسه هر ترانسدیوسر را معرفی کرده و در همه آنها مشترک است. یک الگو ساختار داده‌های بکار رفته در حافظه یک ترانسدیوسر را معرفی می‌نماید. الگوها (در قالب فایل توصیفی الگو) برای انواع گوناگون ترانسدیوسرها تعریف شده و امکان توسیع در آینده را دارا هستند (به پیوست A استاندارد رجوع شود).
ویژیگی‌های ساختار الگو در فلوچارت‌های بعدی شرح داده شده است.

۲-۴-۲- کشف وجود ترانسدیوسرها

اولین کار برای اینکه محتوای TEDS را روی یک اینترفیس بخوانیم سویچ کردن به مد دیجیتال و آماده سازی باس ارتباطی است همانگونه که در فلوچارت شکل ۲-۷ نشان داده شده است.
سیستم سپس باید همه URNها را در نودهای حاضر روی اینترفیس بخواند. اولین ۸ بیت (خواندن و ارسال از lsb به msb است) از شماره شناسایی، همانا کد خانوادگی است که اطلاعاتی درباره اندازه و ساختار حافظه فیزیکی را می-دهد. ۴۸ بیت بعدی باید شامل یک شماره سریال منحصر به فرد بوده و ۸ بیت آخر شامل یک کد CRC برای اولین ۵۶ بیت ارتباطی (کد خانواده و شماره سریال) است.
پس از آن، حافظه داده ممکن است خوانده شود. ممکن است یک تست checksum باید برای خواندن هر صفحه ساخته شود که پس از پردازش حذف خواهد شد. اگر یک وسیله با وجود شماره شناسه دارای حافظه اضافی نباشد، آن وسیله برای خواندن TEDS در نظر گرفته نمی‌شود.
بعد از فاز فوق، یک مجموعه از داده برای هر نود شامل URN با ۶۴ بیت و حافظه داده که خالی از checksum است کشف خواهد شد.
فلوچارت این کار شکل (۲-۷) نشان داده شده است

DOI: 10.30503/nripress.2020.296



شکل ۲-۷- فلوچارت کشف ترانسدیوسرها

۳-۴-۲- شناسایی ترانسدیوسرها و نودهای آن

DOI: 10.30503/nripress.2020.296

بعد از کشف، اطلاعات حافظه داده به همراه آدرس نود مربوطه به هر ترانسدیوسر در یک مجموعه گردآوری می‌شوند. فلوچارت شناسایی یک ترانسدیوسر در شکل (۲-۸) نمایش داده شده است. این پروسه یک مجموعه از URNهای مربوط هر نود را که در هر ترانسدیوسر وجود دارد ایجاد میکند که شامل حافظه داده مربوطه بوده و خالی از بایت‌های checksum است. این پردازش در سه مرحله صورت می‌گیرد:

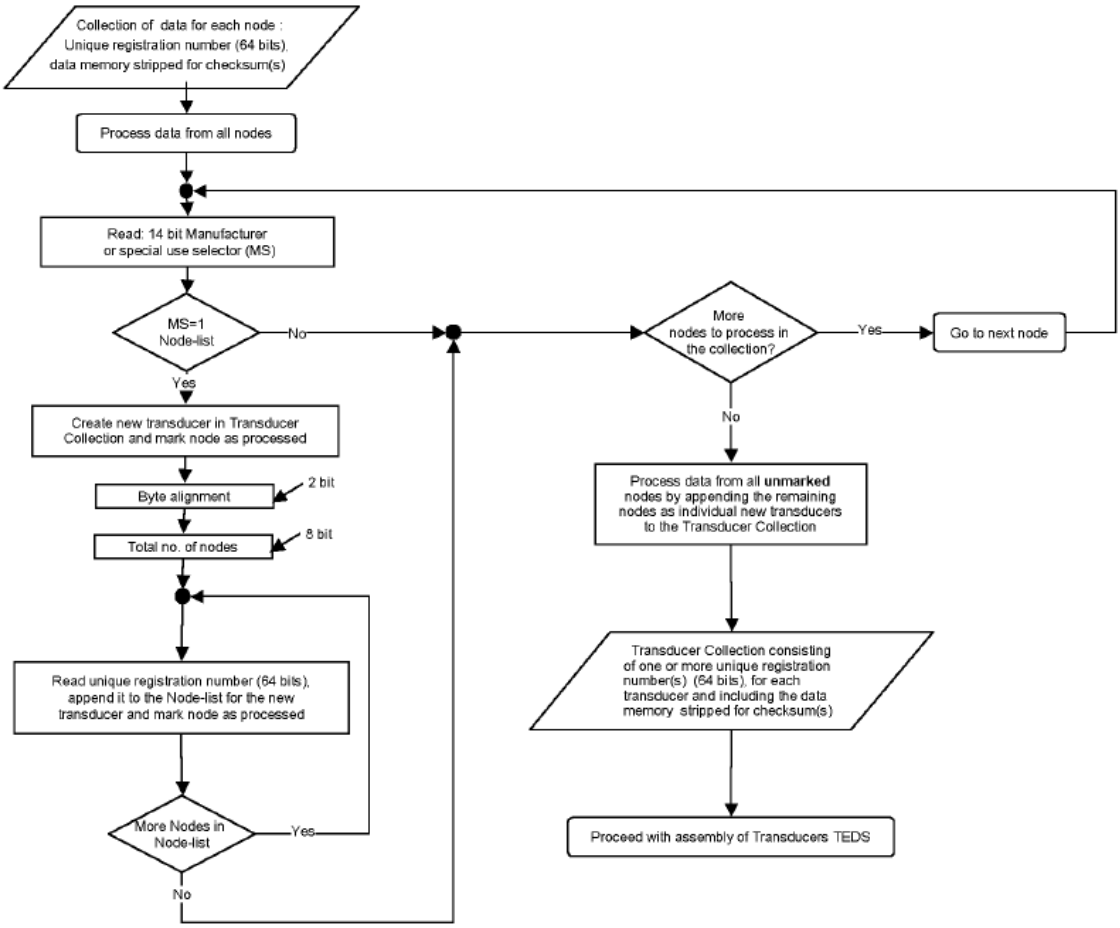
۱-) داده‌ها از هر نود برای آشکار کردن وجود لیست نود، خوانده و بررسی می‌شود. انتخاب کننده ویژه[۱] و یا شماره سازنده در اولین ۱۴ بیت حافظه قرار می‌گیرد. اگر یک نود دارای کد انتخاب کننده یک باشد، این نود شامل لیست نود برای یک ترانسدیوسر IEEE1451 است. اگر یک ترانسدیوسر IEEE1451.4 بیش از یک نود را شامل شود، یکی از نودهای آن باید شامل یک لیست نود باشد. زمانیکه یک لیست نود برای یک ترانسدیوسر مفروض آشکار شد، آن ترانسدیوسر به ساختار داده مجموعه ترانسدیوسرها افزوده می‌شود.

۲-) سپس، همه URNهای (۶۴ بیتی) لیست شده خوانده و به لیست نود برای ترانسدیوسر جدید پیوست می‌شود. و همه این نودها به عنوان نودهای پردازش شده از مجموعه اولیه علامت دار می‌شوند.(به منظور عدم پردازش مجدد)

۳-) هر یک از نودهای باقیمانده (بدون علامت) باید ترانسدیوسرهای اختصاصی باشند، و آنها به همان شکل به مجموعه ترانسدیوسر افزوده می‌شوند. (اگر یک ترانسدیوسر شامل فقط یک نود باشد، یک لیست نود برای این ترانسدیوسر وجود نخواهد داشت، با این وجود URN برای یک نودی که در این ترانسدیوسر وجود دارد به مجموعه ترانسدیوسرها افزوده می‌شود).

[۱] Special use selector

DOI: 10.30503/nripress.2020.296



شکل ۲-۸- فلوچارت شناسایی ترانسدیوسر(ها) و نودهایش

۲-٤-۳-۱- شماره سازنده یا انتخاب کننده ویژه

اولین ۱۴ بیتی که از یک حافظه داده نود خوانده می‌شود، شامل یک انتخاب کننده تعریف شده در جدول (۲-۳) است. بیشتر مقادیر مجاز انتخاب کننده (۱۶۳۸۱-۱۷)، مربوط به شماره سازنده بوده که بوسیله یک لیست در استانداراده IEEE معرفی می‌شوند. این استاندارد توصیف میکند که چگونه TEDS در یک شکل فشرده ذخیره می-شود. حضور یک شماره سازنده از لیست IEEE به شکل غیر مستقیم به یک TEDS فشرده اشاره می‌کند.

یک توصیف شفاهی از معنی دیگر مقادیر این فیلد به شرح زیر است:

- مقادیر 0، 3،5-9 ، 11-13، و 15-16 برای استفاده آینده در نسخه‌های بازنگری شده، در نظر گرفته شده است.
- مقدار 1 نشان دهنده این است که نود شامل لیست نود است.
- مقدار 2 نشان می‌دهد که بخش باقیمانده حافظه شامل TEDS در یک فرمت آزاد است. این مورد نیازمند اطلاعات بیشتری است که در این استاندارد پوشش داده نشده است.
- مقدار 4 نشان می‌دهد که حافظه داده باقیمانده، یک ادامه از حافظه داده و نود قبلی است که باید خوانده و به تصویر TEDS جاری پیوست شود. مقدار انتخاب کننده پیوست نمی‌شود.
- مقادیر 10 و 14 بوسیله نسخه بتای این استاندارد استفاده شده است و نباید بعدا استفاده شود.

DOI: 10.30503/nripress.2020.296

- مقدار 16382 نشان دهنده یک الگوی تعریف شده توسط کاربر است. زمانیکه یک ترانسدیوسر، شامل TEDS پایه از یک سازنده بهمراه اطلاعات ویژه بیشتری که بوسیله کاربر فراهم شده است می‌شود. کاربر باید فایل الگوی توصیفی را محیا کرده و آنرا درکنار فایل توصیفی اصلی ذخیره کند تا سیستم بتواند داده‌های TEDS را خوانده و تفسیر کند.
- آخرین مقدار، 16383 است که نشان می‌دهد، حافظه داده باقیمانده نباید خوانده شود. به عبارت دیگر به انتهای لیست رسیده‌ایم و خواندن باید با نود بعدی که در لیست نود مشخص شده ادامه می‌یابد.

در جدول (۲-۳) مقادیر سازنده و مفاهیم فوق بصورت خلاصه نشان داده شده است

جدول ۲-۳- سازنده یا انتخاب کننده استفاده ویژه

Value	Meaning
0	Future use
1	IEEE 1451.4—Node-List
2	IEEE 1451.4— free-form TEDS only
3	Future use
4	Memory continued
5–9	Future use
10	Reserved by beta format
11–13	Future use
14	Reserved by beta format
15–16	Future use
17–16381	Enumeration of Manufacturers
16382	User defined template
16383	Continue in next node of Node-List

NRI

۲-۴-۴- گردآوری اطلاعات TEDS یک ترانسدیوسر

از مجموعه اطلاعات TESD هر یک از نودهای متصل به یک ترانسدیوسر، اطلاعات TEDS آن ترانسدیوسر ساخته می‌شود. برای به دست آوردن اطلاعات TEDS یک ترانسدیوسر ابتدا اطلاعات TEDS اختصاصی هر نود در یک مجموعه باید گردآوری گردد. فلوچارت تولید این اطلاعات شکل(۲-۹) نشان داده شده است.

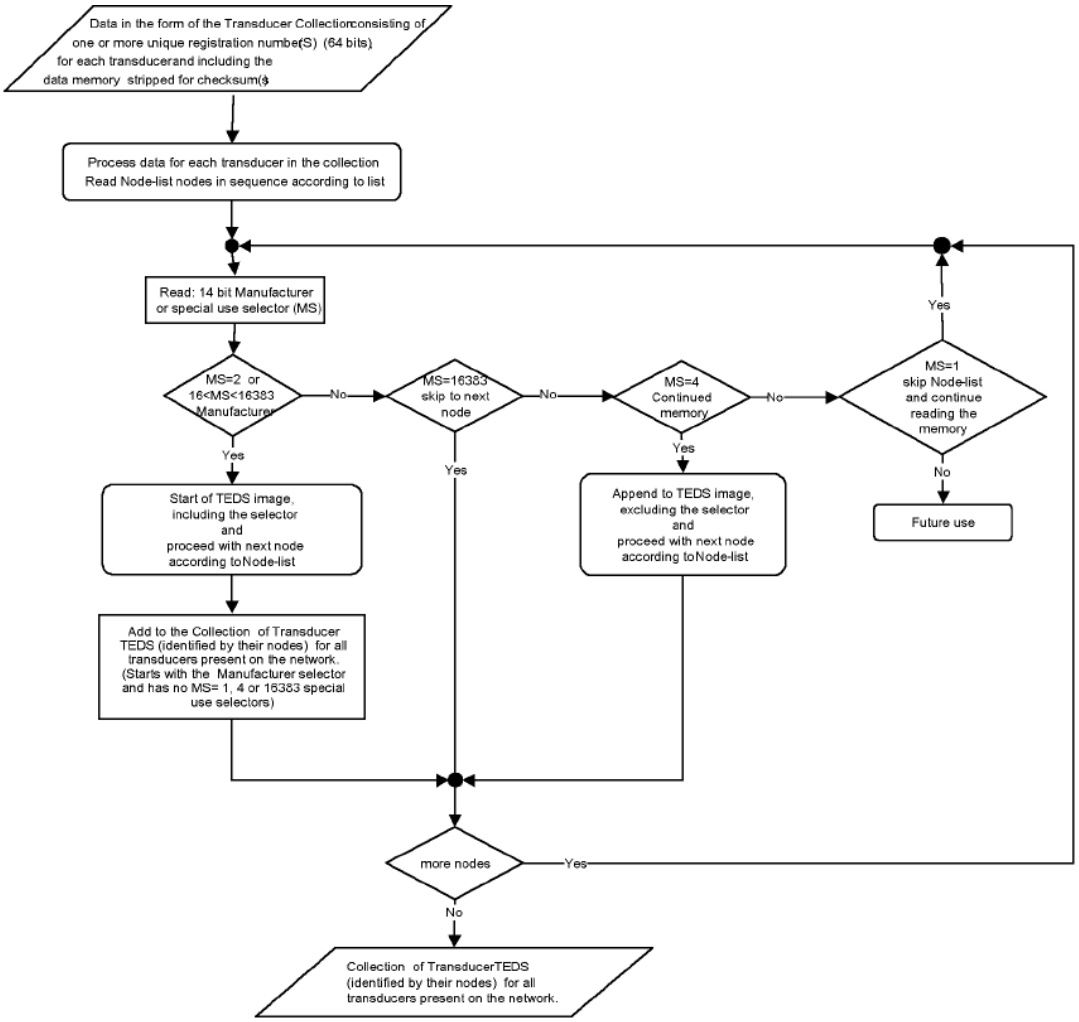
مجموعه نهایی اطلاعات TEDS یک ترانسدیوسر، از اطلاعات TEDS هر یک از نودها و الحاق آنها به همدیگر تولید می‌شود. قوانین الحاق آنها بوسیله شماره سازنده یا انتخاب کننده ویژه که به آنها اشاره شد، بدست می‌آید.

یک نمونه از پروسه استفاده از لیست‌های نود با سه نود حاضر در زیر شرح داده شده است:

- لیست نود را بخوان
- اولین نود در لیست نود را بخوان، که شروع تصویر TEDS، شامل یک شماره شناسه معتبر سازنده است. شمارنده سازنده یا انتخاب کننده ویژه را نگه دارید
- نود دوم را در لیست نود بخوان. سازنده یا انتخاب کننده استفاده ویژه آن برابر 4 است، که به معنی ادامه داشتن حافظه است. حافظه را به تصویر TEDS بدون انتخاب کننده، بیافزایید.

DOI: 10.30503/nripress.2020.296

- محتویات نود بعد از لیست نود، دارای سازنده یا انتخاب کننده استفاده ویژه 16383 است، که به معنی پرش به نود بعدی است، این بدان معنی است که خواندن تمام شده است و نود دیگری وجود ندارد.

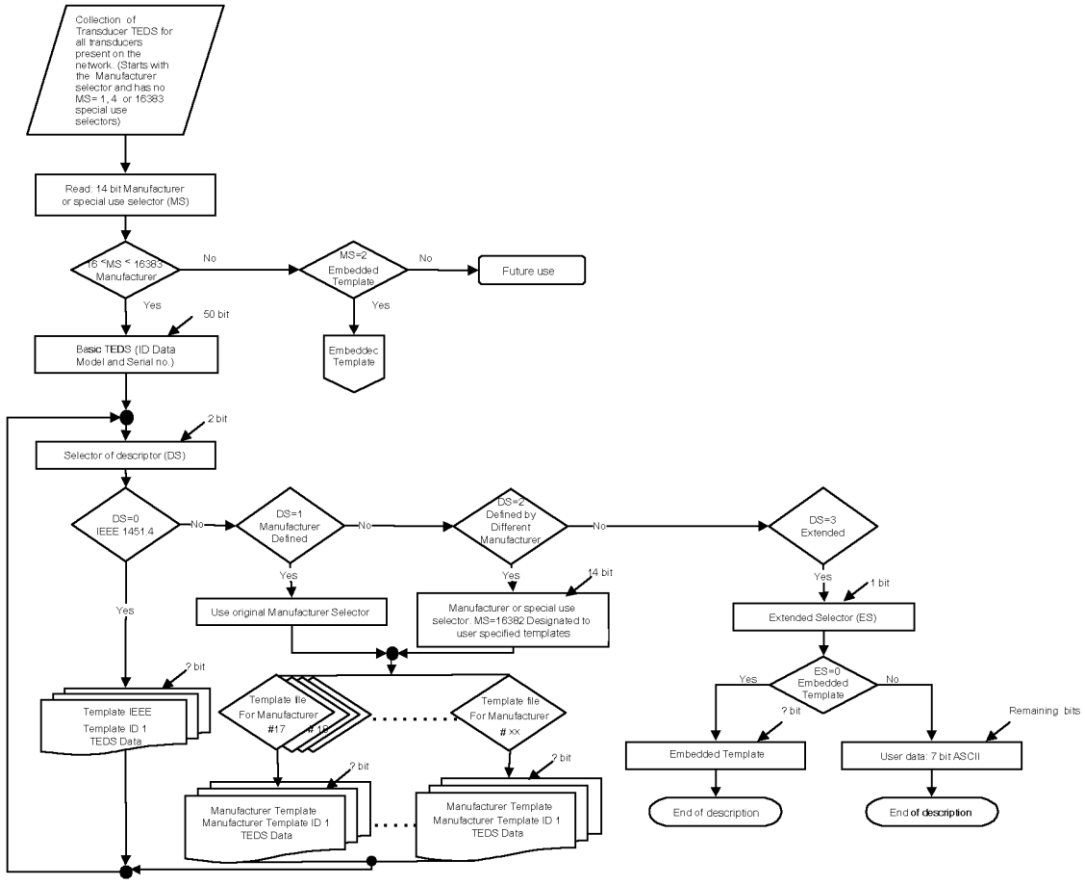


شکل ۲-۹- فلوچارت گردآوری اطلاعات TEDS ترانسدیوسرها

۵-۴-۲- تجزیه کردن اطلاعات TEDS یک ترانسدیوسر

مرحله پایانی در پروسه برای بدست آوردن داده‌های کامل ترانسدیوسر، تجزیه داده‌های TEDS هر یک از ترانسدیوسرهای TEDS است. این مورد در فلوچارت شکل(۲-۱۰) نشان داده شده است.
شماره سازنده یا انتخاب کننده ویژه که پیشتر شرح داده شد، نشان می‌دهد که با یک الگوی embedded (شامل TEDS پایه) یا یک TEDS مربوط به سازنده یا با یک الگوی تعریف شده توسط کاربر سروکار داریم. انتخاب‌کننده و ۵۰ بیت بعدی آن در بخش TEDS پایه تشریح شده است.

DOI: 10.30503/nripress.2020.296



شکل ۲-۱۰- فلوچارتی تفکیک اطلاعات TEDS ترانسدیوسر

یک انتخاب کننده ۲ بیتی شرح دهنده در بخش بعدی معرفی شده است. این انتخاب کننده پس از خواندن الگو، به مجموعه زیرالگوهای صحیحی هدایت می‌شود. مقدار ۳ برای این انتخای کننده نشان دهنده توابع توسعه یافته خواهد بود که ممکن است یک الگوی Embedded (جدا از TEDS پایه) بوده و یا داده کاربر در فرمت ۷ بیتی اسکی (ASCII) باشد.

۲-٤-٥-۱- انتخاب کننده وصفی

انتخاب کننده وصفی، فیلدی است که توصیف می‌کند که یک الگو منطبق بر IEEE1451.4 است یا نه. فیلد مورد نظر یک انتخاب کننده ۲ بیتی با رنج ۰ تا ۳ با مفاهیم زیر می‌باشد:

۱. (صفر)، الگو بوسیله IEEE 1451.4 تعریف شده است.
۲. (یک)، یک ناحیه توصیف شده توسط سازنده دنبال می‌شود. شناسه سازنده بوسیله اولین انتخاب کننده سازنده داده می‌شود برای انتخاب گروه زیرالگوها در بین آن اعدادی که از TEDS خوانده می‌شود، باید در فایل توصیفی که توسط سازنده ایجاد شده، پیدا شود.
۳. (دو)، یک سازنده دیگر یا کاربر الگو را تعریف کرده است. یک سازنده جدید یا انتخاب کننده استفاده ویژه باید دنبال شود. اگر انتخاب کننده ۱۶۳۸۲ است، یک کاربر زیر الگوها (یک کد تایید برای فایل توصیف شده باید در TEDS تعریف شده توسط کاربر در نظر گرفته شود) را تعریف کرده است.

DOI: 10.30503/nripress.2020.296

۴. (سه)، گسترش یافته. یک بیت انتخاب کننده توسعه یافته باید دنبال شود. اگر انتخاب کننده توسعه یافته، 0 است، یک شکل آزاد از TEDS دنبال می‌شود. الگویی که برای خواندن این شکل آزاد از TEDS استفاده می‌شود، باید در فایل توصیفی ایجاد شده بوسیله کاربر یا سازنده این شکل آزاد TEDS پیدا شود.

۲-٤-۵-۲- شناسه الگو

شناسه الگو در ابتدای هر الگو، الگو را شناسایی خواهد کرد. شمار بیت‌های استفاده شده برای شناسایی الگو باید برای توصیف کننده الگو یکتا باشد. توصیف کننده الگو می‌تواند یکی از IEEE، سازنده، یا کاربر باشد.

۵-۲- زبان توصیف الگو[1] TDL

این بخش دستور و مفاهیم زبان توصیفی که در الگوها استفاده می‌شوند، را شرح می‌دهد. با استفاده از این زبان، نقشه بیتی اطلاعات TEDS هر الگو قابل شناسایی است.

۱-۵-۲- کلیات

نکات گرامری رسمی که دستور دقیق TDL را فراهم می‌کند در پیوست C استاندارد IEEE 1451.4 آمده است. استفاده عملکردی و مفهومی این گرامر به تفصیل در این بخش شرح داده می‌شود.
سازندگان ممکن است جزییات مرتبط با یک ترانسدیوسر را در TEDS ذخیره کنند و اطلاعات لازم برای رمزگشایی آنرا که بصورت یک الگو نوشته شده در TDL فراهم کنند. بنابراین یک کاربر میتواند اطمینان حاصل کند که اطلاعات ذخیره شده در TEDS می‌تواند بازیابی شود. این امر به کمک یک الگوی مناسب و یک تجزیه کننده با قابلیت تفسیر TDL را لازم میدارد. این امر مشکلات یک ترانسدیوسر جدید و یا بروز سازی نسل جدیدی از یک ترانسدیوسر را حل می‌نماید. هر گونه بروزسازی تنها با استفاده از یک الگوی جدید براحتی قابل پیاده‌سازی است.

۲-۵-۱-۱- فرمانها

فرمانهای TDL به لحاظ عملکردی تشریح و شامل

- فرمانهای کنترلی- اطلاعاتی را نسبت به جریان عمومی یک الگو فراهم می‌کنند
- فرمانهای شناسایی- به شکل منحصر به فرد الگو و TEDSهای مرتبط را مشخص می‌کنند
- فرمانهای ویژه‌گی- در واقع هسته الگوها هستند بطوریکه مکانیزمی لازم برای تفسیر دادن مقادیر داده‌های TEDS ترانسدیوسر را فراهم می‌کنند

[1] Template Description Language

DOI: 10.30503/nripress.2020.296

۲-۵-۱-۲- نقشه بیتی

مهم است که بدانیم کاری که الگو انجام می‌دهد برنامه ریزی برای TED به شکل بیت به بیت است. با وجود اینکه برخی از فرمانهای TDL هدفی فراعملکردی داشته و هیچ نشانه‌ای از نگاشت بیت وجود ندارد، اغلب فرمانها به شکل شفاف n بیت بعدی را مشخص میکنند، و آنگونه که در زبان فرمانها نشان داده شده است به شکل یک رفتار ویژه باید تفسیر شود. بنابراین، آنگونه که هر فرمان در یک الگو جدا شده و پیاده‌سازی می‌شود، اگر فرمان نگاشت بیت‌ها را در TEDS انجام می‌دهد، آن بیت‌ها از TEDS خوانده شده و اشاره‌گر ورودی به TEDS به بیت بعدی از دنباله بیت‌هایی که در لحظه خوانده شده‌اند ارتقا پیدا می‌کند.

۲-۵-۱-۳- TEDS پایه

پیشتر به این مفهوم اشاره شده است.

۲-۵-۱-٤- حساسیت به حروف بزرگ یا کوچک

این زبان به حروف بزرگ یا کوچک حساس نیست.

۲-۵-۱-۵- رشته‌های کاراکتری

موارد زیادی وجود دارند که یک برنامه نویس الگو ممکن است یک رشته کاراکتر محصور نشده با کاراگتر " را مشخص میکند. به شکل خاص تر این موارد میتوانند به شکل زیر مشخص شوند:

- در فرمان شمارشی
- در آرایه‌ای از ساختارها
- مانند یک پسوند به %MDEF_Properies

یک رشته غیر محصور باید یک مجموعه از کاراکترهای پیوسته (بدون فاصله سفید) شامل کاراکترهای حرفی و عددی (A-Z و 0-9) و هر یک از کاراکترهای ویژه ^، $، _، !،@، #، یا & باشد. اولین کاراکتر نیز نباید رقم (0-9) باشد.
یک رشته محصور با " شامل هر کاراکتر قابل چاپ ASCII به استثنای " و CR و LF است.

۲-۵-۲- فرمانهای تعیین هویت یا شناسایی

این فرمانها اطلاعاتی را فراهم می‌کند که یک الگو به شکل یکتا تعریف شود. یک الگو می‌تواند از پیوستن یک یا چند الگو باشد. بیت‌ها در TEDS برای شناسایی هر الگو به شکل یکتا استفاده می‌شوند.

۲-۵-۲-۱- الگو و خاتمه الگو

این فرمانها شروع و پایان یک الگو را تعریف می‌کند. هر فرمان شروع الگو Template باید به یک فرمان خاتمه الگو EndTemplate ختم شود.

DOI: 10.30503/nripress.2020.296

فرمت و شکل فرمان به شکل زیر است:

```
Template <Manufacturer ID>, <ID number of bits>, <Template ID>,<title>
        <commands>
EndTemplate
```

<Manufacturere ID> باید یک عدد صحیح در مبنای ده باشد. آن می‌تواند برای یک IEEE 1451.4 مقدار صفر داشته باشد; برای یک الگوی تعریف شده بوسیله سازنده، آن می‌تواند مقدار سازنده که الگو را ایجاد کرده است داشته باشد. در زیر بخش ۲-۲-۳-۳-۱ این شناسه سازنده توصیف شده است. دقت کنید که 16382 یک یک شناسه ویژه سازنده است که برای الگوهای تعریف شده بوسیله کاربر رزرو شده است.

<ID number of bits> باید یک عدد صحیح در مبنای ده که نشان دهنده تعداد بیت‌هایی است که باید خوانده شود.

<Template ID> باید یک عدد صحیح در مبنای ۱۰ باشد.

<title> یک رشته با " انتخابی خواهد بود.

<commands> هر مجموعه‌ای از فرمانهای معتبر به استثنای الگو و خاتمه الگو می‌تواند باشد.

فرمان الگو (Template command) باید در ابتدای خط در هر الگویی باشد. فرمان الگو بیت‌های <ID number of bits>در TEDS را نگاشت خواهد کرد. فرمان خاتمه الگو باید آخرین خط الگو باشد. فرمان خاتمه الگو هیچ یک از بیت‌های TEDS را نگاشت نخواهد کرد.

۲-۵-۲-۲- شناسه الگو

الگوهایی که با یک شناسه سازنده در ارتباط هستند با یک شناسه یکتا در بین شناسه سازنده ارتباط خواهند داشت. برای هر شناسه سازنده، تعداد بیت‌هایی که برای شناسه الگو استفاده می‌شود باید یکی باشند. همانگونه که پیشتر گفته شد، بهنگام کار با یک الگوی جدید لازم است که یک الگوی مرتبط با شناسه سازنده جاری جاگذاری شود، برای تشخیص تعداد بیت‌هایی که باید خوانده شوند تا شناسه الگو را تفسیر کند این مقدار باید معلوم باشد. برای نمونه به مثال زیر توجه شود:

Example:

```
Template 0, 8, 13, "Descriptive Title of Template"
EndTemplate
```

where

```
Manufacturer ID = 0
ID number of bits = 8
Template ID = 13
```

۲-۵-۲-۳- الحاق الگو

همانگونه که در شکل (۲-۶) شرح داده شده، نگاشت بیتی(bitmapping) TEDS از مجموعه بهم پیوسته‌ای از یک سری الگوهاست. بطوریکه بعد از خواندن TEDS پایه، چندین الگو ممکن است نیاز باشد تا به شکل کامل

DOI: 10.30503/nripress.2020.296

TEDS را جدا کنند. برای نمونه یک TEDS ممکن است بوسیله TEDS پایه شروع و بدنبال آن یک الگوی استاندارد IEEE و سپس با یک الگوی تعریف شده توسط سازنده و با یک الگوی تعریف شده با کاربر تکمیل شود. همه این الگوها در TEDS معرفی می‌شوند که از شناسه‌های الگو برای شناسایی آنها استفاده می‌شود.

۲-۵-۲-٤- اعتبارسنجی الگو

ممکن است که یک خطا بین خواندن داده از TEDS و تعریف بیت‌ها در یک الگو رخ دهد. محل اعتبارسنجی در انتهای یک الگو رخ می‌دهد. زمانیکه یک خطا در مدت جداسازی یک الگو رخ دهد باعث می‌شود که کل الگو، و تمام الگوهای بعدی، نامعتبر شوند ولی داده‌های خوانده شده از الگوهای قبلی معتبر هستند.

۲-۵-۲-۵- صحت سنجی[۱] فایل الگو

یک فایل ممکن است بیشتر از یک الگو داشته باشد(که هر کدام با یک زوج Template و EndTemplate ترسیم می‌شوند). خط آخر هر فایل الگو باید یک فرمان Validation_Keycode باشد.

Syntax:

Validation_Keycode *<checksum>*

فیلد <checksum> باید یک عدد صحیح در مبنای ده باشد.

الگوریتم عبارت است از:

۱- مقدار متغییر صحیح ۳۲ بیتی checksum را به مقدار صفر آماده سازی کنید.

۲- برای هر خط در فایل تا به خط پایانی، به استثنای خط آخر که شامل فرمان Validation_Keycode است کارهای مرحله ۳ را انجام دهید.

۳- برای هر کاراکتر ACII، شامل CR و LF، موجود در هر خط مراحل ۴ تا ۵ را انجام دهید.

۴- کاراکتر ASCII را به یک مقدار ۸ بیتی unsigned integer تبدیل کنید.

۵- مقدار ۸ بیتی را به checksum ۳۲ بیتی اضافه کنید.

پیوست D استاندارد شامل یک نمونه برنامه C++ برای پیاده سازی اعتبار سنجی checksum است.

۲-۵-۲-٦- شماره نسخه TDL

فرمان TDL_Version_Number مشخص می‌کند کدام نسخه TDL در لحظه ایجاد الگو استفاده شده است.

Syntax:

TDL_Version_Number *<version number>*

<version number> باید یک عدد صحیح مثبت در مبنای ده باشد.

---

[۱] cheksum

DOI: 10.30503/nripress.2020.296

مقدار <version number> نسخه TDL را در لحظه انتشار الگو مشخص می‌کند. مقادیر مشخص و معنی آنها در جدول (۲-۴) فراهم شده است. باید یک و تنها یک TDL_Version_Number در یک الگو وجود داشته باشد. TDL_Version_Number هیچ بیتی از TEDS را نمی‌خواند.

جدول ۲-٤- شماره‌های نسخه

Version number	Meaning
0	Reserved for legacy use. Beta format.
1	Reserved for legacy use. IEEE P1451.4, version 0.9.
2	IEEE Std 1451.4-2004.

۲-۵-۲-۷- UDID[1] (شناسه توصیف یکتا)

UDID یک شناسه یکسان برای یک الگو و انتخاب‌هایی که در این الگو ساخته شده است، می‌باشد. UDID به شکل واضح در TEDS یا در توصیف الگو پیدا نمی‌شود، اما از ساختار TEDS و الگوی استفاده شده محاسبه می‌شود. به عنوان ارجاعی به TEDS استفاده شده در یک ترانسدیوسر و UGID نیز میباشد.
ساختار دستور زبانی UDID به شکل زیر است:

$$\text{UDID} \quad \{\text{I<tid>}\{\text{—}\ \text{<sc>}\}_0^\infty \ \big|\text{M<mid>:<tid>}\{\text{—}\ \text{<sc>}\}_0^\infty \ \text{E}\big|\text{U}\}_1^\infty$$

where

I is an IEEE 1451.4 standard template,
M is a Manufacturer template,
E is an Embedded Template,
U means User data,
<tid> is the template identification number in decimal,
<sc> is the actual selected Select Case value in decimal,
<mid> is the Manufacturer ID number in decimal.

$\{\}_0^\infty$ indicates that this part can occurs between *0* to ∞ times in the parsing of the template.

$\{\}_1^\infty$ indicates at least one occurrence in the parsing of the template.

[1] Unique Description Identification

DOI: 10.30503/nripress.2020.296

بصورت مثال برای تشخیص الگوی ۲۸ با انتخاب‌های زیر

SELECTCASE: "TEDS selection"
With CASE selected: "Combined TEDS for Preamplifier with Attached Microphone" (value = 1)

SELECTCASE "Extended Functionality"
With CASE selected: "Programmable Sens" (value = 1)

SELECTCASE "Additional Basic TEDS"
With CASE selected: "System" (value = 1)

SELECTCASE "System Test (Test Gain)"
With CASE selected: "Not available" (value = 0)

SELECTCASE "Transfer Function"
With CASE selected: "Specified" (value = 1)

و با یک یک داده کاربری به شناسه زیر می انجامد

UDID I28-1-1-1-0-1U

۲-۵-۲-۸- UGID (شناسه یکتای گروه)

شناسه گروه ممکن است به عنوان مکانیزمی برای الگوهای گروه بندی استفاده شود. اینکه چگونه این استفاده می‌شود در این استاندارد تعریف نشده است. با این وجود برای برآورده کردن یکتایی عمومی، قواعد UGID به شکلی که در زیر آمده تعریف می‌شود:

Syntax:

UGID <"*identifier*">, <"*description*">

The <"*identifier*"> shall be a quoted text field.

The <"*description*"> shall be a quoted text field.

این فرمان هیچ بیتی را در TEDS ترسیم نمی‌کند.

۲-۵-۲-۸-۱- قوانین استخراج UGID

<"identifier"> باید UDID در داخل "" از یکی از الگوهای درون گروه باشد.

Example:

UGID "I28-1-1-1-0-1", "Preamp. with programmable gain"

۲-۵-۲-۸-۲- رخ دادن چندین UGID

ممکن است بیشتر از یک UGID در یک الگو (داخل caseهای یک فرمان SelectCase) باشد. اگر بیشتر از یک فرمان UGID درون یک الگو باشد، آنها باید در SelectCases به طریقی باشند که تجزیه کردن الگو همیشه منجر به فقط یک UGID یکتا شود.

DOI: 10.30503/nripress.2020.296

۳-۵-۲- فرمانهای کنترلی

این فرمانها اطلاعات مرتبط با جریان عمومی از مابین الگو بوده و بعلاوه اطلاعات توصیفی را فراهم می‌کند. هیچ یک از این فرمانها، اطلاعات ویژه ترانسدیوسرها را فراهم نمی‌کنند.

۲-۵-۳-۱- چکیده

این فرمان توضیحاتی که الگو را توصیف می‌کند را فراهم می‌کند که در الگوی مکانیزه شده به شکل نرم افزاری برای کاربر نمایش داده شود.

Syntax:

abstract <*description*>

<description> یک متن اختیاری است که به وسیله انتهای خط مجزا می‌شود.
این فرمان در مقایسه با فرمان توضیحی عمومی به این شکل است که باید بوسیله نرم افزار ویرایشی الگو استفاده شود. <description> می‌تواند برای نمایش به کاربر استفاده شود. توضیحاتی که بوسیله فرمان توضیحی (comment command) فقط قابل نمایش برای آنهایی است که دسترسی مستقیم به الگو دارند.
در حالت استفاده عادی، تعداد فراوانی فرمانهای Abstract در ابتدای الگو وجود دارد که یک مرور کلی بر الگو، تاریخچه آن، و کاربرد آن را فراهم می‌کند. به هنگام نمایش هر <description> یک تک خط را اشغال می‌کند. ترتیب این <description> ها خواهد توانست خلاصه‌ای از الگو را تشکیل دهد.

Example:

```
abstract IEEE P1451.4 template version 0.91
abstract rev. 6 date: 990308
abstract rev. 5 was approved via IEEE 1451.4 telcon 990308
```

فرمانهای Abstract هیچ بیتی را در TEDS ترسیم نخواهد کرد.

۲-۵-۳-۲- توضیحها (// comments)

این فرمان کامنت‌های عمومی گنجانده شده و استفاده شده بوسیله نویسنده‌های الگو را فراهم می‌کند.

Syntax:

// <*description*>

<description> یک فیلد متنی انتخابی است.
کامنت‌ها برای استفاده داخلی الگو است. این در برابر فرمان Abstract است، که سعی در فراهم کردن یک توصیفی برای کاربر است. فرمان comment ممکن است بعد از هر فرمانی نمایان شود.(فرمانهای کامنت بین الگوها مجاز هستند. به همین دلیل است که آنها میتوانند بعد از فرمان ENDTEMPLATE و قبل از فرمان TEMPLATE

DOI: 10.30503/nripress.2020.296

نمایان می‌شوند.) متن بین // ها، و انتهای خط باید کامنت‌های متنی باشند. فرمان کامنت هیچ بیتی در TEDS را به تصویر نمی‌کشد. به نمونه زیر توجه شود:

Example:

```
// This is a comment
%Gain, "Amplifier Gain", Cal, 7, UNINT, "", ""// Set Gain
```

۲-۵-۳-۳- SelectCase و Case

این فرمانها قابلیت استفاده از بیت‌ها را در TEDS برای مشخص کردن موارد مختلف فراهم می‌کند.

Syntax:

```
SelectCase <"description">, <access_level>, <number_of_bits>
            Case <"description">, <bit value>
                  <command list>
      EndCase
            …
            Case <"description">, <bit value>
                  <command list>
      EndCase
EndSelect
```

The <*"description"*> shall be a quoted text field.

The <*access_level*> shall be either omitted or be one of the access levels identified in Table 6.

The <*number_of_bits*> shall be a positive decimal integer.

The <*bit value*> shall be a positive decimal integer less than $2^{<number_of_bits>}$.

The <*command list*> shall be a sequence of TDL commands.

این عبارت‌ها یک انتخاب شرطی مبتنی بر یک مجموعه از بیت‌ها در TEDS را پیاده سازی می‌کنند. زمانیکه TEDS تجزیه می‌شود و یک عبارت SelectCase در یک الگو مشاهده می‌شود، تعداد بیت‌هایی که بوسیله <number_of_bits> مشخص می‌شوند باید از TEDS خوانده شود. <bit value> خوانده شده برای مشخص کردن اینست که کدام case باید اجرا شود. فرمانی که در <command list> پیدا می‌شود مرتبط با case ای است که باید اجرا شود. تمام فرمانها در لیست فرمان مربوطه انجام می‌شود و بقیه نادیده گرفته می‌شوند.

مثال زیر فرمان <number_of_bits> مربوط به TEDS را نشان می‌دهد.

Example:

```
SelectCase "Gain Values", Cal, 2
      Case "Gain=16", 0
            %Gain, "Open loop Gain", Cal, 7, UNINT, ,= 10
      EndCase
      Case "Gain=32", 1
            %Gain, "Open loop Gain", Cal, 7, UNINT, ,= 20
      EndCase
      Case "Gain=64", 2
            %Gain, "Open loop Gain", Cal, 7, UNINT, ,= 40
```

NRI

DOI: 10.30503/nripress.2020.296

```
        EndCase
        Case "Gain=128", 3
              %Gain, "Open loop Gain", Cal, 7, UNINT, ,= 80
        EndCase
EndSelect
```

در این مثال، گین بوسیله یک سوییچ ۲ بیتی آماده می‌شود. از آنجایی که گین ۷ بیت را اشغال می‌کند، یک ۵ بیت در TEDS برای خواندن گین به شکل مستقیم از TEDS باقی می‌ماند.

۲-۵-۳-٤- آرایه ساختار[۱]

این فرمان به ایجاد یک مجموعه از خاصیت‌ها (properties) که به شکل یک تک ساختار علاوه بر توانایی ایجاد دینامیکی یک آرایه بوسیله خواندن اندازه آرایه از TEDS اجازه میدهد.

Syntax:

StructArray <*name*>, <*"description"*>, <*access_level*>, <*number_of_bits*>

<*property_list*>

EndStructArray

The <*name*> shall be an unquoted character string.

The <*"description"*> shall be a quoted text field.

The <*access_level*> shall be either omitted or be one of the access levels identified in Table 6.

The <*number_of_bits*> shall be a positive decimal integer.

The <*property_list*> shall be any set of properties.

