به نام خرد

به نام خدا

طراحی یک سیستم امنیتی با مادون قرمز

گردآوری

محمد یارمحمدی

احسان خاندوزی

کلمات کلیدی

مادون قرمز - کنترل دسترسی – میکروکنترلرAVR

چکیده

سیستمی که در اینجا ارایه شده، از امنیتی بالایی برخوردار بوده و هزینه ساخت آن نیز در مقایسه با سایرین بسیار اندک است. عملکرد اصلی این سیستم، کنترل دسترسی افراد به یک مکان یا اطلاعات خاص و ثبت آمار ورود و خروج (ازجمله ID، ساعت، تاریخ و ...) می‌باشد

دانشکده مهندسی

پایان نامه کارشناسی

طراحی یک سیستم امنیتی با مادون قرمز

احسان خاندوزی – محمد یارمحمدی

استادان پروژه:
مهندس سوادی – مهندس شرکا

استاد داور:
دکتر میمندی‌نژاد

مهر ۱۳۸۶

چکیده

امروزه، سیستم های کنترل دسترسی فیزیکی به صورت تلفیقی از سیستم های الکترونیکی و مکانیکی درآمده اند. مشکل اصلی در توسعه اینگونه سیستم ها برقراری تعادل بین امنیت و هزینه ساخت آن است (امنیت بالاتر، موجب صرف هزینه بیشتر می‌شود).

سیستمی که در اینجا ارایه شده، از امنیتی بالایی برخوردار بوده و هزینه ساخت آن نیز در مقایسه با سایرین بسیار اندک است. عملکرد اصلی این سیستم، کنترل دسترسی افراد به یک مکان یا اطلاعات خاص و ثبت آمار ورود و خروج (ازجمله ID، ساعت، تاریخ و ...) می‌باشد. مشابه سایرین، سیستم مذکور نیز شامل دو بخش جدا از هم "*کارت خوان*" و "*کارت*" است. کارت قطعه ایست سخت افزاری که در اختیار کاربر سیستم قرار می گیرد و هر بار اطلاعات موجود در آن، توسط کارت خوان خوانده شده و مجوز های لازم صادر می گردد.

ارتباط بین این دو، از طریق نور "مادون قرمز" برقرار می‌شود. نکته حائز اهمیت در اینجا این است که کارت و کارت خوان، هر یک، تنها ازیک "دیود مادون قرمز" – هم به عنوان فرستنده و هم به عنوان گیرنده - برای ارسال

و دریافت اطلاعات استفاده می‌کنند. همچنین بدین منظور از یک پروتکل "درخواست شناسایی - پاسخ" رمزنگاری‌شده استفاده شده‌است.

در کنار مستندات حاضر، یک نمونه واقعی از سیستم فوق نیز ارایه شده‌است. امنیت و سرعت مناسب و همچنین دقت بالا از ویژگی های این سیستم امنیتی می‌باشد که در عمل نیز قابل مشاهده است.. در مجموع به نظر می‌رسد که سیستم فوق می‌تواند به عنوان یک جایگزین ارزان قیمت و مناسب برای تکنولوژی های کنونی مطرح شود.