NRI

زمانیکه یک فرمان StructArray تجزیه می‌شود، تعداد بیت‌ها که بوسیله <number_of_bits> مشخص می‌شود از TEDS خوانده می‌شود. <property_list> تعداد دفعاتی که به وسیله این مقدار اولیه نمایش داده می‌شود را می‌خواند. دقت شود که فقط خصوصیات (properties) در StructArray وجود دارد و در فرمان دیگری وجود ندارد. نمونه این فرمان در زیر آمده است.

Example:

```
STRUCTARRAY CalTable, "Calibration table", CAL, 7
      %CalPoint_DomainValue, "Domain Calibration Point (% of full
      span)", CAL, 16, ConRes, 0, 0.00153, "0.00", "%"
      %CalPoint_RangeValue, "Range Calibration Deviation (% of
      full span)", CAL, 21, ConRes, -100, 0.0001, "0.0000", "%"
ENDSTRUCTARRAY
```

[۱] StructArray

DOI: 10.30503/nripress.2020.296

۲-۵-۳-۵- فاصله گذاری

این فرمان یک جداسازی در فرمت بندی به کاربردهای سطوح بالاتر بین فیلد قبلی و فیلد بعدی است.

Syntax:

```
Spacing
```

این فرمان فقط برای فرمانهای نمایشی است. این فرمان هیچ یک از بیت‌های در TEDS را ترسیم نمی‌کند.

Example:

```
Spacing
```

در طول خواندن این فرمان، یک برنامه نمایشی الگو ممکن است یک خط را نشان دهد تا به شکل تصویری فرمانهای قبل از فاصله گذاری را از فرمانهای جلوی فاصله گذاری جدا کند.

۲-۵-۳-۶- در یک ردیف قرار دادن[۱]

این فرمان امکان درست قرار دادن اشاره گر ورودی TEDS در طول مرزهای کلمه را می‌دهد، در اصل، به شکل بالقوه از چندین بیت مختلف در TEDS عبور می‌کند.

Syntax:

```
Align <word_size>
```

<word_size> باید یک عدد مثبت صحیح در مبنای ده باشد.

<word_size> باید به شکل اندازه کلمه در حافظه‌ای که TEDS ذخیره شده در نظر گرفته شود. فرمان align اشاره گر ورودی TEDS را به ابتدای کلمه بعدی حرکت می‌دهد. این نیاز به این دارد که پردازنده دو چیز را بداند: جایگاه فعلی در مسیر داده در TEDS (برای نمونه جاییکه اشاره‌گر ورودی در حال حاظر به آن اشاره می‌کند) و <word_size>.

فرض کنید آدرس شروع 0، باشد، اشاره گر ورودی باید با مقدار زیر افزایش یابد:

<word_size> – (Current_Position mod <word_size>)

فرمان Align باید از بیت‌هایی در TEDS به هدف قرار گرفتن در کلمه درست پرش انجام دهد.

به عنوان نمونه

Example:

```
Align 8
```

اگر مکان جاری TEDS برابر ۲۳۸ باشد، دو بیت بعدی TEDS نادیده گرفته می‌شود. این بدان معنی است که اشاره‌گر ورودی TEDS به اندازه ۲ افزایش می‌یابد، بنابراین اشاره گر جاری ۲۴۰ است. یعنی :

---

[۱] align

DOI: 10.30503/nripress.2020.296

8-(238 mod 8) = 2

۴-۵-۲- فرمانهای ویژگی % (property command)

فرمانهای ویژگی با علامت درصد % شناخته می‌شوند. این فرمانها مکانیزمی برای تخصیص دادن مقادیر ویژه ترانسدیوسر به هر از پارامترها را فراهم می‌کنند. این مقادیر در حالت عادی از TEDS خوانده می‌شوند. با این حال، همچنین یک انتخاب تخصیص دادن برای فرمانهای ویژگی وجود دارد که اجازه تخصیص دادن مستقیم یک مقدار ویژه بدون خواندن مقدار از TEDS را مجاز می‌داند.

۲-۵-٤-۱- ساختار فرمان (command stucture)

Syntax:

%<property_tag>,<"description">,<access_level>,<data_type>,<format>,

<physical_unit>

or

%<property_tag>,<"description">,<access_level>,<data_type>,<format>,

<physical_unit> = <value>

or

%<property_tag><[subproperty]>,<"description">,<access_level>,<data_type>,

<format>,<physical_unit>

or

%<property_tag><[subproperty]>,<"description">,<access_level>,<data_type>,

<format>,<physical_unit> = <value>

<property_tag> باید یکی از برچسب‌های مشخص شده در پیوست B باشد.
<”description”> باید فیلد متنی داخل کوتیشن باشد.
<acess_level> باید یکی از سطوح دسترسی تعریف شده در جدول ۲-۵ باشد.
<data_type> باید یکی از موارد تعریف شده در ۲-۲-۷-۵ باشد.
<format> باید یکی از فرمت‌های توصیفی داخل کوتیشن بند ۲-۲-۷-۶ باشد.
<physical_unit> باید یک فیلد متنی داخل کوتیشن یا آرایه به شکلی که در ۲-۲-۷-۷ آمده باشد.
<subproperty> باید یک زیر ویژگی با عملکرد بسط یافته باشد که در ۲-۲-۷-۹ شرح داده شده است.

DOI: 10.30503/nripress.2020.296

<value> باید یک مقدار عددی یا متنی مناسب برای نوع داده مرتبط با آن باشد، برای جزییات بیشتر به ۲-۲-۸-۷ توجه شود.

<property_tag> همیشه با یک پیشوند % بدون هیچ فاصله‌ای همراه است. هیچ کامایی بین <property_tag> و <subproperty> وجود ندارد. هیچ کامایی اطراف علامت مساوی = وجود ندارد. بقیه پارامترهای دیگر با یک کاما جدا شده‌اند. همه پارامترها به استثنای <property_tag> و <data_type> اختیاری هستند. با این حال اگر یک پارامتر حذف شود، یک کاما باید فیلد پارامتر را تعیین کند.

توصیفهای جزییات هر یک از پارامترها در پاراگراف زیر ارایه شده است. استفاده از علامت مساوی در زیر بند ۲-۲-۸-۷ توضیح داده شده است.استفاده از <[subproperty]> در بند ۲-۲-۷-۹ شرح داده شده است. فرمانهای ویژگی تخصیص داده نشده باید بیت‌ها از TEDS را با توجه به اندازه مشخص شده در <data_type> بخوانند. این بیت‌ها با توجه به پارامتر <data_type>، و مقدار منتج شده که به این ویژگی اختصاص داده می‌شوند، تفسیر می‌شوند. فرمانهای ویژگی تخصیص داده شده هیچ بیتی را از TEDS نخواهند خواند.

۲-۵-٤-۲- لیست ویژگی (property list)

پیوست B استاندارد برچسب‌های ویژگی تعریف شده بوسیله این استاندارد را شرح داده و لیست می‌کند. سازنده‌ها این امکان را دارند که ویژگی خودشان را آنگونه که در ۲-۲-۷-۱۰-۱ شرح داده شده ایجاد کنند.

۲-۵-٤-۳- توصیف (description)

پارامتر <"description"> یک فیلد متنی کوتیشن دار است. هدف آن این است که به شکل یک توصیف خلاصه برای کاربر نمایش داده شود.

۲-۵-٤-٤- دسترسی سطح (Acess_level)

پارامتر <acess_level> یک سطح دسترسی داده را برای مقدار تخصیص داده شده به این ویژگی اختصاص می‌دهد. این بدان منظور است که یک سطح امنیتی برای داده فراهم کند. به این معنی که، کاربرهای مختلف فقط در صورتی قادر به نوشتن داده هستند که دسترسی کافی برای این منظور داشته باشند. جدول(۲-۵) مقدار سطح دسترسی مجاز را فراهم می‌کند.

جدول ۲-۵- سطوح دسترسی برای ویژگی‌های ترانسدیوسر

Access level	Description
ID	Data used to identify the specific IEEE 1451.4 Transducer.
Cal	Data that may be changed when the transducer is calibrated.
Usr	Data that may be changed by the user.

۲-۵-٤-۵- نوع داده (Datatype)

DOI: 10.30503/nripress.2020.296

پارامتر <data_type> روشی را تعریف می‌کند که بوسیله آن بیت‌های خوانده شده از TEDS برای آن ویژگی رمزگشایی می‌شوند. انواع داده مجاز در جدول(۲-۶) آمده است. جزییات بیشتر در این بند فراهم شده است.

Syntax:

<number_of_bits>, <dtype>

or

<number_of_bits>, <float_type>, <start_value>, <tolerance>

or

<number_of_bits>, single

or

<number_of_bits>, <enum_type>

<number_of_bits> باید یک عدد صحیح در مبنای ده باشد.
<dtype>، <float_type>، single، و <enum_type> باید یکی از انواع نشان داده شده در جدول ۲-۶ باشد.
<start_value> باید یک عدد دهدهی باشد.
<tolerance> باید یک عدد دهدهی باشد.

۲-۵-٤-۵-۱- تعداد بیت‌ها (Number_of_bits)

پارامتر <number_of_bits> طول بیت را در TEDS برای مقدار این نوع داده مشخص می‌کند. بنابراین، برای این نوع داده، <number_of_bits> بیت‌های از TEDS خوانده شده، مطابق با پارامتر <dtype>، <float_type>، یا <enum_type> تفسیر شده، و مقدار نهایی به ویژگی جاری تخصیص داده می‌شود.

جدول ۲-٦- انواع داده

DOI: 10.30503/nripress.2020.296

<dtype>	Description
Date	Number of the day since 1 Jan 1998.
UnInt	Unsigned Integer.
Bit_Digit	Binary digit by 2 bits.
BitBin	Array of Bit_Digits.
Chr5	Character by five bits.
ASCII	Standard 7-bit ASCII.
Unicode	International Unicode. See ISO/IEC 10646-1:2002 and *The Unicode Standard, Version 2.0* [B14].
String5	Describes a structure that consists of a string length and an array of CHR5 characters.
String7	Describes a structure that consists of a string length and an array of ASCII characters.
String16	Describes a structure that consists of a string length and an array of Unicode characters.
<float_type>	**Description**
ConRelRes	Constant Relative Resolution. Only positive values. Provides a logarithmic mapping of an interval.
ConRes	Constant Resolution. Provides a linear mapping of an interval.
Single	IEEE 754-defined 4-byte single precision floating-point number.
<enum_type>	**Description**
(Defined enumeration type)	References an enumerated data type defined by the Enumeration command. for details on enumeration.

NRI

۲-۵-٤-۵-۲- انواع داده استاندارد و رشته <dtype>

تاریخ (Date)

نوع داده تاریخ یک عدد صحیح است که تعداد روزها را از تاریخ 1 january 1998 را می‌شمرد. اولین روز در همان تاریخ روز صفر است. همه '1' در TEDS به شکل خام (باینری) نشاندهنده این است که ویژگی استفاده نشده است.

Example:

```
%CalDate, "date of calibration", CAL, 14, Date, "", ""
```

ورودی باینری برای این باید:

00000000011111 باشد که به 1 February 1998 ترجمه می‌شود.

دقت کنید ۱۴ بیت، تاریخ را تا سال ۲۰۳۲ فراهم می‌کند.

UnInt

این نوع داده یک عدد صحیح بدون علامت مستقیم را از باینری به دسیمال (دهدهی یا مبناهای دیگر) مشخص می‌کند. همه '1' در TEDS خام (باینری) نشاندهنده اینست که ویژگی ای استفاده نشده است.

Example:

```
%Gain, "Amplifier Gain", Cal, 7, UnInt, "", ""
```

DOI: 10.30503/nripress.2020.296

ورودی باینری برای این ممکن است عدد زیر باشد:
0000010 که به دسیمال به عدد ۲ ترجمه خواهد شد.

Bit_Digit

Bit_Digit یک رقم باینری است که می‌تواند مقادیر بیتی "0"، "1"، "x"، و "," کاما را آنگونه که در جدول(۲-۷) آمده شامل می‌شود.

جدول ۲-۷- تعریف Bit_Digit

Binary value	Bit_Digit
00	0
01	1
10	x
11	,

BitBin

BitBin یک آرایه از Bit_Digitها است.

Chr5

این یک مجموعه از کاراکتر به هدف استفاده از یک حداقل تعداد بیت‌ها برای کاراکترها است. جدول(۲-۸) رمزگشایی را فراهم می‌کند.

Example:

```
%CalInitials, "person who calibrated", CAL, 15, Chr5, "", ""
```

ورودی باینری ممکن است:
00011 00010 00001 باشد که به ABC ترجمه می‌شود.

جدول ۲-۸- نوع داده کاراکتری Chr5

		lsb							
		000	001	010	011	100	101	110	111
msb	00	Space	A	B	C	D	E	F	G
	01	H	I	J	K	L	M	N	O
	10	P	Q	R	S	T	U	V	W
	11	X	Y	Z	,	.	/	_	@

ASCII

بیت‌های استفاده شده در این نوع داده باید با استفاده از کد ۷ بیتی استاندارد ASCII تفسیر شوند.
<number_of_bits> برای این نوع داده باید مضرب ۷ باشد.

DOI: 10.30503/nripress.2020.296

Example:

```
%CalInitials, "person who calibrated", CAL, 21, ASCII, "", ""
```

ورودی باینری برای این مورد می‌تواند 1000001 1000010 1000011 باشد که به ABC ترجمه می‌شود.

Unicode

بیت‌ها با استفاده از این نوع داده با استفاده از ISO/IEC 10646-1:2002 (به استاندارد Unicode ویرایش ۲ مراجعه شود.) تفسیر می‌شوند.

<number_of_bits> برای این نوع داده باید مضرب ۱۶ باشد.

Example:

```
%CalInitials, "person who calibrated", CAL, 48, Unicode, "", ""
```

ورودی باینری برای این مورد می‌تواند

0000000001000001 0000000000010000010 00000000001000011 باشد که به مقدار ABC ترجمه می‌شود.

StringN

انواع داده String5، String7، String16 برای مجاز دانستن متغییر رشته با طول متغییر بین TEDS مهیا می‌شود. به این معنی که، ویژگی در الگو یک تعداد بیت‌هایی را شامل می‌شود که نشان دهنده طول بیت‌های یک فیلد است که تعداد کاراکترهای درون رشته را نشان می‌دهد.

زمانیکه یک فرمان StringN تجزیه می‌شود، تعداد بیت‌هایی که با <number_of_bits> که از TEDS خوانده می‌شود، نشان داده می‌شود. مقداری که خوانده می‌شود، تعداد کاراکترهای در رشته را نشان می‌دهد، که سپس از TEDS خوانده می‌شود.

Example:

```
%CalInitials, "person who calibrated", CAL, 4, String5, "", ""
```

در این مثال، ۴ بیت از TEDS خوانده خواهد شد. و مقدار این ۴ بیت تعداد کاراکتر ۵ بیتی Chr5 که باید برای رشته خوانده شود، را نشان می‌دهد. اگر بیت‌های TEDS برای این مثال به شکل زیر باشد:

0011 00001 00010 00011

بنابراین، اولین ۴ بیت مقدار ۳ را دارد، پس سه کاراکتر Chr5 باید (۱۵ بیت تمام) برای مقداردهی به رشته خوانده شود: ABC

۲-۵-٤-۵-۳- انواع ممیز شناور <Float_Type>

دو نوع داده وجود دارد که مقادیر فشرده ممیز شناور را فراهم می‌کند. یکی یک نگاشت خطی در یک بازه را فراهم می‌کند و دیگری یک نگاشت لگاریتمی را مهیا می‌کند. این انواع داده، ConRelRes و ConRel دو پارامتر <start_value> و <tolerance> را به همراه دارد.

برای انواع ممیز شناور، مقدار باینری خام TEDS که همگی یک هست نشان دهنده این است که یک عدد نیست (NaN). این منطبق بر IEEE Std 754-1985 با در نظر گرفتن تک نوع است. زمانیکه NaN خوانده شود، مقدار ویژگی تعریف نشده است و ویژگی استفاده نخواهد شد.

- مقدار اولیه (Start_Value)

DOI: 10.30503/nripress.2020.296

مقدار اولیه، مقداری که به ویژگی تخصیص داده می‌شود را تعریف میکند، زمانیکه بیت‌های خوانده شده از TEDS همگی صفر هستند و برای استفاده در محاسبه مقدار نهایی که برای تخصیص به ویژگی برای دیگر مقادیر بیت‌های TEDS در نظر گرفته می‌شوند.

- تلرانس

تلرانس برای اندازه گام در ConRes استفاده می‌شود.

در ConRelRes، تلرانس به عنوان بیشینه تفاوت قابل قبول بین یک مقدار مطلوب و مقداری که باید در TEDS ذخیره شود، تعریف می‌شود. این برای استفاده در مشخص کردن رزولوشن عددی مورد نیاز است، که دو برابر تلرانسی است که در فرمول برای ConResRel آمده است. این مقادیر باید با در نظر گرفتن عدم قطعیت کالیبراسیون انتخاب شود. برای توضیحات بیشتر درباره ConRes و ConRelRes می‌توان به متن استاندارد مراجعه کرد.

- دقت معمولی (single)

این نوع داده یک IEEE 754 تعریف شده ۴ بایتی با دقت معمولی عدد ممیز شناور است.

Example:

```
%MaxPhysVal, "Max Val of the Physical Range", ID, 32, Single, "",
"V/(m/s²)"
```

ورودی باینری برای این مثال می‌تواند:

0 10000001 01000000000 به شکل رابطه زیر برای محاسبه استفاده می‌شود:

$$\text{MaxPhysVal} = -1 \times 2 \times (129-127) \times 1.625 = -6.5$$

۲-۵-٤-۵-٤- انواع شمارشی <enum_type>

نوع شمارشی یک نگاشت صحیح به مجموعه مقادیر گسسته فراهم می‌کند. تعیین شمارش در ابتدا در فرمان شمارشی تعریف می‌شود. نوع تعریف شده به عنوان یک نوع داده در فرمان ویژگی استفاده می‌شود.

Syntax:

Enumerate <*defined_type*>, <*enum_list*>

<defined_type> باید یک رشته کاراکتر بدون کوتیشن باشد.

<enum_list> یک لیست رشته‌های کاراکتر یا مقادیر عددی که با کاما به ترتیب جدا شده‌اند، می‌باشند.

رفرنس به یک آیتم در یک لیست شمارشی ممکن است بوسیله مکان به عنوان یک عدد صحیح یا بوسیله خود لیست آیتم باشد. یک رفرنس عدد صحیح به یک آیتم در یک لیست شمارشی باید اولین آیتم در لیست را به عنوان صفر ۰ در نظر بگیرد.تعداد بیت‌های لازم برای یک نوعشمارشی از تعداد مقادیر محتمل تعیین می‌شود. هنگامیکه یک مقدار ثابت در یک ویژگی شمارشی در نظر گرفته می‌شود، مقدار ثابت ممکن است یک مقدار شمارشی باشد یا مقدار کاراکتری.

نوع شمارشی هیچ بیتی را در TEDS تصویر نمی‌کند.

DOI: 10.30503/nripress.2020.296

Example:

The following defines an enumerated data type called "mycolors."

```
Enumerate mycolors, "blue", "black", "gray", "green"
```

This enumeration assigns integer values of 0 to blue, 1 to black, 2 to gray, and 3 to green.

این فرمان ویژگی رنگ منتخب را می‌خواند.

```
%MDEF_Color, "Choice of color", USR, 2, mycolors, "", ""
```

ورودی باینری خوانده شده از TEDS برای این فرمان می‌تواند 01 باشد، که رنگ منتخب را سیاه تعریف می‌کند.
فرمان وپیگی برابر با فرمان قبل برای استفاده از تخصیص ثابت می‌تواند به شکل زیر باشد:

```
%MDEF_Color, "Choice of color", USR, 2, mycolors, "", "" = black
```

۲-۵-٤-٦- پارامتر فرمت

پارامتر فرمت یک فرمت نمایش پیشنهاد شده برای مقدار یک ویژگی است. فرمت‌ها همیشه رشته‌های داخل کوتیشن هستند. فرمت‌ها ممکن است با ایجاد یک رشته از کاراکترهای ویژه توصیف شده باشند.
جدول(۲-۹) لیست‌های کاراکترهای با فرمت مختلف را نشان می‌دهد.

جدول ۲-۹- نشانه‌ها و علایم فرمت بندی

DOI: 10.30503/nripress.2020.296

Numeric formatting symbols	Description
. (period)	Decimal separator. The actual character displayed is dependent on the display program.
, (comma)	Thousand separator. The actual character displayed is dependent on the display program.
0	Digit placeholder. Display a digit or 0.
#	Digit placeholder. Display a digit or nothing.
$	Display the literal character $.
%	Percentage. The value is multiplied by 100 and a percent sign is appended.
E– or e–	Scientific notation with a minus sign (–) next to negative exponents and nothing next to positive exponents. This symbol shall be used with other symbols, as in 0.00E–00 or 0.00E00.
E+ or e+	Scientific notation with a minus sign next to negative exponents and a plus sign (+) next to positive exponents. This symbol shall be used with other symbols, as in 0.00E+00.
p	Displays a number using the SI prefixes
r	Relative (displays the number with a resolution corresponding to the resolution given by the Con-RelRes statement). This is generally not used by itself, but in conjunction with the "p" format. See 7.4.6.1.
rp	Relative with prefix. A combination of the preceding "r" and "p."
String format	**Description**
S	Display as a standard string.

جدول ۲-۹- ادامه نشانه‌ها و علایم فرمت بندی

Date formats	Description
/	Date separator.
D	Day of the month in one or two numeric digits, as needed (1 to 31).
Dd	Day of the month in two numeric digits (01 to 31).
M	Month of the year in one or two numeric digits, as needed (1 to 12).
mm	Month of the year in two numeric digits (01 to 12).
mmm	First three letters of the month (Jan to Dec).
mmmm	Full name of the month (January to December).
yy	Last two digits of the year (01 to 99).
yyyy	Full year (0000 to 9999).
Enumerated format	**Description**
E or e	Displays the enumerated name rather than the enumerated index.

نمونه‌های از مثالهای فرمت بندی در جدول (۲-۱۰) نشان داده شده است

جدول ۲-۱۰- مثالهای فرمت

DOI: 10.30503/nripress.2020.296

Format	Example output
“0000.00”	0234.50
“0.00”	234.50
“####.##”	234.5
“#,##0.00”	2,345.60
“m/dd/yy”	3/26/98
“d-mmm”	26-Mar
“mmmm-yy”	January-98
“s”	This is a string.
“e”	Blue

۲-۵-٤-٦-۱-پیشوند و فرمت‌های نسبی p-، r-، و rp

این فرمت‌ها یک روش ساده برای مشخص کردن ایجاد مقیاس در صفحه نمایش باید مقیاس به منظور عدم نیاز به نمایش صفرهای زیاد است. فرمت ”p“ استفاده از پیشوند را برای نشان دادن مقیاس نشان می‌دهد. فرمت ”r“ نشان می‌دهد که دقت نشان داده شده باید بسته به دقت عدد نمایش داده شده باشد.

این مخصوصا زمانی سودمند است که از انواع داده ConRel یا ConRelRes استفاده می‌شود از آنجایی که آنها برای تهیه اعداد مقیاس شده طراحی شده‌اند.

فرمت ”p“ عموما با دیگر فرمت‌های عددی استفاده می‌شود. بدان معنی که، فرمت ”p“ یا با کاراکترهای فرمت که تعداد رقمها را نشان می‌دهد برای نمایش در هر سمت جدا کننده‌های دسیمال استفاده می‌شود یا فرمت ”p“ در ترکیب با فرمت ”r“ استفاده می‌شوند. زمانی که فرمت ”p“ حاضر است پیشوند مناسب باید در صفحه نمایش جا داده شود.

فرمت ”r“ عموما بوسیله خودش استفاده نمی‌شود. نمایش فرمت ”p“ باید از پیشوندهای در جدول(۲-۱۱) استفاده کند. توجه کنید که اینها با سیستم بین المللی واحدها (SI) همخوانی دارند.

جدول ۲-۱۱- پیشوندهای عددی

DOI: 10.30503/nripress.2020.296

Factor	Name	Symbol	Factor	Name	Symbol
10^{24}	Yotta	Y	10^{-3}	milli	m
10^{21}	Zetta	Z	10^{-6}	micro	μ
10^{18}	Exa	E	10^{-9}	nano	n
10^{15}	Peta	P	10^{-12}	pico	p
10^{12}	Tera	T	10^{-15}	femto	f
10^{9}	Giga	G	10^{-18}	atto	a
10^{6}	Mega	M	10^{-21}	zepto	z
10^{3}	Kilo	k	10^{-24}	yocto	y

NRI

Examples:

Given the following template command:

```
%Sens@Ref, "Sensitivity@ref.", Cal, 16, ConRelRes, 5E-7, 0.00015,
"rp", "V/(m/s_)"
```

مقادیر آنگونه که در جدول(۲-۱۲) نشان داده شده است نمایش داده خواهد شد.

جدول ۲-۱۲- مثالهای فرمت‌های پیشوندی و نسبی (با دقت بالا)

Actual value in the range		Will be stored in memory as	Resulting value	Displayed with format "rp"	Displayed with format "0.00p"
5,00225E–7	5,00375E–7	2	5,00300E–7	500,30n	500,30n
9,99508E–4	9,99808E–4	25339	9,99658E–4	999,7μ	999,66μ
9,99808E–4	1,00011E–3	25340	9,99957E–4	1000,0μ	999,96μ

یک فرمان الگوی مشابه با دقت پایین تر مقادیر را آنگونه که در جدول(۲-۱۳) آمده است نمایش می‌دهد:

```
%Sens@Ref, "Sensitivity@ref.", Cal, 14, ConRelRes, 5E-7, 0.015,
"rp", "V/(m/s_)"
```

دول ۲-۱۳- مثالهای فرمت‌های پیشوندی و نسبی (با دقت پایین)

DOI: 10.30503/nripress.2020.296

Actual value in the range		Will be stored in memory as	Resulting value	Displayed with format “rp”	Displayed with format “0.00p”
5,22668E–7	5,38348E–7	2	5,30450E–7	530n	530,45n
9,55398E–3	9,84060E–3	334	9,69623E–3	9,7m	9,70m
9,84060E–3	1,01358E–2	335	9,98712E–3	10,0m	9,99m
3,38670E6	3,48830E6	1000	3,43712E6	3,44M	3,44M

۲-۵-٤-۷- واحدهای فیزیکی

پارامتر واحدهای فیزیکی در ارتباط با یک رشته متنی (unit_name) با تعریف یک بعد فیزیکی و واحدها است. رشته unit_name سپس در فرمان ویژگی برای تخصیص دادن واحدهای فیزیکی به مقادیر پارامتر استفاده می‌شود. ارتباط یک unit_name با یک تعریف از واحدها به شکل محلی مربوط به الگو است.

ابعاد فیزیکی بوسیله تخصیص یک مقدار نمایی برای هر یک از واحدهای مبتنی بر SI، شامل رادیان و استرادیان و واحدهای مبتنی بر SI در نظر گرفته شده‌اند، اما واحدهای بدون SI باید به شکل استفاده از آفست و فاکتورهای مقیاس شرح داده شوند. سرانجام، واحدها ممکن است یا به شکل ابعاد، نسبت مقادیر آن ابعاد، یا لگاریتم ابعاد تعریف شوند.

این مکانیزم اجازه تعریف واحدهای اختیاری را می‌دهند، و بعلاوه به شکل کامل واحدهایی که قابل تبدیل به واحدهای مبتنی بر SI بوسیله نرم افزار کاربردی هستند را تعریف می‌کند.

۲-۵-٤-۷-۱- پیوند واحد فیزیکی به رشته کاراکتری

مکانیزم برای پیوند جایگزینی رشته کاراکتری با واحد فیزیکی از طریق فرمان Physical_Unit فراهم می‌کند. این پیوندها به شکل محلی مربوط به الگو جاری هستند و برای نشان دادن پیوندها در الگوهای دیگر برنامه ریزی نشده‌اند. فرمان Physical_Unit فقط برای اهداف کاربردی استفاده می‌شوند. این فرمان هیچ بیتی در TEDS را تصویر نمی‌کند.

Syntax:

Physical_Unit <“unit_name”>, <unit_definition>

<”unit_name”> باید یک رشته کاراکتری داخل کوتیشن باشد که در پیوند با واحدهایی است که توسط فیلد <unit_definition> برای باقیمانده الگو تعریف می‌شود.

<unit_definition> یک مجموعه ۱۲ عضوی داخل پرانتز هست که واحدها را به شکلی که در بند زیر آمده تعریف می‌کند.

۲-۵-٤-۷-۲- تعریف واحد

فیلد تعریف واحد از ۱۲ المان تشکیل شده است. این فیلد در الگو به شکل زیر نوشته می‌شود:

(enumeration, exp1, exp2, exp3, exp4, exp5, exp6, exp7, exp8, exp9, scale, offset)

DOI: 10.30503/nripress.2020.296

جدول(۲-۱۴) توصیف می‌کند که چگونه هر یک از این فیلدها باید استفاده شوند. توان[۱] باید همه اعداد دسیمال در گامهای افزایشی ½ بین 64- و 63 باشد. نه(۹) فیلد توان، یک واحد U را مطابق معادله زیر تعریف می‌کند:

$$U = \text{radians}^{exp1} \cdot \text{steradians}^{exp2} \cdot \text{meters}^{exp3} \cdot \text{kilograms}^{exp4} \cdot \text{seconds}^{exp5} \cdot \text{amperes}^{exp6} \cdot \text{kelvin}^{exp7} \cdot \text{moles}^{exp8} \cdot \text{candelas}^{exp9}$$

اولین فیلد نشان می‌دهد چگونه این بعد U در تعریف واحد استفاده استفاده می‌شود. انتخاب‌های مختلف در جدول(۲-۱۴) لیست شده‌اند. واحد با جزییات بیشتر از راه استفاده از فاکتورهای Offset و Scaling (دو فیلد آخر) برای مجاز داشتن سیستم‌های واحد به غیر از سیستم واحدهای SI، تعریف می‌شوند.

جدول ۲-۱۴- ساختار انواع داده واحدهای فیزیکی

Element no.	Description
1 Enumeration Field	0: Unit is described as m·(U+b) where U is defined by the next 9 fields, m is a dimensionless scaling factor in field 11, and b is an offset in field 12 taken to have units of U.
	1: Unit is the ratio m·(U/U +b), where U is defined by the next 9 fields, m is a dimensionless scaling factor in field 11, and b is an offset in field 12 taken to have units of U/U.
	2: Unit is m·($\log_{10}$(U/U_0)+b), where U is defined by the next 9 fields, U_0 is a reference quantity having a magnitude of exactly 1.0 with units of U, and m and b are dimensionless scaling factors in fields 11 and 12.
	3: Unit is m·($\log_{10}$(U/U)+b), where the next 9 fields define U, and m and b are dimensionless scaling factors in fields 11 and 12.
	4: The data is a bit vector (digital data) and has no unit.
	5: Unit is m·($\log_{10}^{-1}$(U/U_0)+b), where U is defined by the next 9 fields, U_0 is a reference quantity having a magnitude of exactly 1.0 with units of U, and m and b are dimensionless scaling factors in fields 11 and 12.
	6: Unit is m·($\log_{10}^{-1}$(U/U)+b), where the next 9 fields define U, and m and b are dimensionless scaling factors in fields 11 and 12.
	7–255: Reserved
2	Exponent of radians

جدول ۲-۱۴- ساختار انواع داده واحدهای فیزیکی (ادامه)

Element no.	Description
3	Exponent of steradians
4	Exponent of meters
5	Exponent of kilograms
6	Exponent of seconds
7	Exponent of amperes
8	Exponent of kelvins
9	Exponent of moles
10	Exponent of candelas
11	Scaling factor m
12	Offset factor b

[۱] exponent

DOI: 10.30503/nripress.2020.296

assigns the string "Hz" to a physical unit definition. The exponent fields define a base SI unit U as

$U = radians^0 \cdot steradians^0 \cdot meters^0 \cdot kilograms^0 \cdot seconds^{-1} \cdot amperes^0 \cdot kelvin^0 \cdot moles^0 \cdot candelas^0$

which simplifies to $U = seconds^{-1}$.

The enumeration field of 0, the scaling factor 1, and the offset factor of 0 define the units as

$"Hz" = 1.0 \cdot (U + 0\ seconds^{-1})$

این فرمان رشته "Hz" را برای واحد فیزیکی تعریف می‌کند. فیلدهای توان یک واحد U مبتنی بر SI به شکل زیر تعریف می‌کند.

$U = radians^0 \cdot steradians^0 \cdot meters^0 \cdot kilograms^0 \cdot seconds^{-1} \cdot amperes^0 \cdot kelvin^0 \cdot moles^0 \cdot candelas^0$

فیلد شمارشی 0، فاکتور بزرگی(مقیاس) 1، و فاکتور آفست 0 واحد را به شکل زیر تعریف می‌کند:

$"Hz" = 1.0 \cdot (U + 0\ seconds^{-1})$

نمونه‌های دیگر در زیر آمده است:

```
Physical_Unit "mm", (0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0.001, 0)
// "mm" millimeter = 0.001 m
Physical_Unit "°C", (0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1.0, -273.15)
// "°C" Celsius is (kelvin - 273.15 K)
Physical_Unit "°F", (0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1.8, -
255.372222)
// "°F" Fahrenheit is 1.8 · (kelvin – 255.3722 kelvin)
Physical_Unit "ppm/°C", (0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1e-6, 0)
// "ppm/°C" 10-6 · (kelvin^-1)
Physical_Unit "microstrain", (1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1e-6,
0)
// "microstrain" 10-6 · (meters/meters)
Physical_Unit "pH", (2, 0, 0, -3, 0, 0, 0, 0, 1, 0, -1.0, -3)
// "pH" -1.0 · (log10[mole/meter^3] - 3) = -log10(10^-3 · mole/meter^3)
= -log10(mole/liter)
Physical_Unit "%/decade", (6, 0, 0, 0, 0, -1, 0, 0, 0, 0, 100, 0)
// "%/decade" 100 / (log10[second^-1/second^-1] + 0) = 100 /
log10(Hz/Hz)
```

۲-۵-٤-۸- اختصاص دهی اختیاری

به جای خواندن یک مقدار ویژگی از TEDS، این امکان پذیر است که یک مقدار را به شکل مستقیم در الگو اختصاص دهیم. این امر با افزودن علامت مساوی "=" و مقدار بعد از پارامترهای ویژگی انجام می‌شود. دستور این مورد در بندهای پیشین آمده است. مقدار باید با نوع داده مشخص شده همخوانی داشته باشد. مقدار به شکل یکی از موارد زیر مشخص می‌شود:

- عدد یا عدد صحیح دسیمال
- عدد باینری با استفاده از پیشوند "0b"
- عدد هگزادسیمال با استفاده از پیشوند "0x"
- رشته داخل کوتیشن

بدون در نظر گرفتن تعداد بیت‌های فیلد، یک فرمان ویژگی با تخصیص دهی اختیاری نباید هیچ بیتی را از TEDS بخواند.

DOI: 10.30503/nripress.2020.296

مثال: این فرمان ویژگی می‌تواند در یک الگو باشد:

```
%CalInitials, "person who calibrated", CAL, 21, ASCII, "s", "" =
"CHJ"
```

۲-٤-٥-۹- عملکردهای گسترش یافته با استفاده زیرویژگی‌ها

عملکرد گسترش یافته مکانیزمی را برای مشخص کردن اطلاعات درباره عملکرد یک ترانسدیوسر با قابلیت ماورا اندازه‌گیری یا تحریک ساده را فراهم می‌کند. این به شکل استفاده از زیرویژگی‌ها که اصلاح‌کننده ویژگی‌ها هستند پیاده سازی می شوند. اغلب موارد این توابع کنترل سویچها در ترانسدیوسر و یا فعال و غیر فعال کردن ویژگی‌های مرتبط را انجام می‌دهند.

زیرویژگی‌ها از نظر نگارشی بوسیله یک براکت مربعی [] بعد از برچسب ویژگی مشخص می‌شود. یک تابع کنترلی بوسیله یک مجموعه‌ای از زیرویژگی‌ها تعریف می‌شود که باید بوسیله زیر ویژگی [Initialize] شروع شود و زیرویژگی [CtrlFunctionMask] را لزوماً شامل خواهد شد. هر ویژگی ممکن است با زیرویژگی‌ها اصلاح شود. ممکن است بیشتر از یک تابع کنترلی با همان برچسب ویژگی وجود داشته باشد.

۲-٥-٤-۹-۱- تعریفها

در زیر تعریفهایی که در ۲-۴-۵-۹ استفاده می‌شوند آمده است.

- **عملگرهای بیتی کاربردی با نوع داه Bit_Digit**

جدول ۲-۱۵- جدول توابع منطقی

SET	0	1	x
0	1	1	0
1	1	1	0
x	0	0	0

XOR	0	1	x
0	0	1	0
1	1	0	0
x	0	0	0

XNOR	0	1	x
0	1	0	1
1	0	1	1
X	1	1	1

- **بیت‌های توابع**

بیت‌های توابع، بیت‌های در صفحه‌های ویژه یک قطعه با کد خانواده "حافظه عمومی" بوده و برای این موارد توابع این بیت‌ها در الگوی ترانسدیوسر تعریف شده‌اند.

- **رجیستر توابع (FR)**

FR یک رجیستر باینری است که طولش برابر تعداد بیت‌های مرتبط با سوییچ‌های سخت افزاری یا بیت‌های تابع هست که بوسیله کد خانواده تعریف شده و بوسیله لیست نود ترانسدیوسر IEEE 1451.4 رتبه بندی شده است. رجیستر باید شامل وضعیت جاری تمام موجودیتهای مستقل در ترانسدیوسر IEEE 1451.4 باشد. FR تنها از نوع داده BitBin بوده که فقط برای "1" و "0" دارای اعتبار است. با ارزشترین بیت سمت چپ Bit_Digit رجیستر باید مطابق با مقدار اولین موجودیت تابع (FR1) در سیستم باشد. برای نمونه، یک FR با یک مقدار "1100"

DOI: 10.30503/nripress.2020.296

نمایانگر یک سیستم با چهار موجودیت FR که FR1 و FR2 دارای یک مقدار "1" بوده و باقیمانده موجودیت FR (FR3 و FR4) دارای مقدار "0" هستند.

- **ویژگی کنترل (control property)**

یک ویژگی کنترل به شکل مستقیم یک یا چندین موجودیت FR را در سخت افزار مورد نظر کنترل می‌کند. یک ویژگی کنترل دارای زیرویژگی [Function] است. ویژگی‌های کنترل با موجودیتهای FR مقادیر ویژگیشان مرتبط است. آن می‌تواند یک لیستی از دو مقدار باشد زمانیکه تابع [FunctionType] 0 است، یک "checkmark" را تعریف می‌کند. یک مقدار ویژگی تمام سوییچ‌ها را، همچنین آنهایی که کنترل نمی‌شوند را مشخص می‌کند. اینها با یک "x" نشان داده می‌شوند.

برای نمونه، در یک سیستم با ویژگی کنترل غیر فعال و مقادیر ویژگی کنترل داده شده با:

State 1: "xx0x"
State 2: "0010"

مقادیر FR به غیر از آنهایی که در State 1 و State 2 تعریف شده‌اند قابل اعمال کردن بوسیله ویژگی کنترل غیر فعال نخواهند بود.

یک ویژگی کنترل زمانی انتخاب می‌شود که یک اپراتور XNOR بین مقدارش و FR نتیجه اش برای همه Bit_Digitها "1" شود. این بر این دلالت دارد که ویژگی تابع به شکل منحصر به فردی بوسیله FR با در نظر گرفتن [FunctionType] (جدول ۲-۱۵) تعریف می‌شود.

زمانیکه کاربر تنظیم تابع کنترلی را تغییر می‌دهد، FR که به شکل بیتی است مقادیر ویژگی کنترل را به استثنای موردی که آنها "x" هستند تخصیص می‌دهد و مرحله زیر برای همه توابع کنترلی بعدا در الگو تعریف می‌شوند:

۱- یک ویژگی کنترلی، زمانیکه یک XOR باینری بیتی بین هر مقدار ویژگی کنترل انتخاب شده قبلی و این مقدار ویژگی کنترلی که به ازای همه Bit_Digit ها مقدارش "1" باشد، باید غیر فعال شود.

۲- یک ویژگی کنترلی، زمانیکه یک XNOR بین مقدارش و FR نتیجه اش برای همه Bit_Digit ها "1" باشد، باید مقداردهی (set) شود. در غیر اینصورت نباید مقداردهی شود.

اگر هر تغییری در FR اعمال شود، سخت افزار متناظر باید مطابق آن به روز شود.

برای مورد خاص یک زیرویژگی تابع، فیلد توصیفی در فرمان ویژگی باید با فیلد توصیفی در مقدار تابع فرمان ویژگی دیگر مرتبط باشد. آنگونه که در مثال نشان داده شده است، فیلد توصیفی می‌تواند با یک برچسب ویژگی و زیرویژگی مرتبط "Register mask" جایگزین شود. این نشان می‌دهد فیلد توصیفی ویژگی مرتبط که با مقدار فرمت شده صحیح خوانده شده از فرمان ویژگی مرتبط ترکیب شده است، باید در صفحه نمایش کاربر استفاده شود. مثال فرمان ویژگی زیر به متن نمایش داده شده در مثال منجر می‌شود.

```
%Sens[function], %Sens@Ref["100x"],  USR, 8, BitBin, "","" = "1000"
%Sens[function], %Sens@Ref["010x"],  USR, 8, BitBin, "","" = "0100"

%Sens@Ref ["100x"], "Sensitivity @ F ref", CAL, 16, ConRelRes, 100E-6,
0.0001, "rp", "V/Pa"
%Sens@Ref ["010x"], "Sensitivity @ F ref", CAL, 16, ConRelRes, 100E-6,
0.0001, "rp", "V/Pa"
```

این منجر به یک تابع کنترل متنی مانند مورد زیر (وابسته به مقادیر در TEDS) می‌شود:

```
o Sensitivity @ F ref   200mV/Pa
x Sensitivity @ F ref   20mV/Pa
```

- **ویژگی مرتبط با FR**

DOI: 10.30503/nripress.2020.296

یک ویژگی می‌تواند وابسته به مقدار جاری FR باشد. این آن امر از طریق یک نقاب رجیستری[1] مرتبط میشود. نقاب رجیستری از نوع داده BitBin با یک طول برابر FR است. نقاب رجیستری مشخص میکند چه زمانی این ویژگی فعال شود. آن می‌تواند شامل کاراکتر جایگزین شونده باشد که بوسیله یک "x" نشان داده می‌شود. این ویژگی‌ها به لحاظ دستور زبانی داخل یک براکت [ ] نشان داده شده و بعد از برچسب ویژگی مانند زیر ویژگی هستند. ویژگی زمانی می‌تواند فعال شود که یک XNOR بیتی بین نقاب رجیستری و FR به مقدار "1" برای همه Bit_Digits منتج شود، در غیر اینصورت از غی فعال می‌شود. اگر یک لیست جداشده با کاما بکار گرفته شود، زمانی فعال می شود که نقاب رجیستری آن فعال شود. زمانیکه یک تغییر در FR بوسیله ابزار ویژه سیستمی تشخیص داده شد ویژگی‌های مرتبط می‌تواند به روز شود. زمانیکه یک ویژگی از کار انداخته شد (disabled) آن نمیتواند به هر روشی استفاده شود.
برای نمونه، یک سیستم با ویژگی مرتبط

```
%gain["100x,0x00"], "Gain", Cal, 12, ConRes, -40, 0.025, "0.0p",
"dB"
```

ویژگی مرتبط را برای مقادیر FR: 1000، 1001، 0000، 0100 توانا خواهد کرد.