فهرست مطالب

عنوان **صفحه**

۱- تاریخچه و بیان مساله..... ۹

۲- مقدمه..... ۱۰

۲-۱-طراحی سیستم..... ۱۱

۲-۲- توپولوژی سیستم‌های کنترل دسترسی..... ۱۱

۳- ارتباط بین کارت و کارت خوان ۱۲

۳-۱- استفاده از یک دیود نوری به عنوان سنسور..... ۱۲

۳-۲- جزییات پروتکل ارتباطی ۱۴

۴- سخت افزار کارت خوان..... ۱۷

۴-۱-RTC (Real Time Clock) ۱۷

۴-۲- ارتباط سریال دو سیمه (I2C یا TWI)..... ۱۸

۴-۳- معرفی تراشه DS1307..... ۲۳

۴-۳-۱- تقویم و ساعت..... ۲۵

۴-۳-۲- رجیستر کنترل DS1307..... ۲۷

۴-۳-۳- نرم افزار ارتباط با DS1307 ..... ۲۸

۴-۴- ارتباط با رایانه ۲۹

۴-۴-۱- ارتباط سریال..... ۲۹

۴-۴-۲- ارتباط UART..... ۳۱

۴-۴-۳- سازگاری USART با UART در AVR..... ۳۲

۴-۴-۴- سخت افزار لازم برای ارتباط با رایانه..... ۳۵

۴-۴-۵- توضیحاتی راجع به پورت سریال رایانه..... ۳۶

۴-۵- نرم افزار میکروکنترلر برای ارتباط با رایانه..... ۳۸

۴-۶- LCD کاراکتری..... ۳۹

۴-۶-۱- توضیح مختصری راجع به پایه ها..... ۴۱

۴-۶-۲- نحوه اتصال LCD به میکروکنترلر..... ۴۱

۴-۶-۳- برنامه نویسی برای LCD ۴۲

۴-۶-۴- کنترل نور پس زمینه LCD..... ۴۴

۴-۷- صفحه کلید..... ۴۵

۶- سخت‌افزار کارت..... ۴۷

۷- برنامه اصلی دستگاه کارت خوان..... ۴۷

۷-۱- پیکره بندی دستگاه کارت خوان..... ۴۴

۷-۲- وقفه تایمر..... ۴۹

۷-۳- وقفه خارجی..... ۴۹

۷-۴- وقفه UART..... ۴۹

۷-۵- تبدیل تاریخ میلادی به شمسی..... ۵۰

۷-۶- حلقه بی پایان..... ۵۱

۸- برنامه اصلی کارت..... ۵۳

۸-۱- پیکره بندی کارت..... ۵۴

۸-۲- وقفه تایمر..... ۵۴

۸-۳- وقفه خارجی..... ۵۵

۸-۴- حلقه بی پایان..... ۵۵

۹- نرم افزار گزارش اطلاعات ورود و خروج..... ۵۶

۹-۱- مشخصات فنی..... ۵۶

۹-۲- مشخصات کاربری..... ۵۷

۱۰- نتایج و جمع‌بندی..... ۵۷

۱۰-۱- سرعت..... ۵۷

۱۰-۲- دقت..... ۵۸

۱۰-۳- امنیت..... ۵۸

۱۰-۴- تداخل با نور محیط..... ۵۸

۱۰-۵- سهولت استفاده..... ۵۸

۱۱- مراجع..... ۵۹

پیوست‌ها..... ۶۰

تاریخچه و بیان مساله

انسان همواره حق مالکیت بر اشیا و لوازم اطرافش را برای خود قایل بوده‌است و برای محافظت از آنها نیز روشهای مختلفی را به کار برده و می برد. این روش‌ها با پیشرفت علم و دانش بشر تغییر کرده و به مرور زمان از اشکال سنتی خود (مانند قراردادن یک تخته سنگ جلوی غار محل زندگی، انواع حصار، پرچین، قفل وکلید و ...) خارج شده و به صورت سیستم های جدید و پیشرفته الکترونیکی و کامپیوتری که امروزه مشاهده می‌کنیم، درآمده اند. ابته استفاده از سیستم‌های امنیتی الکترونیکی هنوز فراگیر نشده و از این حیث در ابتدای امر قرار دارند (در مقایسه با سیستم " قفل - کلید" که تقریبا" همه جا دیده می‌شود). اما با توجه به سرعت و دقت زیاد و استهلاک کم اینگونه سیستم‌ها، در آینده احتمالا" شاهد استفاده از آنها به‌جای قفل و کلیدهای معمول خواهیم بود.

سیستم‌های امنیتی الکترونیکی که تا به امروز معرفی شده‌اند، در کنار مزایای فراوان، معا یبی نیز دارند. مهمترین مشکل در توسعه اینگونه سیستم‌ها، برقرای تعادل بین هزینه ساخت و امنیت آن است به طوری که امنیت سیستم از حد معقول و استاندارد پایینتر نباشد (طبیعتا" امنیت بالاتر هزینه بیشتری می‌طلبد). برای توضیح بیشتر این مورد در ادامه به معرفی کلی واجمالی این سیستم‌ها می‌پردازیم.

همه سیستم‌های امنیتی الکترونیکی، مانند سیستم " قفل – کلید" از دو قسمت عمده تشکیل شده اند:

- یک دستگاه مرکزی که در مکان مورد نظر نصب می‌شود و به آن کارت‌خوان می‌گوییم (عملکرد آن همانند قفل در سیستم "قفل – کلید" است).
- تعدادی قطعه الکترونیکی جدا که در اختیار افراد کاربر سیستم قرار میگیرد و ما در اینجا به آن کارت می‌گوییم (عملکرد آن مانند کلید در سیستم "قفل – کلید" است).

کارت معمولا" اطلاعات از پیش ذخیره شده‌ای در خود دارد و در هنگام نیاز این اطلاعات به طریقی – که بسته به نوع سیستم متفاوت است – به دستگاه مرکزی ارسال می‌شود و در صورت تایید، مجوزهای لازم صادر می-گردد.

این سیستم‌ها به طور عمده به دو نوع کلی "تماس‌دار" و "بدون تماس" تقسیم می‌شوند. در سیستم‌های تماس‌دار جهت تبادل اطلاعات لازم است که بین کارت و کارت‌خوان ارتباط فیزیکی برقرار شود و یک مشکل عمده این سیستم‌ها از همین ناشی می‌شود. معمولا" پس از مدتی کارت و کارت‌خوان از محل اتصال دچار خرابی میشوند و خواندن اطلاعات یا زمان زیادی می‌برد و یا کلا" غیر ممکن می‌شود. ضمن اینکه سرعت سیستم‌های تماس‌دار به نسبت انواع بدون تماس پایینتر است؛ زیرا برقراری اتصال مناسب بین کارت و کارت‌خوان زمان‌بر است.

در سیستم‌های بدون تماس، مانند RFID، مشکل به صورت دیگری است. با توجه به اینکه در یک سیستم RFID سیگنال‌ها در محیط اطراف پخش می‌شوند، لازم است اطلاعات قبل از ارسال رمزنگاری شوند. در غیر اینصورت ممکن است به راحتی مورد سوء استفاده قرار بگیرند و امنیت سیستم پایین می آید. اما در صورت رمزنگاری نیز باید هم در کارت و هم در کارت‌خوان حداقل یکبار عمل رمزنگاری/رمزگشایی انجام شود که این امر پردازشگر قویتر، توان مصرفی بالاتر و حافظه بیشتری را می‌طلبد. بنابراین هزینه ساخت کارت و کارت‌خوان بالا می-رود (به ویژه کارت) و از لحاظ اقتصادی مقرون به صرفه نمی‌باشد.

سیستمی که ما در اینجا ارایه کرده‌ایم، علاوه بر امنیت بالا، هزینه ساخت اندکی داشته و می‌تواند به عنوان یک راهکار امنیتی مناسب در موسسات و کارخانجات مورد استفاده قرار گیرد.