۲-۵-٤-۹-۲- لیست زیر ویژگی

لیست زیرویژگی‌های معتبر در جدول(۲-۱۶) آمده است.

جدول ۲-۱٦- زیرویژگی‌ها

[1] Register Mask

DOI: 10.30503/nripress.2020.296

Subproperty Accessor	Description
`Default`	The default value for the function. (Not the default value of the property.) This is either the value of the function upon start-up or, if the function must be initialized, the value to initialize the function to. The `Default` value is of datatype `BitBin`. Shall not be present if `%DefaultFR` is defined. If not present and default is not defined by `%DefaultFR`, then Default = 0.
`Initialize`	A single bit that indicates whether this function shall be initialized upon start-up. It also initiates a new Control Function. The `Initialize` value is of datatype `UNINT`. 0: Do not initialize 1: Initialize Shall be present.
`CtrlFunctionMask`	The mask on the FR indicating which bits of the FR are associated with the Control Function. The `CtrlFunctionMask` value is of datatype `BitBin`. The mask shall be a bit-wise SET of all the Control Property values defined in the Control Function. Shall be present.

NRI

DOI: 10.30503/nripress.2020.296

جدول ۲-۱۶- زیرویژگی‌ها (ادامه)

Subproperty Accessor	Description
ReadWrite	Two bits indicating what level of programmability the function has. The ability to read means that the current setting can be electronically read from the transducer. The ability to write means that the setting can be changed electronically. The ReadWrite value is of datatype UNINT. 0: Read and Write 1: Can not read or write (physical switch only) 2: Read only 3: Write only If not present, then assume can read and write.
FunctionType	Indicates how many options can be, or shall be, selected at one time. (The widget for displaying these function types is left to the application program.) The FunctionType value is of datatype UNINT. 0: Can have either 0 or 1 settings set. 1: Shall have exactly 1 setting set. 2: Can have any number of settings set including none. 3: Can have any number of settings set but shall have at least one. If not present, then assume there shall be exactly 1 setting set.
Function	Controls the behavior of the Extended Functionality. A Control Property has the subproperty [Function]. Control Properties are associated to FR entities by their property values. The property value is of datatype BitBin defining the FR setting that applies to a given function state.
"Register mask"	This is the one subproperty where the subproperty name, "Register mask," is not actually used in a template. Instead, a Register mask, or a list of register masks is provided in the form of a value of datatype BitBin. The property shall be enabled when a bitwise XNOR between the Registermask and the FR results in "1" for all Bit_Digits, otherwise disabled. If a list is applied, it shall be enabled if one Register mask in the list applies.

NRI

مثال:

موارد آمده در زیر یک مثال جامع از توانایی تعریف تابع گسترش یافته است. که تا پایان این زیر بخش ادامه می‌یابد. سیستم شرح داده شده در شکل(۲-۱۱) را در نظر بگیرید که دو مرحله دارد: یک فیلتر با گین و یک مالتی پلکسر. وضعیتهای زیر که در سه تابع گروه بندی شده است در زمان طراحی تعریف شده‌اند:

The gain:

Gain 0dB (default value)

DOI: 10.30503/nripress.2020.296

Gain 10 db (amp 1: +10db, amp2: +0dB)

Gain 20 dB (amp 1: +10db, amp2: +10dB)

The filter:

Filter OFF (default value)

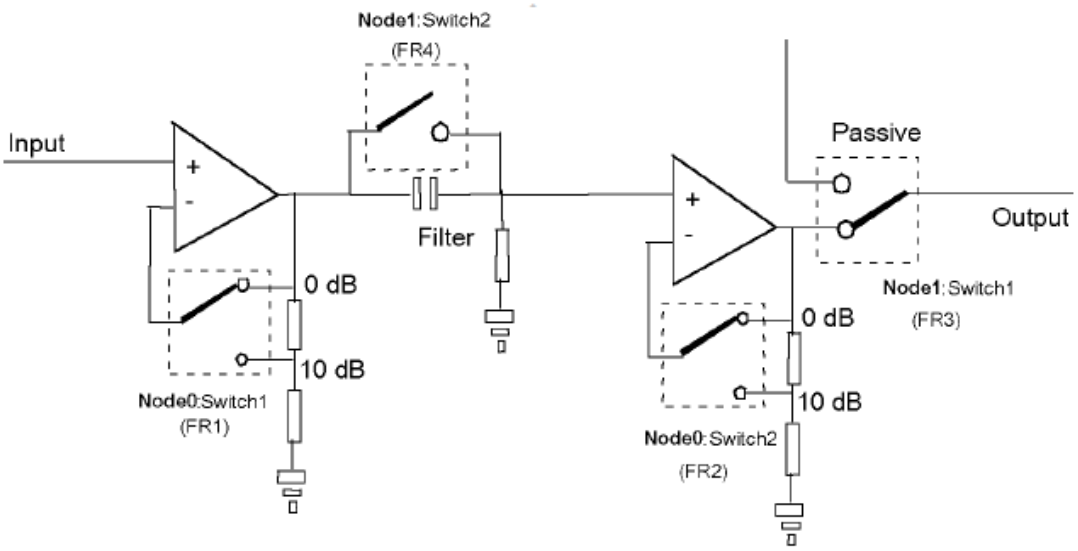
Filter ON (only valid for gain>0db)

The multiplexer:

Passive-off (default value)

Passive-on

در شرایط بهره (gain) 20dB+، یک محدودیت در پهنای باند اتفاق می‌افتد، این امر به یک فیلتر پایین گذر منتج می‌شود. تقویت‌کننده و فیلتر زمانیکه وسیله در مود پسیو (passive) است همچنان در وضعیت معتبر می‌باشد.

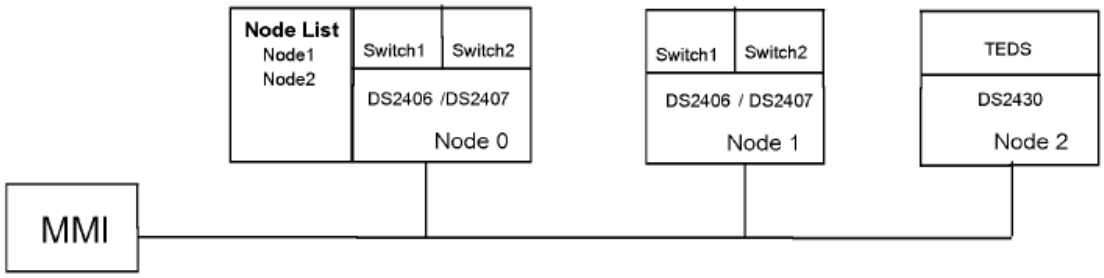


شکل ۲-۱۱- فیلتر با گین و سیستم مالتی پلکسر

یک پیکربندی محتمل نود برای کنترل تقویت کننده با فیلتر در شکل(۲-۱۱) میتواند به شکل(۲-۱۲) توصیف شود. یک سویچ به عنوان یک وسیله سخت افزاری تعریف شده است که می‌تواند مقادیر 0 و 1 را داشته باشد. سوییچ‌ها اجزای عملکردی یک نود سخت افزاری مانند DS2406/DS2407 هستند و به شکل کامل بوسیله فایلهای تعریف شده XML استاندارد IEEE1451.4 توصیف شده‌اند. PIO-A و PIO-B در یک DS2406/2407 نمونه‌هایی از سوییچ‌ها هستند. FR به عنوان یک سوییچ سخت افزاری مجرد تعریف شده است. بحث TDL فقط در مورد موجودیتهای FR هست.

DOI: 10.30503/nripress.2020.296

دقت کنید که Node0 شامل یک لیست نود در نقشه حافظه خود است که شامل نودهایی است که به تقویت کننده متعلق است.



شکل ۲-۱۲- پیکربندی نود فیلتر با گین و مالتی پلکسر

یک تابع گین FR1، FR2 به شکل FR هایی که در ارتباط با گینهای 0 dB، 10 dB، و 20 dB هستند تعریف می‌کند.

یک تابع فیلتر FR4 را به شکل یک FR ای که در ارتباط با فیلتر است تعریف می‌کند.

یک تابع پسیو (passive) یک تابع در ارتباط با FR3 تعریف می‌شود که اگر مالتی پلکسر در حالت passive on یا passive off باشد.کنترل مالتی پلکسر در ارتباط با ویژگی‌های passive تعریف شده در الگوی زیر است.

یک الگوی TDL توصیف کننده تقویت کننده در شکل ۲-۷ در زیر آمده است:

DOI: 10.30503/nripress.2020.296

```
// function control definitions
%passive[Initialize], “Always initialize this”, ID, 1, UnInt, “”,
“” = 1
%passive[CtrlFunctionMask], “Control Function Mask”,ID, 8, BitBin,
“”, “” = “1111”
%passive[ReadWrite], “Write only”, ID, 2, UnInt, “”, “” = 3
%passive[Default], “Default not passive”, ID, 8, BitBin, “”, “” =
“xx0x”
%passive[FunctionType], “passive control type”, ID, 2, unint,
“”,“” = 0//checkmark
%passive[Function], "Passive mode", USR, 10, BitBin, “”,“” =
“xx0x,0010”

%gain[Initialize], “Always initialize this”, ID, 1, UnInt, “”, “”
= 1
%gain[CtrlFunctionMask], “Control Function Mask”,ID, 8, BitBin,
“”, “” = “1101”
%gain[ReadWrite], “Write only”, ID, 2, UnInt, “”, “” = 3
%gain[Default], “Default gain: 0dB”, ID, 8, BitBin, “”,“” = “00x0”
%gain[FunctionType], “Gain control type”, ID, 2, unint, “”, “” =
1//One Exactly

%gain[Function], “This is 20dB”, USR, 8, BitBin, “”,“” = “11xx”
//State1
%gain[Function], "This is 10dB", USR, 8, BitBin, “”,“” = “01xx”
//State2
%gain[Function], "This is  0dB", USR, 8, BitBin, “”,“” = “00x0”
//State3

%filter[Initialize], “Always initialize this”, ID, 1, UnInt, “”,
“” = 1
%filter[CtrlFunctionMask], “Control Function Mask”,ID, 8, BitBin,
“”, “” = “0001”
%filter[ReadWrite], “Write only”, ID, 2, UnInt, “”, “” = 3
%filter[Default], “Default disabled”, ID, 8, BitBin, “”, “” =
“xxx0”
%filter[FunctionType], “Control type”, ID, 2, unint, “”,“” =


0//checkmark
%filter[Function], "HP filter", USR, 10, BitBin, “”,“” =
“xxx0,xxx1”
%Passive, "Supports multiplexer via no gain selected", Cal, 1,
UNINT, “”,“”
%Gain[“1x0x”], "This is stage 1. 10dB", Cal, 12, ConRes, -40,
0.025, "0.000p", "dB"
%Gain[“0x0x”], "This is stage 1.  0dB", Cal, 12, ConRes, -40,
0.025, "0.000p", "dB" = 0
%Gain[“x10x”], "This is stage 2. 10dB", Cal, 12, ConRes, -40,
0.025, "0.000p", "dB"
%Gain[“x00x”], "This is stage 2. 0dB", Cal, 12, ConRes, -40,
0.025, "0.000p", "dB" = 0
%TF_HP_S[“xx00”], "High Pass filter off", Cal, 1, unint,
"0.000p", "Hz" = 0
%TF_HP_S[“xx01”], "High Pass filter on (only valid for
gain>0dB)", Cal, 12, ConRelRes, 0.01, 0.001,"0.000p", "Hz"
%TF_SP[“110x”], "Bandwidth limiting at +20dB", Cal, 5,
ConRelRes, 10, 0.01,"0.000p", "Hz"
```

DOI: 10.30503/nripress.2020.296

مورد دیگری از زیرویژگی پیش فرض، پیش فرض می‌تواند بوسیله ویژگی %DefaultFR به شکلی تعریف شود که تنظیم کل FR باید در آن تعریف شده باشد. تعریف این در وسیله فشرده تر از استفاده از زیر ویژگی است به دلیل اینکه آن بدون در نظر گرفتن کارکترهای عام (wildcard) تعریف می‌شود.
برای نمونه:

```
%DefaultFR, "Default Function setting", Cal, 4, UNINT, "",""
```

۲-۵-٤-۱۰- برخی ویژگی‌های خاص

برای این ویژگی‌ها نیاز به اطلاعات بیشتر برای استفاده صحیح وجود دارد.

۲-۵-٤-۱۰-۱-ویژگی‌های تعریف شده توسط کارخانه %MDEF

سازنده‌ها یا دیگر کاربران ممکن است ویژگی‌های خودشان را با استفاده از پیشوند %MDEF تعریف کنند. بدین ترتیب ممکن است یک سازنده یک ویژگی را با ضمیمه کردن یک رشته کاراکتر بدون کوتیشن به پیشوند %MDEF ایجاد کند. این به سازنده اجازه الحاقیه‌های ویژه را به لیست ویژگی‌ها می‌دهد. این ویژگی‌ها در حالت کلی بوسیله تجزیه کننده‌های الگو قابل تشخیص نخواهند بود، بنابراین فقط باید زمانی استفاده شوند که ضروری هستند. تجزیه کننده‌های الگو باید قادر به شکستن این ویژگی‌ها و خواندن تعداد مشخص بیت‌ها از TEDS باشند. اگر تجزیه کننده نتواند ویژگی را تجزیه کند، ممکن است آن را نادیده بگیرد.
مثال : یک ویژگی ممکن است به شکل زیر تعریف شود:

```
%MDEF_myprop, “Special manufacturer property”, USR, 6, UNINT, “”, “”
```

7.4.10.2 Appended TEDS properties

An Appended TEDS is a TEDS not located on the transducer. It provides a mechanism for defining properties without memory size constraints.

۲-۵-٤-۱۰-۲- ویژگی‌های TEDS ضمیمه شده

یک TEDS ضمیمه شده یک TEDS ای هست که داخل ترانسدیوسر جای داده نشده است. آن یک مکانیزمی را برای تعریف ویژگی بدون محدودیت در فضای حافظه فراهم می‌کند.

- %Appended TEDS

اگر یک TEDS ضمیمه شده برای این ترانسدیوسر وجود دارد، این ویژگی مشخص می‌شود. TEDS باید در فرمت الگوی Embedded باشد.در زیر قرارداد نامگذاری فایل برای TEDS ضمیمه شده می‌باشد. استفاده از ”+“ در اینجا به معنی الحاق رشته است.

```
“IEEE1451_4_” + Manufacturer Code + “_” + Model Number + “_” + Version
Letter + “_” + Serial Number + “.ted”
```

اگر TEDS ضمیمه شده، داده محصول یا داده‌ای که مربوط به یک شماره سریال خاص است را توصیف کند، سپس شماره سریال در فایل با ”PD“ جایگزین می‌شود.

```
“IEEE1451_4_” + Manufacturer Code + “_” + Model Number + “_” + Version
Letter + “_” + “PD.ted”
```

مثال:

DOI: 10.30503/nripress.2020.296

این فرمان ویژگی می‌تواند در TEDS باشد:

```
%Appended_TEDS, "Appended TEDS", USR, 1, YesNo, "", ""
```

If the bit read indicates that an Appended TEDS exists, then the associated filename might be:

اگر بیت خوانده شده نشان دهنده این باشد که یک TEDS ضمیمه شده وجود دارد، پس نام فایل مرتبط ممکن است به شکل زیر باشد:

```
"IEEE1451_4_17_4394_A_0_3235.ted"
```

که اعداد معرف موارد زیر است:

```
ManufacturerID = 17
Model Number   = 4394
Version Letter = a
Version Number = 0
Serial Number  = 3235
```

اگر TEDS ضمیمه شده شامل اطلاعات عمومی باشد، نام فایل به شکل زیر میتواند باشد:

```
"IEEE1451_4_17_4394_A_0_PD.ted"
```

7.4.10.2.2 %Appended_TEDS_location

This property provides a location for the Appended TEDS if it exists. Most likely this would read in an URL or a directory path.

- %Appended_TEDS_location

این ویژگی یک مکانی را برای TEDS ضمیمه شده اگر وجود داشته باشد، فراهم می‌کند. به احتمال زیاد این یک URL یا یک مسیر دایرکتوری را می‌خواند.

مثال:

این فرمان ویژگی می‌تواند در یک فرمان TEDS باشد.

```
%Appended_TEDS_location, "Appended TEDS location", USR, 5,
String5, "", ""
```

ممکن است چیزی مشابه مورد زیر از TEDS خوانده شود:

```
www.ieee1451.org
```

۲-۵-٤-۱۱- فرمت الگوی Embedded

یک الگوی Embedded یک TEDS خود توصیف بوده بطوریکه شامل توصیف الگو است.

به این ترتیب، TEDS شامل فرمانهای TDL قانونی است، که تمام ویژگی‌ها به شکل شفاف مقداردهی شده‌اند. این یک بخش متکی به خود TEDS است که نیازی به الگوی از پیش تعریف شده ندارد.

هر خط با carriage return و line feed خاتمه می‌یابد.

قوانین زیر برای داده‌ها در این بخش از TEDS باید رعایت شود:

۱. هر دستور ویژگی که
۲. دستورات زیر
   - دستورات شروع و خاتمه الگو
   - دستور نسخه TDL

DOI: 10.30503/nripress.2020.296

- UGID
- Comments
- Abstract
- SelectCase و case
- StructArray
- Align

نباید استفاده شوند

۳. ویژگی‌های زیر

- %user
- %xml
- %EmbTpl

نباید بکار روند

۲-۵-٤-۱۱-۱- %EMBTPL

ویژگی %EMBTPL به یک بخشی از TEDS اجازه می‌دهد که در فرمت الگوی Embedded باشد. در ساختار زبان فرمان %EMBTPL، اگر <number_of_bits> برای <data_type> 0 باشد، پس مابقی TEDS باید به شکل فرمانهای TDL از نوع ASCII خوانده شود. در غیر اینصورت، <number_of_bits> میتواند نشان دهنده تعداد بیت‌های اختصاص داده شده برای الگوی Embedded باشد. اگر همه حافظه اختصاص داده شده استفاده نشده است الگوی Embedded باید به صفر ختم شود.

مثال:

این مثال نشان می‌دهد که بخش باقیمانده TEDS در فرمت الگوی Embedded است.

```
%EMBTPL, "TDL Commands", Usr, 0, ASCII, "", ""
```

۲-۵-٤-۱۲- شکل آزاد ویژگی‌های TEDS

ویژگی‌های متعدد به یک بخش TEDS اجازه می‌دهد که در یک شکلی باشد که به شکل سختگیرانه بوسیله فرمانها در یک الگو تعریف نشده باشند. اینها برای انعطاف پذیری فراهم شده‌اند، اما باید با احتیاط استفاده شوند از آنجاییکه همه تجزیه کننده‌ها قادر به پشتیبانی یا تفسیر داده فراهم شده در این سبک نخواهند بود.

افزون بر آن، این فرمهای ذخیره سازی داده در TEDS عموما فضای حافظه زیادی را اشغال می‌کنند.

هشدار!

داده در این نواحی شکل آزاد بوسیله کاربر تعریف شده‌اند و با ریسک خود کاربر به کار گرفته می‌شود!

۲-۵-٤-۱۲-۱- %user

ویژگی %user به یک بخش از TEDS اجازه می‌دهد که در فرمت تعریف شده توسط کاربر باشند. اگر طول بیت نوع داده 0 باشد، پس مابقی TEDS در فرمت تعریف شده توسط کاربر است. در غیر اینصورت، طول بیت نشان دهنده کل طول فیلد تعریف شده توسط کاربر است.

مثال:

DOI: 10.30503/nripress.2020.296

این مثال نشان می‌دهد که بخش باقیمانده TEDS در یک فرمت تعریف شده توسط کاربر با کاراکترهای ASCII هفت بیتی است.

```
%user, "User data (ASCII 7-bit)",  Usr, 0, ASCII, “”, “”
```

۲-۵-٤-۱۲-۲- %XML

ویژگی %XML به یک بخشی از TEDS اجازه می‌دهد که در فرم XML مبتنی به ASCII باشد. دقت کنید که استفاده از این نیازمند یک XML DTD یا Schema خارجی است. در ساختار زبانی فرمان %XML اگر <number_of_bits> برای <data_type> 0 باشد، پس مابقی TEDS باید به شکل ASCII XML خوانده شود. در غیر اینصورت <number_of_bits> می‌تواند نشان دهنده مقدار کل واقعی بیت‌هایی که باید خوانده شود باشد. در مورد آخر باید مراقب بود چرا که اندازه یک فایل XML فقط بعد از نوشتن کامل آن تعیین می‌شود.

مثال:

این مثال نشان می‌دهد که بخش باقیمانده TEDS در فرمت XML تعریف شده توسط کاربر است.

```
%XML, "XML Commands",  Usr, 0, ASCII, “”, “”
```

۲-۵-٤-۱۳- رخدادهای چند باره از یک ویژگی

زمانیکه یک ویژگی در یک الگو بیشتر از یک بار اتفاق می‌افتد، اینگونه فرض می‌شود که ویژگی نیاز به یک آرایه‌ای از داده دارد. بنابراین، مقادیر رخدادهای چندباره از یک ویژگی یکسان به شکل ترتیبی در یک آرایه ذخیره می‌شوند. این همچنین درباره ویژگی‌هایی که بوسیله زیرویژگی‌ها اصلاح می‌شوند نیز صادق است. یک ویژگی اصلاح شده به عنوان یک ویژگی منحصر به فرد برای خودش در نظر گرفته می‌شود و ممکن است نیاز به یک آرایه برای ذخیره‌سازی مقادیر مرتبط داشته باشد. مهم است که ترتیب نیز حفظ شود.

این بدان معنی است که آرایه باید همان ترتیبی را برای ویژگی‌ها پشتیبانی کند که آنها در الگو پیدا می‌شوند.

اگر ارتباط قویتری از ویژگی‌ها نیاز باشد، توصیه می‌شود که آنها به وسیله فرمان StructArray گروهبندی شوند.

دقت شود که ویژگی‌هایی که به شکل یکسان نشانه دار شده‌اند ممکن است در الگوهای مختلف اما به هم الحاق شده باشند. اینگونه ویژگی‌ها باید به شکل جداگانه که با الگوهای ویژه در ارتباط هستند، برخورد شوند.

۲-۵-٤-۱٤- اداره کردن ویژگی‌های غیر رسمی

برای دلایل مختلف، یک تجزیه کننده TDL ممکن است یک ویژگی را تشخیص ندهد. برخی از دلایل عبارتند از:

۱- ویژگی جدید زمانیکه تجزیه کننده در حال ایجاد شدن است، افزوده شود.

۲- ویژگی تعریف شده توسط سازنده باشد (%MDEF).

۳- تجزیه کننده همه ویژگی‌ها را پشتیبانی نکند.

زمانیکه این اتفاق بیافتد، اگر تجزیه کننده TDL در خدمت یک برنامه دیگری باشد، سپس تجزیه کننده باید برچسب ویژگی و مقدار مرتبط با آن که از TEDS خوانده شده را به برنامه منتقل می‌کند.هر استفاده‌ای از اطلاعات برای برنامه است که تشخیص بدهد. یک سناریوی محتمل این است که نرم افزار مورد نظر به سادگی اطلاعات را به سود کاربر نمایش دهد.

DOI: 10.30503/nripress.2020.296

۲-٦- ویژگی‌های اینترفیس ترانسدیوسر در مود ترکیبی (MMI)

۲-۶-۱- مقدمه

MMI استاندارد IEEE1451.4 تبادل پایدار یک سیگنال ترانسدیوسر و سیگنال داده TEDS بین یک ترانسدیوسر و یک NCAP یا سیستم جمع آوری داده را تضمین می‌کند.

۲-٦-۱-۱- تعریف MMI

MMI استاندارد IEEE1451.4 ارتباطی برای هر دو سیگنال‌های آنالوگ و داده TEDS دیجیتال، بین یک ترانسدیوسر و یک NCAP یا سیستم جمع آور داده است. پرتکلها، زمان بندی، و مشخصه‌های الکتریکی در این استاندارد دو کلاس MMI را تعریف می‌کند، که اجازه می‌دهد سیگنال‌های آنالوگ و دیجیتال یا به ترتیب یک ارتباط را به اشتراک بگذارند یا به شکل همزمان روی اتصالهای جداگانه بکار روند.
برای این استاندارد، IEEE1451.4-2004، یک اتصال با دو سیم هادی تعریف می‌شود.
حافظه TEDS در یک وسیله دو ترمینالی، به نام نود جای میگیرد، این قطعه یک پرتکل تبادل داده سریال، جند اتصالی را که در بخش‌های پیشین شرح داده شده‌است پشتیبانی و به تغذیه و داده اجازه می‌دهد یک خط را به اشتراک گذاشته و از خط دیگر به عنوان خط برگشت استفاده شود. نودها از درایور خط کلکتور باز استفاده کرده و به شکل پارازیتی از خط داده تغذیه می‌شوند. پیاده سازی موفقیت آمیز MMI نیاز دارد که کاربران جزئیات این پرتکل را که در بخش‌های بعدی گفته شده با دقت مطالعه کنند. پرتکل تبادل داده‌ای که در این استاندارد استفاده شده است مطابق با پروتکلی است که در قطعه حافظه DS2430A بکار رفته است. لذا رجوع به برگه اطلاعاتی این قطعه و مستند "1- Wire Master Device Configuration" [2]می تواند کمک زیادی کند.

۲-٦-۱-۲- عملکرد MMI

یک MMI کلاس ۱، ارسال یک سیگنال ترانسدیوسر یا داده TEDS دیجیتال را به شکل ترتیبی روی یک تک اتصال تعریف خواهد کرد. کلاس ۱ یک سیگنال ترانسدیوسر و تغذیه، یا داده TEDS دیجیتال و تغذیه را اجازه خواهد داد تا از همان اتصال استفاده کند. در یک MMI کلاس ۱، سطوح منطقی دیجیتال صفر همان صفر ولت و سطح دیجیتال یک ۵- ولت تعریف شده است. در حالت کلی MMI کلاس ۱ از منطق معکوس استفاده می‌کند و دارای یک اتصال داده مشترک است.
یک MMI کلاس ۲ از اتصالهای جدا برای سیگنال‌های ترانسدیوسر و داده TEDS دیجیتال استفاده خواهد کرد. کلاس ۲ اجازه دسترسی همزمان سیگنال ترانسدیوسر و داده TEDS دیجیتال را داده، و ممکن است یک خط برگشت مشترک را استفاده کند. در یک MMI کلاس ۲، سطح منطقی دیجیتال صفر همان صفر ولت و سطح یک ۵ ولت می باشد. در حالت کلی، MMI کلاس ۲ از منطق مثبت استفاده می‌کند و یک اتصال داده مجزا دارد.

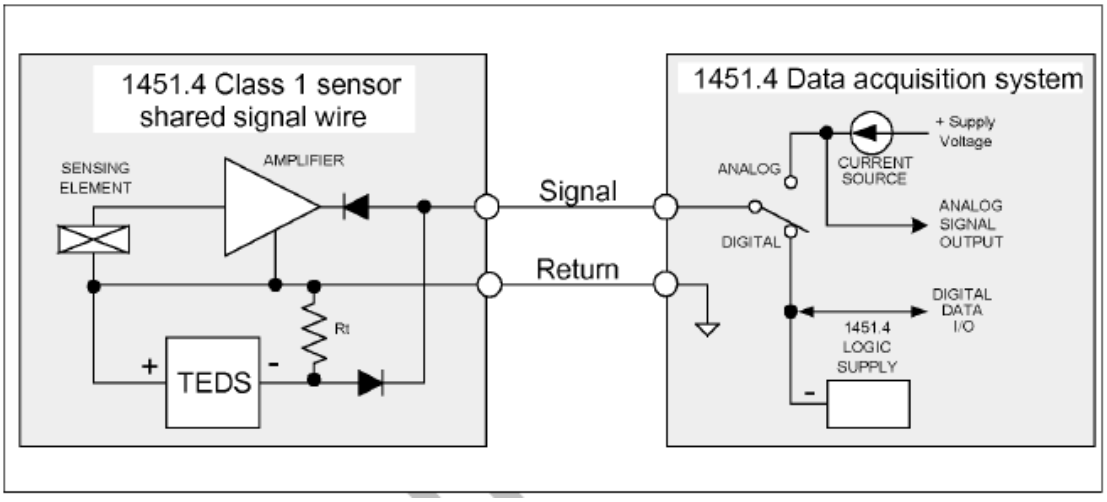
۲-٦-۱-۲-۱- کلاس ۱

DOI: 10.30503/nripress.2020.296

یک MMI کلاس ۱ با استفاده از یک سیم سیگنال و یک سیم بازگشت، انتقال یک سیگنال ترانسدیوسر و یا داده TEDS دیجیتال بصورت غیرهمزمان فراهم ساخته که به عنوان یک باس اینترفیس ۲ سیمه شناخته می شود. پیاده سازی عمومی MMI کلاس ۱ در شکل ۲-۱۳، ۲-۱۴، و ۲-۱۵ شرح داده شده است.

بلوک دیاگرام شکل(۲-۱۳) یک سنسور نوعی که با جریان ثابت تغذیه می‌شود و شامل یک تقویت کننده داخلی است را شرح می‌دهد. سیگنال آنالوگ ولتاژی روی همان خطی که جریان تغذیه به تقویت‌کننده اعمال می‌کند موجود است. یک قطعه حافظه اطلاعات TEDS را در خود دارد. دو دیود هدایت شونده با جریان، یا سوییچ‌های کنترل شونده آنالوگ، دسترسی ترتیبی به تقویت کننده یا حافظه TEDS را با معکوس کردن پلاریته تغذیه جریانی یا با انتخاب سوییچ، فراهم می کنند. استفاده از سوییچهای آنالوگ مزیت حذف افت ولتاژ دیودی را خواهد داشت. منبع جریان مثبت در سیستم جمع آوری داده، تغذیه جریانی را به تقویت کننده از طریق اتصال سیگنال و دیود بالایی زمانیکه سوییچ در وضعیت آنالوگ است، تامین می‌کند.

زمانیکه سوییچ کنترل در وضعیت دیجیتال است، تغذیه منطق دیجیتال از دیود پایینی جاری می شود، تا حافظه عملکرد خود را انجام دهد.

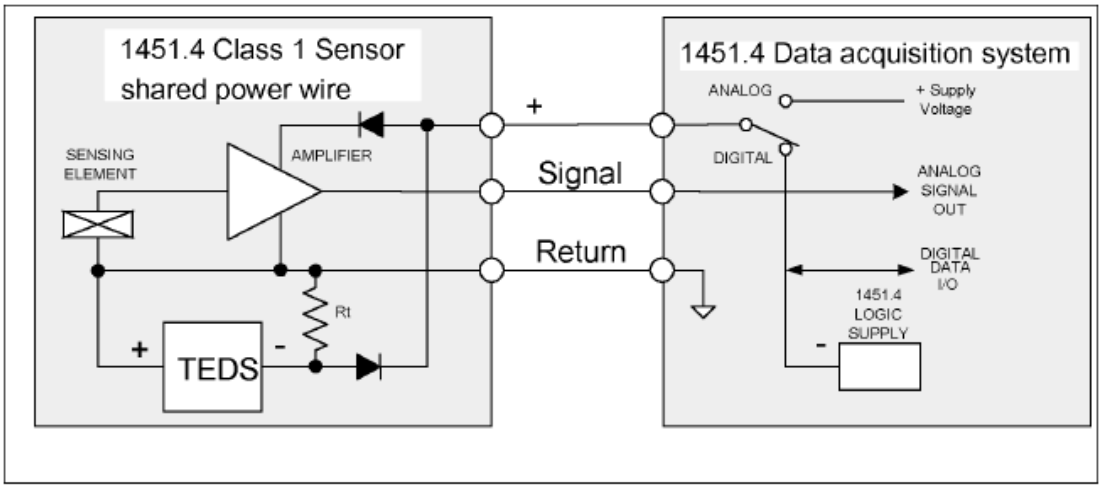


شکل ۲-۱۳- MMI کلاس ۱ استاندارد IEEE 1451.4، سیم سیگنال مشترک

به دلیل اینکه قطعه حافظه برای تشخیص وضعیت 0 ، نیازمند یک کاهش ولتاژ با سرعت مناسب است، لذا به کمک یک مقاومت پایین کشنده Rt این شرایط فراهم می شود. این مقاومت، خازن مدار حافظه و سیم بندی‌ها را با سرعت مناسب که منطبق با مشخصه قطعه است فراهم می سازد. طبیعت پروتکل داده و توانایی آن در کار با تغذیه معکوس اجازه میدهد توابع آنالوگ و دیجیتال از طریق تغییر پلاریته یک سیم را به اشتراک بگذارند.

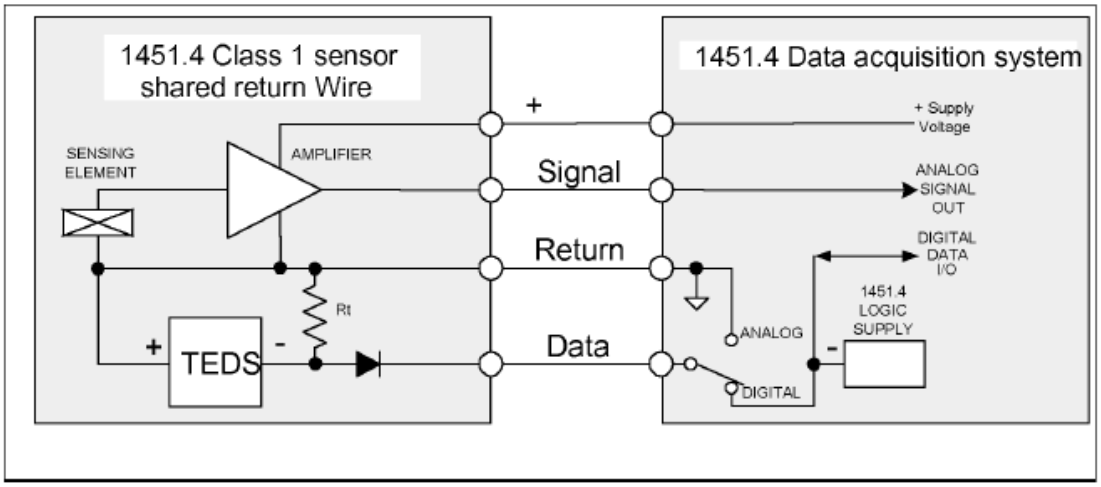
نمونه دیگری از یک MMI کلاس ۱ با سیم تغذیه و زمین مشترک در شکل(۲-۱۴) نشان داده شده است.

DOI: 10.30503/nripress.2020.296



شکل ۲-۱۴- MMI کلاس ۱ استاندارد IEEE 1451.4 ، سیم تغذیه مشترک

شکل(۲-۱۵) پیکربندی یک ترانسدیوسر با یک خط برگشت مشترک آنالوگ و دیجیتال، در روی اتصال برگشت زمین، یا شیلد را نشان می دهد. در این حالت باید احتیاط لازم جهت جلوگیری از اختلال‌های متقابل بین سیگنال‌های آنالوگ و داده TEDS دیجیتال با توجه به افت ولتاژ روی خط برگشت مشترک در نظر گرفته شود، بنابراین در مود آنالوگ، داده دیجیتال را با یک سوییچ به هدف کاهش نویز از کار می‌اندازند.



شکل ۲-۱۵- MMI کلاس ۱ استاندارد IEEE 1451.4، سیم برگشت مشترک

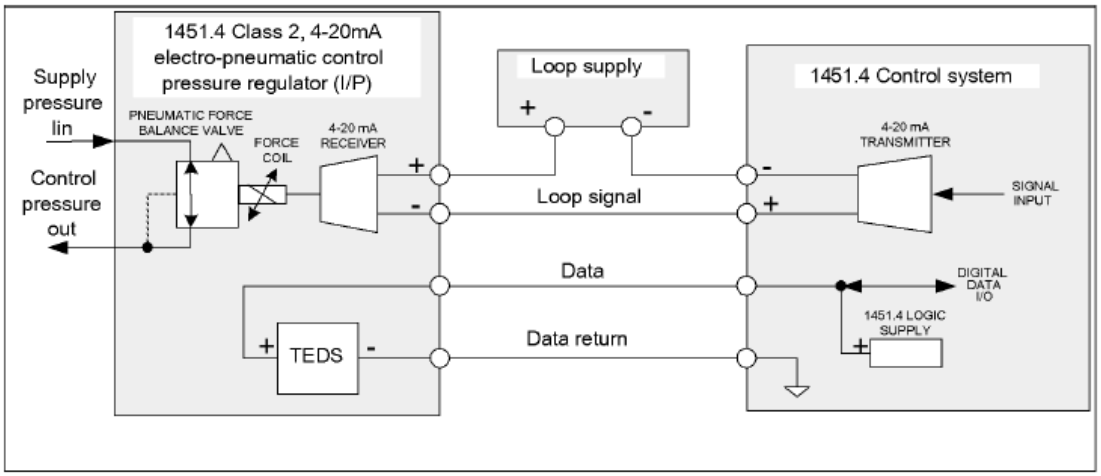
۲-۶-۱-۲-۲- کلاس ۲

اینترفیس کلاس ۲ استاندارد IEEE 1451.4 از اتصالهای جداگانه برای تبادل سیگنال‌های ترانسدیوسر و داده TEDS دیجیتال استفاده خواهد کرد. با اتصالهای جدا، دسترسی به سیگنال و داده به شکل همزمان رخ خواهد داد. اتصال داده دیجیتال کلاس ۲ از سطوح منطقی مثبت استفاده خواهد کرد، لذا منطق صفر با صفر ولت و منطق یک با ۵ ولت نشان داده می شود. برای حفاظت پلاریته ممکن است از یک دیود سوییچینگ و مقاومت terminating استفاده شود. اتصالهای جداگانه کلاس ۲ به ترانسدیوسرهایی که با اتصالهای دوسیمه مشترک کار می کنند مطابقت ندارد.

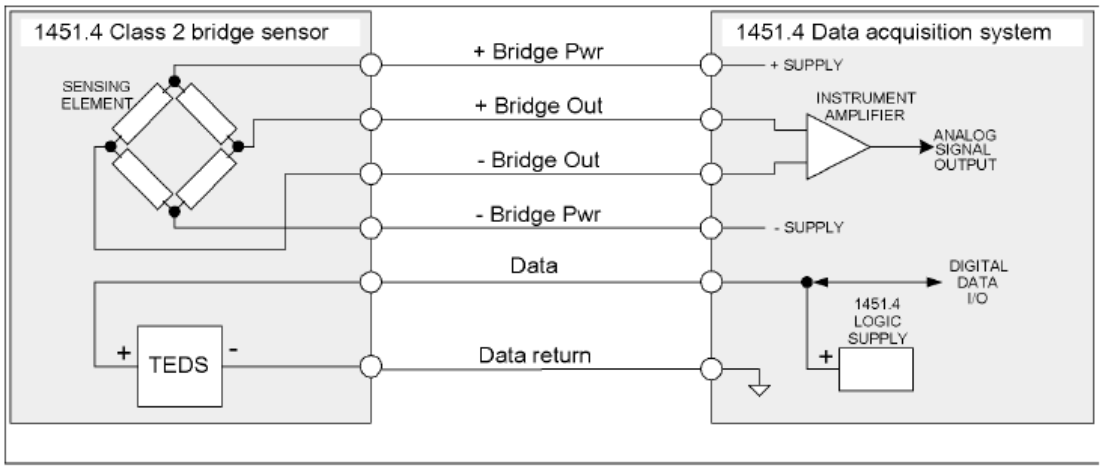
تمام زمانبندی‌های منطقی برای اینترفیس دیجیتال کلاس ۲ در بخش‌های بعدی شرح داده شده و برابر با زمانبندی کلاس ۱ این اینترفیس است.

DOI: 10.30503/nripress.2020.296

نمونه‌هایی از اینترفیسهای کلاس ۲ استاندارد IEEE 1451.4 در شکل ۲-۱۶ و شکل ۲-۱۷ نشان داده شده‌اند. شکل(۲-۱۶) اینترفیس TEDS در یک کاربرد حلقه جریانی 4-20mA تصویر می‌کشد. شکل(۲-۱۷) کاربرد اینترفیس TEDS در یک سنسور پل مقاومتی را نشان داده است.



شکل ۲-۱۶- MMI کلاس ۲ استاندارد IEEE 1451.4 محرک(actuator) 4-20mA



شکل ۲-۱۷- MMI کلاس ۲ استاندارد IEEE 1451.4، سنسور پل مقاومتی

۲-۶-۲- مود آنالوگ

۲-۶-۲-۱- کلاس ۱

مود عملکردی یک ترانسدیوسر کلاس ۱ استاندارد IEEE 1451.4 با پلاریته منبع تغذیه ولتاژی یا جریانی که به ترانسدیوسر ارسال می‌شود، کنترل می‌شود. ترانسدیوسر کلاس ۱ برای ولتاژ و یا جریان مثبت نسبت به برگشت در مود آنالوگ خواهد بود.

شکل(۲-۱۳) یک سنسور جریان ثابت کلاس ۱ که شامل یک تقویت کننده داخلی است را نشان می‌دهد. دو دیود هدایت شونده با جریان اجازه دسترسی ترتیبی به تقویت کننده یا حافظه TEDS را با معکوس کردن پلاریته جریان تغذیه می‌دهد. منبع جریان مثبت در سیستم جمع آوری داده 1451.4 جریان تغذیه مثبت ثابت را از طریق اتصال سیگنال و دیود بالایی زمانیکه سوییچ در وضعیت آنالوگ است، فراهم می‌کند.

DOI: 10.30503/nripress.2020.296

ترانسدیوسر تغذیه شده با ولتاژ کلاس ۱ که در شکل(۲-۱۴) شرح داده شده است به شکل مشابه زمانیکه سیستم جمع آوری داده به تغذیه مثبت سوییچ می کند، در مد آنالوگ قرار می گیرد.
سیستم کلاس ۱ نشان داده شده در شکل(۲-۱۵) یک اتصال مداوم به تقویت کننده آنالوگ را پشتیبانی می‌کند. همزمان خط داده TEDS دیجیتال جهت دسترسی دیجیتال نیز می تواند وصل می‌شود، اما برای جلوگیری از تداخل نویز جریانی آنالوگ و دیجیتال از سویچ استفاده شده است.

۲-۶-۲-۲- کلاس ۲

یک اینترفیس کلاس ۲ از اتصالهای جداگانه برای تبادل سیگنال‌های ترانسدیوسر و داده TEDS دیجیتال استفاده خواهد کرد. با اتصالهای جداگانه، دسترسی به سیگنال و داده به شکل همزمان ممکن است.
بدلیل جدایی توابع آنالوگ و دیجیتال، عملکرد آنالوگ ترانسدیوسر هیچ تاثیری روی عملکرد ارتباط دیجیتال نداشته و لذا در مطالب بعدی به آن اشاره نمی شود.

۲-۶-۳- مود دیجیتال

۲-۶-۳-۱- کلاس ۱

مود عملکردی یک ترانسدیوسر کلاس ۱ استاندارد IEEE 1451.4 بوسیله پلاریته منبع تغذیه ولتاژ یا جریانی که به ترانسدیوسر منتقل می‌شود کنترل می‌شود. برای ولتاژ منفی محدود به ۶- ولت نسبت سیم برگشت که توسط تغذیه جریان ثابت و یا یک منبع ولتاژ تولید می شود ترانسدیوسر در مد دیجیتال قرار می گیرد. تغذیه ولتاژی با جریان محدود شده بوده و سطح ولتاژ صفر ولت برای منطق صفر و سطح ولتاژ ۵- ولت برای منطق یک خواهد بود.
نودهای IEEE1451.4 به عنوان وسایل دو سر قابل دسترس خواهند بود و به پرتکل تبادل داده، اجازه می‌دهد تغذیه و داده روی یک سیم وجود داشته درحالیکه سیم دیگر به عنوان سیم بازگشت استفاده می‌شود. نودها از پورت کلکتور باز استفاده می‌کنند و تغذیه پارازیتی را از خط داده فراهم می سازند. پرتکل ارسال یک تکنیک اختصاصی است که برای استفاده در این استاندارد مطابقت داده شده است. قطعه حافظه نیاز دارد که وضعیت صفر با محدودیتهای مشخص در fall time اعمال شود. این زمان بندی با دیود هدایت شونده نمی‌تواند حافظه را از خط درایور داده جدا کند. بنابراین یک مقاومت پایین کشنده[۱] در بخش پایانی (R_t) باید در مدار وجود داشته باشد شکل(۲-۱۳). این مقاومت، خازن مدار حافظه و سیم بندی را تخلیه کرده و تضمین می‌کند که منطق صفر ولتاژ با نیازمندیهای زمان بندی صحیح مطابقت داشته باشد.

۲-۶-۳-۲- کلاس ۲

MMI کلاس۲ استاندارد IEEE 1451.4 از یک اتصال داده TEDS دیجیتال جدا از اتصالهای آنالوگ استفاده خواهد کرد.

[۱] pull down Resistor

DOI: 10.30503/nripress.2020.296

سطوح منطقی صفر برابر صفر ولت و سطح منطقی یک برابر ۵ ولت است. منبع لاجیکی باید به لحاظ جریانی محدود شود و زمانبندی و پرتکل دسترسی داده برای کلاس ۲ همانند کلاس ۱ است. اگر از دیود در اتصال داده MMI کلاس ۲ برای حفاظت از پلاریته استفاده می شود لازم است با یک مقاومت همراه شود.

۴-۶-۲- تعریفهای خط

۲-٦-٤-۱- کلاس ۱

ارتباط دیجیتال با یک ترانسدیوسر کلاس یک از IEEE 1451.4 روی یک اتصال اشتراکی با تغذیه یا سیگنال آنالوگ خواهد بود.
اتصالهای کلاس ۱ در جدول (۲-۱۶) خلاصه شده است.
ارتباط دیجیتال نیاز دارد که خط داده با 5V- نسبت به خط برگشت بوسیله یک منبع تغذیه با جریان محدود شده درایو شود.

۲-٦-٤-۲- کلاس ۲

در استاندارد IEEE 1451.4، کلاس ۲، ترانسدیوسرها از یک اتصال جدا برای ارتباط دیجیتال استفاده می‌کنند. توابع دیجیتال و آنالوگ ممکن است به شکل همزمان رخ دهند. جدول (۲-۱۷) را مشاهده کنید.
در این مد خط داده با 5V+ نسبت به خط برگشت بوسیله یک منبع جریان محدود شده، درایو می‌شود. عملکرد آنالوگ ممکن است بدون وقفه و مستقل از عملکرد دیجیتال ادامه یابد.

جدول ۲-۱۷- تعریف خطوط برای MMI کلاس ۱

Mode		Physical wire name			
		+ Power	Signal	Data	Return
Shared signal wire	Analog Mode	Not applicable	Analog signal	Not applicable	Return
	Digital Mode	Not applicable	(–) Data	Not applicable	Return
Shared power wire	Analog Mode	(+) Power	Analog signal	Not applicable	Return
	Digital Mode	(–) Data	Undefined	Not applicable	Return
Shared return wire	Analog Mode	(+) Power	Analog signal	Not used	Return
	Digital Mode	(+) Power	Analog signal	(–) Data	Return

جدول ۲-۱۸- تعریف خطوط برای MMI کلاس ۲

Physical wire name		
Data	Data return	"n" Analog
(+) Data	Return	Analog signals

DOI: 10.30503/nripress.2020.296

۵-۶-۲- پرتکل ارسال داده دیجیتال MMI

پروتکل ارسال داده زیر در پیاده سازی ترانسدیوسر مود ترکیبی IEEE1451.4 و سیستم‌های جمع آوری داده استفاده می‌شوند. این پروتکل، اعمال تغذیه و تبادل داده را به یک یا چند وسیله IEEE 1451.4 را نشان می‌دهد که قوانین زیر مدیریت می‌شوند:

۱- پروتکل باید Master/Slave، multidrop، و سریال باشد. یک وسیله master یکتا (عموما سیستم جمع آوری داده) باید تغذیه را تامین کرده و هر تراکنشی با هر نود مطابق با یک ترتیب زمان‌بندی روی یک زوج سیم سیگنال و سیم برگشت آغاز می‌کند. نودها باید مجهز به پورت کلکتور باز بوده و از تغذیه پارازیتی خط داده استفاده کنند.

۲- هر نود باید شامل یک آدرس URN ۶۴ بیتی شامل ۸ بیت کد خانواده، ۴۸ بیت شماره سریال و ۸ بیت کد CRC بوده که قابل شناسایی توسط master برای کنترل دسترسی در بین چندین نود روی یک باس مشترک استفاده می‌شوند.

۳- یک سلسله مراتب از فرمانها که بوسیله کد خانوادگی مشخص شده و بوسیله master صادر می‌شود، هر تراکنش بین master و نود را کنترل خواهند کرد. فرمانها با یک ترتیب آماده‌سازی شروع و با ترتیبی از سیگنال‌ها، شامل یک فرمان تابعب به ROM، سپس یک فرمان تابعی به حافظه، و در انتها خواندن و نوشتن داده‌های مربوطه ختم می شود. البته این سلسه دستورات با دریافت پالس حضور آغاز شده و ختم این محاوره با پالس آغاز سازی بعدی خواهد بود.

۴- یک مجموعه تعریف شده از ترتیبهای خاص و سیگنالینگ اینترفیس، مبتنی بر زمانبندی، توسط master، تولید می شوند تا امکان تبادل داده بین مرکز و تجهیزات فراهم شود.

۵- در ارتباط سریال فوق، ارسال هر داده باید از بیت با ارزش پایین شروع شود.

۶- برای سیستم‌هایMMI کلاس ۱ استاندارد IEEE 1451.4، داده باید در فرمت معکوس خوانده و نوشته شود، اما درکلاس ۲ استاندارد IEEE 1451.4، داده باید در فرمت مثبت نوشته و خوانده شود.(قبلا تشریح شده)

۷- سیگنال ولتاژی بیشتر از 0 V در MMI کلاس ۱ باید سیگنال‌های آنالوگ باشند

نودها ممکن است هر کدام چندین تابع آنالوگ و یا دیجیتال را داشته باشند. توابع دیجیتال شامل RAM، EEPROM، OTP، و ۶۴ بیت حافظه ROM است. توابع آنالوگ مانند سوییچ‌های آنالوگ، فرستنده/ گیرنده‌های داده، ADC، DAC، و ترانسدیوسرهای دمایی ممکن است در نودها وجود داشته باشند.

هر نود یک حافظه دایم ROM از پیش برنامه ریزی شده، شامل شماره ثبت الکترونیکی یکتا، که شامل Family code و CRC code است را خواهد داشت. این شماره سریال یکتا پایه ارتباط دیجیتال قابل آدرس دهی را درون یک معماری باس multidrop فراهم می‌سازد. کد خانوادگی Family code مجموعه‌ای از فرمانها را تعریف می‌کند، که به وسیله master در تمام تراکنشهای داده دیجیتال با نود استفاده می‌شود.

۲-۶-۵-۱- مراحل تراکنش

مراحل تراکنش زیر، تراکنشهای مود دیجیتال IEEE 1451.4 را کنترل می‌کنند:

۱- آماده سازی:

DOI: 10.30503/nripress.2020.296

هر تراکنشی با یک مراحل آماده سازی آغاز می‌شود.master یک پالس ریست را میفرستد، که بعد از آن نود یا نودها با یک پالس حضور (presence pulse) پاسخ می دهند. با پالس حضور master تشخیص می‌دهد که یک یا چندین نود در باس وجود دارد.

۲- فرمان تابع ROM

بعد از آماده سازی،master یکی از چهار فرمان تابع ROM را با استفاده از بخش‌های زمانبندی خواندن و نوشتن، ارسال می‌کند. هر فرمان ROM نیازمند آن است که master از یکسری مراحل الگوی خواندن و نوشتن ویژه بکار برده تا نودها را از طریق توابع به شکل صحیح هدایت کند. فرمان جستجوی ROM به master اجازه می‌دهد شماره سریال هر نودی که در باس وجود دارد تشخیص دهد. فرمان تطبیق ROM به master اجازه می‌دهد یکی از نودهای روی باس را جهت یک محاوره انتخاب کند. دستور خواندن ROM به master اجازه می‌دهد که محتویات ۶۴ بیتی ROM، خوانده شود. دستور پرش از ROM (Skip ROM) یک میانبری برای پایان دادن به توابع آدرس دهی بوده و مستقیما می توان توابع RAM را بکار برد. یک دیتاشیت جاری روی وسایل نود ویژه، انعطاف پذیرترین منبع برای فرمانهای معتبر برای آن نود است. داده توصیفی در "DS2430A 256 bit 1-wire EEPROM" اجازه خواهد داد خواننده یک فهم ابتدایی از پردازشهای منطقی در فرمانهای ROM را کسب کند.

۳- فرمان تابع حافظه و تبادل داده

بعد از عبور از فرمان ROM، نود وارد وضعیت فرمانهای تابع حافظه می‌شود. master سپس فرمان مناسب را با مجموعه صحیحی از سیگنالیگ خواندن و نوشتن را برای خواندن و یا نوشتن می‌فرستد. بسته به کد خانوادگی هر نود یک مجموعه دستورات مشخص وجود دارد که مرجع قابل اطمینال برای دستورات کاربردی این بخش از حافظه است. بصورت نمونه دیتا شیت مربوطه به قطعه DS2430A معرفی می گردد.

۲-٦-۵-۱-۱- مراحل آماده سازی

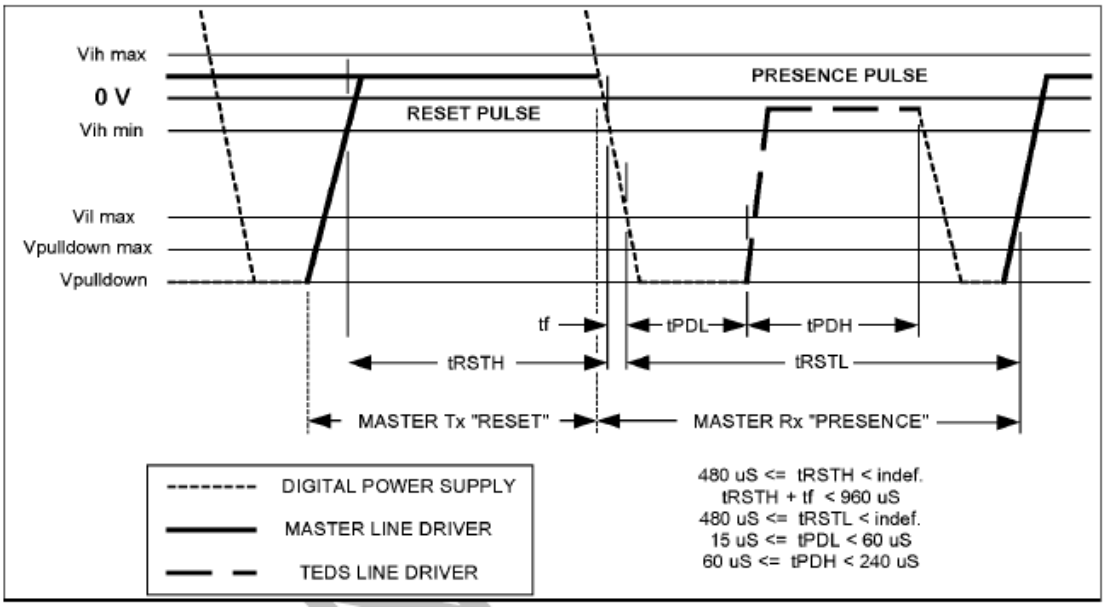
توجه: دیاگرامهای زمانبندی که در ادامه آمده است همگی از برگه اطلاعاتی قطعه DS2430A گرفته شده که در کلاس ۲ مستقیما قابل استفاده بوده ولی در کلاس ۱ اینترفیس، لازم است با تغییر سطوح منطقی همانطور که قبلا اشاره شده بکار روند.

master (سیستم جمع آوری داده) هر تراکنش داده دیجیتالی را با ارسال یک پالس ریست آماده‌سازی می‌کند. البته باس بایستی در وضعیت ایده آل 5- V، برای شروع مراحل آماده سازی باشد (به شکل ۲-۱۸ توجه کنید).

برای تضمین اینکه همه نودها روی باس ریست شوند، پالس باید حداقل 480 us طول زمانی داشته باشد. در طول 15-60 us بعد از یک پالس ریست، نودها یک پالس حضور را اعمال کرده و باس را برای یک مدت زمان بین 15 us و 60 us صفر می کنند.

از وضعیت غیر فعال 5- V، master پالس ریست 480 us را اعمال خواهد کرد، و نود یک پالس حضور، بعد از 15- 60 us را بازخواهد گرداند. هر زمان که یک تراکنش به شکل موقت متوقف می‌شود، باس در وضعیت ایده‌آل 5- V برای برگشتن به تراکنش، خواهد ماند. اگر باس بیشتر از 120 us در وضعیت 0 V باقی بماند، نود یک مراحل ریست را آغاز می‌کند، و تراکنش پایان خواهد پذیرفت.

DOI: 10.30503/nripress.2020.296



شکل ۲-۱۸- مراحل آماده سازی

۲-۶-۵-۱-۲- مراحل سیگنالینگ

پس از مراحل آماده سازی، master با استفاده از یکی از چندین فرمان ROM، و سپس با فرمان حافظه RAM شروع به سیگنالینگ با نود می‌کند. سیگنالینگ در قالبهای زمانی ثابت بوده که حاوی پالس‌های مشخص نوشتن و خواندن از نود است.

همواره در یک قالب زمانی ثابت ۶۰ میکروثانیه، هر ارتباط با یک پالس صفر شونده برای مدت زمان 1-15us از طرف master آغاز می شود. در ادامه پس از آن، اگر ارسال داده برای نود باشد وضعت باس توسط master کنترل شده و نود اطلاعات را می خواند ولی اگر ارسال داده از نود به master باشد وضعیت باس توسط نود کنترل شده و در این حال master داده را می خواند.

برای نمونه، بعد از مرحله آماده سازی، یک فرمان خواندن ROM (33h) از قطعات زمانی نوشتن 1 و نوشتن 0 تشکیل شده است.

W1, W1, W0, W0, W1, W1, W0, W0 (33h, lsb first)

بعد از آن 64 عدد بازه زمانی خواندن وجود دارد که شامل کد خانوادگی، شماره سریال، و کد CRC است.

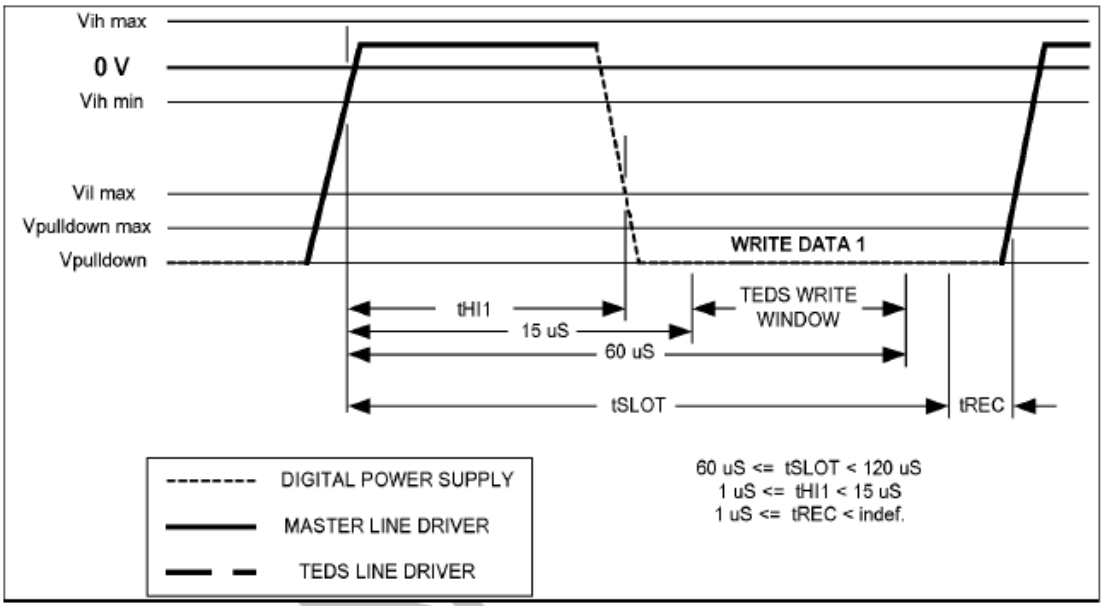
مراحل فوق، نود را در نقطه شروع فرمانهای حافظه RAM قرار می‌دهد. اگر master از صحت CRC اطمینان حاصل کند می تواند یک فرمان حافظه شروع شود. یک خطای CRC نشان دهنده این است که بیشتر از یک وسیله به فرمان خواندن ROM پاسخ داده است، و عکس العمل درست برای master این است که یک مرحله ریست جدید را اعمال کند و با یک فرمان جستجوی ROM (F0h) برای تشخیص شناسه‌های نودها روی باس به کار ادامه دهد.

این عملیات با جزییات بوسیله فلوچارت‌هایی که در "DS2430A 256-bit 1-Wire EEPROM" آمده، شرح داده شده‌اند.

DOI: 10.30503/nripress.2020.296

۲-۶-۵-۱-۳- قطعات زمانی Time Slot ها

قطعه زمانی نوشتن یک، با پالس صفر شونده 1-15us توسط master شروع و سپس باس رها شده تا به سطح ۵- ولت (غیرفعال) خود برسد. پس از این مدت، در یک پنجره زمانی مشخص اشاره شده در شکل (۲-۱۹) وضعیت باس توسط نود خوانده می شود لذا وضعیت باس باید در این پنجره ثابت بماند.



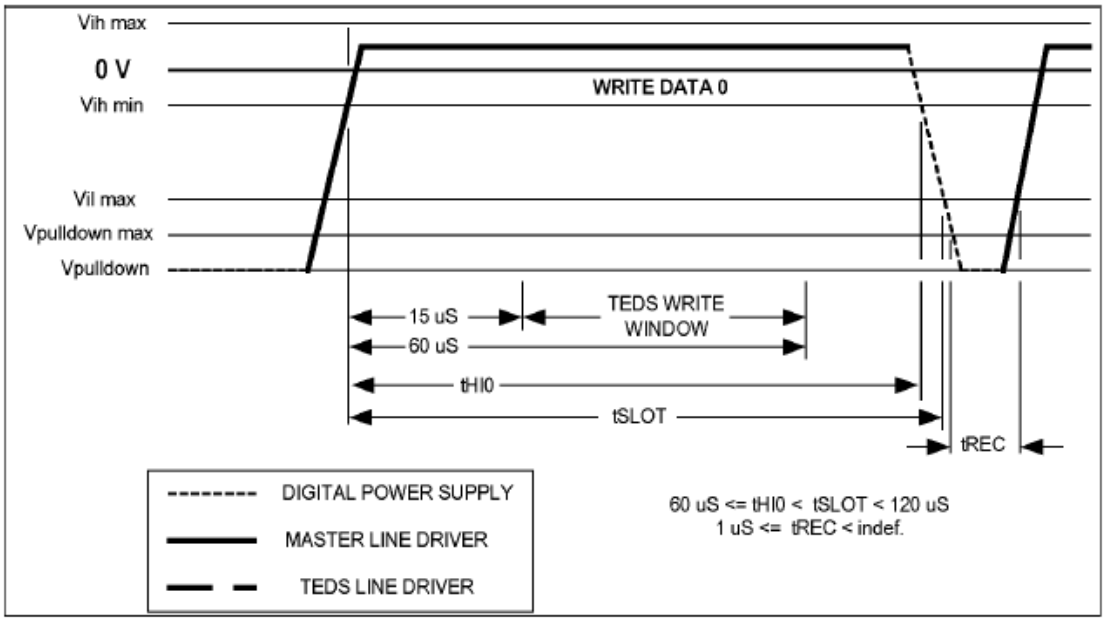
شکل ۲-۱۹- Time Slot نوشتن یک

قطعه زمانی نوشتن صفر، با پالس صفر شونده 1-15us توسط master شروع و سپس وضعیت باس تا انتهای قطعه زمانی ۶۰ میکروثانیه در صفر ولت باقی می‌ماند. یک پالس حداقل ۱ میکروثانیه به وضعیت ۵- ولت برای اتمام قطعه زمانی بوده و قطعه بعدی می تواند شروع شود.شکل(۲-۲۰)

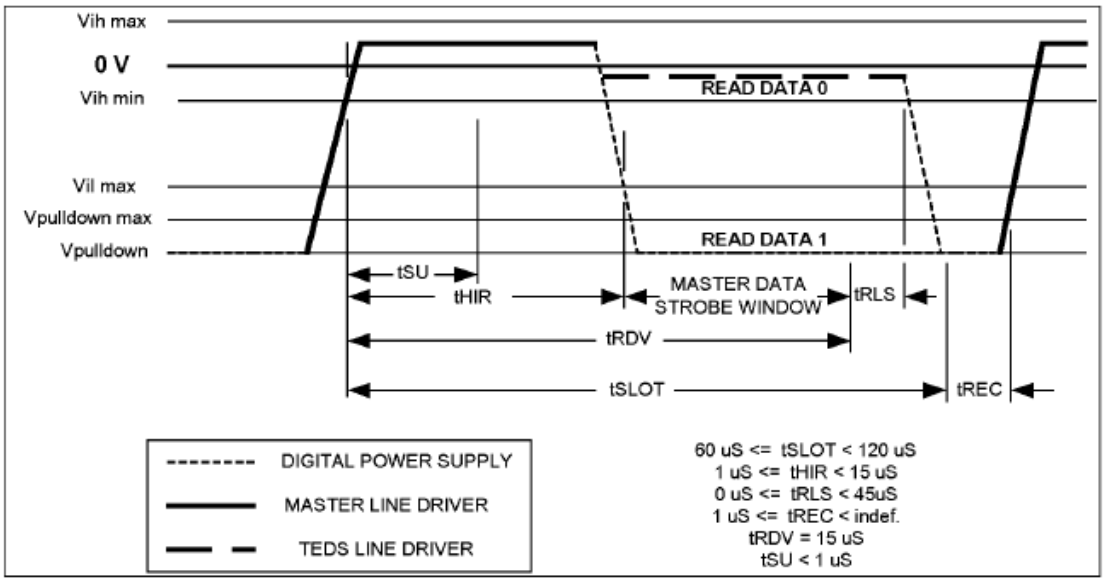
شایان توجه است که در قطعات زمانی نوشتن یک و یا صفر، وضعیت باس تنها توسط master کنترل شده و نودهای ارتباطی در وضعیت خواندن قرار دارند.

قطعه زمانی خواندن داده، با پالس صفر شونده 1-15us توسط master شروع و سپس باس رها شده تا به سطح ۵- ولت (غیرفعال) خود برسد. پس از این مدت، در یک پنجره زمانی مشخص اشاره شده در شکل (۲-۲۱) وضعیت باس توسط نود کنترل شده (صفر کردن باس و یا رها کردن آن) و master در این شرایط در وضعیت خواندن است. در انتهای قطعه زمانی ۶۰ میکروثانیه یک پالس با حداقل پهنای ۱ میکروثانیه به منزله خاتمه قطعه زمانی بوده می تواند قطعه زمانی بعدی شروع شود.

DOI: 10.30503/nripress.2020.296



شکل ۲-۲۰- time slot نوشتن صفر



شکل۲-۲۱- time slot خواندن

۲-۶-۵-۲- ویژگی‌های الکتریکی

جدول ۲-۱۸ شامل اطلاعاتی با جزییات روی سیگنال ولتاژها است، که در پیاده سازی MMI کلاس ۱ IEEE 14511.4 در نظر گرفته خواهد شد. سطح سیگنال‌ها منطقی صفر و یک و بیشترین و کمترین آنها آمده است. ولتاژها برای اینترفیس کلاس ۲ IEEE 1451.4 همان اندازه‌ها را خواهند داشت، اما با پلاریته مثبت. جزییات زمانبندی در جدول ۲-۱۹ به هر دو اینترفیس کلاس ۱ و کلاس ۲ اعمال خواهد شد.

جدول ۲-۱۹- ویژگی‌های الکتریکی DC

DOI: 10.30503/nripress.2020.296

(–40 °C to +85 °C; Vpulldown = –3.4 V to - 6.0 V)

Parameter	Symbol[a]	Min	Typ	Max	Units
Analog Mode (Class 1)		0	2 to 30		V
Logic 1[b]	Vil	–6.0	–5	–3.4	V
Logic 0[b]	Vih	–0.8		0	V
Output[c] Logic 0 @ –4mA	Voh	–1		0	V
Output[c] Logic 1		–6.0[d]		Vpulldown	V
Supply current limit[b] (during read)	IL	–1.5		–4.0	mA
Operating charge[c]	QOP			30	nC
Programming current[c]	IP			600	µA
Capacitance data pin[c]	CD			800	pF

[a]Refer to logic diagrams.
[b]Master line driver.
[c]TEDS line driver.
[d]Voltages more negative might damage the digital devices.

جدول ۲-۲۰- ویژگی‌های الکتریکی AC

(–40 °C to +85 °C; Vpull-up/down = –3.4 V to –6.0 V)

Parameter	Symbol	Min	Max	Units
Time slot	tSLOT	60	120	µs
Write 1 HIGH time	tHI1	1	15	µs
Write 0 HIGH time	tHI0	60	120	µs
Read HIGH time	tHIR	1	15	µs
Read Data Valid	tRDV	Exactly 15		µs
Release time	tRLS	0	45	µs
Release time @ Vpulldown = –4.6V to –6.6V	tRLS	7	30	µs
Read data setup	tSU		1	µs
Recovery time	tREC	1		µs
Reset time high	tRSTH	480	960[a]	µs
Reset time low	tRSTL	480	Indefinite	µs
Presence detect high	tPDH	60	240	µs
Presence detect low	tPDL	15	60	µs

[a]tRSTH may be extended indefinitely, but if > 960 µs, it might mask devices attempting to assert an interrupt signal on the bus.

NRI

DOI: 10.30503/nripress.2020.296

۲-۶-۵-۳- منبع تغذیه لاجیکی IEEE 1451.4

نودهایی که از MMI استاندارد IEEE 1451.4 استفاده می‌کنند، مجهز به پورت کلکتور باز بوده و تغذیه خود را بصورت پارازیتی از خط داده تامین می کنند. تغذیه برای نودها در طول خواندن، بوسیله یک منبع محدود شده جریانی فراهم شده و در وضعیت یک منطقی حداکثر ۱.۵ میلی آمپر و در وضعیت صفر منطقی، حداکثر به ۴ میلی آمپر خواهد رسید. برای MMI کلاس ۱، همه ولتاژها منفی و برای اینترفیس کلاس ۲، همه ولتاژها مثبت خواهند بود.

DOI: 10.30503/nripress.2020.296

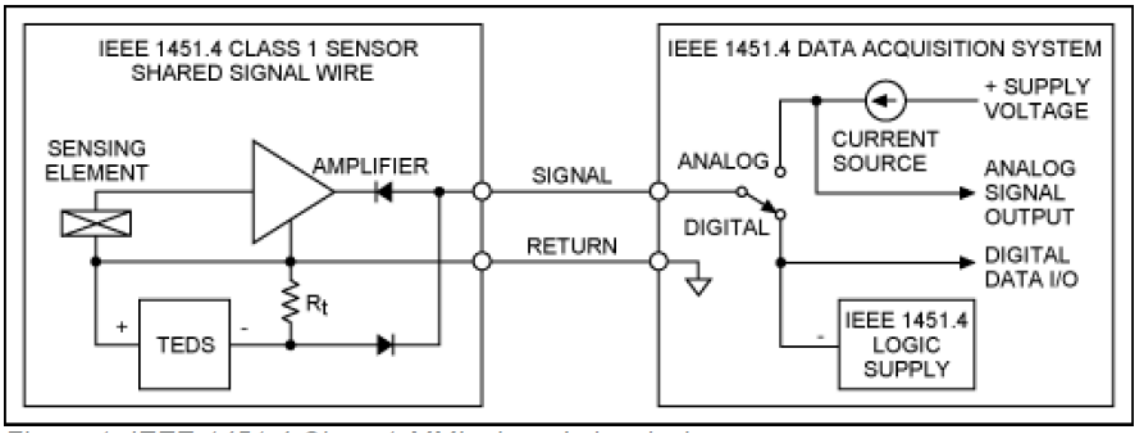
فصل سوم: طراحی درایور و مبدل منطبق بر استاندارد IEEE1451.4

DOI: 10.30503/nripress.2020.296

۱-۳- سازگارسازی شبکه 1-Wire برای استاندارد IEEE1451.4

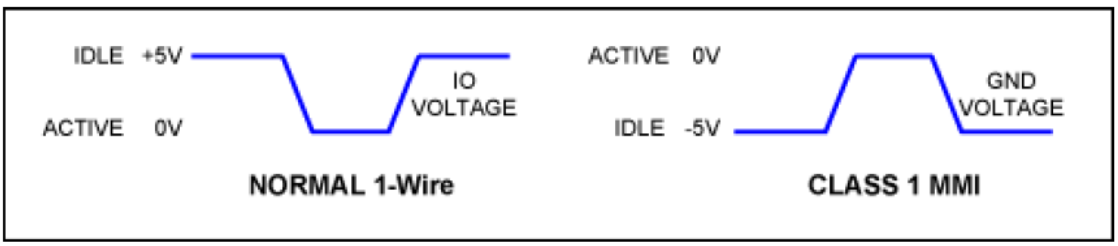
همانطور که در فصل دوم اشاره شده در اینترفیس MMI کلاس ۲ از این استاندارد، ارتباط بخش دیجیتال ترانسدیوسر با Ncap دقیقا مطابق با باس استاندارد 1-Wire می‌باشد. اما در کلاس ۱ همین شبکه با اندکی تفاوت در سطح سیگنال‌های منطقی بکار میرود.

در این پروژه هدف پیاده سازی استاندارد IEEE1451.4 برای مبدل بار الکتریکی دوسیمه‌ای است که قبلاً طراحی و ساخته شده است. از آنجایی که این مبدل کاملا آنالوگ و دو سیمه می‌باشد، بنابراین در این حالت بایستی از همان دو سیم ارتباطی آنالوگ، برای ارتباط دیجیتال نیز استفاده شود. بنابراین اینترفیس کلاس ۱ مد مختلط استاندارد IEEE1451.4 که مطابق شکل (۳-۱) است برای این مبدل کاربرد دارد. همانطور که از شکل پیداست بخش دیجیتال و آنالوگ با استفاده از دیود مجزا شده‌اند بطوریکه سیگنال با پلاریته مثبت باعث فعال شدن بخش آنالوگ و منفی شدن این پلارتیه باعث فعال کردن بخش دیجیتال سنسور خواهد شد. البته این اینترفیس بصورت همزمان میسر نیست.



شکل ۳-۱- کلاس ۱ اینترفیس MMI

پروتکل ارتباطی در مد دیجیتال دقیقا منطبق با باس 1-Wire با اندکی تفاوت در سطح سیگنال منطقی است. یک باس استاندارد تک سیمه با سطوح ولتاژ ۵+ و صفر ولت کار می‌کند اما در در استاندارد IEEE1451.4 این سطوح ۵- و صفر ولت می‌باشند. این تفاوت و تغییر سطح ولتاژ در شکل (۳-۲) نشان داده شده است.



شکل ۳-۲- سطوح سیگنال 1-Wire در مقابل کلاس ۱ استاندارد IEEE1451.4

با توجه به شکل مشاهده می‌شود که باس استاندارد 1-Wire در حالت غیرفعال ۵ ولت بوده و به هنگام فعال شدن صفر می‌شود در حالی که در استاندارد IEEE1451.4 وضعیت باس در حالت غیر فعال ۵- ولت بوده و با فعال شدن صفر می‌شود. لذا در طراحی درایور مورد نظر برای این استاندارد این مسئله باید در نظر گرفته شود.

DOI: 10.30503/nripress.2020.296

بنابراین برای داشتن یک درایور Ncap منطبق با استاندارد IEEE1451.4 می‌توان از همان طرحهای درایور شبکه 1-Wire با لحاظ تبدیل سطح فوق استفاده نمود. شایان ذکر است که Master شبکه 1-Wire در واقع همان نقش Ncap را در استاندارد IEEE1451.4 ایفا می‌کند.

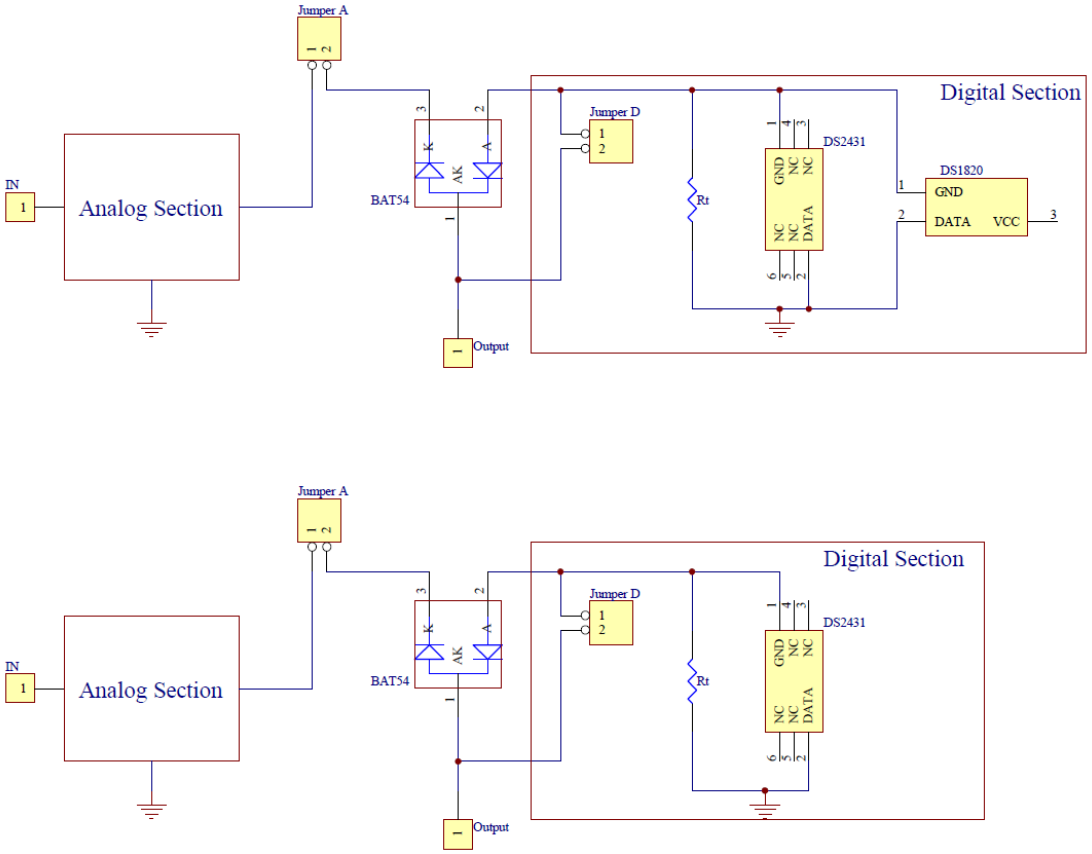
۲-۳- طراحی مبدل و درایور استاندارد IEEE1451.4

در بخش ترانسدیوسر (که در اینجا بعنوان تجهیزات Slave هستند) بایستی نکته فوق در نظر گرفته شده و پلاریته قطعات کاربردی بصورت صحیح و دقیقا مطابق با شکل (۳-۱) بکار روند. این قطعات در واقع حافظه‌های EEPROM با باس 1-Wire هستند که اطلاعات دیتا شیت الکترونیک (TEDS) تجهیز را مطابق با الگوها و فرمت استاندارد IEEE1451.4 در خود ذخیره می‌کنند. امکان استفاده از دیگر سنسورهای 1-Wire به همراه حافظه در این استاندارد دیده شده است.
استفاده از مقاومت Rt در شکل (۳-۱) جهت تخلیه سریع ولتاژ خازنی ضروری بوده و مقدار آن نیز عملا با ملاحظات سرعت پاسخ باس باید تنظیم گردد.
در بخش درایور نیز همانطور که قبلا اشاره شده می‌توان از طراحی درایورهای شبکه 1-Wire کمک جست و با استفاده از یک مدار تبدیل سطح در خروجی درایور، از آن برای درایور استاندارد IEEE1451.4 استفاده نمود. ضمناً مطابق شکل (۳-۱) از یک سویچ برای تغییر وضعیت آنالوگ به دیجیتال در درایور استفاده می‌شود.

۱-۲-۳- طراحی مبدل مجهز به بخش دیجیتال منطبق با استاندارد IEEE1451.4

همانطور که در بخش ۳-۱ اشاره شد مطابق استاندارد IEEE1451.4 لازم است ارتباط مد مخلوط در کلاس ۱ صورت گیرد که همانند شکل (۳-۱) خواهد شد. سطح ولتاژ باس ارتباطی با منطق منفی و ۵- ولت بوده و مطابق با استاندارد IEEE1451.4 می‌باشد.
در این پروژه برای تامین حافظه TEDS از چیپ DS2431 استفاده شده است. این چیپ دارای حافظه ۱۰۲۴ کیلوبیت بوده و می‌توان علاوه بر اطلاعات پایه‌ای TEDS اطلاعات کالیبراسیون را نیز در آن جای داد. به منظور پیاده سازی لیست ترانسدیوسر نیز سعی شده است با استفاده از یک چیپ سنسور دمای 1-Wire به شماره DS1820 بر روی یک باس مشترک با حافظه امکان تولید لیست ترانسدیوسر را نیز بررسی نمود.
به منظور ساده‌سازی تست‌ها، دو مبدل با دو بخش دیجیتال متفاوت، طراحی شده است بطوریکه در یکی تنها حافظه و در دیگری علاوه بر حافظه سنسور دما نیز وجود دارد تا حالتهای مختلف قابل تست باشند. شکل (۳-۳) طراحی شماتیک دو مبدل بصورت یکپارچه و قابل نصب در یک کیس را نشان میدهد.

DOI: 10.30503/nripress.2020.296



شکل ۳-۳- شماتیک مبدل دو کاناله پیزوالکتریک به همراه بخش TEDS

بخش آنالوگ مبدل همانا طرح مبدل آنالوگ از پروژه قبلی بوده و با افزودن بخش دیجیتال به همراه قطعات جانبی مطابق شکل (۳-۳) سخت افزار یک مبدل مجهز شده به TEDS را مطابق استاندارد IEEE1451.4 فراهم ساخته است.