۱. مقدمه

سیستم تشریح شده در این پروژه، یک سیستم کنترل دسترسی فیزیکی است که از دو بخش عمده تشکیل شده است: بخش کارت‌خوان و کارت. کارت یک قطعه مدار الکترونیکی است که در اختیار کاربر قرار دارد و مجوزهای لازم تنها به دارنده کارت داده خواهد شد.در هر کارت اطلاعات مربوط به همان کاربر - که در اینجا فقط شامل ID می‌شود - قرار دارد. دستگاه کارت خوان هم وظیفه خواندن اطلاعات هر کارت و تایید اعتبار آن را به عهده دارد. این سیستم از نوع بدون تماس[1] است. به این معنی که جهت تبادل اطلاعات بین کارت و کارت‌خوان،

1- ContactLess

هیچ گونه تماس فیزیکی لازم نیست. این تبادل پیام از طریق دیودهای مادون قرمز صورت می‌گیرد. نکته جالب توجه در این سیستم این است که ارسال ودریافت پیام در هر یک از دستگاه‌ها توسط تنها یک دیود انجام می‌شود. نحوه انجام این کار در فصول بعد توضیح داده شده‌است.

۱-۱. طراحی سیستم

داشتن دو دیدگاه کلی و جزیی نسبت به یک سیستم کمک زیادی به تجزیه و تحلیل ساده‌تر و طراحی بهتر آن می‌کند. در طراحی این سیستم نیز این دو دیدگاه لحاظ شده‌اند. برای درک بهتر ابتدا موضوعات کلی راجع به این پروژه بحث شده‌است. و در ادامه جزییات آن مورد بررسی قرار گرفته‌اند. سیستم شامل دو بخش کلی است:

۱. طراحی سخت‌افزار

۲. طراحی نرم‌افزار

به دلیل اهمیت سیستم ارسال و دریافت اطلاعات از طریق دیود های مادون قرمز – که بین کارت و کارت خوان مشترک می‌باشد – به طور جداگانه بحث شده است. در ادامه هر یک از قسمت‌های فوق تشریح شده است. در ادامه ابتدا با نگرش کلی به سیستم به تحلیل قسمت‌های مختلف آن می‌پردازیم.

۱-۲. توپولوژی سیستم‌های کنترل دسترسی

سیستم های کنترل دسترسی مرسوم از سه بخش تشکیل شده‌اند:

۱. وسیله کنترل دسترسی (کارت) مانند تگ‌های RFID یا کارت‌های مغناطیسی.

۲. دستگاه خواننده اطلاعات که اطلاعات را از کارت خوانده و به سرور می‌فرستد (کارت‌خوان).

۳. سرور کنترل دسترسی که عمل تایید هویت و انجام عمل متناسب با آن را به عهده دارد.

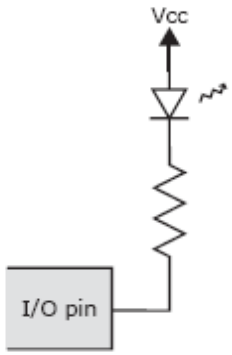
در این پروژه دستگاه کارت خوان و سرور کنترل دسترسی در قالب یک دستگاه در نظر گرفته شده‌اند.

۲. ارتباط بین کارت و کارت خوان

در این پروژه برای ارتباط بین کارت و کارت خوان از یک دیود مادون قرمز استفاده شده‌است. در نمونه های مشابه این ارتباط با یک گیرنده و یک فرستنده مادون قرمز مجزا انجام می‌شود. اما در این پروژه یک ارتباط دو طرفه با کمک تنها یک دیود مادون قرمز ایجاد شده‌است. در واقع می توان به کمک یک دیود فرستنده مادون قرمز (و یا LED های معمولی) از وجود مادون قرمز آگاهی یافت (از فرستنده مادون قرمز به عنوان سنسور مادون قرمز استفاده کرد).

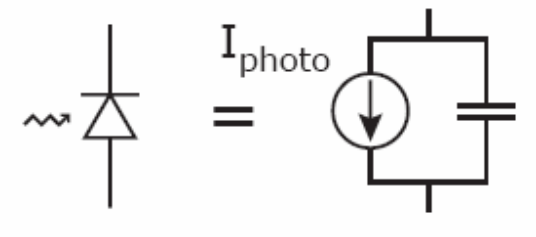
۲–۱. استفاده از یک دیود نوری به عنوان سنسور

یک دیود نوری معمولی را می توان به راحتی به صورت زیر با استفاده از میکروکنترلر راه اندازی کرد.



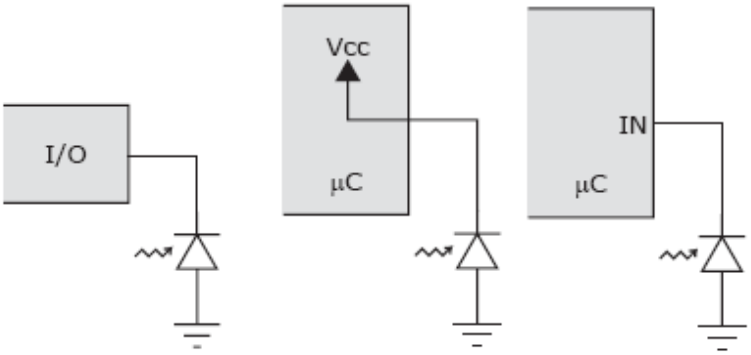
شکل ۳-۱ راه اندازی دیود معمولی

این روش راه اندازی دیود نوری به صورت مستقیم است،اما در حالت بایاس معکوس،دیود نوری را می توان به صورت ساده شده با یک خازن موازی با منبع جریان مدل کرد که این همان جریانی است که متناسب با نور رسیده به دیود است.



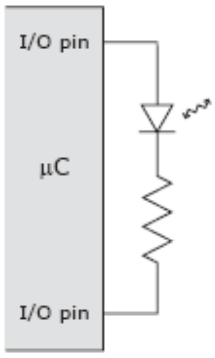
شکل ۳-۲ مدل دیود نوری در حالت بایاس معکوس

یک را ساده برای بکار بردن دیود نوری به عنوان سنسور این است که آنود آن را به زمین و کاتود را به یک پایه ورودی – خروجی میکروکنترلر متصل کنیم.با یک کردن این پایه و معکوس شدن بایاس دیود، خازن شروع به شارژ شدن می‌کند. با تغییر پایه مربوطه به عنوان ورودی، به خازن اجازه دشارژ شدن داده می‌شود، با اندازه‌گیری زمان دشارژ خازن می توان اندازه ای از نور تابیده شده به دیود نوری را داشت. این مراحل در شکل زیر مشخص شده‌اند.



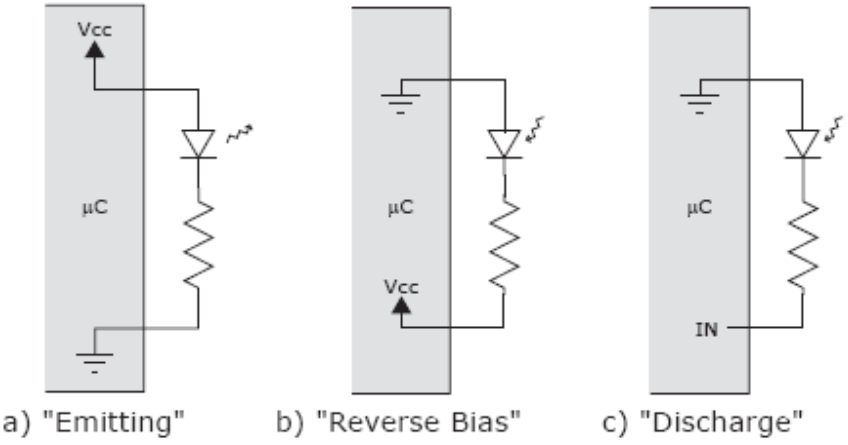
شکل۳-۳ مراحل اندازه‌گیری نور به کمک دیود نوری

در این پروژه برای ارسال و دریافت اطلاعات لازم است که دیود نوری هم در حالت مستقیم و هم در حالت معکوس بایاس شود،برای انجام این کار دیود را به صورت زیر به میکروکنترلر متصل کردیم.



شکل ۳-۴ اتصال دیود نوری به میکرو برای ارتباط دو طرفه

در شکل زیر حالتهای مختلف استفاده از دیود نوری به کمک دو پایه ورودی-خروجی میکروکنترلر نمایش داده شده‌است.



شکل ۳-۵ حالت های مختلف بایاس دیود نوری توسط میکروکنترلر

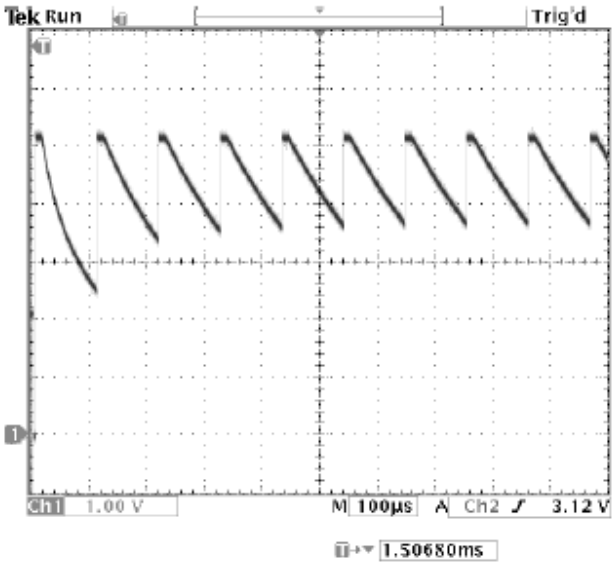
۲-۲. جزییات پروتکل ارتباطی

در این قسمت به تشریح پروتکل ارتباطی می‌پردازیم:

نحوه کار به این صورت است که دو دستگاه به نوبت از خود نور مادون‌قرمز بیرون می‌دهند. یک تابش کوتاه نور، به منزله صفر یا SPACE و یک تابش طولانی نشانگر ۱ یا MARK خواهدبود. شروع پروتکل به این

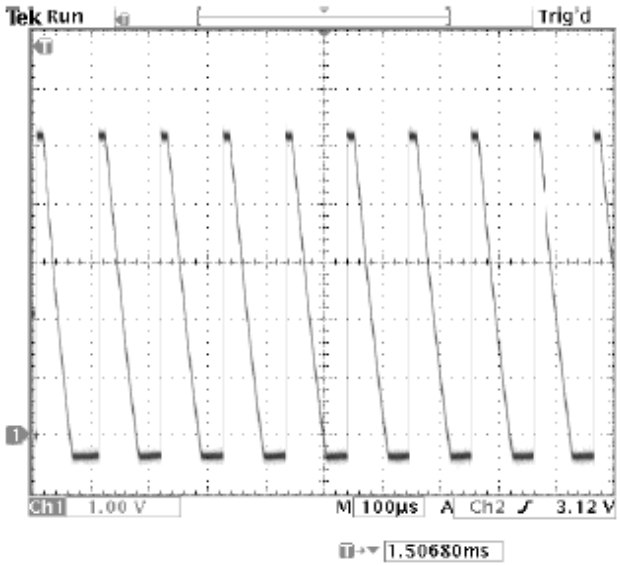
صورت است که دستگاه یک چرخه " بیکاری " را اجرا می‌کند که در آن به مدت ۱ میلی‌ثانیه نور تابیده می‌شود و در ادامه برای ۴ میلی‌ثانیه منتظر دریافت نور می‌شود. در طی مدت دریافت، دستگاه ۴۰ بار نور دریافتی از محیط را اندازه‌گیری می‌کند که یعنی هراندازه‌گیری، ۱۰۰ میکروثانیه طول می‌کشد.