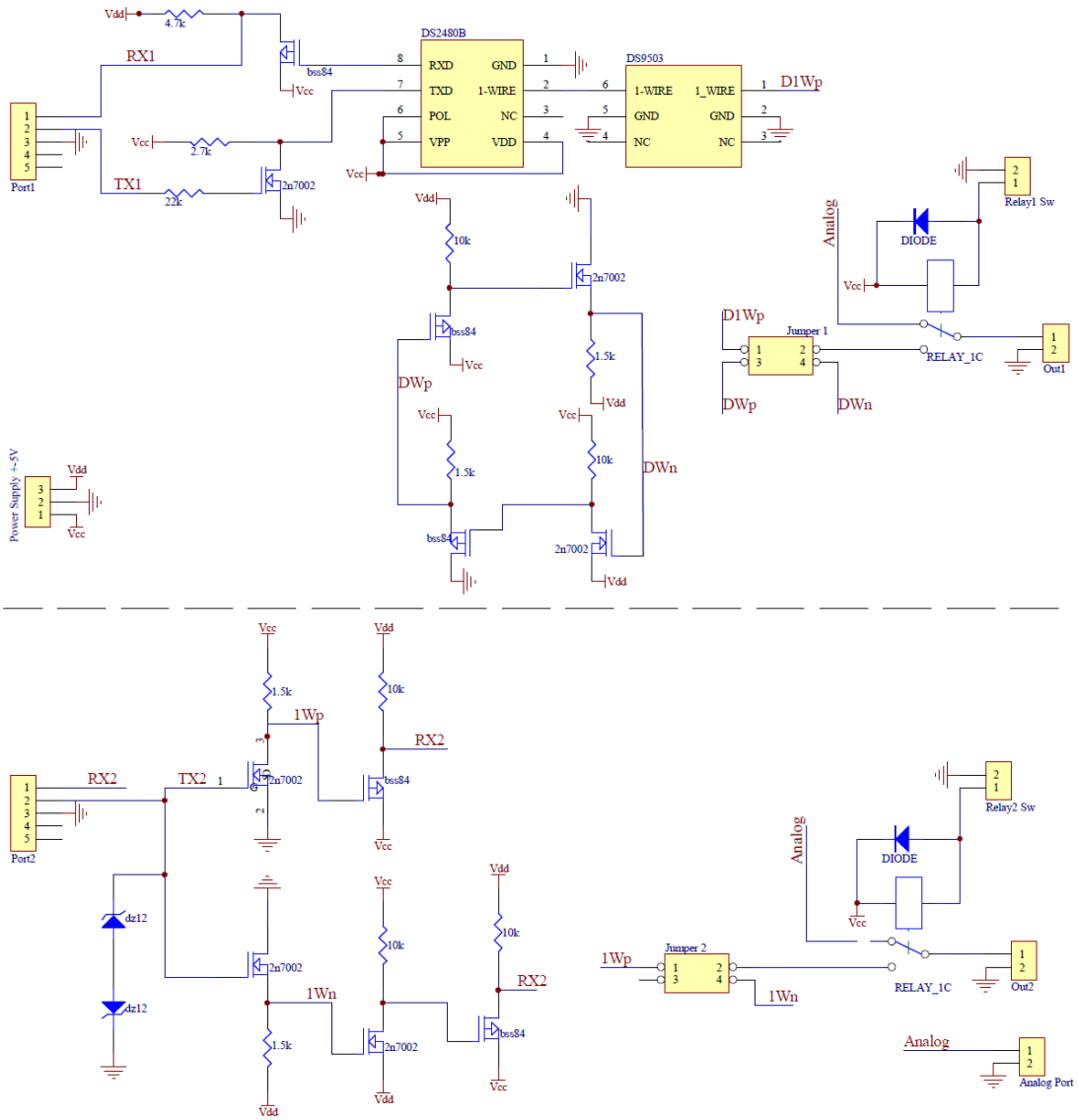
۳-۲-۲- طراحی درایور شبکه منطبق با استاندارد IEEE1451.4

در این پروژه باید درایوری مطابق استاندارد IEEE1451.4 برای مبدل فوق مهیا کنیم. همانطور که در بخش ۳-۱ اشاره شد درایورهای استاندارد 1-Wire می‌توانند برای این منظور با انجام اصلاحات ذکر شده بکار روند. نمونه درایور اشاره شده در فصل اول که با استفاده از چیپ DS2480 کار می‌کند و در شکل (۱-۹) معرفی شده برای استفاده بدین منظور دارای مشخصات مناسب عملیاتی است. درایور عملیاتی ارتباط دیجیتال در استاندارد IEEE1451.4 دارای سطح ولتاژ منفی است بطوریکه سطح ولتاژ شبکه بایستی در حالت غیر فعال ۵- ولت بوده و در حالت فعال به صفر ولت برسد. لذا برای تبدیل درایور 1-Wire به درایور استاندارد IEEE1451.4 لازم است یک بخش مناسب‌سازی به خروجی طرح درایور 1-Wire مطابق شکل (۳-۴) اضافه شده تا عمل تبدیل سطح سیگنال را انجام دهد. در این پروژه از طرح تغذیه خارجی برای مبدل استفاده شده است.

به منظور مقایسه و یا داشتن یک درایور جایگزین ساده، از درایور دیگری که بدون چیپ DS2480 است نیز در طراحی کلی درایور استفاده شده است. در این قسمت، از یک مبدل مستقیم سطح سیگنال پورت سریال به TTL استفاده شده تا هم درایوری مطابق استاندارد 1-Wire را ساخته و هم با تغییر سطح سیگنال نهایی به ۵- ولت، این

DOI: 10.30503/nripress.2020.296

مبدل برای استاندارد IEEE1451.4 تبدیل شود. شکل (۳-۴) شماتیک این درایور دو کاناله را نشان می‌دهد. یک کانال با چیپ DS2480 بوده و کانال دیگر بصورت مستقیم است.



شکل ۳-٤- شماتیک مبدل دو کاناله درایور کلاس ۱ استاندارد

با درایور فوق می‌توان شبکه 1-Wire و شبکه منطبق بر IEEE1451.4 را به دو صورت راه‌اندازی نمود. در هر دو کانال سویچ تغییر آنالوگ به دیجیتال بصورت یک رله با کنترل تحریک دستی فراهم شده است. کنتاکت نرمال بسته رله برای حالت آنالوگ و با فعال شدن رله بخش دیجیتال فعال می‌شود.

شایان ذکر است که بر مبنای قطعات استفاده شده در طرحهای فوق لیست قطعات لازم این پروژه استخراج و سفارش خرید آن صورت گرفته است.

۳-۲-۳- لیست اقلام موردنیاز

با توجه به اشکال (۳-۳) و (۳-۴) لیست اقلام مورد نیاز مبدل و درایور در جداول (۳-۱) و (۳-۲) آمده است.

جدول ۳-۱- اقلام مورد نیاز برد مبدل

DOI: 10.30503/nripress.2020.296

Part Type	Footprint	Description
3.2k	805	resistor
BAT54	sot23/bat54	diode
DS1820	BCY-W3/E4	1wire temprature sensor
DS2431	DSO-C6/D4	1wire 1kbit eeprom
Output	sip1	output port
CON1	sip1	output port
Jumper A	sip2	jumper
Jumper D	sip2	jumper

جدول ۳-۲- اقلام مورد نیاز برد درایور

Part Type	Footprint	Description
1.5k	805	resistor
2.7k	805	resistor
4.7k	805	resistor
10k	805	resistor
22k	805	Resistor
dz12	thd	zener
DIODE	1206	diode
DS2480B	soj-8	1 wire driver chip
2n7002	SOT-23	n ch mosfet
bss84	SOT-23	p ch mosfet
RELAY_1C	RELAY_1C_N5	a/d relay
RELAY_1C	RELAY_1C_N5	a/d relay
Jumper 1	IDC4	manual selection jumper of P/N one wire
Jumper 2	IDC4	manual selection jumper of P/N one wire
Out1	sip2	output port
Out2	sip2	output port
Port1	sip5	input port
Port2	sip5	input port
Power Supply +-5V	sip3	power supply port
Analog Port	sip2	analog port
Relay1 Sw	sip2	relay control Sw
Relay2 Sw	sip2	relay control Sw

تعداد اقلام سفارشی برای مونتاژ و ساخت حداقل ۵ نمونه برد از هریک از بردهای فوق منظور شده است.

DOI: 10.30503/nripress.2020.296

فصل چهارم: طراحی و ساخت بخش دیجیتال مبدل و تست‌های اولیه

٤-١- مروری بر مشخصات سنسورهای شتابسنج و مبدل آنها

سنسورهای شتاب سنج پیزوالکتریک با توجه به خروجیشان به دو دسته کلی تقسیم می‌شوند:

DOI: 10.30503/nripress.2020.296

سنسور شتاب سنج با خروجی بار الکتریکی[1]

سنسور شتاب سنج با خروجی ولتاژ[2]

سنسور شتاب سنج با خروجی بار الکتریکی همانگونه که از نام آن پیداست دارای خروجی بار الکتریکی متناسب با شتاب می‌باشد، از آنجایی که این سنسور هیچ مدار الکترونیکی نداشته و ماهیتا یک قطعه پیزوالکتریک بوده که جرمی روی آن قرارگرفته است. لذا این نوع سنسور محدودیت دمایی کمتری داشته و برای دماهای بالاتر از ۱۵۰ تا حدود ۵۰۰ درجه سانتیگراد قابل استفاده است. اما برای استفاده از خروجی این نوع سنسورها باید به کمک مدارات الکترونیک مناسب سازی و تقویت سیگنال صورت گرفته تا امکان انتقال آن برای سیستمهای اندازه گیری فراهم شود. به این مدار اصطلاحاً مبدل بار به ولتاژ یا تقویت‌کننده بار اطلاق می‌گردد. استفاده از این ترکیب سنسور و مبدل بیرونی امکان اندازه گیری شتاب را برای محیطهای با دمای بالا فراهم می‌سازد.

سنسور نوع دوم زمانی است که مجموعه سنسور به همراه برد مناسب سازی سیگنال در داخل کیس سنسور قرار میگیرند. به این سنسورها که حاوی الکترونیک داخلی هستند، IEPE نیز اطلاق می‌گردد. این عمل محدودیت دمایی بخش الکترونیک را به سنسور تحمیل نموده و ماکزییم دمای کار آن بستگی به مدار الکترونیک داخلی خواهد داشت. اما کار با این سنسور راحتتر بوده و خروجی آن از نوع ولتاژ و مستقیما به بخش پردازشی قابل اتصال است.

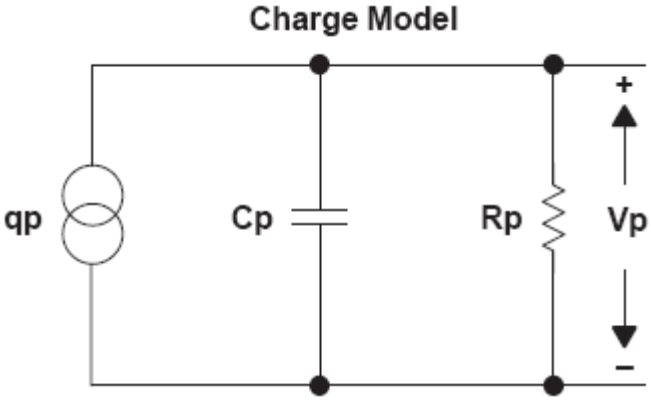
نمونه مبدل بار به ولتاژ ساخته شده در پژوهشگاه به منظور کاربردهای دما بالا بوده و بنابراین بصورت خارج از سنسور طراحی شده است. آخرین نسخه از مبدل بار به ولتاژ ساخته شده، بر اساس نمونه‌های صنعتی دارای اینترفیس خروجی دوسیمه است. ورودی این مبدل با یک زوج سیم کواکسیال به یک سنسور شتابسنج پیزوالکتریک متصل می‌شود و سیگنال خروجی مبدل به همراه تغذیه روی یک زوج سیم کواکسیال در پورت خروجی قرار می‌گیرد. چگونگی ارتباط تغذیه و سیگنال بصورت همزمان روی یک زوج سیم به کمک استفاده از یک منبع جریان ثابت صورت می‌پذیرد. از آنجا که ولتاژ این منبع می‌تواند در یک محدود متغیر باشد لذا سیگنال خروجی مبدل بصورت ولتاژ روی این جریان مدوله شده است. بطوریکه با جریان ثابت تغذیه و بدون تحریک ورودی (بدون شتاب)، یک ولتاژ بایاس و ثابت (معمولا ۱۰ تا ۱۲ ولت) که نقطه کار مبدل است تولید می‌شود. حال هر گونه نوسان حول این ولتاژ بایاس متاثر از ورودی مبدل که همانا شتاب است خواهد بود. ماکزیمم نوسان حول ولتاژ بایاس $\pm 5v$ برای ماکزیمم شتاب اعمالی به سنسور لحاظ شده است. برای حذف ولتاژ بایاس خروجی از یک خازن بلوکه کننده و مقاومت مناسب در مدار خروجی استفاده می‌شود. بدین ترتیب سیگنالی متناسب با شتابهای مثبت و منفی تولید می‌شود.

وظیفه مبدل مناسب سازی و تبدیل سیگنال خروجی سنسور پیزوالکتریک به یک سیگنال ولتاژی با امپدانس محدود است. همانطور که می‌دانیم خروجی یک سنسور پیزوالکتریک بصورت بارالکتریکی با امپدانس بالاست. حساسیت این سنسورهای شتابسنج بصورت میزان بار تولیدی بر واحد شتاب بیان می‌شود که بصورت نمونه این بار خروجی در حد چند پیکوکلمب بر واحد شتاب جاذبه g بیان می‌شود. ورودی مبدل بایستی به گونه ای باشد که کمترین اثر بارگذاری را روی سنسور داشته و یک خروجی ولتاژی متناسب با بار ورودی تولید نماید. بنابراین ورودی مبدل نیز باید بسیار امپدانس بالا بوده و با طراحی مناسب، بار روی این ورودی را به ولتاژ تبدیل نماید. بدین ترتیب ولتاژ خروجی مبدل متناسب با شتاب خواهد شد که این گین در نمونه های صنعتی معمولاً مقدار ۱۰ یا ۱۰۰ میلی ولت بر g بکار میرود.

مدل الکتریکی ساده سنسور شتاب سنج پیزو الکتریک که تا حد خوبی پاسخگوی نیازهای مداری می‌باشد در شکل (۴-۱) آمده است. این مدل دارای یک منبع بار متغییر qp موازی با یک خازن C_{PE} و مقاومت R_{PE} است.

[1] Charge Mode Accelerometer
[2] Voltage Mode Accelerometer or IEPE Accelerometer

DOI: 10.30503/nripress.2020.296



شکل ٤-١- مدل الکتریکی سنسور شتاب سنج پیزوالکتریک

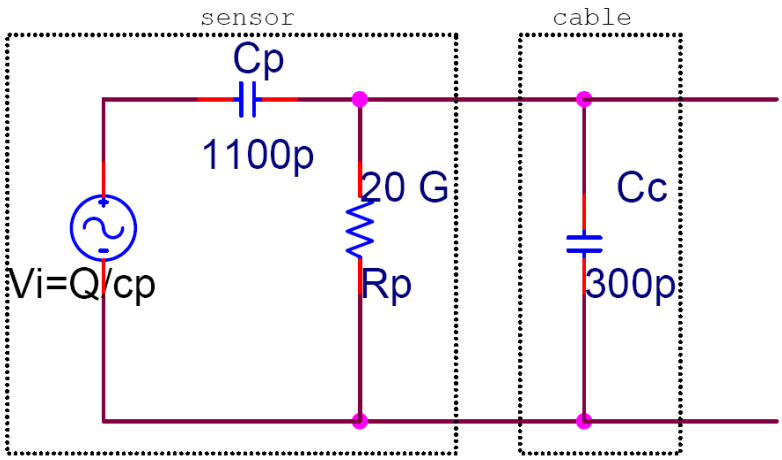
میزان بار تولیدی ناشی از شتاب با qp نشان داده شده و میزان حساسیت بر حسب pc/g در دیتاشیت سنسور اعلام می‌گردد.
در شکل (۴-۱) شاخه منبع بار و خازن موازی آن می‌تواند به منبع ولتاژ با مقدار زیر و خازن سری به صورت شکل (۴-۲) تبدیل شود.

رابطه (۴-۱)
$$V_{PE} = \frac{Q_{PE}}{C_{PE}}$$

در رابطه فوق V_{PE} حساسیت ولتاژی سنسور بر حسب mv/g می‌باشد.

پارامتر دیگری که برای سنسور پیزوالکتریک در محاسبات مداری و طراحی‌های مبدل بار به ولتاژ باید در نظر گرفت، مقاومت (R_{PE}) است که به شکل موازی با مقاومت ورودی طبقه اول مبدل بار به ولتاژ ظاهر می‌شود.
از دیگر مشخصات سنسورحداقل و حداکثر شتاب اندازه گیری و فرکانس شتاب اعمالی و میزان حساسیت سنسور به شتابهای جانبی و یا به دما بصورت پارمتر درصدی اشاره می‌گردد. البته دیگر پارامترهای فیزیکی همانند جرم و ابعاد و نصب و .... در دیتاشیت معرفی می‌شوند.
سنسور پیزوالکتریک مربوط به این پروژه مدل 4371 ساخت شرکت B&K است. مدار معادل این سنسور به همراه کابل آن در شکل (۴-۲) نشان داده شده است. مقادیر پارامترهای اشاره شده در این شکل از برگه اطلاعاتی سنسور اخذ شده است. و خازن کابل 300pF در نظر گرفته شده است.

DOI: 10.30503/nripress.2020.296



شکل٤-٢- مدار معادل سنسور انتخابی و کابل آن

ولتاژ سنسور از تقسیم حساسیت باری سنسور به خازن آن بدست می‌آید. که برای این سنسور همانطور که قبلا اشاره شده حدود 9 mV/g می‌باشد. و برای ماکزیمم شتاب 50g این ولتاژ برابر 450 mV می‌شود.

می‌توان براحتی رابطه ولتاژ خروجی دوسر سنسور به همراه کابل را مطابق رابطه زیر بدست آورد.

$$\frac{V_o}{V_i}=\frac{R_pC_ps}{R_p(C_p+C_c)s+1}$$ رابطه (۴-۲)

پاسخ فرکانسی این سنسور بصورت فیلتر بالاگذری با پارامترهای زیر است.

$$G_{hiPass}=\frac{R_pC_p}{R_p(C_p+C_c)}=\frac{C_p}{C_p+C_c}$$ رابطه (۴-۳)

$$\omega_c=\frac{1}{R_p(C_p+C_c)}$$ رابطه (۴-۴)

همانطور که ملاحظه می‌شود وجود خازن کابل گین ولتاژی سنسور را کم می‌کند.

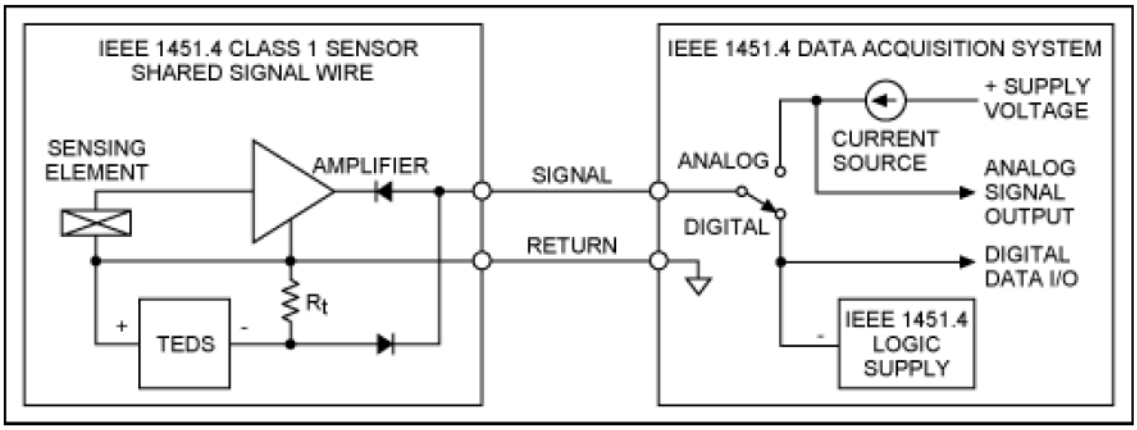
مطابق آنچه که در گزارشات پروژه های قبلی آمده است مبدل را نیز می‌توان بصورت یک تقویت‌کننده بار با گین و پاسخ فرکانسی مشخص در نظر گرفت. گین این تقویت‌کننده با توجه به خازن فیدبک آن بصورت 1/Cf می‌باشد که گین آنرا برحسب v/c و یا mv/pC می‌توان تنظیم نمود. برای مبدل ساخته شده این خازن 100pf و بنابراین گین آن 10mv/pC می‌باشد. با توجه به حساسیت باری سنسور که در حدود 10pc/g است می‌توان از مجموعه سنسور و مبدل به گین 100mv/g رسید. از طرف دیگر خازن سنسور در پاسخ فرکانسی مجموعه سنسور و مبدل تاثیرگذار بوده و روابط آن در پروژه‌های قبل بدست آمده‌است. طراحی مبدل به لحاظ مشخصات فرکانسی باید به گونه‌ای باشد که از ماکزیمم رنج پاسخ فرکانسی سنسور بهره برده و تا حد ممکن این بازه را محدود ننماید.

علاوه بر مشخصات فنی اشاره‌شده از یک مبدل، میزان نویز خروجی و اعواجاج هارمونیکی آن نیز از دیگر مواردی است که به تفصیل در گزارشات پروژه‌های قبلی ساخت مبدل به آن اشاره شده‌است. میزان نویز و اعوجاج با دمای کار مبدل می‌تواند جابجا شود لذا بصورت پارامتر با ماکزیمم مقدار در رنج دمای کاری بیان می‌شود. موارد فوق همگی بحثهای مربوط به بخش آنلوگ مبدل بوده که در این پروژه بیشتر از این به آن اشاره نمی‌شود. اما در این پروژه هدف اضافه‌کردن یک بخش دیجیتال حاوی حافظه است تا اینکه بتوان منطبق با استاندارد IEEE1451.4 دیتاشیت الکترونیک را برای مبدل فراهم ساخت که در ادامه گزارش این مبحث به تفصیل آمده‌است.

٢-٤- طراحی بخش دیجیتال مبدل

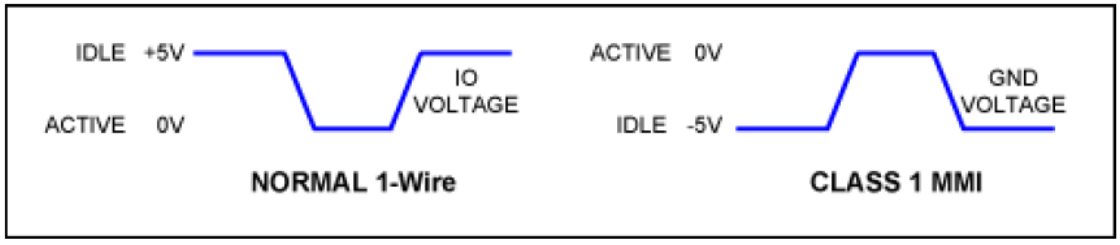
DOI: 10.30503/nripress.2020.296

در این پروژه هدف پیاده‌سازی استاندارد IEEE1451.4 برای مبدل بار الکتریکی دوسیمه‌ای است که قبلاً طراحی و ساخته شده‌است. از آنجایی که این مبدل کاملا آنالوگ و دو سیمه می‌باشد، بنابراین در این حالت بایستی از همان دو سیم ارتباطی آنالوگ، برای ارتباط دیجیتال نیز استفاده شود. بنابراین اینترفیس کلاس ۱ مد مختلط استاندارد IEEE1451.4 که مطابق شکل (۳-۱) است و برای این مبدل کاربرد دارد. همانطور که از شکل پیداست بخش دیجیتال و آنالوگ مبدل با استفاده از دیود مجزا شده‌اند بطوریکه سیگنال با پلاریته مثبت باعث فعال شدن بخش آنالوگ و منفی شدن این پلارتیه باعث فعال کردن بخش دیجیتال سنسور خواهد شد. بدیهی است که در این اینترفیس استفاده همزمان آنالوگ و دیجیتال میسر نیست.



شکل ٤-۳- کلاس ۱ اینترفیس MMI

پروتکل ارتباطی در مد دیجیتال دقیقا منطبق با باس 1-Wire با اندکی تفاوت در سطح سیگنال منطقی است. یک باس استاندارد تک سیمه با سطوح ولتاژ ۵+ و صفر ولت کار می‌کند اما در در استاندارد IEEE1451.4 این سطوح ۵- و صفر ولت می‌باشند. این تفاوت و تغییر سطح ولتاژ در شکل (۱-۴) نشان داده شده‌است.



شکل ٤-٤- سطوح سیگنال 1-Wire در مقابل کلاس ۱ استاندارد IEEE1451.4

با توجه به شکل مشاهده می‌شود که باس استاندارد 1-Wire در حالت غیرفعال ۵ ولت بوده و به هنگام فعال شدن صفر می‌شود در حالی که در استاندارد IEEE1451.4 وضعیت باس در حالت غیر فعال ۵- ولت بوده و با فعال شدن صفر می‌شود. لذا در طراحی درایور مورد نظر برای این استاندارد این مسئله باید در نظر گرفته‌شود.

نکته‌ای که در بخش دیجیتال مبدل باید دقت داشت پلاریته صحیح قطعات و کاربرد مقاومت Rt همانند شکل () است. این قطعات در واقع حافظه‌های EEPROM با باس 1-Wire هستند که اطلاعات دیتاشیت الکترونیک (TEDS) تجهیز را مطابق با الگوها و فرمت استاندارد IEEE1451.4 در خود ذخیره می‌کنند. امکان استفاده از دیگر سنسورهای 1-Wire به همراه حافظه در این استاندارد دیده شده‌است.

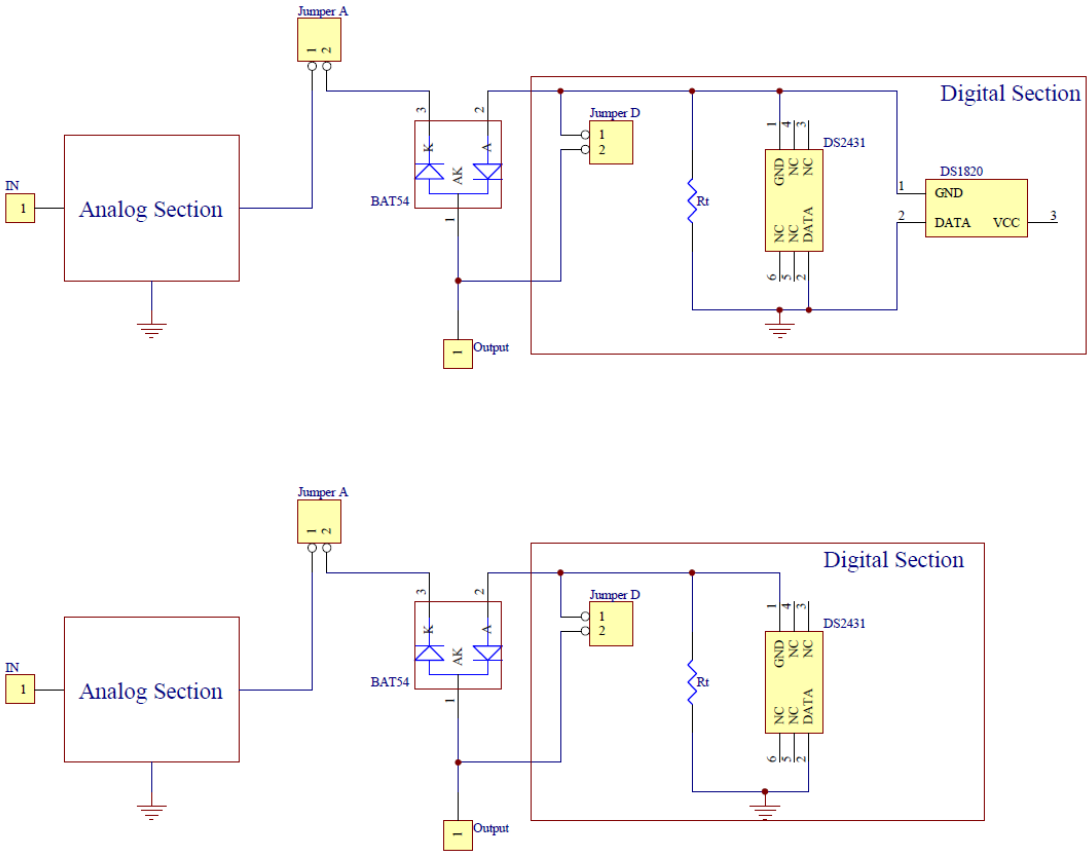
برای داشتن یک درایور منطبق با استاندارد IEEE1451.4 می‌توان از همان طرحهای درایور شبکه 1-Wire با لحاظ تبدیل سطح فوق استفاده نمود. شایان ذکر است که Master شبکه 1-Wire در واقع همان نقش درایور را در استاندارد IEEE1451.4 ایفا می‌کند.

DOI: 10.30503/nripress.2020.296

در بخش درایور نیز همانطور که اشاره‌شده می‌توان از طراحی درایورهای شبکه 1-Wire کمک جست و با استفاده از یک مدار تبدیل سطح در خروجی درایور، از آن برای درایور استاندارد IEEE1451.4 استفاده نمود. ضمناً مطابق شکل (۴-۳) از یک سویچ برای تغییر وضعیت آنالوگ به دیجیتال در درایور استفاده می‌شود.

در این پروژه برای تامین حافظه TEDS از چیپ DS2431 استفاده شده‌است. این چیپ دارای حافظه ۱۰۲۴ کیلوبیت بوده و می‌توان علاوه بر اطلاعات پایه‌ای TEDS اطلاعات کالیبراسیون را نیز در آن جای داد. به منظور پیاده‌سازی لیست ترانسدیوسر نیز سعی شده‌است با استفاده از یک چیپ سنسور دمای 1-Wire به شماره DS18B20 بر روی یک باس مشترک با حافظه، امکان تولید لیست ترانسدیوسر را نیز بررسی نمود.

به منظور ساده‌سازی تست‌ها، دو مبدل با دو بخش دیجیتال متفاوت، طراحی شده‌است بطوریکه در یکی تنها حافظه و در دیگری علاوه بر حافظه سنسور دما نیز وجود دارد تا حالتهای مختلف قابل تست باشند. شکل (۳-۳) طراحی شماتیک دو مبدل بصورت یکپارچه و قابل نصب در یک کیس را نشان میدهد.



شکل ٤-٥- شماتیک مبدل دو کاناله پیزوالکتریک به همراه بخش TEDS

بخش آنالوگ مبدل همانا طرح مبدل آنالوگ از پروژه قبلی بوده و با افزودن بخش دیجیتال به همراه قطعات جانبی مطابق شکل (۴-۵) سخت‌افزار یک مبدل مجهز شده به TEDS را مطابق استاندارد IEEE1451.4 فراهم ساخته‌است.

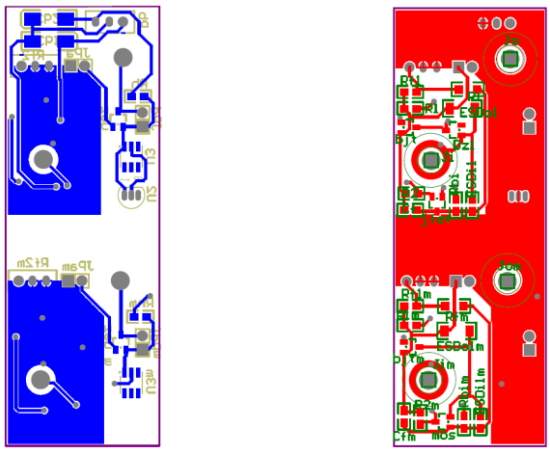
همانطور که از شماتیک (۴-۵) پیداست با قراردادن جامپرهایی در مبدل امکان جداکردن بخش آنالوگ و استفاده از بخش دیجیتال در پلاریته مثبت همانند شبکه 1-Wire نیز دیده شده است. این کار به خاطر ساده‌سازی تستهای مبدل و کمک به عیب‌یابی ساده‌تر صورت گرفته‌است.

۳-۴- ساخت برد جدید مبدل

DOI: 10.30503/nripress.2020.296

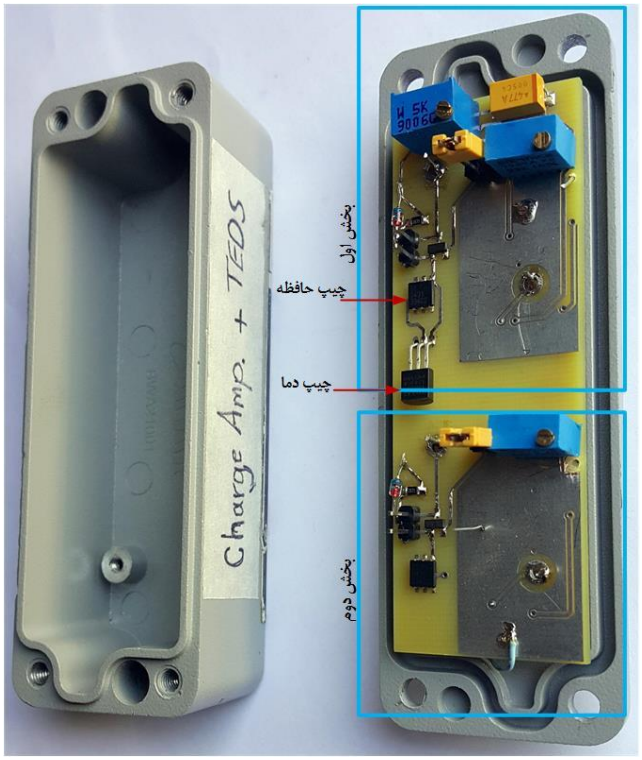
در پروژه قبلی دو مبدل با طراحی متفاوت در داخل یک کیس طراحی و ساخته شدند در اینجا نیز طرح ساخت بخش دیجیتال برای این دو مبدل صورت می‌گیرد و بگونه‌ای است که باز هم همان کیس قبلی برای مبدل استفاده شود. بعلاوه سعی شده‌است در طرح جدید ارتباطات کابلی حذف و کانکتورهای ورودی و خروجی مستقیما روی برد نصب و در کیس قبلی قابل بهره‌برداری باشند. طراحی نقشه ساخت برد به گونه‌ای است که قطعات بخش آنالوگ در یک سطح برد و قطعات بخش دیجیتال در سطح دیگر برد مونتاژ شوند. نقشه ساخت برد در شکل (۴-۶) نشان داده شده‌است. تصویر (۴-۷) کیس نهایی ساخته شده و برد درون آنرا نشان میدهد.

تست و راه اندازی بخش آنالوگ مبدل همانند پروژه های قبل انجام شده اما برای تست بخش دیجیتال مبدل نیاز به یک درایور است لذا تست بخش دیجیتال مبدل را در بخش طراحی و ساخت درایور اشاره خواهیم کرد.



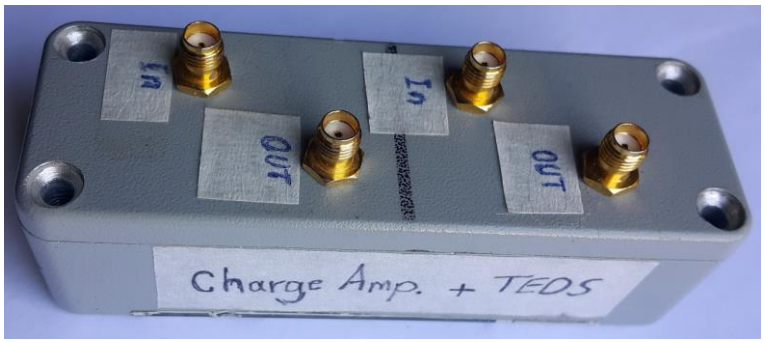
الف – پشت برد ب- روی برد

شکل ٤-٦- نقشه برد ساخته شده از مبدل (الف –پشت برد ب- روی برد)



شکل ٤-٧- تصویر برد مونتاژه شده مبدل و کیس مربوطه بصورت باز شده

DOI: 10.30503/nripress.2020.296



شکل ٤-٨- تصویری از مبدل ساخته شده نهایی و پورتهای ارتباطی آن (مجهز به حافظه TEDS)

DOI: 10.30503/nripress.2020.296

فصل پنجم : طراحی و ساخت بخشهای کمکی جهت ارتباط مبدل با کامپیوتر و قرائت داده‌ی TEDS و ارسال و دریافت دستورات مختلف

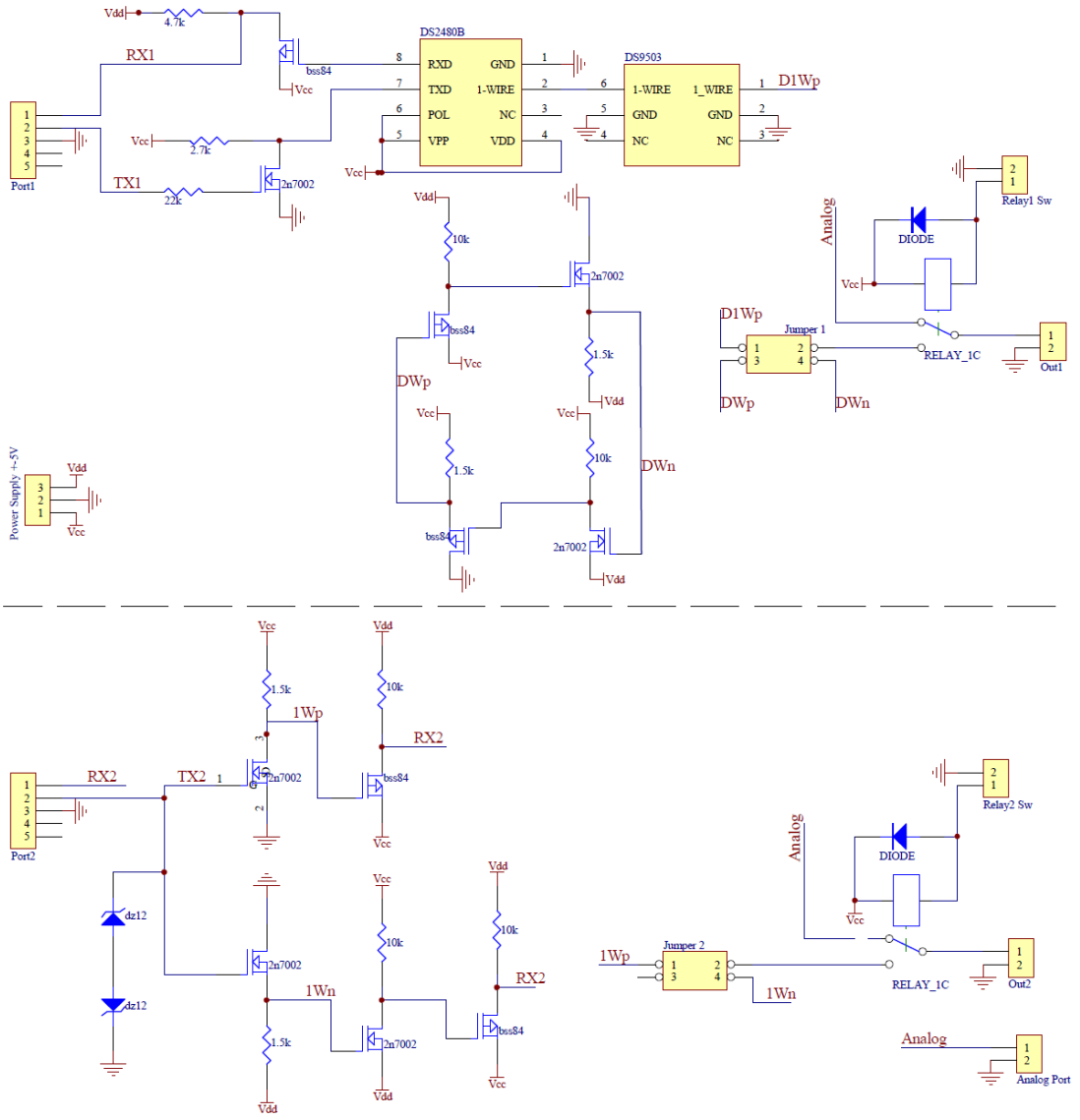
DOI: 10.30503/nripress.2020.296

۱-۵- طراحی درایور ارتباطی مبدل با کامپیوتر

در این پروژه باید درایوری مطابق استاندارد IEEE1451.4 برای مبدل پروژه مهیا کنیم. همانطور که در گزارش اول پروژه اشاره شد، درایورهای استاندارد 1-Wire می‌توانند برای این منظور با انجام اصلاحات لازم بکار روند. بنابراین درایور مبتنی بر چیپ DS2480 بدین منظور دارای مشخصات مناسب عملیاتی است. اما درایور عملیاتی مبتنی بر استاندارد IEEE1451.4 دارای سطح ولتاژ منفی است بطوریکه سطح ولتاژ شبکه بایستی در حالت غیر فعال ۵- ولت بوده و در حالت فعال به صفر ولت برسد. لذا برای تبدیل درایور 1-Wire به درایور استاندارد IEEE1451.4 لازم است یک بخش مناسب‌سازی به خروجی طرح درایور 1-Wire اضافه شده تا عمل تبدیل سطح سیگنال را انجام دهد. بخش بالایی شکل (۵-۱) طرح این مبدل با بخش تبدیل‌کننده سطح ولتاژ نشان میدهد. در طراحی امکان انتخاب خروجی تک سیمه و یا خروجی منطبق با IEEE14514.4 نیز دیده شده است.

به منظور مقایسه و یا داشتن یک درایور جایگرین ساده، از درایور دیگری که بدون چیپ DS2480 است نیز در طراحی کلی درایور استفاده شده‌است. در این قسمت، از یک مبدل مستقیم سطح سیگنال پورت سریال به TTL استفاده شده تا هم درایوری مطابق استاندارد 1-Wire را ساخته و هم با تغییر سطح سیگنال نهایی به ۵- ولت، این مبدل را برای استاندارد IEEE1451.4 کاربرد داشته‌باشد. بخش پایینی شکل (۵-۱) شماتیک مربوط به این قسمت از درایور را نشان می‌دهد.