در محیط عادی، مقدار کافی نور مادون‌قرمز جهت دشارژ خازن تا زیر حد آستانه، در طول بازه دریافت، وجود ندارد. در شکل زیر، مقدار ولتاژ پایه کاتد خازن در نور محیط عادی نشان داده شده‌است. محور عمودی به بازه‌های ۱ ولتی و محور افقی به بازه‌های ۱۰۰ میکروثانیه‌ای تقسیم شده‌است. در ابتدا خازن تا ۵ ولت شارژ شده و سپس دشارژ می‌شود ولی همانطور که می‌بینید، سطح ولتاژ هیچ‌گاه از حد آستانه پایینتر نمی‌آید و میکروکنترلر همیشه آن را پایه را ۱ – منطقی – می‌خواند.



شکل ۳-۶ نور اندازه‌گیری‌شده در محیط عادی

شکل زیر نشان‌دهنده همان ولتاژ پایه کاتد است با این تفاوت که نور یک دیود دیگر به نور محیط اضافه شده‌است. مشاهده می‌شود که خازن در طول ۱۰۰ میکروثانیه به طور کامل تخلیه می‌شود و سطح ولتاژ پایه کاتد آن تا زیر حد آستانه پایین می‌آید و در نتیجه میکروکنترلر آن پایه را به عنوان صفر – منطقی- می‌خواند.



شکل ۳-۷ نور اندازه‌گیری‌شده هنگام وجود یک دیود دیگر در محیط اطراف

چرخه بیکاری ادامه پیدا می‌کند تا زمانی که حداقل در ۲ بار اندازه‌گیری پیاپی نور محیط (۲۰۰ میکروثانیه) مشخص شود که نور غیرعادی در آن وجود دارد. در این هنگام دستگاه فرض می‌کند که این دو پالس نور دریافتی، از یک دستگاه مشابه دیگر ارسال شده‌است. بنابراین از چرخه بیکاری – که ۱میلی‌ثانیه تابش نور و ۴ میلی‌ثانیه دریافت آن بود – به چرخه " قفل‌کردن فاز" تغییر وضعیت می‌دهد.

چرخه قفل‌کردن فاز نسبت به چرخه قبل کمی سریعتر است؛ به این صورت که در این حالت نیز زمان ارسال و تابش نور ۱ میلی‌ثانیه خواهدبود ولی در ادامه، تعداد متغیری از دفعات اندازه‌گیری (۱۰۰ میکروثانیه) را درپی دارد. در چرخه قفل‌کردن فاز میکروکنترلر در دو حالت از بازه دریافت خارج می‌شود:

- اگر یک لبه پایین‌رونده مشاهده شود. لبه پایین‌رونده عبارتست از دو پالس نور متوالی (۲۰۰ میکروثانیه) و در ادامه آن عدم مشاهده نور در ۱۰ بار اندازه‌گیری پیاپی نور محیط.
- اگر ۴۰ بار اندازه‌گیری (۴ میلی‌ثانیه) انجام شود و لبه پایین‌رونده مشاهده نشود.

بنا براین الگوی اجرای چرخه قفل‌کردن فاز به ترتیب به شکل زیر است :

- تابش نور از یک دستگاه به مدت ۱ میلی‌ثانیه
- سپس ۱ میلی‌ثانیه هر دو دستگاه منتظردریافت نور می‌شوند
- پس از آن دستگاه دیگر برای ۱ میلی‌ثانیه نور می‌تاباند
- مجدداً هر دو دستگاه به مدت ۱ میلی‌ثانیه منتظر دریافت نور می‌شوند
- کلیه مراحل فوق از ابتدا تکرار می‌شود

حتی اگر دو دستگاه ۲۵٪ خطای فرکانس کلاک داشته‌باشند، باز هم می‌توانند عمل قفل‌کردن فاز را انجام دهند. سرعت اسمی این چرخه ۲۵۰ هرتز است.

در هنگام ارتباط، بیت‌های داده به صورت غیرهمزمان یا آسنکرون فرستاده می‌شوند. یک پالس یک میلی-ثانیه‌ای نور، به معنای ۱ یا MARK و یک پالس ۰,۵ ثانیه‌ای نشانگر صفر یا SPACE است. در چرخه بیکاری، دستگاه مدام MARK ارسال می‌کند. فرمت بسته‌های داده به این صورت است که ابتدا یک SPACE به نشانه آغاز بسته ارسال می‌شود. سپس ۸ بیت داده (اعم از ۰ یا ۱) فرستاده می‌شود و در انتها یک MARK به نشانه پایان بسته ارسال می‌شود. این نوع قالب‌بندی داده‌ها شبیه به قالب مورد استفاده در پروتکل RS232 8-N-1 است.

۳. سخت افزار کارت خوان

۳-۱. RTC (Real Time Clock)

امروزه با پیشرفت علم،زمان اهمیت بیشتری پیدا کرده است.در حال حاضر اغبل دستگاههای الکترونیکی به گونه ای ساخته می شوند تا زمان و تاریخ را در خود نگه دارند.این وسایل برای حفظ تاریخ و ساعت و اعلام آن از یک RTC (Real Time Clock) استفاده می‌کنند. با افزایش درخواست برای استفاده از RTC ،میکروکنترلرهای جدید نظیر AVR نیز از این قابلیت به شیوه های گوناگون حمایت می‌کنند. در اکثر میکروکنترلرهای AVR ،تایمر ۲ میتواند به صورت یک RTC عمل کند،که در اینصورت کلاک مورد نیاز را به صورت خارجی و از یک کریستال ساعت 32KHz تامین می‌کند. علاوه بر این تراشه هایی نیز به عنوان RTC وجود دارند که غالبا زمان، شامل ثانیه، دقیقه ،ساعت و تاریخ شامل روز،ماه،سال را با در نظر گرفتن سال کبیسه در خود نگه می‌دارند. این تراشه ها را می توان به صورت I2C ، SPI و ... به میکروکنترلر متصل کرد.