DOI: 10.30503/nripress.2020.296



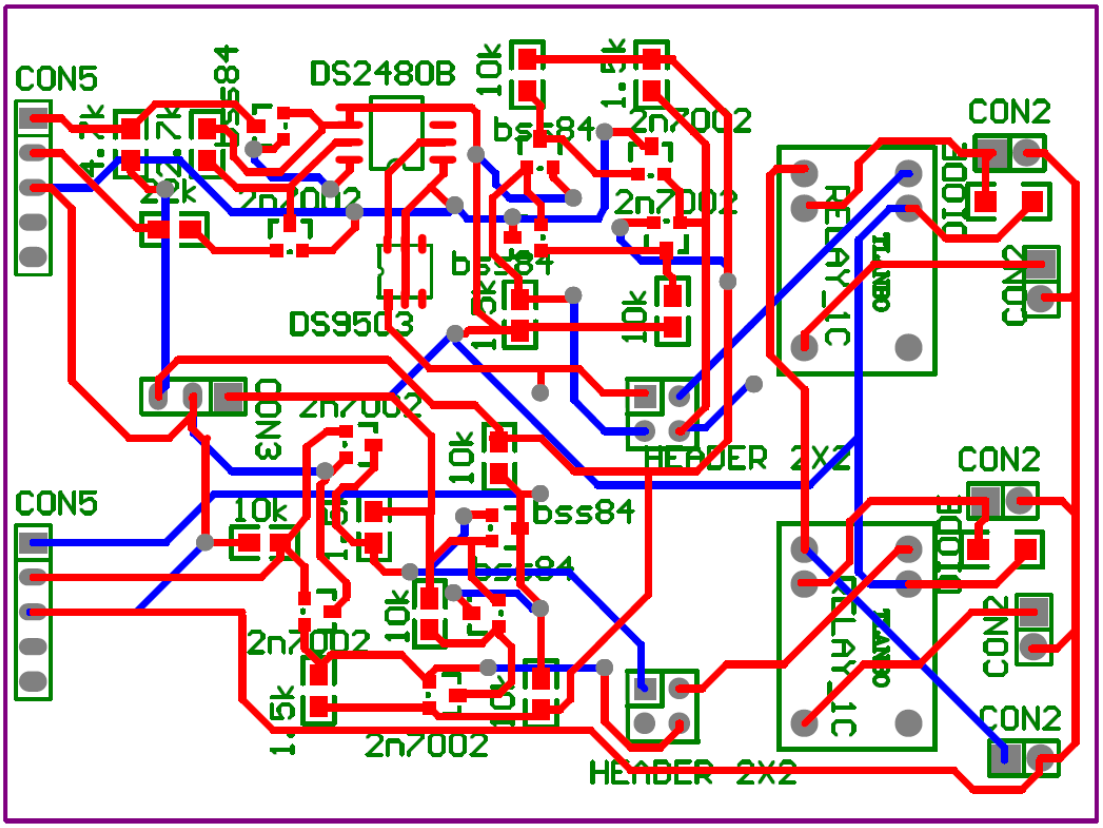
شکل ۵-۱- شماتیک مبدل دو کاناله درایور کلاس ۱و ۲ استاندارد

با درایور فوق می‌توان هم شبکه 1-Wire و هم شبکه منطبق بر IEEE1451.4 را به دو صورت مستقیم و مبتنی بر چیپ DS2480 راه‌اندازی نمود. در هر دو کانال سویچ تغییر آنالوگ به دیجیتال بصورت یک رله با کنترل تحریک دستی فراهم شده‌است. کنتاکت نرمال بسته رله برای حالت آنالوگ بکار رفته و با فعال شدن رله بخش دیجیتال آن فعال می‌شود.

۵-۲- ساخت برد درایور ارتباطی

نقشه ساخت برد طرح فوق بر مبنای استفاده از یک کیس ساده پلاستیکی تهیه شده‌است تا مجموعه درایور و پورتهای ارتباطی آن به شکل منسجمی داخل آن قرار گیرند در شکل (۵-۲) نقشه ساخت این برد نشان داده شده‌است.

DOI: 10.30503/nripress.2020.296



شکل ۵-۲- نقشه برد مدارچاپی درایور دو کاناله

۳-۵- تست و اصلاح درایور در مد شبکه تک سیمه

در بخش اول درایور که از قطعه DS2480 استفاده شده چون طراحی مطابق پیشنهاد سازنده قطعه برای شبکه 1-Wire صورت گرفته احتمال اشکال در آن کم است ولی کاربرد آن منوط به وجود نرم‌افزاری است که بتواند با این چیپ کار کند. لذا نیاز به درایور نرم‌افزاری برای کار با این قطعه وجود دارد که این نرم‌افزار یا باید توسط کاربر تولید شود و یا اینکه ممکن است درایور آماده‌ای وجود داشته باشد و باید آنرا تهیه نمود.

برای بخش دوم مبدل که بصورت مستقیم کار میکند می‌توان از همان الگوریتمهای اشاره شده در گزارش مرحله اول پروژه استفاده نمود که باز هم نیاز به داشتن یک درایور نرم‌افزاری مشهود است.

در هر حال چون تنوع قطعاتی که با این شبکه کار می‌کنند زیاد است و متناسب با هر نوع قطعه (کد خانوادگی مختلف) بایستی از کدهای مناسب آن قطعه استفاده نماییم، لذا تهیه و تولید نرم‌افزار مبتنی بر پروتکل کاربردی و دارای قابلیت شناسایی انواع قطعات، کاری زمانبر است که در پروژه حاضر دیده نشده‌است.

همانطور که در گزارش اول پروژه اشاره شده‌است پروتکل ارتباطی استاندارد IEEE1451.4 دقیقا مانند شبکه 1-wire می‌باشد لذا با توجه به این نکته، با کمک جستجوهای صورت گرفته، به یک مجموعه درایور و نرم‌افزار شناسایی برای این شبکه دست پیدا کردیم که این مجموعه نیز توسط سازنده شبکه 1-Wire بصورت مجانی در سایت این شرکت قرار گرفته است. این درایور هم می‌تواند با چیپ DS2480 کار کرده و هم می‌تواند درایور مستقیم و یا چیپ تحت usb را پشتیبانی نماید. نرم‌افزار شناسایی شبکه نیز به نام OneWireViewer بوده و تقریبا تمامی انواع قطعات را می‌شناسد. این مجموعه متعلق به شرکت Maxim بوده و در اختیار همگان قرار گرفته است. با این نرم‌افزار براحتی می‌توان یک شبکه تک سیمه شامل انواع قطعات را به لحاظ عملکردی تست و راه‌اندازی نمود. درایور این مجموعه بعد از نصب و راه‌اندازی قابلیت شناسایی اتوماتیک نوع درایور سخت‌افزاری را نیز داراست. نرم‌افزار

DOI: 10.30503/nripress.2020.296

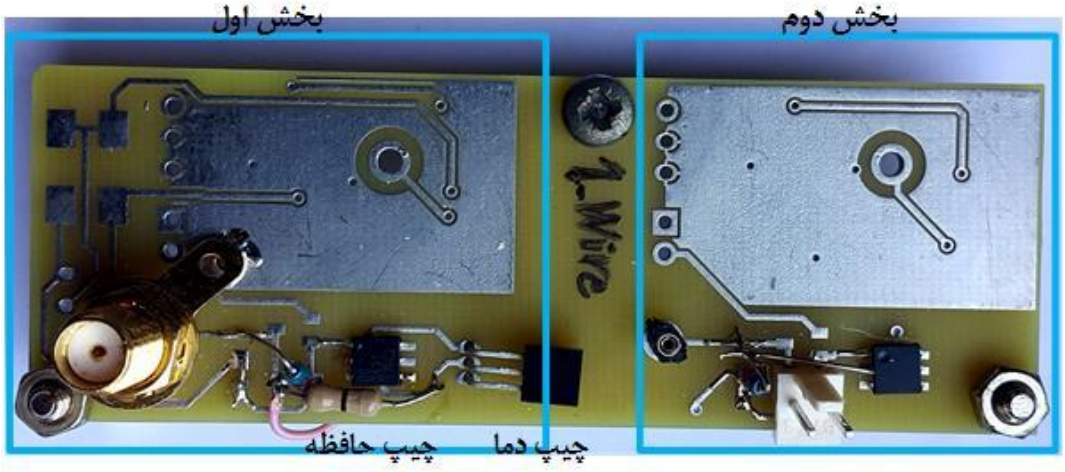
OneWireViewer انواع قطعات حافظه را شناسایی و امکان برنامه‌ریزی حافظه و یا پاک کردن و یا قفل کردن آنها را مطابق دیتاشیت فراهم می‌سازد. علاوه بر قطعات حافظه، این نرم‌افزار قابلیت شناسایی و نمایش وضعیت قطعاتی همچون سنسور دما و رطوبت و سویچ و ساعت و ... را نیز داراست.
با وجود این درایور و نرم‌افزار شناسایی آن، می‌توان مجموعه مبدل و درایور را تست و راه‌اندازی نمود.

۱-۳-۵- راه‌اندازی درایور

همانطور که در بخش قبل اشاره شد درایور طراحی شده دارای دو بخش است بخش مجهز به چیپ DS2480 و دیگری که بدون چیپ و بصورت مستقیم می‌باشد. هر دو درایور وظیفه راه‌اندازی یک شبکه تک سیمه را با استفاده از پورت سریال فراهم می‌آورند که جزئیات آن در گزارش مرحله اول [6] پروژه آمده‌است.
مطابق نقشه (۵-۱) یک نمونه از درایور مونتاژ و راه‌اندازی شد. جهت تست سخت‌افزاری بخش مستقیم از این شرط استفاده می‌کنیم که شبکه تک سیمه باید با عرض پالس حدود ۸ میکروثانیه کار کند. بر این اساس می‌توان مبنای صحت سخت‌افزار را عملکرد مناسب شبکه به پالس ۴ میکروثانیه‌ای قرار داد. بدین ترتیب با تحریک ورودی TX پورت سریال با چنین پالسی، باید سیگنالی مشابه را در سمت شبکه تک سیمه داشته باشیم. بدین ترتیب از پاسخ فرکانسی مناسب درایور مطمئن خواهیم شد و در صورت عملکرد نامناسب با تغییر المانهای مدار می‌توان به پاسخ مناسب دست یافت.
تست فوق برای بخش مجهز به چیپ DS2480 امکان پذیر نیست، چون این چیپ تغییر پروتکل روی سیگنالهای پورت سریال بوجود می‌آورد. اما از آنجا که طرح مدار مطابق با پیشنهاد سازنده چیپ است لذا به لحاظ سخت‌افزاری اطمینال زیادی بر صحت آن وجود دارد.
با اندازه‌گیری سطح ولتاژ خروجی شبکه تک سیمه در هر دو بخش درایور، باید در حالت نرمال ولتاژ مثبت ۵ ولت را داشته باشیم. از آنجایی که همواره سطح ولتاژ شبکه تک سیمه از طریق پورت سریال قابل خواندن است لذا در حالت نرمال بایستی در RX پورت سریال سطح ولتاژی بین ۴- تا ۱۱- را داشته باشیم.
همانطور که قبلا اشاره شده در هردو بخش درایور یک تبدیل سطح ولتاژی از سطح ۵ ولت به ۵- ولت وجود دارد که مطابقت آنرا برای استاندراد IEEE1451.4 فراهم می‌سازد. لذا سطوح ولتاژ این بخش نیز در حالت نرمال بایستی ۵- ولت باشد.
پس از تست‌های سخت‌افزاری فوق، ادامه تستهای درایور با نرم‌افزار درایور و نرم‌افزار شناسایی شبکه تک سیمه می‌تواند صورت پذیرد. با اتصال بخش DS2480 به پورت سریال همانطور که اشاره شد درایور فوق باید بصورت اتوماتیک آنرا شناسایی کند. با فعال سازی درایور به کمک نرم‌افزار مربوطه بنام Defut 1-Wire Net و تنظیم مناسب شماره پورت سریال روی آن می‌توان با انتخاب یکی از چهار نوع درایور موجود روی آن، عمل شناسایی را آغاز نمود. که در مورد درایور این پروژه، بخش دارای چیپ با نوع DS9097U و بخش مستقیم آن با نوع DS9097E شناسایی می‌شود.
پس از شناسایی موفق درایور توسط نرم‌افزار مربوطه، لازم است یک شبکه تک سیمه را راه‌اندازی کنیم. برای این منظور لازم است از برد مدارچاپی خام مبدل استفاده کرده و تنها قطعات بخش دیجیتال را به گونه‌ای مونتاژ کنیم که برای یک شبکه 1_Wire تنظیم و آماده گردد. از طرف دیگر با تنظیم درایور سخت‌افزاری در مد 1_wire میتوان شبکه‌ای تک سیمه از اتصال این دو ساخته و با استفاده از نرم‌افزار OneWireViewer به شناسایی قطعات متصل به شبکه دست یافت و بدین ترتیب از صحت عملکرد مجموعه مبدل و درایور در مد شبکه تک سیمه اطمینان پیدا می‌کنیم.

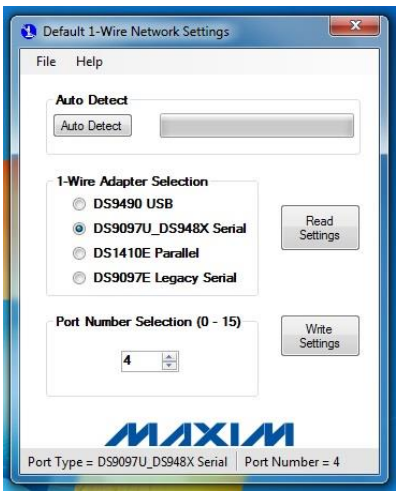
DOI: 10.30503/nripress.2020.296

در شکل (۵-۳) تصویری از یک برد کمکی را که در مد 1_wire مونتاژ و تنظیم شده‌است نشان می‌دهد. همانطور که اشاره شد درایور نیز باید در حالت 1_Wire تنظیم شده باشد.



شکل ۵-۳- مونتاژ بخش دیجیتال برد کمکی منطبق با شبکه 1-Wire

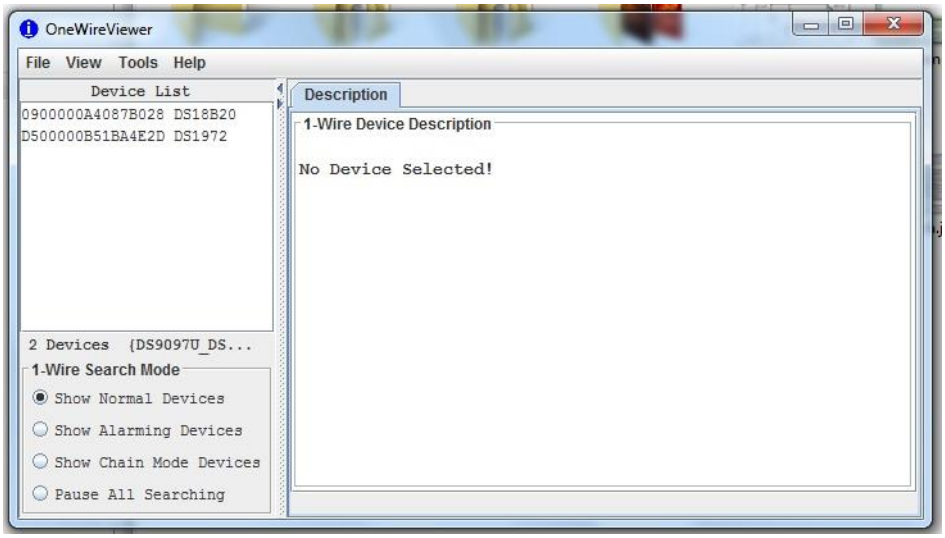
همانطور که اشاره شد نرم‌افزار اجرایی درایور شبکه به نام Default 1-Wire Network بوده که پس از اجرا و انتخاب پورت و آداپتور مناسب نتیجه شناسایی موفق شبکه تک سیمه اعلان می‌کند. نمایی از این نرم‌افزار در شکل (۵-۴) نشان داده شده‌است.



شکل ۵-٤- راه اندازی درایور نرم‌افزاری و شناسایی آداپتور شبکه تک سیمه

با اطمینان از صحت درایور شبکه می‌توان به شناسایی قطعات روی این شبکه توسط نرم‌افزار OneWireViewer پرداخت که در شکل (۵-۵) لیست قطعات آشکار شده به همراه آدرس و شماره قطعه را نشان می‌دهد. همانطور که مشاهد می‌شود دو قطعه DS1972 و DS18B20 در روی شبکه (بخش دیجیتال مبدل) آشکار شده‌است. شایان ذکر است در نرم‌افزار قطعه DS2431 با قطعه معادل آن به شماره DS1972 شناخته شده‌است.

DOI: 10.30503/nripress.2020.296

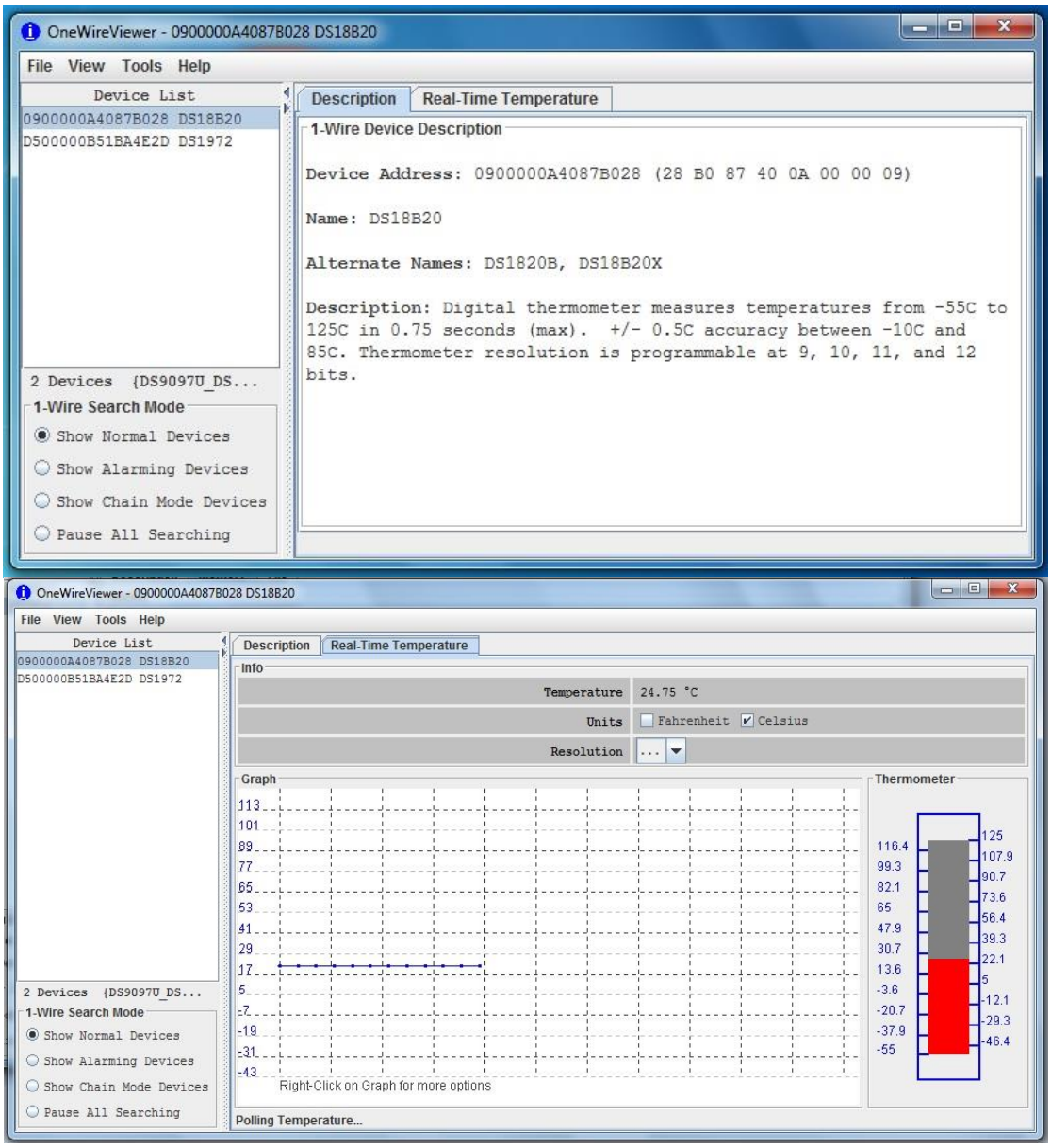


شکل ۵-۵- شناسایی قطعات روی شبکه تک سیمه

نتیجه شناسایی با هر دو بخش سخت‌افزار درایور، بایستی یکسان بوده و تنها تفاوت در نوع آداپتور تنظیمی روی نرم‌افزار درایور خواهد بود.

با توجه به شکل (۵-۵) دو قطعه روی شبکه تک سیمه که یکی حافظه و دیگری سنسور دماست شناسایی شده‌است. با انتخاب هر قطعه از بخش Device list اطلاعات مربوطه به آن قطعه از بخش دیگر نرم‌افزار براحتی قابل دستیابی است. با انتخاب قطعه دماسنج، میزان دمای آشکار شده توسط قطعه را در منوهای نرم‌افزار براحتی می‌توان مشاهده نمود. در شکل (۵-۶) -۷) تصاویری از شناسایی قطعات دماسنج و حافظه موجود روی شبکه و صفحات مربوطه به آنها نشان داده شده‌است.

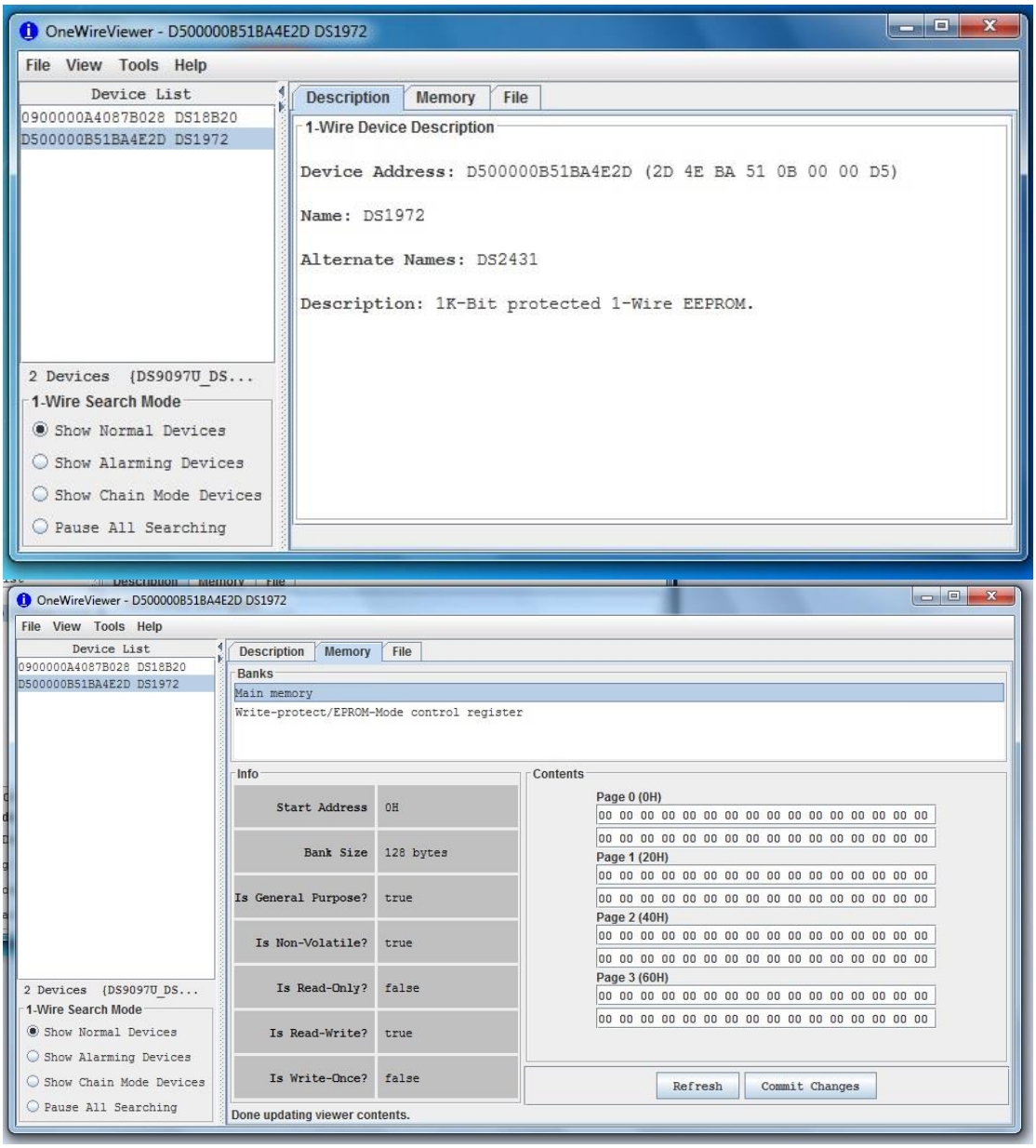
DOI: 10.30503/nripress.2020.296



شکل ۵-٦- شناسایی قطعه دماسنج و توضیحات و اندازه دمای آشکار شده

با اتخاب حافظه، می‌توان علاوه بر توضیات کلی قطعه، اطلاعات ثبت شده در حافظه را به دلخواه برنامه‌ریزی و یا اصلاح نمود و در نهایت نیز چیپ را قفل نموده تا از دسترسی کاربر جهت تغییر خارج شود(مطابق دیتاشیت قطعه). در شکل (۵-۷) منوهای مربوط به قطعه حافظه را نشان میدهد.

DOI: 10.30503/nripress.2020.296



شکل ۵-۷- توضیحات مربوط به قطعه حافظه و نمایش داده‌های ذخیره شده در آن

بدین ترتیب تا بحال از صحت درایور و برد کمکی در حالت شبکه تک سیمه (غیر فعال بودن بخش تبدیل سطح ولتاژ) اطمینان پیدا شده‌است. لذا در ادامه با فعال کردن بخش تبدیل سطح ولتاژ، تست درایور و مبدل را ادامه خواهیم داد.

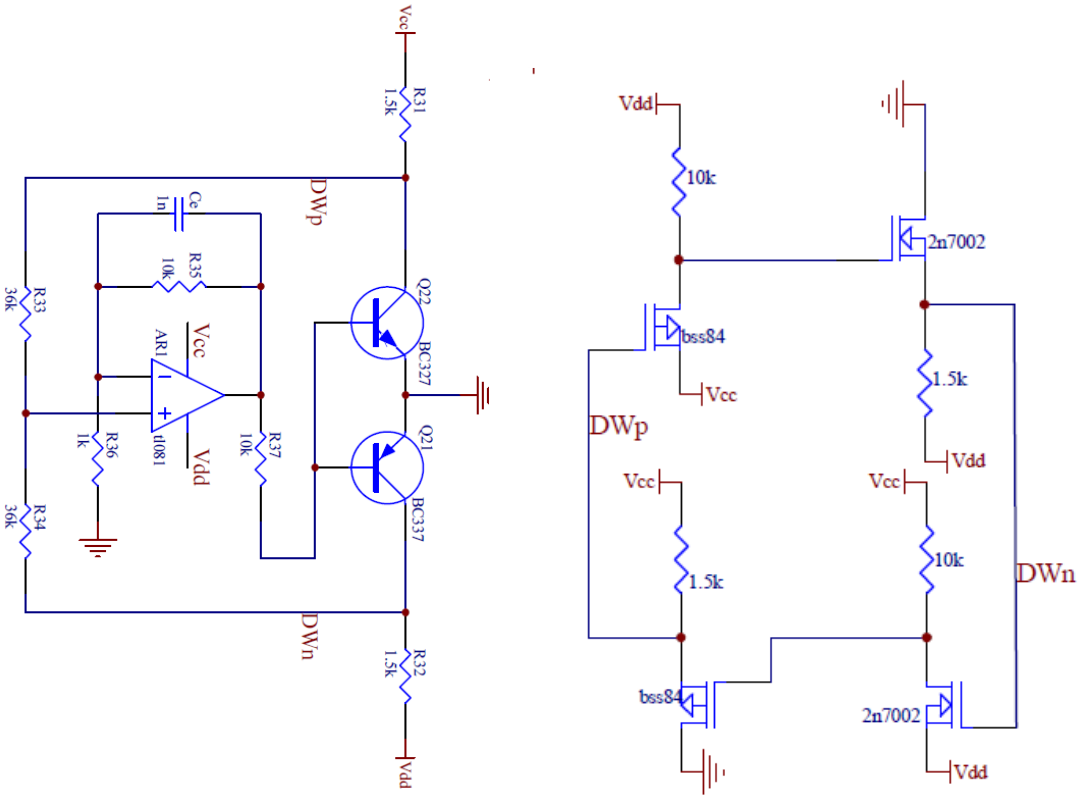
٤-٥- تست و اصلاح درایور و مبدل در استاندارد IEEE1451.4

پس از اطمینان از صحت کارکرد درایور در مد 1_Wire می‌توان به تست مبدل و درایور در مد IEEE1451.4 پرداخت. برای این منظور کافی است در هر دو بخش درایور از تبدیل کننده سطح ولتاژ استفاده کرده و در واقع این بخش مدار در مسیر خروجی قرار گیرد. حال با این درایور می‌توان مبدل ساخته‌شده این پروژه را که مبتنی بر استاندراد IEEE1451.4 می‌باشد نیز تست نمود. برای اطمینان، لازم است قبل از اتصال مبدل به درایور به کمک ولتمتر از وجود سطح ولتاژ ۵- ولت در خروجی درایور مطمئن شویم.

DOI: 10.30503/nripress.2020.296

۱-۴-۵- راه‌اندازی و تست مبدل و درایور

با اتصال مبدل به هر یک از دو بخش درایور متاسفانه عملکرد درستی از طریق نرم‌افزار OneWireViewer مشاهده نشد و نرم افزار نتوانست قطعات روی شبکه را شناسایی کند. حال در ادامه به عیب یابی موضوع می‌پردازیم.
برای عیب یابی بهتر است از بخش درایور نوع مستقیم که تبدیل پروتکل ندارد شروع کنیم. این مسئله عیب یابی را سرعت می‌بخشد. با قطع مبدل از درایور و اعمال پالس مستقیم یک مگاهرتزی به جای سیگنال TX و بررسی وضعیت باس معلوم شد که عملکرد این بخش درایور صحیح است بنابراین باید مشکل از مبدل بوده باشد. در مبدل نیز تنها تفاوت با تست حالت 1-Wire این است که در مدار مقاومت Rt اضافه شده‌است با تغییر و مناسب سازی این مقاومت (3.3k) و تست مجدد به پاسخ مناسب از مجموعه مبدل و درایور مستقیم می‌رسیم. حال که از صحت مبدل اطمینال یافتیم می‌توان آنرا با درایور مجهز به چیپ DS2480 نیز مورد تست قرار دهیم. با اتصال مبدل به این بخش از درایور، پاسخ صحیحی دریافت نمی‌شود. با اطمینال از صحت کار مبدل، مشکل الزاما در بخش درایور آن است و چون صحت بخش اولیه درایور در حالت 1-Wire در تستهای قبلی معلوم شده‌بود لذا تنها بخشی که به این تست اضافه شده می‌تواند ایجاد کننده مشکل باشد. به عبارت دیگر بخش تبدیل کننده سطح ولتاژ از ۵ ولت به ۵- ولت بایستی ایراد داشته باشد. به منظور تست دقیقتر این بخش از مدار، ورودی آنرا از بخش ماقبل جدا کرده و توسط یک مولد سیگنال پالسی یک مگاهرتزی با سطح ۵ ولت تحریک می‌کنیم در صورت صحت مدار، سیگنال خروجی آن نیز باید پالسی همانند ورودی با سطح ولتاژ ۵- ولت باشد. اما آنچه که مشاهد شد این بود که خروجی همواره صفر باقی مانده و تغییر وضعیتی نداشت. با بررسی دقیقتر مدار، معلوم شد که اشکال در طراحی مدار بوده و لذا طراحی این بخش به صورت دیگری که در شکل (۵-۸) نشان داده شده‌است تغییر یافت.

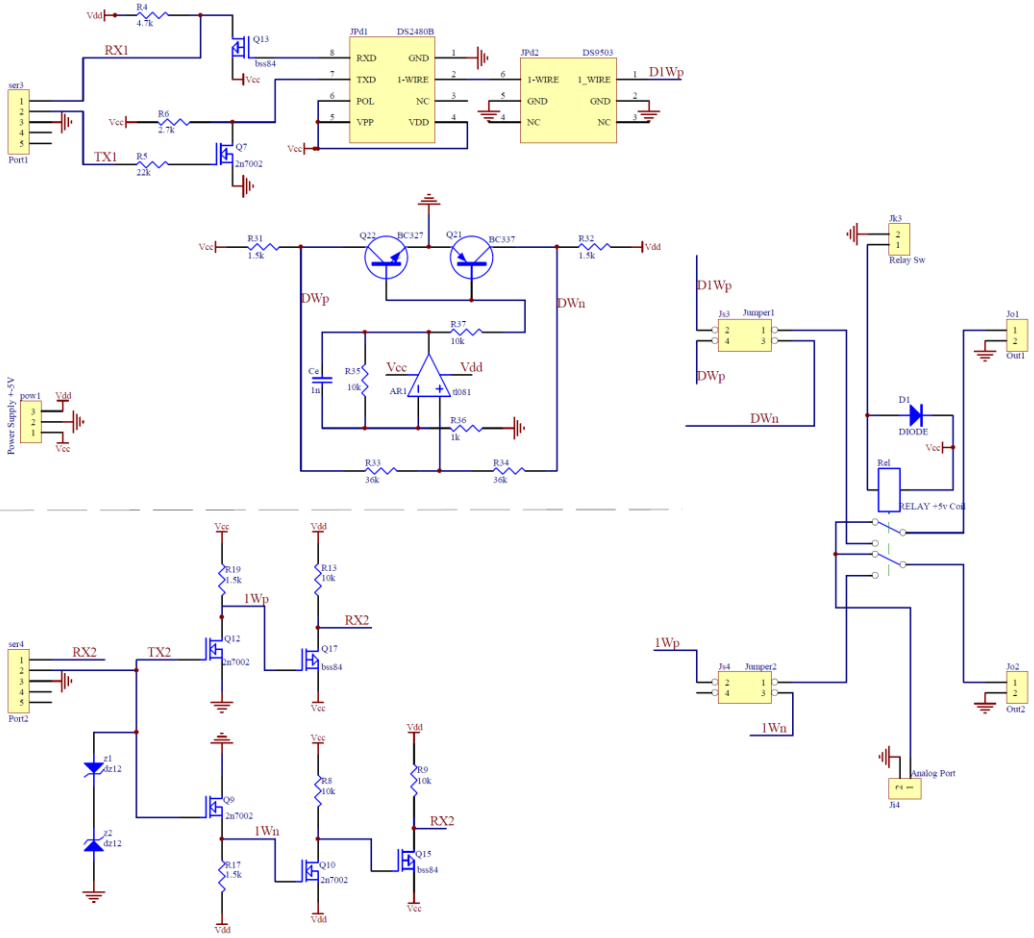


ب- طرح اصلاحی الف -طرح اولیه

شکل ۵-۸- نقشه شماتیک بخش تبدیل سطح ولتاژ در نمونه اولیه(الف) و نمونه اصلاحی (ب)

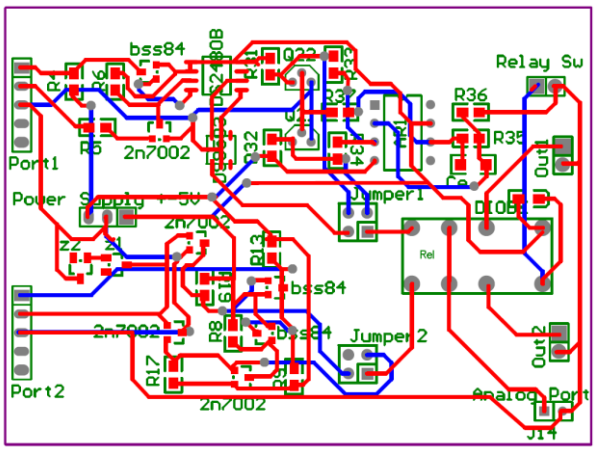
DOI: 10.30503/nripress.2020.296

با ساخت یک نمونه مدار تستی جدید و قراردادن در مدار مبدل عملکرد مناسب آن نیز مورد تایید واقع شد. بدین ترتیب طراحی مبدل در این بخش مطابق فوق اصلاح نهایی گردید. بعلاوه در اصلاح نهایی درایور، به جای دو رله از یک رله با دو کنتاکت مجزا استفاده شده تا مصرف و هزینه مدار نیز کاهش یابد. در شکل (۵-۹) نقشه شماتیک از طرح نهایی درایور نشان داده شده‌است.



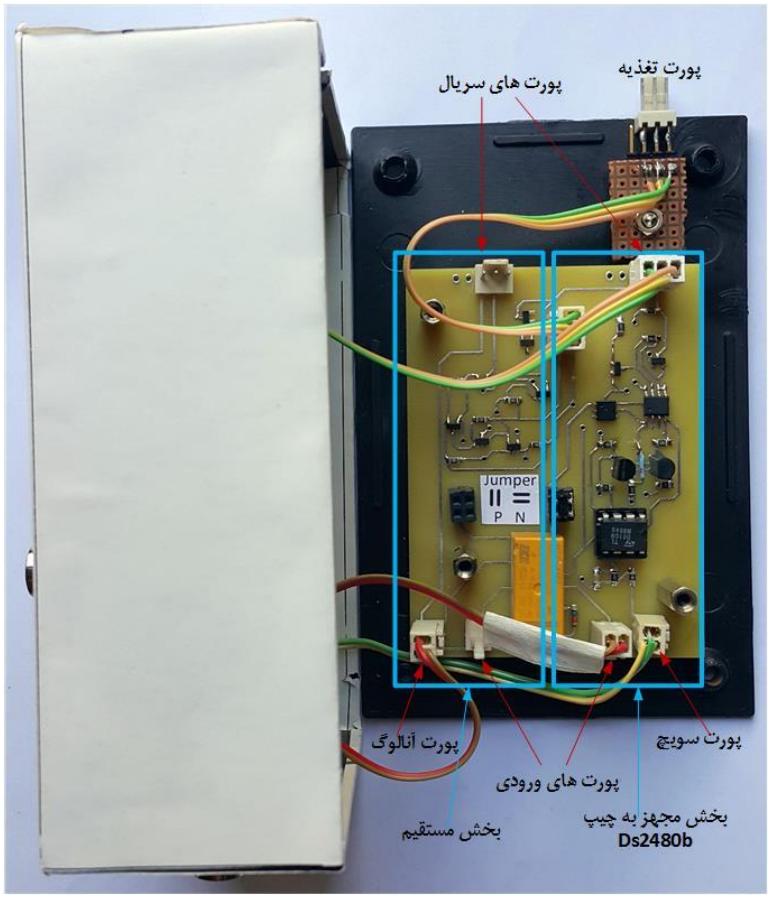
شکل ۵-۹- شماتیک مدار نهایی درایور

نقشه ساخت مدار درایور در شکل (۵-۱۰) نشان داده شده‌است و در شکلهای (۲-۱۱) و (۲-۱۲) تصویر این برد را در داخل کیس مربوطه و تصویر نهایی دستگاه را نشان داده است.



شکل ۵-۱۰- نقشه ساخت برد از درایور نهایی پروژه

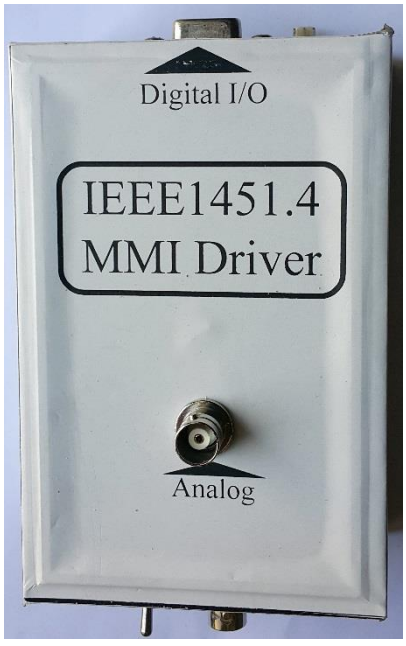
DOI: 10.30503/nripress.2020.296



شکل ۵-۱۱- تصویر برد درایور مونتاژی و چگونگی پیاده‌سازی در کیس





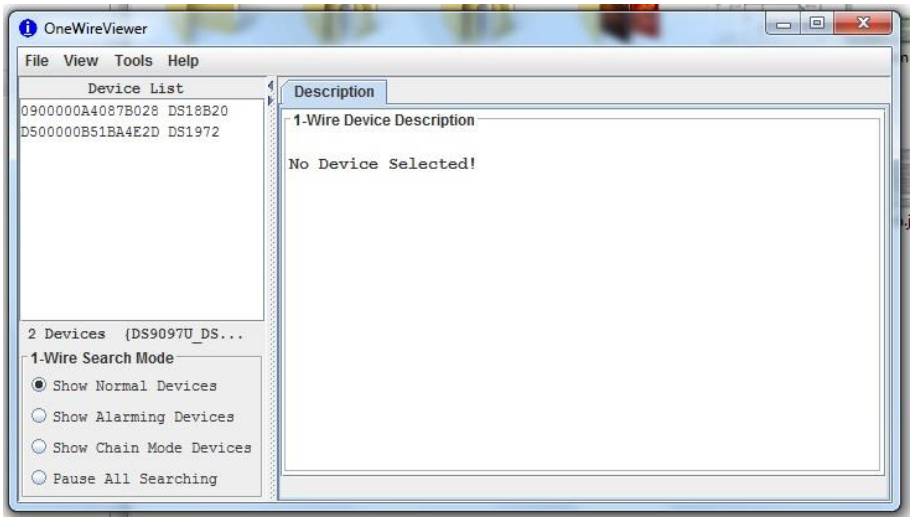


شکل ۵-۱۲- تصویر نماهای مختلف و پورتهای داریور نهایی

DOI: 10.30503/nripress.2020.296

۲-۴-۵- تست نهایی عملکرد درایور و مبدل پروژه

با اصلاح مبدل و درایور در مد IEEE1451.4 نتیجه تست اتصال ایندو به همدیگر و آشکارسازی آنها در نرم‌افزار OneWireViewer در شکل (۵-۱۳) نشان داده‌است.



شکل ۵-۱۳- شناسایی قطعات و نودهای مبدل در مد دیجیتال

همانطور که شکل پیداست دو نود و یا دو قطعه شناسایی شده‌اند که یکی حافظه DS2431 بوده و دیگری سنسور دماسنج به شماره DS1820 می‌باشد.

تا بحال از صحت سخت‌افزاری مبدل و درایور اطمینال یافته و به کمک نرم‌افزارهای اشاره شده توانستیم یک شبکه را راه‌اندازی و قطعات روی آنرا شناسایی و به اطلاعات هر یک دسترسی یافته و یا آنها را تغییر دهیم. در ادامه می‌خواهیم با پیاده‌سازی دیتاشیت الکترونیک در قطعه حافظه، مبتنی بر الگوهای استاندارد IEEE1451.4 ، یک ترانسدیوسر منطبق بر این استاندارد را خواهیم داشت. با توجه به تستهای فوق بخش دیجیتال مبدل، به لحاظ سخت‌افزاری پتانسیل لازم برای پیاده‌سازی دیتاشیت الکترونیک TEDS را داراست. آنچه که باقی مانده در واقع تماما نرم‌افزاری بوده و باید اطلاعات قطعات حافظه را منطبق بر الگوهای استاندارد برنامه‌ریزی نماییم. مشروح این الگوهای استانداردی در گزارش مرحله اول [6] پروژه بیان شده‌است. بنابراین در ادامه سعی می‌کنیم که اطلاعات فنی یک ترانسدیوسر را مطابق الگوهای تعریف شده در استاندارد، در حافظه دیجیتال ترانسدیوسر ثبت نماییم. تا به یک ترانسدیوسر حاوی دیتاشیت الکترونیک TEDS دست یابیم که هدف نهایی این پروژه است.

DOI: 10.30503/nripress.2020.296

فصل ششم: پیاده‌سازی دیتاشیت الکترونیک منطبق بر الگوهای استانداردی

DOI: 10.30503/nripress.2020.296

۱-٦- مقدمه

همانطور که در گزارش اول پروژه اشاره شده‌است یک ترانسدیوسر منطبق بر استاندارد IEEE1451.4 دارای یک بخش دیجیتال است که گذرگاه ارتباطی آن اگر با بخش آنالوگ بصورت مشترک استفاده شود دارای اینترفیس کلاس ۱ و اگر کاملا مستقل از آنالوگ باشد دارای اینترفیس کلاس ۲ است. در کلاس ۲ ، شبکه ارتباطی با بخش دیجیتال و پروتکل ارتباطی دقیقا منطبق بر استاندارد شبکه 1-Wire است اما در کلاس ۱، که مربوط به این پروژه نیز هست سطح لاجیک ارتباطی با شبکه تک سیمه تفاوت دارد(سطح ولتاژ ۵- ولت به جای ۵ ولت). بخش دیجیتال یک ترانسدیوسر می‌تواند از چندین نود داخلی که حداقل دارای یک نود حافظه است تشکیل یافته‌باشد. در حالت مینیمم، ترانسدیوسر می‌تواند دارای یک نود از نوع حافظه بوده که اطلاعات دیتاشیت الکترونیک را مطابق الگوهای معرفی شده در استاندارد IEEE1451.4 در حافظه خود ثبت و ذخیره نماید. الگوهای استانداردی به گونه‌ای طراحی شده‌اند که در یک حجم کم، بیشترین اطلاعات را میتوان در آن ذخیره نمود. برای پیاده‌سازی اطلاعات روی حافظه دیجیتال لازم است نرم‌افزاری منطبق بر الگوهای استانداردی تهیه شوند تا این امر اجرایی گردد. در تهیه و تفسیر الگو از زبان الگونویسی TDL استفاده می‌شود که در گزارش مرحله اول پروژه به تفصیل به آن اشاره شد. به کمک این زبان امکان پیاده‌سازی انواع الگوهای ثبت اطلاعات و یا تفسیر الگوهای خوانده شده براحتی میسر میشود. اما با توجه به تنوع و تعداد زیاد الگوها، تهیه و تولید این نرم‌افزار کاری زمانبر بوده که در این پروژه دیده نشده‌است. لذا برای انجام فعالیت های باقی مانده، جستجوهای زیادی به دنبال نرم‌افزارهای آماده‌ای که با این استاندارد کار می‌کنند صورت گرفت. خوشبختانه دو نرم‌افزار از شرکتهای سازنده سنسور یافت شد این دو شرکت خود دارای سنسورهایی با قابلیت TEDS منطبق بر استاندارد IEEE1451.4 هستند. باید اشاره کرد که خوشبختانه این دو نرم‌افزار نیز بر اساس درایور شرکت Maxim می‌توانند کار کنند. بدین ترتیب با اجرای این درایور، این دو نرم‌افزار نیز قابل اجرا خواهند بود. نرم‌افزار اول از شرکت معتبر و معروف B&K بوده و بنام IEEE1451.4 TEDS editor می‌باشد. این نرم‌افزار فقط با درایور خود این شرکت تمام قابلیتهای خود را ارائه میدهد و الا فقط محدود به شناسایی ترانسدیوسرهای منطبق با این استاندارد و یا برنامه‌ریزی الگوها با کد سازنده B&K می‌باشد. بعلاوه این نرم‌افزار تمامی الگوهای استاندری را پوشش نمی‌دهد بلکه به خاطر محدوده کاری شرکت (حوزه سنسورهای ارتعاشی) فقط الگوهای مربوط سنسورها ارتعاشی را پشتیبانی می‌کند.

DOI: 10.30503/nripress.2020.296

اما نرم‌افزار دوم از شرکت DeweSoft بوده و بنام DeweSoft TEDS editor می‌باشد. در این نرم‌افزار نیز برای دستیابی به تمامی قابلیتهای آن، نیاز به خرید مجوز از شرکت مربوطه می‌باشد. و الا می‌توان با آن انواع الگوهای استانداردی را تنها با شماره شناسه ID این شرکت پیاده‌سازی نمود بعلاوه یک نمونه از الگوی کالیبراسیون را نیز داراست که پیاده‌سازی و استفاده از آنها در بخش بعدی نشان داده خواهد شد.
با بررسیهای بعمل آمده نرم‌افزار DeweSoft بروزتر از نرم‌افزار شرکت B&K بوده و تقریبا تمامی الگوهای استاندرادی را پشتیبانی میکند. به عبارت دیگر تقریبا اکثر الگوهای پیاده‌سازی شده توسط نرم‌افزار B&K ، با نرم‌افزار DeweSoft قابل شناسایی هستند اما برعکس آن همواره میسر نیست.

۲-٦- قابلیت های نرم‌افزار IEEE1451.4 TEDS editor از شرکت B&K

این نرم‌افزار همانطور که اشاره شده بر اساس درایور شرکت Maxim کار میکند. با نصب و اجرای این نرم‌افزار، قابلیتهای زیر از قابل ارائه هستند:

- شناخت نودهای اتصال یافته به شبکه شامل آدرس و نوع قطعه در هر نود (نودهای حافظه بدون اطلاعات و یا مغایر با استاندارد بصورت not assigned معرفی میشوند)
- شناخت نودهای دارای اطلاعات TEDS شامل نام الگو و نودهای مربوط به آن
- شناخت نود لیست و نمایش آنها بعنوان یک مجموعه واحد
- محدودیت الگوهای قابل شناسایی به سنسورهای پیزوالکتریک و تقویت‌کننده‌های بار
- قابلیت برنامه‌ریزی الگوهای موجود با شناسه کارخانه نامشخص
- قابلیت کار با ترانسدیوسرهای ویژه این شرکت با قابلیت برنامه‌ریزی (با وجود مجوز)
- قابلیت تولید لیست نود (با وجود مجوز)
- در اختیار گرفتن تمامی قابلیتهای نرم‌افزار (با وجود مجوز)

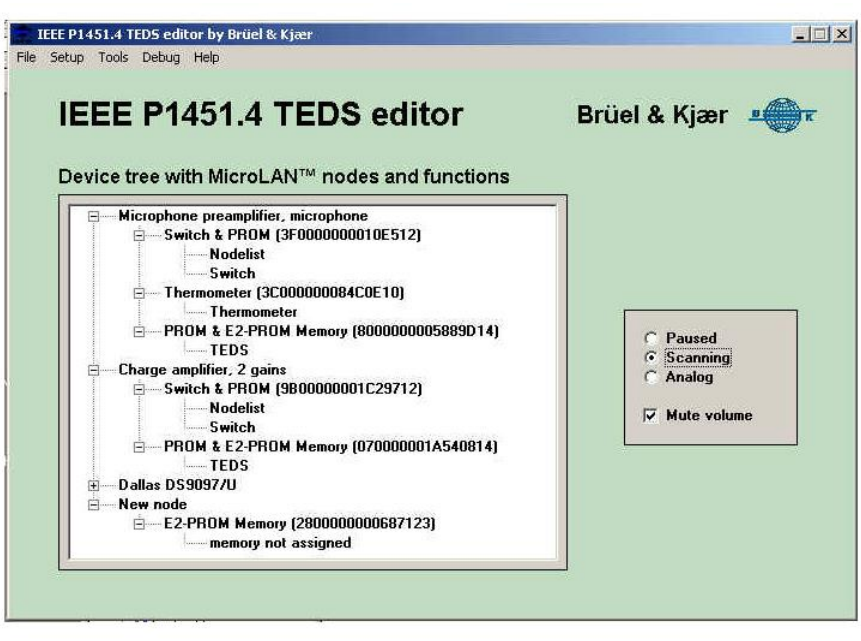
در شکل (۶-۱) لیست الگوهای قابل برنامه‌ریزی و شناسایی و در شکل (۶-۲) صفحه اصلی نرم‌افزار بصورت نمونه نشان داده شده‌است. با توجه به شکل (۶-۲) مشخص است که این نرم‌افزار قابلیت شناسایی نود لیست را دارا بوده و نودهای زیر مجموعه هر یک از ترانسدیوسرها را نیز نشان می‌دهد. بعلاوه نوع قطعه متصل به هر نود و شماره آدرس آنها نیز نشان داده می‌شود. اگر نود حافظه مربوط به یک ترانسدیوسر بدون اطلاعات و یا منطبق بر استاندارد نباشد آنرا بصورت یک حافظه بدون برنامه ریزی Not Assigned نشان میدهد. برای این قطعات حافظه، امکان اختصاص یک الگو از کتابخانه خود نرم‌افزار وجود داشته و تولید الگو را با پارامترهای پیش فرض خود امکان پذیر می‌سازد. اما تغییر پارامتر و یا استفاده از قابلیت های پیشرفته‌تر نیازمند اخذ مجوز از شرکت B&K می‌باشد.

DOI: 10.30503/nripress.2020.296

Accelerometer
Accelerometer, transfer function
Accelerometer, 2 gains
Accelerometer, 2 gains, transfer function
Charge amplifier
Charge amplifier, accelerometer
Charge amplifier, 2 gains
Charge amplifier, 2 gains, accelerometer
Charge amplifier, transfer function, accelerometer
Charge amplifier, 2 gains, transfer function, accelerometer
Microphone, integrated preamplifier
Microphone, integrated preamplifier, transfer function
Microphone, integrated preamplifier, 2 gains
Microphone, integrated preamplifier, 2 gains, transfer function
Microphone preamplifier
Microphone preamplifier, microphone
Microphone preamplifier, 2 gains
Microphone preamplifier, 2 gains, microphone
Microphone preamplifier, transfer function
Microphone preamplifier, transfer function, microphone
Microphone preamplifier, 2 gains, transfer function
Microphone preamplifier, 2 gains, transfer function, microphone
Microphone, integrated preamplifier - ext. sens.
TachoProbe
CCLD - Tacho Probe
Accelerometer, V1.0
Accelerometer, transfer function, V1.0
Force transducer, V1.0
Force transducer, transfer function, V1.0
Charge amplifier (ID=26), V1.0
Charge amplifier, attached Accelerometer, V1.0
Charge amplifier, attached Accelerometer, transfer function, V1.0
Charge amplifier (ID=26), 2 gains, V1.0
Charge amplifier, attached Accelerometer, 2 gains, V1.0
Charge amplifier, attached Accelerometer, 2 gains, transfer function, V1.0
Charge amplifier (ID=43), V1.0
Charge amplifier, attached Force transducer, V1.0
Charge amplifier, attached Force transducer, transfer function, V1.0
Charge amplifier (ID=43), 2 gains, V1.0
Charge amplifier, attached Force transducer, 2 gains, V1.0
Charge amplifier, attached Force transducer, 2 gains, transfer function, V1.0
Microphone with integrated preamplifier, V1.0
Microphone with integrated preamplifier, systemtest, V1.0
Microphone with integrated preamplifier, transfer function, V1.0
Microphone with integrated preamplifier, transfer function, systemtest, V1.0
Microphone, integrated preamplifier, 2 gains, V1.0
Microphone, integrated preamplifier, systemtest, 2 gains, V1.0
Microphone, integrated preamplifier, transfer function, 2 gains, V1.0
Microphone, integrated preamplifier, transfer funct., systemtest, 2 gains, V1.0
Microphone preamplifier, V1.0
Microphone combined with preamplifier, V1.0
Microphone combined with preamplifier, systemtest, V1.0
Microphone combined with preamplifier, transfer function, V1.0
Microphone combined with preamplifier, transfer function, Systemtest, V1.0
Microphone preamplifier, 2 gains, V1.0
Microphone & preamplifier, 2 gains, V1.0
Microphone & preamplifier, systemtest, 2 gains, V1.0
Microphone & preamplifier, transfer function, 2 gains, V1.0
Microphone & preamplifier, transfer function, Systemtest, 2 gains, V1.0
Capacitive microphone, V1.0

شکل ٦-١- لیست الگوهای پشتیانی شده توسط نرم‌افزار B&K

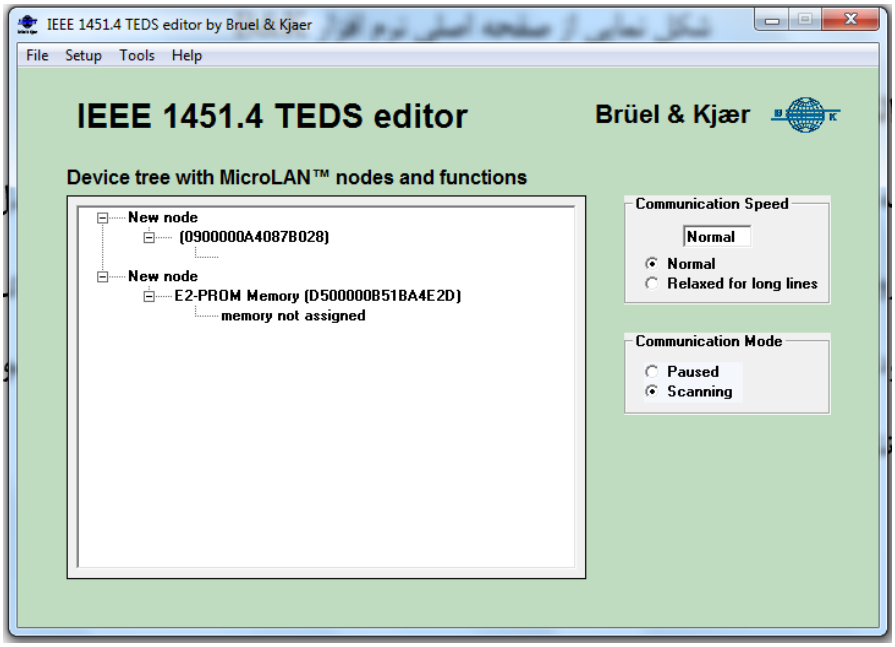
DOI: 10.30503/nripress.2020.296



شکل ٦-٢- نمایی از صفحه اصلی نرم‌افزار B&K

۱-۲-۶- پیاده‌سازی الگوهای نمونه استانداردی روی مبدل پروژه با نرم‌افزار شرکت B&K

در این بخش به کمک این نرم‌افزار به پیاده‌سازی الگوهای استانداردی روی حافظه مبدل می‌پردازیم. با اتصال مبدل پروژه که هنوز حافظه آن برنامه‌ریزی نشده‌است تنها دو نود جدا از همدیگر روی نرم‌افزار مطابق شکل (۶-۳) آشکار می‌شود که نود حافظه بصورت بدون برنامه‌ریزی و یا Not Assigned نشان داده شده و نود سنسور دما چون شناسایی نشده فقط بصورت آدرس نود بدون اطلاعات مربوطه نشان داده شده‌است. علت عدم شناسایی نود دماسنج بخاطر بروز نبودن نرم‌افزار در شناسایی قطعات جدید است.



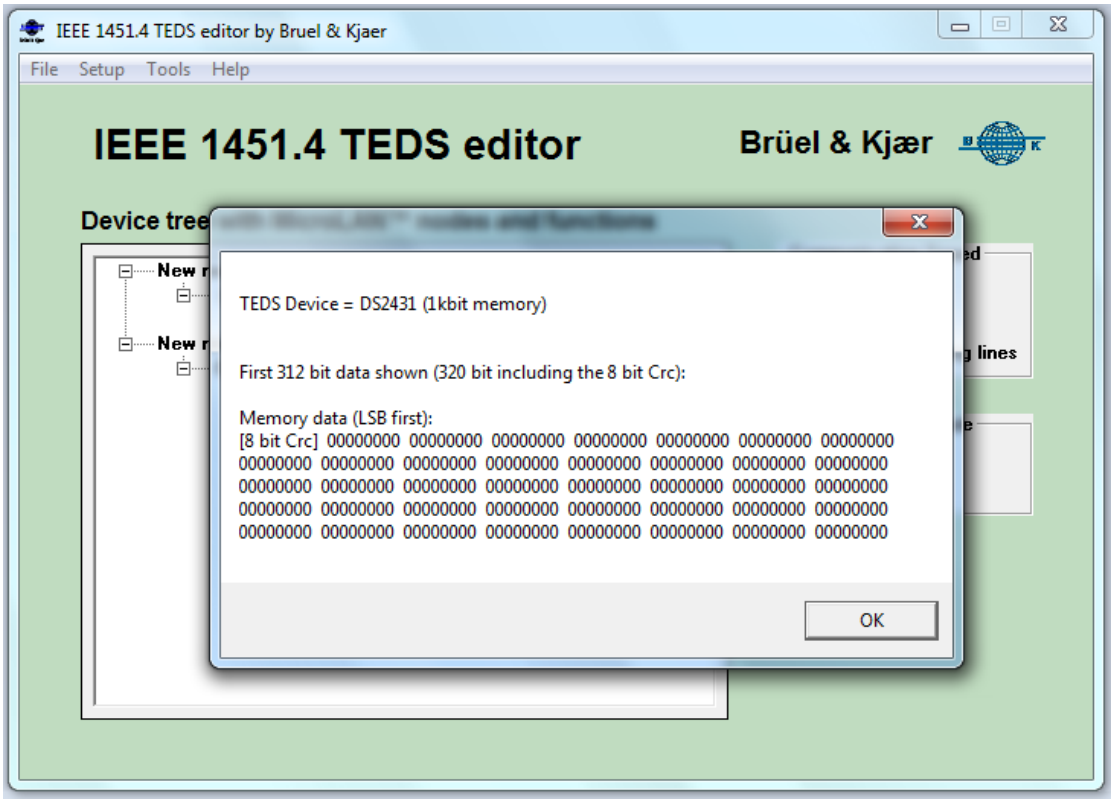
شکل ٦-٣- وضعیت شناسایی مبدل بدون دیتاشیت با نرم‌افزار B&K

DOI: 10.30503/nripress.2020.296

با انتخاب حافظه برنامه‌ریزی نشده ابتدا منویی از وضعیت داده‌های درون حافظه را مطابق شکل (۶-۴) نشان میدهد. با زدن کلید Ok وارد منوی انتخاب الگو مطابق شکل (۶-۵) می‌شویم. در این منو با انتخاب نوع الگو و سازنده آن می‌توانیم الگوی مورد نظر مطابق شکل (۶-۶) بسازیم. همانطور که از شکل (۶-۵) پیداست در اینجا با توجه با فانکشن مبدل این پروژه، دسته Charge amplifier انتخاب و الگوی استانداردی با شماره شناسه ۲۶ که مربوطه به تقویت‌کننده بار است نهایی شده‌است. سپس با انتخاب یک سازنده، اطلاعات لازم برای تولید چارچوب الگو کامل شده و میتوان با زدن کلید Get new template الگوی مورد نظر در منویی جدید با پارامترهای پیش فرض همانند شکل (۶-۶) تولید نمود. در این منو با زدن زبانه Node و انتخاب Write node امکام برنامه‌ریزی حافظه مذکور فراهم شده‌است.

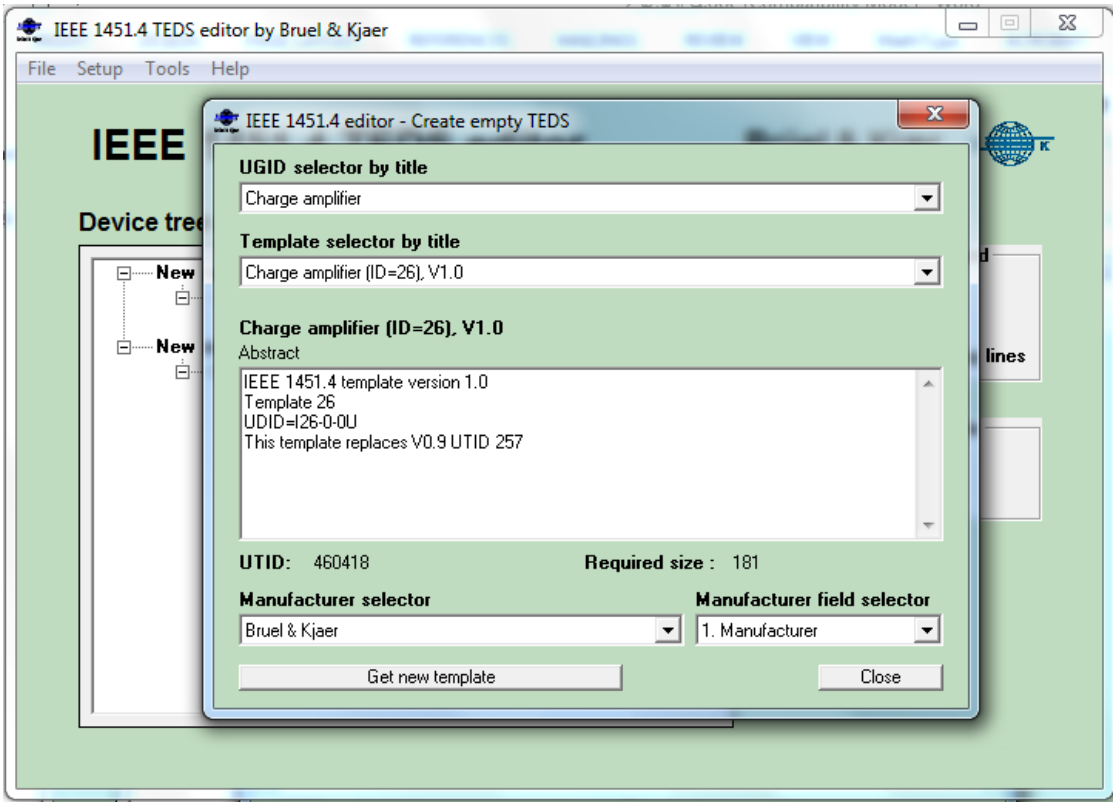
همانطور که از شکل (۶-۶) پیداست مجموعه اطلاعات الگو شامل بخش اولیه یعنی الگوی پایه و سپس یک الگوی استانداردی با پارامترهای اطلاعاتی مربوطه است و در انتها با یک الگوی کاربری خاتمه یافته است. الگوی پایه در تمامی ترانسدیوسرهای منطبق با این استاندارد باید وجود داشته باشد. اما الگوهای بعدی اختیاری هستند. الگوی پایه مطابق آنچه در گزارش اول پروژه آمده شامل شناسه سازنده و شماره مدل و کارکتر سریال و شماره سریال ترانسدیوسر است.

همانطور که از شکل (۶-۶) پیداست پارامترهای اطلاعاتی الگوی انتخاب شده و واحد آنها با مقادی پیش فرض نشان داده شده‌است که شامل گین و پلاریته خروجی و فرکانسهای قطع بالا و پایین و فرکانس مرجع و .... می‌باشد. در این نرم افرار برای تنظیم مقادیر پارامترهای الگو نیاز به داشتن مجوز نرم‌افزار از شرکت B&K است. آخرین الگوی کاربردی همانا اطلاعات مربوط به کاربر است که بسته به فضای موجود درحافظه می‌توان یک سرس اطلاعات نوشتاری برای آن اختصاص داد. این بخش نرم‌افزار قابل دسترس بوده و امکان تغییر مقادیر آن وجود دارد.

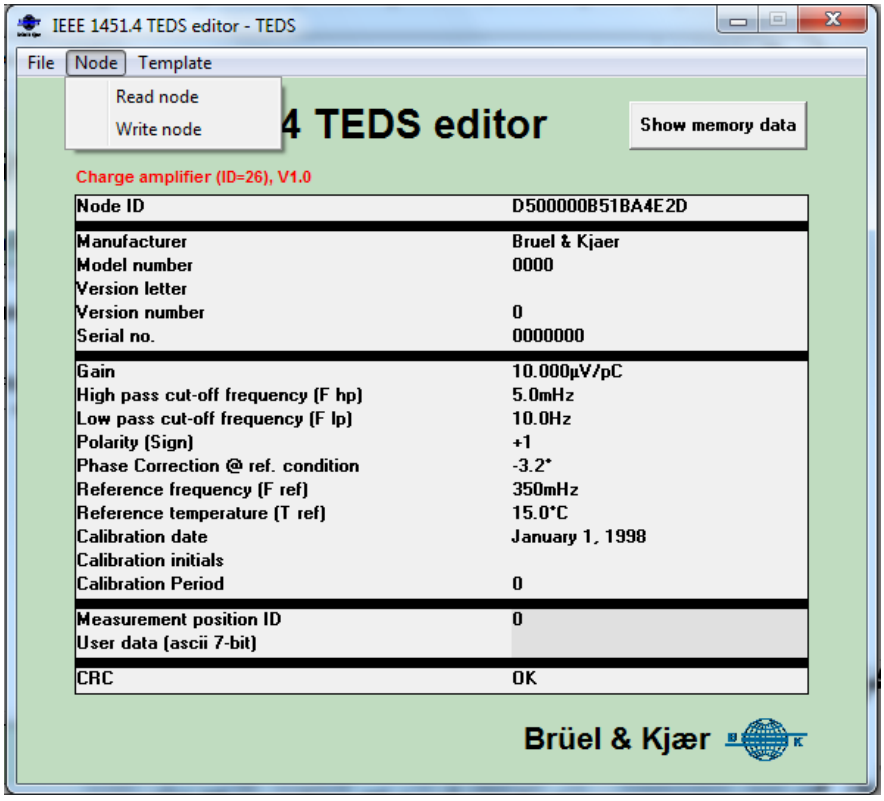


شکل ٦-٤- منوی انتخاب قطعات حافظه

DOI: 10.30503/nripress.2020.296



شکل ٦-٥- منوی انتخای الگوهای استانداردی

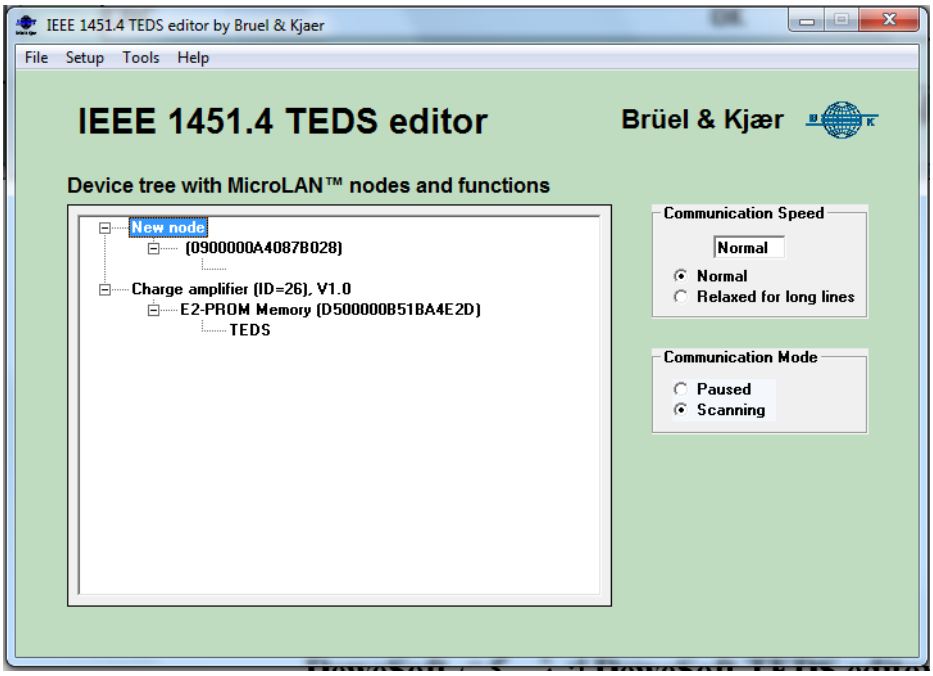


شکل ٦-٦- منوی اطلاعاتی از پارامترهای الگوی انتخابی

بعد از برنامه‌ریزی نودحافظه مطابق الگو ، نمایش کلی نودها در نرم‌افزار به شکل (۶-۷) تغییر پیدا می‌کند. وجود اطلاعات TEDS در روی نود حافظه حاکی از موفقیت در برنامه‌ریزی الگو مطابق استاندارد می‌باشد. اگر روی نود

DOI: 10.30503/nripress.2020.296

حافظه TEDS انتخاب شود منویی همانند شکل (۶-۶) به نمایش در می‌آید که محتویات درون حافظه را نشان می‌دهد.



شکل ٦-۷- شناسایی مبدل با الگوی پیاده‌سازی شده

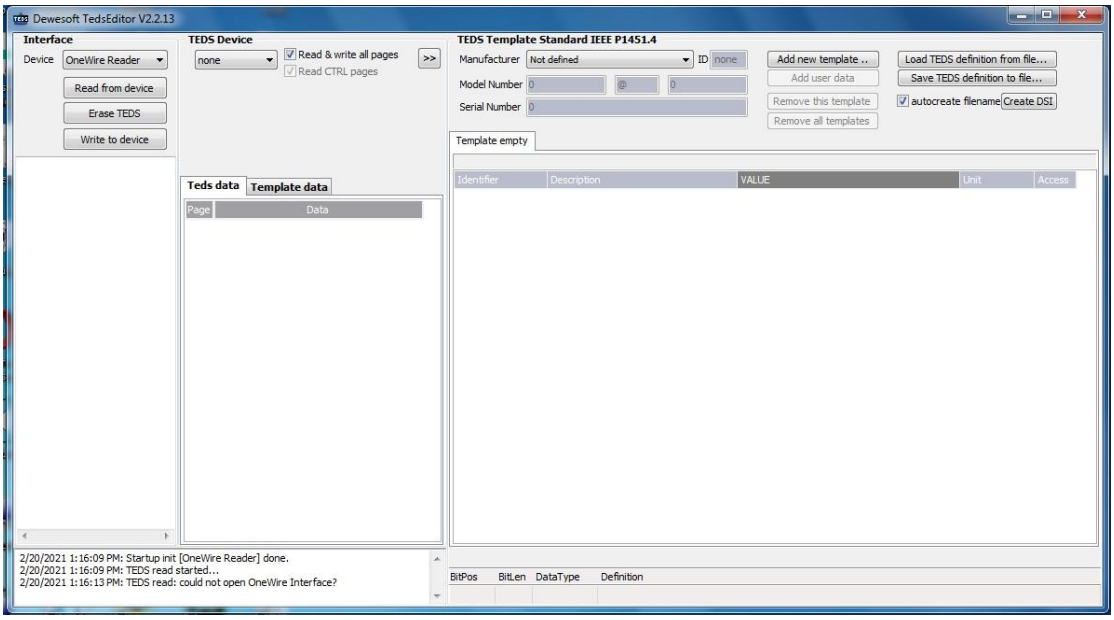
۳-٦- قابلیت های نرم‌افزارDeweSoft TEDS editor از شرکت DeweSoft

این نرم‌افزار نیز با درایور شرکت Maxim کار میکند و دارای قابلیت های زیر است

- قابلیت شناسایی انواع قطعات حافظه
- کار با تنها یک قطعه حافظه و امکان پیاده‌سازی الگو در یک قطعه (عدم تشخیص نودهای روی یک ترانسدیوسر و لیست نود)
- پیاده‌سازی تمامی الگوهای استاندارد به همراه یک الگوی کالیبراسیون با شناسه خود این شرکت
- دارای چند الگو ویژه مربوط به این شرکت
- امکان پیاده‌سازی الگوی کاربری
- امکان تغییرات مقادیر پارامترهای داده شده در الگو

در شکل (۶-۸) نمایی از صفحه اصلی نرم افرار نشان داده شده‌است. همانطور که از شکل آن پیداست نرم‌افزار دارای سه بخش است که بخش تشخیص درایور و انتخاب درایور در سمت چپ و در قسمت میانی امکان انتخاب شماره قطعه حافظه و نمایش محتویات داخلی آن و در سمت راست نرم‌افزار، امکان انتخاب انواع الگوها و محتویات و پارامترهای هر الگو فراهم شده‌است.

DOI: 10.30503/nripress.2020.296



شکل ٦-٨- نمای کلی نرم‌افزار DewSoft

با تنظیم اینترفیس نرم‌افزار در حالت OneWire Reader براحتی امکان ارتباط با درایور این پروژه فراهم می‌شود. همانطور که از شکل پیداست با انتخاب قطعه حافظه از منوی وسط نرم‌افزار امکان قرائت اطلاعات و یا نوشتن و پاک کردن کل حافظه فراهم شده‌است. لیست الگوهای پشتیبانی شده توسط این نرم‌افزار در شکل (۶-۹) نشان داده شده‌است.

(IEEE 1451.4 V1.0) ID(#25) Accelerometer/Force transducer w. const. curr. ampl.
(IEEE 1451.4 V1.0) ID(#26) Charge Amplifier (incl. attached accelerometer)
(IEEE 1451.4 V1.0) ID(#27) Microphones w. built-in preamp.
(IEEE 1451.4 V1.0) ID(#28) Microphone Preamps. w. attached micr. or system
(IEEE 1451.4 V1.0) ID(#29) Microphones (capacitive)
(IEEE 1451.4 V1.0) ID(#30) High-Level Voltage Output Sensors
(IEEE 1451.4 V1.0) ID(#31) Current Loop Output Sensors
(IEEE 1451.4 V1.0) ID(#32) Resistance Sensors
(IEEE 1451.4 V1.0) ID(#33) Bridge Sensors
(IEEE 1451.4 V1.0) ID(#34) AC Linear/Rotary Variable Differential Transformer (LVDT/RVDT) Sensors
(IEEE 1451.4 V1.0) ID(#35) Strain Gage
(IEEE 1451.4 V1.0) ID(#36) Thermocouple
(IEEE 1451.4 V1.0) ID(#37) Resistance Temperature Detectors (RTDs)
(IEEE 1451.4 V1.0) ID(#38) Thermistor
(IEEE 1451.4 V1.0) ID(#39) Potentiometric Voltage Divider
(IEEE 1451.4 V1.0) ID(#43) Charge Amplifier (incl. attached force transducer)
(IEEE 1451.4 V0.9) ID(#0) Accelerometer
(IEEE 1451.4 V0.9) ID(#1) Accelerometer transfer function
(IEEE 1451.4 V0.9) ID(#12) Microphone, integrated preamplifier
(IEEE 1451.4 V0.91) ID(#24) Accelerometer, transfer function
(IEEE 1451.4 V1.0 Wrong printed) ID(#28) Microphone Preamps. w. attached micr. or system
(Dewesoft V1.0) ID(#7) Dewesoft Sensor Properties

(IEEE 1451.4 V1.0) ID(#42) Frequency Response Table
(Dewesoft V1.0) ID(#2) Dewesoft Frequency Response Table
(Dewesoft V1.0) ID(#6) Dewesoft Amplifier Properties
(Dewesoft V1.0) ID(#7) Dewesoft Sensor Properties
(Dewesoft V1.0) ID(#11) Dewesoft Channel Properties

شکل ٦-٩- انواع الگوهای قابل پشتیبانی با نرم‌افزار DewSoft

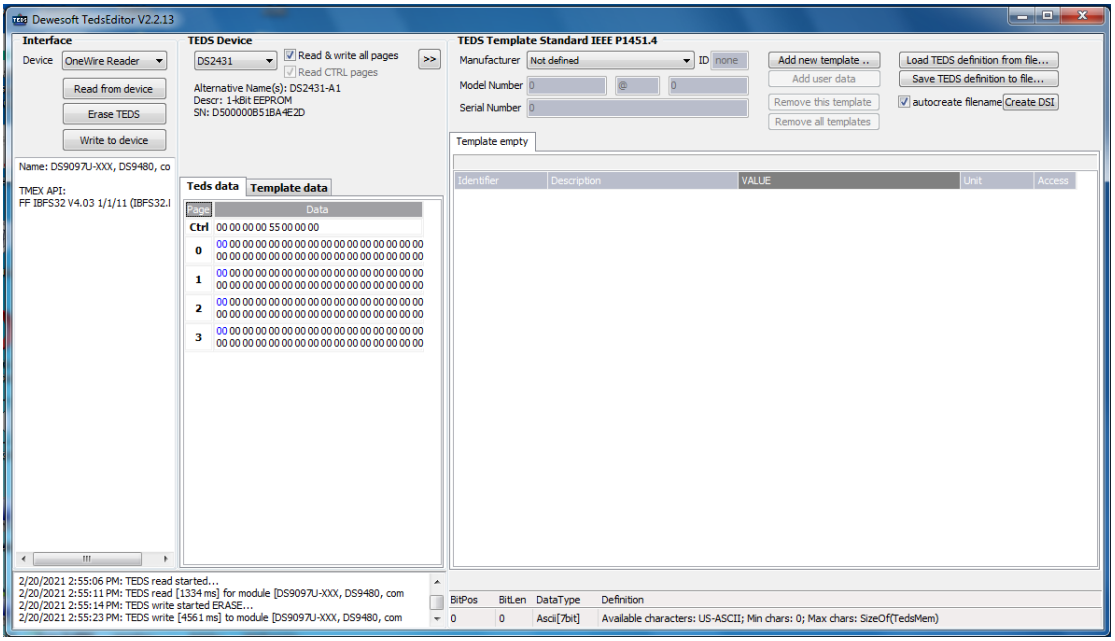
DOI: 10.30503/nripress.2020.296

همانطور که مشاهد می‌شود تمامی الگوهای استانداردی و یک الگوی کالیبراسیون و چند نمونه از الگوهای خاص شرکت DeweSoft دیده می‌شود. هر الگوها علاوه بر عنوان دارای شماره شناسه الگو بوده و شماره نسخه استاندارد مربوطه نیز آمده‌است.

یکی از ویژگیهای جالب این نرم‌افزار مشاهده محتوای هر الگو و میزان حافظه اشغالی هر پارامتر و آدرس قرارگیری هر پارامتر در بخش پایینی الگو بخوبی مشهود است. بخوبی با تغییر پارامترهای انتخابی الگو تغییر حجم و پارامترهای افزوده شده نیز قابل مشاهده هستند.

۱-۳-۶- پیاده‌سازی الگوهای نمونه استانداردی روی مبدل پروژه با نرم‌افزار DeweSoft

اگر با نرم‌افزار DeweSoft به مجموعه مبدل و درایور اتصال یابیم. منویی مانند شکل (۶-۱۰) از نرم‌افزار خواهیم داشت. همانطور که از شکل پیداست این نرم‌افزار با اینترفیس OneWireReader می‌تواند به تبادل اطلاعات بپردازد بعلاوه این نرم‌افزار قطعه حافظه موجود روی مبدل که همانا DS2431 می‌باشد را شناسایی و در قسمت میانی نشان داده است. چون این قطعه برنامه‌ریزی نشده است محتوای اطلاعاتی آن پاک بوده و در قسمت الگو نیز موردی را نشان نمی‌دهد. در ادامه یک الگوی استاندرادی همانند بخش قبل با این نرم‌افزار تولید و برنامه‌ریزی خواهیم نمود. پس از انتخاب شناسه یک سازنده، انتخاب الگو با گزینه Add new template صورت میگیرد. همانطور که اشاره شد این نرم‌افزار تقریبا تمامی الگوهای استانداردی را پشتیبانی می‌نماید. حال با انتخاب الگوی شماره ۲۶ یک الگوی یک تقویت‌کننده بار تولید می‌کنیم بطوریکه فرمت و پارامترهای این الگو در شکل (۶-۱۱) نشان داده شده‌است. این الگو برای وضعیت تقویت کنند بار بدون سنسور است.

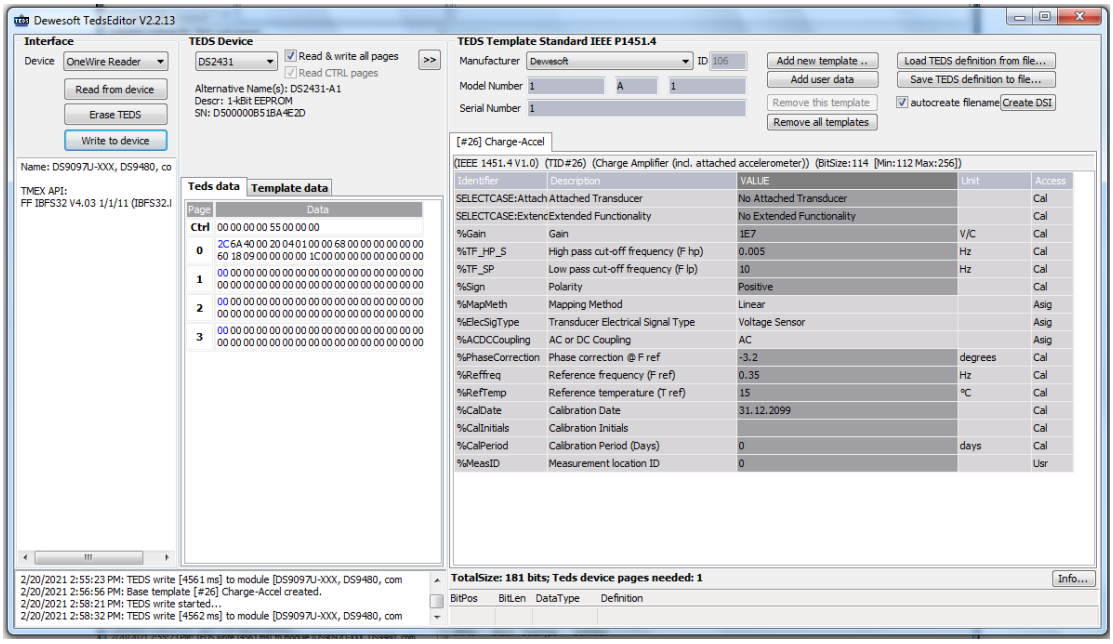


شکل ٦-۱۰- شناسایی مبدل خام جهت برنامه‌ریزی الگوهای متفاوت

همانطور که از شکل (۶-۱۱) پیداست الگوی انتخابی شامل پارامترهایی نظیر گین و فرکانسهای قطه بالا و پایین و فرکانس مرجع و پلاریته خروجی و .... به همراه مقادیر پیش فرض آنها و واحد هر پارامتر است. در این نرم‌افزار امکان تغییر مقادیر پارامترها وجود دارد. این امکان در بخش اطلاعات دیتاشیت پایه یعنی شماره مدل و شمارخ سریال و کاراکتر سریال نیرز وجود دارد.