در این پروژه با توجه به حجم بالای محاسبات میکروکنترلر برای سایر قسمتهای دستگاه به جای استفاده از تایمر ۲ از یک تراشه DS1307 استفاده شده‌است. این تراشه از طریق باس I2C به میکروکنترلر متصل می‌شود.

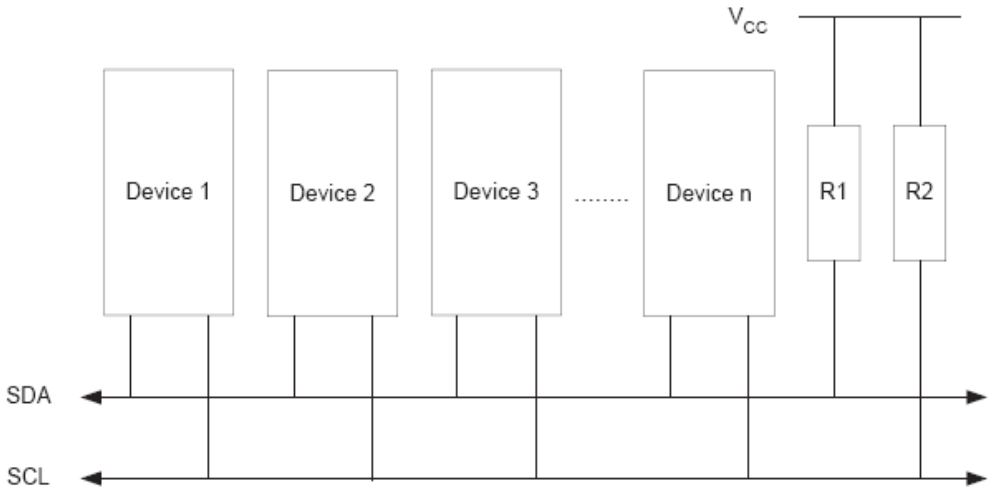
۳-۲. ارتباط سریال دو سیمه(TWI یا I2C)

ارتباط سریال دو سیمه یک پروتکل ارتباطی سریال است که توسط شرکت Philips ارائه شده‌است که قابلیت‌های آن به صورت زیر است:

- یک واسطه ارتباطی بسیار ساده و در عین حال انعطاف پذیر و قدرتمند،که تنها به دو خط باس نیاز دارد.
- می‌تواند به صورت Master و Slave کار کند.
- وسیله می‌تواند به صورت فرستنده و یا دریافت کننده عمل کند.
- با داشتن فضای آدرس ۷ بیتی،می‌تواند تا حداکثر ۱۲۸ وسیله Slave را آدرس دهی کند.
- از وجود چند Master حمایت می‌کند.

- حداکثر سرعت آن می‌تواند به 400KHz برسد.
- به مدار حذف کننده نویز داخلی که اسپایکهای موجود بر روی خطوط باس را حذف می‌کند مجهز است.
- امکان تنظیم کامل آدرس Slave و حمایت از فراخوانیهای عمومی را دارد.
- در صورتی که آدرس مربوط به خود را تشخیص دهد می‌تواند AVR را از حالت Sleep بیدار کند.

عموما واسطه ارتباط دو سیمه برای کار با میکروکنترلر ها مناسب است، پروتکل TWI این امکان را به طراح می‌دهد تا حداکثر ۱۲۸ وسیله مختلف را تنها با استفاده از دو خط باس دو طرفه، یکی برای پالس ساعت (SCL) و دیگری برای داده (SDA)،به یکدیگر متصل کند. تنها سخت افزار خارجی که برای ایجاد این باس مورد نیاز است یک مقاومت بالاکش (Pull up)برای هر یک از خطوط باس است. تمامی وسایل متصل به باس، آدرسهای خاص خود را دارند و نحوه ارتباط بیت آنها نیز توسط پروتکل TWI مشخص می‌شود.

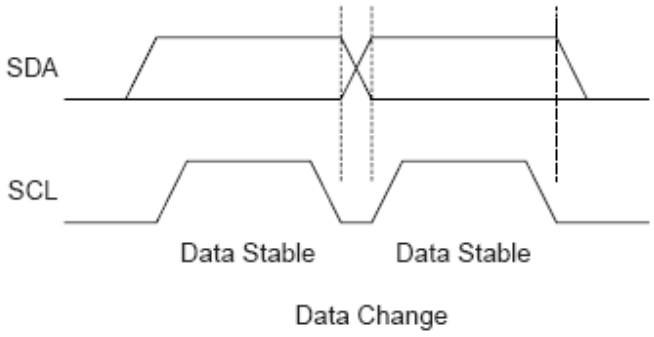


شکل ۴-۱ اتصالات باس TWI

در تمامی وسایلی که از TWI حمایت می‌کنند، درایورهای باس به صورت Open-Drain یا Open-Collector می‌باشند. این ویژگی موجب می‌شود تا آنها به صورت Wire-AND عمل کنند، که این برای عملکرد صحیح باس

ضروری است. بنابراین یک سطح پایین در خط باس TWI، زمانی تولید می‌شود که خروجی یک یا چند وسیله صفر باشد و سطح بالای آن نیز تنها زمانی که تمام وسایل TWI در حالت امپدانس بالا باشند، حاصل می‌گردد. تعداد وسایل مجاز برای اتصال به یک بای تنها توسط محدودیت ظرفیت باس که حداکثر باید 40pF باشد و نیز فضای آدرس ۷ بیتی Slave،مشخص می گردد.

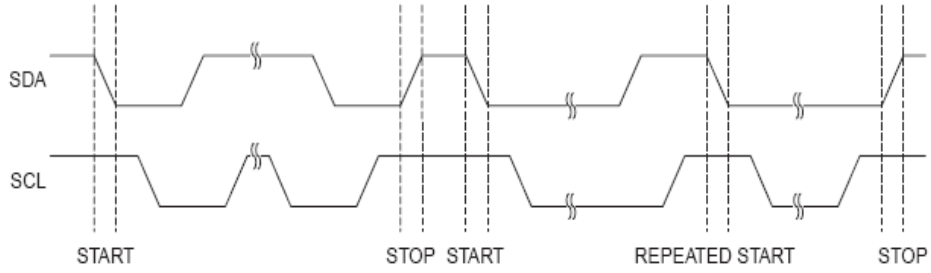
به ازای هر بیت داده که بر روی باس TWI فرستاده می‌شود، یک پالس در خط کلاک آن را همراهی می‌کند. زمانی که خط کلاک بالاست، سطح خط داده باید ثابت باقی بماند. تنها استثنای این قانون در تولید حالتهای شروع و توقف است.



شکل۴-۲ / ارزیابی داده‌ها

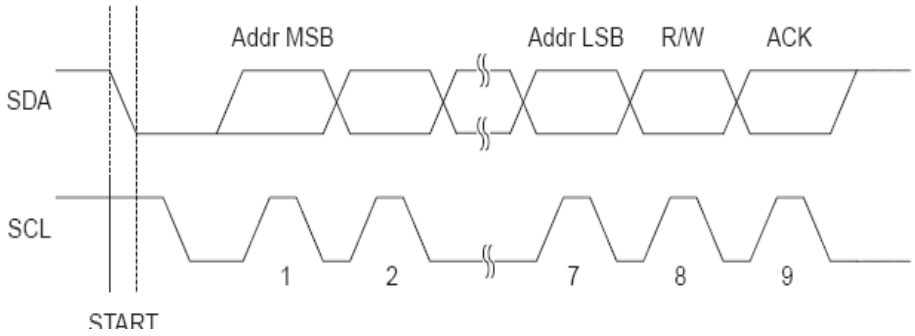
آغاز و توقف ارسال داده توسط Master صورت می گیرد، زمانی که Master، حالت شروع را روی باس ایجاد می‌کند، ارسال آغاز می‌شود و زمانی که حالت توقف را ایجاد می‌کند، ارسال پایان می‌پذیرد. در بین حالتهای شروع و توقف،باس مشغول، در نظر گرفته می‌شود و Master دیگری نباید سعی در کنترل باس نماید. یک حالت خاص زمانی رخ می‌دهد که در بین یک حالت شروع و توقف، حالت شروع جدیدی ایجاد شود. به این حالت حالت شروع مکرر (Repeated Start) گفته می‌شود و زمانی اتفاق می‌افتد که Master بخواهد بدون از دست دادن کنترل باس، ارسال جدیدی آغاز کند. بعد از یک شروع مکرر، باس تا رسیدن حالت توقف بعدی،مشغول در نظر گرفته می‌شود.

همانطور که در شکل زیر نشان داده شده‌است، حالت‌های شروع و توقف با تغییر سطح خط SDA در زمانی که خط SCL بالاست، انجام می‌شود.



شکل ۴-۳ حالت شروع، شروع مکرر و خاتمه

تمامی بسته های آدرس فرستاده شده بر روی بای TWI ، ۹ بیت طول دارند که از ۷ بیت آدرس،یک بیت کنترل خواندن / نوشتن و یک بین تصدیق (Acknowledge) تشکیل شده اند.



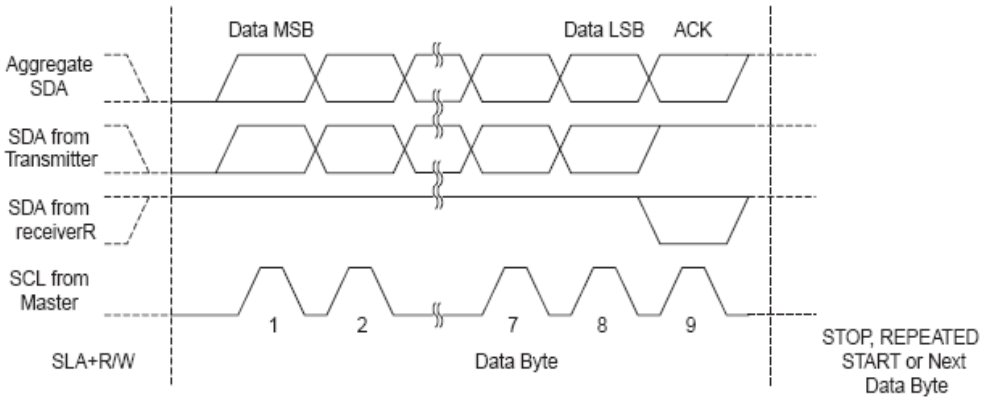
شکل ۴-۴ قالب بسته آدرس

اگر بیت خواندن/ نوشتن یک شود،پس از آن عمل خواندن انجام می‌شود و در غیر این صورت عمل نوشتن انجام می‌شود. زمانی که Slave تشخیص دهد که آدرس روی باس به آن تعلق دارد،باید در سیکل نهم SCK،با زمین کردن SDA با آن پاسخ دهد. اگر Slave آدرس دهی شده، مشغول باشد و یا به هر دلیلی نتواند به درخواست Master پاسخ دهد، خط SDA باید در سیکل کلاک ACK بالا باقی بماند. پس از آن Master می‌تواند یک حالت توقف، یا یک حالت شروع مکرر را برای آغاز یک ارسال مجدد بفرستد.یک بسته آدرس که شامل یک آدرس Slave و یک بیت خواندن و نوشتن باشد، به ترتیب به صورت SLA+R و SLA+W نشان داده می‌شود.