DOI: 10.30503/nripress.2020.296

هر تغییر در مقادیر، در محتوای اطلاعاتی بخش میانی نرم‌افزار که مربوطه به داده حافظه است منعکس می‌گردد. پس از نهایی شدن الگوی انتخابی میتوان با انتخاب کلید write to device اطلاعات را روی حافظه برنامه‌ریزی نمود.



شکل ٦-١١- انتخاب یک الگوی نمونه و برنامه‌ریزی آن

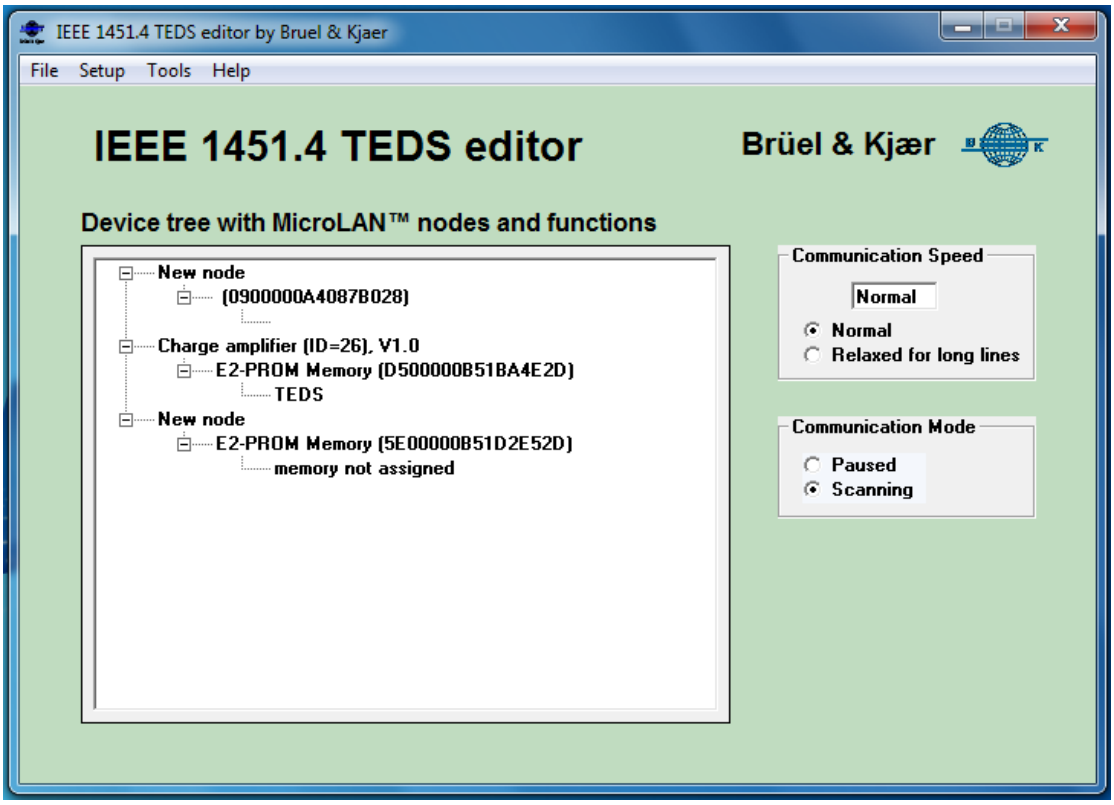
تا کنون به کمک دو نرم‌افزار پیاده‌سازی یک نمونه الگو نشان داده شده‌است. در ادامه به پیاده‌سازی دیگر قابلیتهای آمده در استاندارد همانند لیست نود می‌پردازیم.

٤-٦- پیاده‌سازی لیست نود مطابق استاندارد

همانطور که قبلا اشاره شده، نرم‌افزار DeweSoft قابلیت نمایش تمامی نودهای متصل به شبکه را نداشته و تنها می‌تواند نودها دارای حافظه را شناسایی کرده و هر یک از آنها را بصورت جداگانه مطابق الگوهای خود برنامه‌ریزی نماید. پس لیست نود را که مربوط به مجموعه نودهای داخلی یک ترانسدیوسر است پشتیبانی نمی‌کند. اما نرم‌افزار B&K بخش لیست نود را پشتیبانی می‌نماید اما بدون مجوز مربوطه قادر به تولید لیست نود به کمک آن نمی‌باشیم. ولی قابلیت تشخیص و نمایش آن نیازی به مجوز ندارد بدین معنا که اگر لیست نود را به هر شکلی مطابق استاندارد پیاده‌سازی کنیم باید در این نرم‌افزار این لیست نود قابل نمایش باشد.

مطابق گزارش اول پروژه برای پیاده‌سازی لیست نود از نودهای داخلی یک ترانسدیوسر لازم است حداقل دو نود دارای قابلیت حافظه در ترانسدیوسر وجود داشته باشند که یکی برای پیاده‌سازی لیست نود بکار فته و دیگری برای پیاده‌سازی الگوهای استانداردی می‌باشد. به منظور مهیا سازی یک ترانسدیوسر با دو نود حافظه، مجددا از برد کمکی بخش استفاده می‌کنیم و با اتصال بخش دیجیتال دو مبدل به همدیگر، مجموعه‌ای با دو حافظه و یک قطعه سنسور دما فراهم شود. بدین ترتیب به لحاظ سخت‌افزاری یک مجموعه با امکان پیاده‌سازی لیست نود آماده شده‌است. اگر نود حافظه اضافه شده به نودهای برنامه‌ریزی قبلی را در محیط نرم‌افزار B&K مشاهد کنیم شکلی مانند (۶-۱۲) را خواهد داشت که هر نود برای خود مستقل بوده و زیر مجموعه دیگری قرار نگرفته است.

DOI: 10.30503/nripress.2020.296



شکل ٦-١٢- مجموعه نودهای آماده شده جهت پیاده‌سازی لیست نود

حال سعی می‌کنیم مطابق با استاندارد لیست نود را در حافظه خام جدید پیاده‌سازی کنیم. در واقع می‌خواهیم که سه نود فوق در قالب یک مجموعه و تحت یک لیست نود باشند. همانطور که در گزارش مرحله اول پروژه آمده است برای لیست نود شناسه سازنده ۱۴ بیتی در دیتاشیت پایه مقدار یک را به خود می‌گیرد و پس از آن دو بیت رها شده و یک بایت بعدی تعداد نودهای مربوط به یک ترانسدیوسر را نشان می‌دهد. سپس آدرس ۶۴ بیتی هر یک از نودها پشت سر هم قرار می‌گیرند. لازم به ذکر است که آدرس خود لیست نود در این مجموعه نباید باشد بنابراین برای مجموعه تست فوق دو آدرس ۶۴ بیتی بعنوان نودهای داخلی معرفی شده در لیست نود باید باشند. در زیر مجموعه این اطلاعات را نشان میدهد

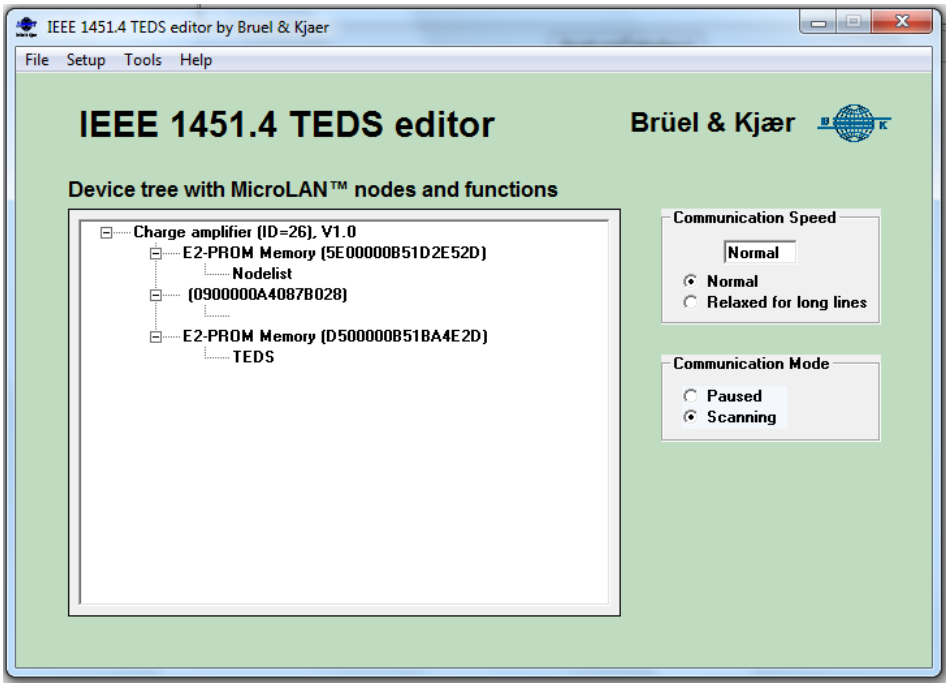
01 00 03
28 B0 87 40 0A 00 00 09
2D 4E BA 51 0B 00 00 D5

در نهایت مجموعه دیتای فوق در ابتدای حافظه نود لیست قرار داده‌می‌شود. شایان ذکر است که مطابق استاندارد، همواره چیدن اطلاعات داخل حافظه با بایت کم ارزش شروع می‌شود و بعلاوه هر بلوک ۲۵۶ بیتی باید دارای یک بایت کد خطایابی CRC در ابتدای بلوک باشد. با محاسبه این کد برای اطلاعات فوق (کد E4) این بایت در ابتدای حافظه و بقیه در ادامه حافظه برنامه‌ریزی می‌شوند. بنابراین مجموعه اطلاعات زیر

E401000328B087400A0000092D4EBA510B0000D5

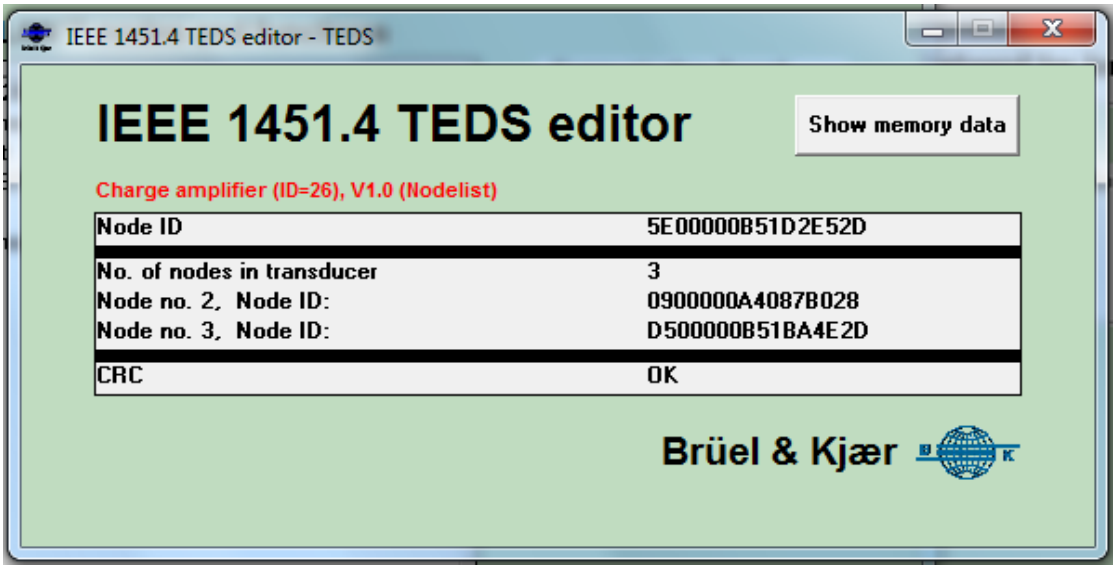
در حافظه لیست نود باید برنامه‌ریزی شوند. که برای برنامه‌ریزی فوق از نرم‌افزار OneWireViewer استفاده شده‌است. بعد از پیاده‌سازی لیست نود اگر مجددا با نرم‌افزار B&K اطلاعات آنرا بخوانیم یک مجوعه سه نودی مطابق شکل (۶-۱۳) خواهیم داشت که تحت یک عنوان بنام تقویت‌کننده شارژ آشکار شده و دیگر نودها زیرمجموعه آن هستند.

DOI: 10.30503/nripress.2020.296



شکل ٦-١٣- پیاده‌سازی لیست نود برای یک ترانسدیوسر

همانطور که از شکل (۶-۱۳) پیداست در یک لیست نود، نودهای داخلی یک ترانسدیوسر بصورت زیر شاخه‌های آن آشکار می‌شوند و عنوان دیتاشیت به مجموعه آنها اطلاق شده‌است. اگر در نرم‌افزار Nodelist انتخاب شود منویی مانند شکل (۶-۱۴) ظاهر می‌شود که حاوی اطلاعات لیست نود بوده و اطلاعات تعداد نودها و آدرسهای مربوطه به آنها آمده است و بعلاوه نقشه حافظه داده‌های ذخیره شده در حافظه را نیز می‌توان مشاهد نمود.

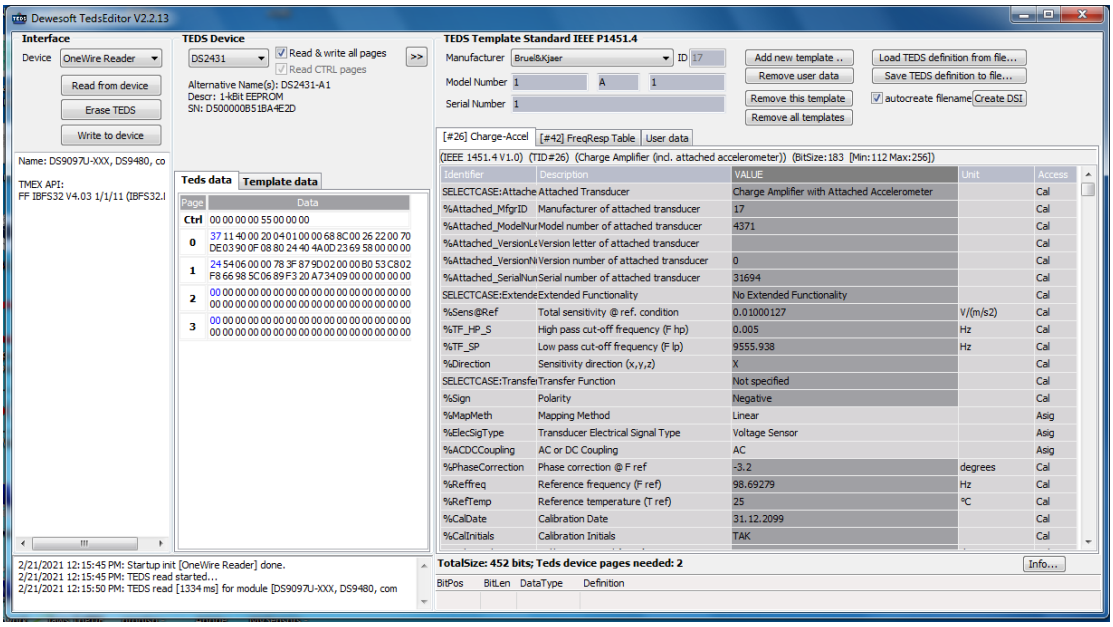


شکل ٦-١٤- اطلاعات یک لیست نود

٦-٥- پیاده‌سازی الگوهای کالیبراسیون و کاربری

DOI: 10.30503/nripress.2020.296

در این بخش سعی در پیاده‌سازی الگوی کالیبراسیون و کاربری به کمک نرم‌افزار DeweSoft داریم. برای این منظور حافظه بخش دیجیتال مبدل را پاک کرده و مجددا الگوهای مورد نظر را تولید و روی آن برنامه‌ریزی می‌کنیم. ابتدا الگوی شماره ۲۶ مربوط به تقویت‌کننده بار را انتخاب می‌کنیم. سپس با اضافه کردن الگوی جدید منوی دیگری باز شده که الگوهای قابل ارائه را در این مرحله نشان می‌دهد. با انتخاب الگوی شماره ۴۲ که مربوط به کالیبراسیون است یک زبانه جدید مربوطه به این الگو نیز ساخته می‌شود. در انتها با اضافه کردن الگوی کاربری مجموعه الگوهای مبدل مطابق شکل (۶-۱۵) کامل می‌شود. در ادامه به تکمیل اطلاعات هر یک از الگوهای مبدل می‌پردازیم.



شکل ٦-۱۵- مجموعه الگوهای انتخابی در نرم‌افزار DeweSoft

در ادامه مطالب به منظور وضوح بیشتر تصاویر ارائه شده فقط محدود انتخاب الگو و پارامترهای آن از نرم‌افزار نشان داده شده‌است. در شکل (۶-۱۶) تنظیم پارامترهای الگوی استاندارد با شناسه ۲۶ در وضعیت اتصال سنسور شتاب به آن نشان داده شده‌است. همانطور که مشاهده می‌شود با فعال کردن اتصال سنسور یک بخش به الگو اضافه شده که دقیقا اطلاعات دیتاشیت پایه مربوط به سنسور است. لذا شماره سازنده و مدل و سریال سنسور باید به صورت صحیح تنظیم شوند چون مبدل این پروژه برای سنسور B&K به شماره مدل ۴۷۳۱ و سریال ۳۶۵۱۲ است لذا این اطلاعات در بخش پارامترهای سنسور وارد شده است. دیگر پارامترهای الگو مطابق با اطلاعاتی که از انجام پروژه‌های قبلی بدست آمده تنظیم شده است. بعنوان مثال حساسیت 100mv/g و فرکانسهای قطع بالا و پایین و پلاریته منفی و فرکانس مرجع 100hz و.... با توجه به اطلاعات اخذ از انجام پروژه های قبلی پروژه تنظیم شده است.

DOI: 10.30503/nripress.2020.296

[#26] Charge-Accel | [#42] FreqResp Table | User data

(IEEE 1451.4 V1.0) (TID#26) (Charge Amplifier (ind. attached accelerometer)) (BitSize: 183 [Min: 112 Max: 256])

Identifier	Description	VALUE	Unit	Access
SELECTCASE:Attach	Attached Transducer	Charge Amplifier with Attached Accelerometer		Cal
%Attached_MfgrID	Manufacturer of attached transducer	17		Cal
%Attached_ModelN	Model number of attached transducer	4371		Cal
%Attached_Version	Version letter of attached transducer			Cal
%Attached_Version	Version number of attached transducer	0		Cal
%Attached_SerialNu	Serial number of attached transducer	31694		Cal
SELECTCASE:Extenc	Extended Functionality	No Extended Functionality		Cal
%Sens@Ref	Total sensitivity @ ref. condition	0.01000127	V/(m/s2)	Cal
%TF_HP_S	High pass cut-off frequency (F hp)	0.005	Hz	Cal
%TF_SP	Low pass cut-off frequency (F lp)	9555.938	Hz	Cal
%Direction	Sensitivity direction (x,y,z)	X		Cal
SELECTCASE:Transfi	Transfer Function	Not specified		Cal
%Sign	Polarity	Negative		Cal
%MapMeth	Mapping Method	Linear		Asig
%ElecSigType	Transducer Electrical Signal Type	Voltage Sensor		Asig
%ACDCCoupling	AC or DC Coupling	AC		Asig
%PhaseCorrection	Phase correction @ F ref	-3.2	degrees	Cal
%Reffreq	Reference frequency (F ref)	98.69279	Hz	Cal
%RefTemp	Reference temperature (T ref)	25	°C	Cal
%CalDate	Calibration Date	31.12.2099		Cal
%CalInitials	Calibration Initials	TAK		Cal

TotalSize: 452 bits; Teds device pages needed: 2 Info...

BitPos	BitLen	DataType	Definition
224	12	UnInt	Min: 0; Max: 4095; Res: 1

شکل ٦-١٦- نمایش و تنظیم پارامترهای الگوی انتخابی اول

در شکل (۶-۱۷) الگوی کالیبراسیون فرکانسی در ۳ نقطه برای این مبدل تنظیم شده است. این اطلاعات شامل درصد انحراف از دامنه مرجع در فرکانس های کالیبراسیون ۱و ۱۰۰ و ۱۰ کیلوهرتز وارد شده است.

[#26] Charge-Accel | [#42] FreqResp Table | User data

(IEEE 1451.4 V1.0) (TID#42) (Frequency Response Table) (BitSize: 125 [Min: 15 Max: 4601])

Identifier	Description	VALUE	Unit	Access
STRUCTARRAY:TF_1	Transfer function table	3		Cal
%TF_Table_Freq1	Frequency	1	Hz	Cal
%TF_Table_Ampl1	Amplitude	-5	%	Cal
%TF_Table_Freq2	Frequency	99.99998	Hz	Cal
%TF_Table_Ampl2	Amplitude	-100	%	Cal
%TF_Table_Freq3	Frequency	9999.995	Hz	Cal
%TF_Table_Ampl3	Amplitude	5	%	Cal

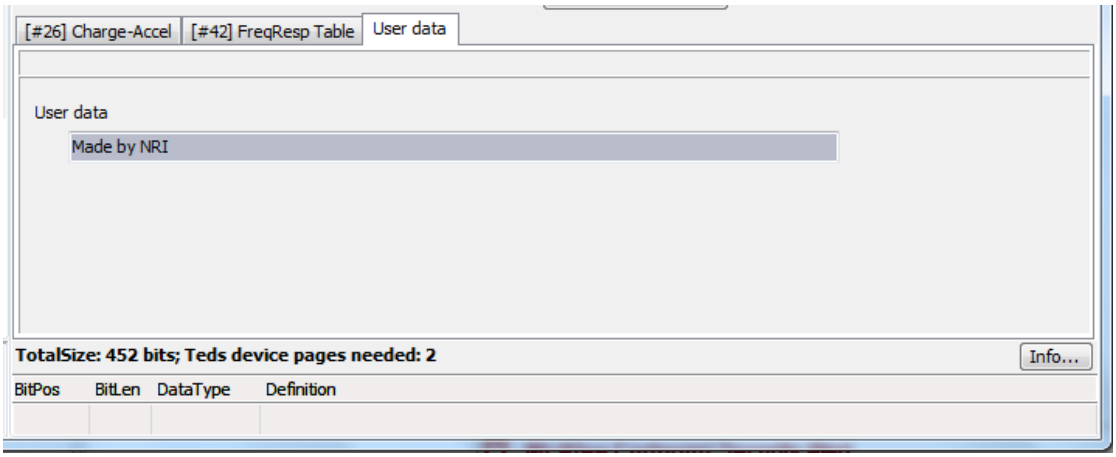
TotalSize: 452 bits; Teds device pages needed: 2 Info...

BitPos	BitLen	DataType	Definition

شکل ٦-١٧- نمایش و تنظیم پارامترهای الگوی انتخابی دوم (الگوی کالیبراسیون)

در شکل (۶-۱۸) الگوی کاربری را نشان داده شده و می‌توان هر متن نوشتاری را با توجه به محدودیت حافظه دستگاه، در این بخش وارد نمود که در اینجا عبارت Made by NRI وارد شده است.

DOI: 10.30503/nripress.2020.296



شکل ٦-١٨- نمایش وضعیت الگوی کاربری

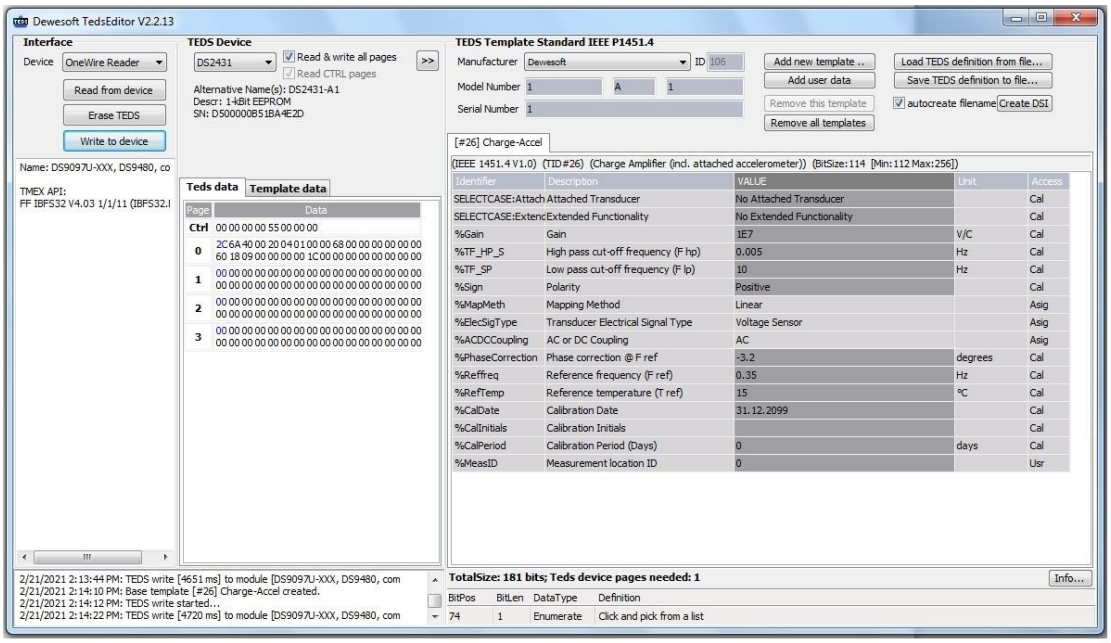
همانطور که از شکل (۶-۱۵) پیداست مجموعه الگوهای انتخابی با پارامترهای تنظیم شده برای آنها، فضایی معادل ۴۵۲ بیت در حافظه اشغال میکنند. مقدار فضای اشغالی الگوهای انتخابی از بخش پایینی نرم‌افزار قابل دستیابی است.

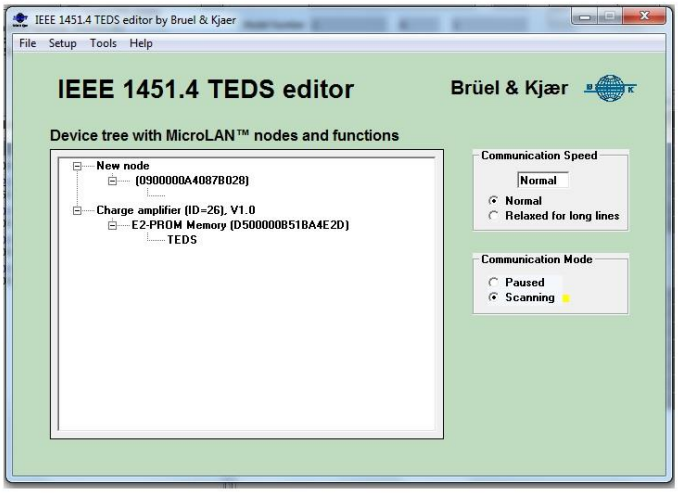
٦-٦- بررسی تشخیص الگوی تولید شده توسط یک نرم‌افزار در محیط نرم افرار دیگر

همانطور که اشاره شد الگوهای نرم‌افزار B&K محدود به کاربردهای مختلف از پیزوالکتریک بوده و شامل سنسورهای ارتعاشی ، میکروفن، تاکومتر، سنسور نیرو و تقویت‌کننده بار هستند. با وجود این محدودیت، در این بخش به تولید الگوهایی با یک نرم‌افزار و مشاهده آن با نرم‌افزار می‌پردازیم تا تفاوت آنها آشکار گردد.

برای نمونه الگوی استانداردی با شناسه ۲۶ را که بدون سنسور و فانکشهای اضافی را که با نرم‌افزار DeweSoft برنامه ریزی کرده‌ایم، در محیط نرم افرار B&K مشاهده می‌کنیم. نتیجه این کار در شکل (۶-۱۹) نشان داده شده است. تصویر بالایی الگوی تولیدی با نرم‌افزار DeweSoft را نشان داده و تصویر زیرین نتیجه مشاهده آن با نرم‌افزار B&K است.

DOI: 10.30503/nripress.2020.296

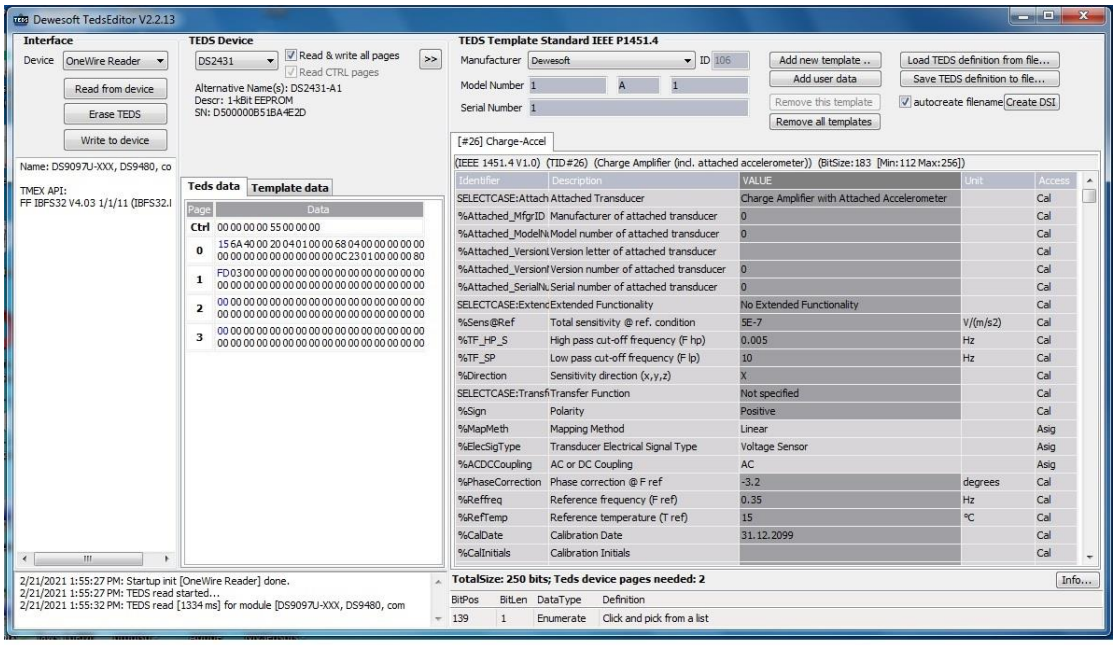


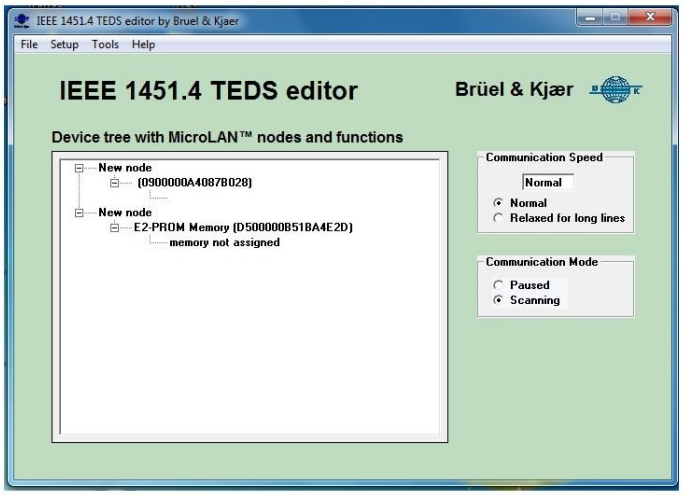


شکل ٦-١٩- تولید الگوی اول با نرم‌افزارDeweSoft و مشاهده آن با نرم‌افزار B&K

همانطور که از شکل پیداست الگو تولیدی بدستی توسط نرم‌افزار B&K آشکار شده است. حال تست فوق را تنها با تغییر در تنظیم الگو از حالت بدون سنسور به حالت اتصال یافته به سنسور تکرار می‌کنیم. در شکل (۶-۲۰) نتیجه این تست نشان داده شده‌است.

DOI: 10.30503/nripress.2020.296



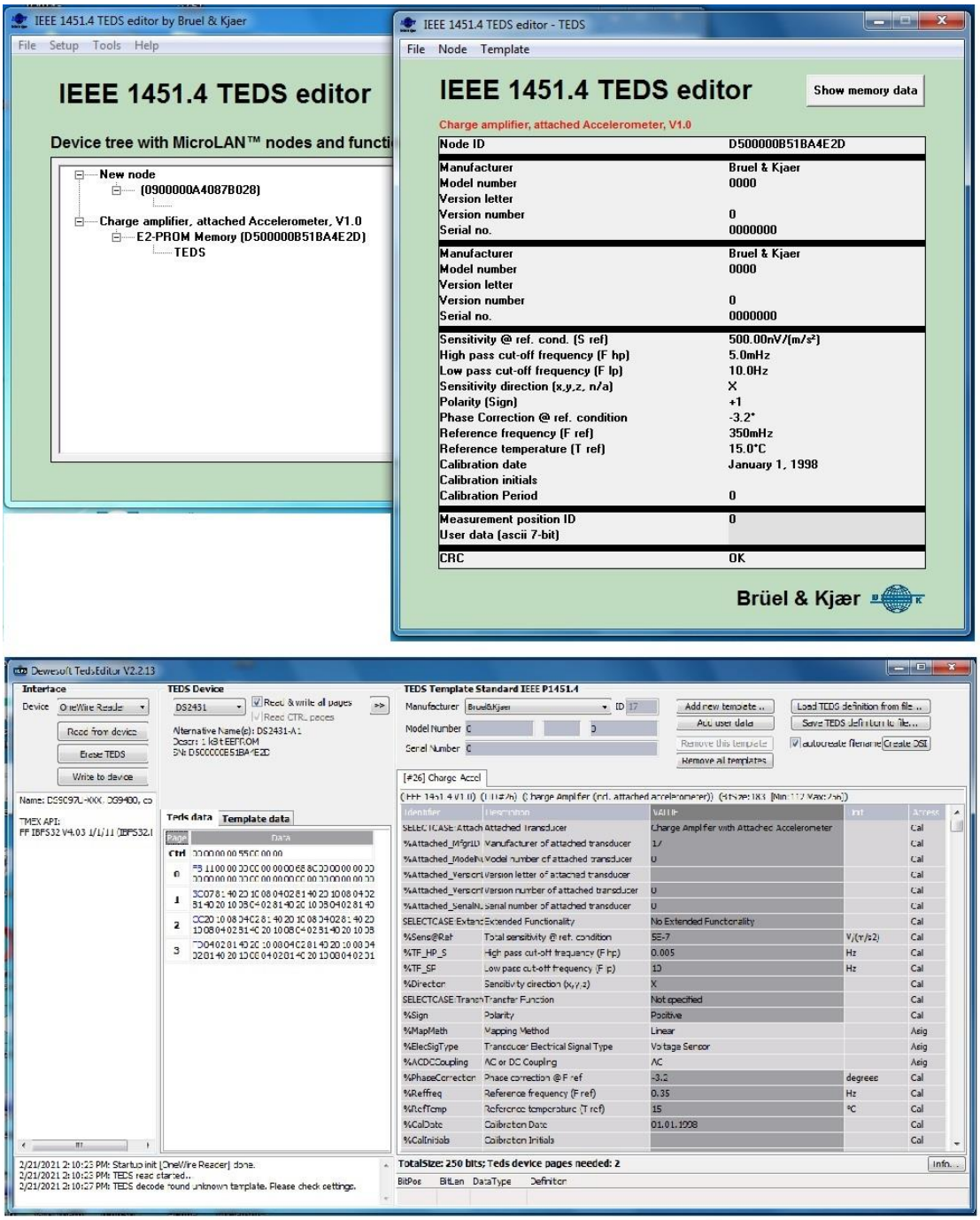


شکل ٦-٢٠- تولید الگوی دوم با نرم‌افزارDeweSoft و مشاهده آن با نرم‌افزار B&K

می‌بینیم که در این حالت، الگو بدرستی توسط B&K آشکار نشده و قابلیت شناسایی آنرا ندارد.

حال مورد فوق را برعکس انجام میدهیم یعنی الگوی شماره ۲۶ را به همراه سنسور از طریق نرم‌افزار B&K برنامه‌ریزی کرده و با برنامه DeweSoft به آن مشاهده می‌کنیم. در شکل (۶-۲۱) نتیجه نشان داده شده‌است

DOI: 10.30503/nripress.2020.296



شکل ٦-٢١- تولید الگو با نرم‌افزار B&K و مشاهده آن با نرم‌افزار DeweSoft

همانطور که از شکل (۶-۲۱) پیداست در حالت عکس، نرم‌افزار DeweSoft قابلیت شناسایی الگو را داراست. بنابراین نرم‌افزار DeweSoft بروز از نرم‌افزار B&K بوده و بعلاوه همانطور که قبلا اشاره شده الگوهای بیشتری را پشتیبانی میکند.

نتیجه گیری:

DOI: 10.30503/nripress.2020.296

در فصل اول مشخصات یک باس ویژه که به باس تک سیمه معروف است مطالعه و بررسی شد. در ادامه نیازمندی‌های الکتریکی و سیگنالینگ لازم برای این شبکه معرفی شده و سپس پروتکل ارتباطی حاکم مورد بررسی قرار گرفت. بعلاوه چند نوع از چیپ‌های درایور و حافظه و سنسور را که بر مبنای این باس کار می کنند معرفی گردید. چند طرح سخت افزاری برای ساخت مرکز اصلی شبکه معرفی و چگونگی کار با یک نوع ساده آن تشریح شد.

در فصل دوم پس از یک بررسی مختصر در خصوص خانواده استانداردهای IEEE1451 به نقش و کاربرد هر بخش استاندارد اشاره شد. این استاندارد با معرفی ایده دیتاشیت الکترونیک TEDS در ترانسدیوسرها، قابلیت هوشمند سازی را برای آنها فراهم ساخت. بخش چهارم این استاندارد با شماره IEEE1451.4 به ترانسدیوسرهای آنالوگ اختصاص داشته و با ترکیب مدارات آنالوگ به دیجیتال، بدنبال هوشمندسازی آنهاست. یک نمونه موجود از ترانسدیوسر آنالوگ، می تواند مبدل بار به ولتاژی باشد که در پروژه قبلی در پژوهشگاه ساخته شده است. در واقع در پروژه حاضر بدنبال پیاده سازی استاندارد فوق روی این مبدل هستیم.

در فصل سوم گزارش با توجه به مطالب فصول اول و دوم طراحی مبدل بار به ولتاژ منطبق بر این استاندارد صورت گرفته است. بعلاوه طراحی یک درایور منطبق با این استاندارد برای اتصال به کامپیوتر صورت گرفته است. به کمک این درایور و نرم افزار کامپیوتری امکان پیاده سازی یک ارتباط با مبدل فراهم می شود و می توان اطلاعات روی حافظه دیجیتال این مبدل را منطبق بر استاندارد برنامه ریزی و یا خواند.

در طی فصلهای چهارم و پنجم گزارش طراحی سخت‌افزار مبدل و درایور انجام و تستهای سخت‌افزاری انجام گرفته است. بعلاوه تستهای نرم‌افزاری درایور در دو مد شبکه تک سیمه و استاندارد IEEE1451.4 نیز به کمک درایورهای نرم‌افزاری اشاره شده در متن گزارش انجام پذیرفته و در نهایت یک مجموعه سخت‌افزاری و نرم‌افزاری جهت پیاده سازی دیتاشیت الکترونیک آماده شده است. در فصل ششم پروژه فعالیتهای مربوطه به چگونگی پیاده‌سازی دیتاشیت الکترونیک منطبق با الگوهای استانداردی تشریح شده است. در این فصل به کمک دو نرم‌افزار، دو نمونه الگوی استانداردی بر روی مبدل پروژه پیاده‌سازی شده است سپس چگونگی پیاده‌سازی لیست نود منطبق با استاندارد آمده و به دنبال آن پیاده‌سازی الگوی کالیبراسیون و الگوی کاربری بر روی مبدل ای مبدل پروژه با مقادیر مناسب ارائه شده است. در انتها مقایسه‌ای از عملکرد این دو نرم‌افزار نیز در گزارش آورده شده است.

DOI: 10.30503/nripress.2020.296

مراجع:

[1] “Tech brief 1: 1-Wire Design Guide,” Maxim/Dallas Semiconductor Corp. Application Note 570, http://www.maxim-ic.com/AN570.
[2] “1-Wire Master Device Configuration,” Maxim/Dallas Semiconductor Corp. Application Note 2965, http://www.maxim-ic.com/AN2965.
[3] “DS2430A 256-bit 1-Wire EEPROM” Maxim/Dallas Semiconductor Corp, Datasheet, http://www.maxim-ic.com/DS2430A
[4] “1-WIRE SEARCH ALGORITHM” Maxim/Dallas Semiconductor Corp. Application Note 187, http://www.maximintegrated.com/AN187.
[5] IEEE 1451.4-2004 - IEEE Standard for a Smart Transducer Interface for Sensors and Actuators--Mixed-Mode Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats
[6]– گزارش PCNPN40/T1 تحت عنوان "طراحی و ساخت درایور دیجیتال سنسور هوشمند بر اساس استاندارد IEEE1451.4 بصورت نمونه برای مبدل پیزوالکتریک" ، گروه ابزار دقیق ،پژوهشگاه نیرو