در ارسال آدرس Slave، ابتدا بیت MSB اسرال می‌گردد. آدرسهای Slave می‌توانند هر یک از مقادیر ۱ تا ۱۲۷ را به خود بگیرند ولی از آدرس صفر برای فراخوانی عمومی استفاده می‌شود. زمانی که یک فراخوانی عمومی

انجام می‌شود، بیاد تمامی Slave ها در سیکل ACK با زمین کردن خط SDA به آن پاسخ دهند. .هنگامی که Master بخواهد تا یک پیغام را برای تمامی Slave های موجود ارسال کند، از یک فراخوانی عمومی استفاده می‌کند. در صورتی که یک آدرس فراخوانی عمومی و به دنبال آن یک بیت نوشتن ارسال گردد، تمامی Slave هایی که می-توانند به فراخوانی پاسخ دهند، در سیکل ACK خط SDA را زمین می‌کنند. در این صورت بسته های داده بعدی توسط تمامی Slave هایی که به فراخوانی عمومی پاسخ داده اند، دریافت می گردد. باید توجه شود که ارسال آدرس فراخوانی عمومی و به دنبال آن، یک بیت خواندن، بدون معناست؛ چرا که در این صورت چندین Slave داده های مختلفی را بر روی باس قرار می‌دهند که باعث مختل شدن باس می‌شود.

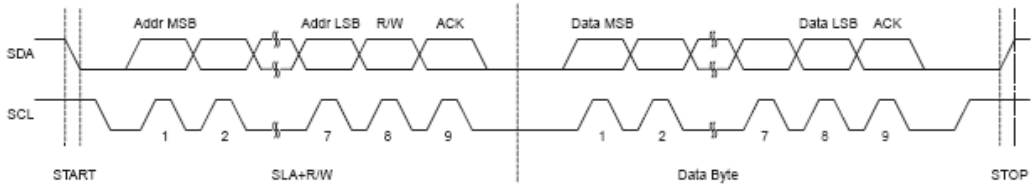
تمامی بیت های داده که بر روی باس TWI ارسال می شوند ۹ بیتی هستند که شامل یک بایت داده و یک بیت تصدیق (Acknowledge)می باشند.



شکل ۴-۵ قالب بسته داده

در حین ارسال داده، Master کلاک و حالتهای شروع و توقف را تولید می‌کند، در صورتی که دریافت کننده تنها باید دریافت داده را تصدیق کند. یک تصدیق توسط دریافت کننده در سیکل نهم SCL و با زمین کردن خط SDA اعلان می‌گردد. اگر دریافت کننده دریافت را به درستی انجام ندهد، خط SDA را در حالت بالا رها می‌کند (NACK). زمانی که دریافت کننده آخرین بایت را دریافت کند یا به هر دلیلی نتواند بایت دیگری را دریافت کند، باید با ارسال یک NACK بعد ار آخرین بایت باخبر کند.

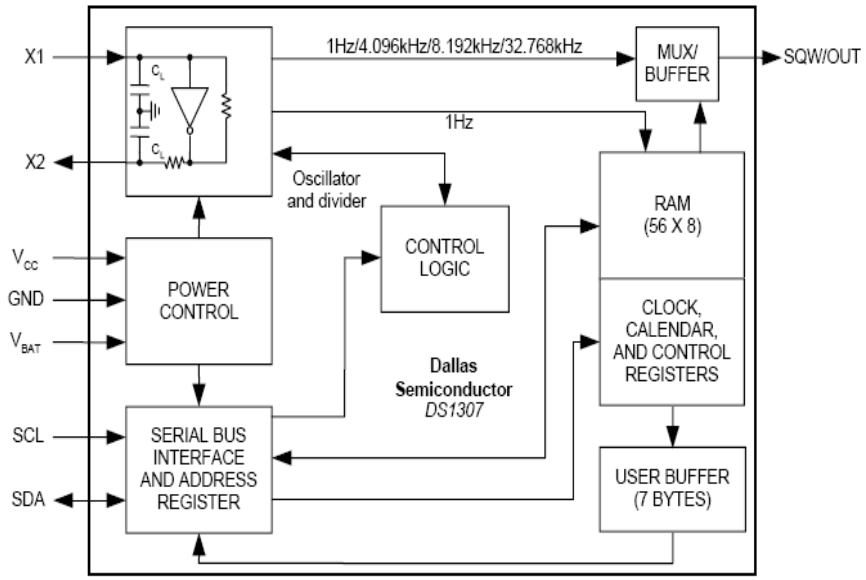
اصولا یک ارسال، از یک حالت شروع، یک SLA+R/W و یک یا چند بسته داده تشکیل شده‌است. یک پیغام تهی که از یک حالت شروع و به دنبال آن یک حالت توقف تشکیل شده باشد غیر قانونی است.



شکل ۴-۶ نحوه تبادل داده‌ها

۳-۳. معرفی تراشه DS1307

DS1307 یک تراشه RTC سریال، با توان مصرفی کم و خروجی های BCD (Binary Coded Decimal) با ۵۶ بایت حافظه SRAM است. آدرس و داده به صورت سریال و از طریق باس I2C منتقل می شوند .این تراشه اطلاعات ثانیه،دقیقه، ساعت، روز هفته،روز،ماه و سال را در خود نگاه می دارد. علاوه بر این تعداد روزهای ماه،برای ماه های با کمتر از ۳۱ روز و همچنین سال کبیسه به صورت خودکار در این تراشه در نظر گرفته شده‌است. البته تقویم موجود در این تراشه و تراشه های مشابه همگی به صورت میلادی تنظیم شده‌است. ساعت داخلی تراشه می-تواند به صورت ۲۴ ساعته و یا ۱۲ ساعته با نمایانگر AM/PM تنظیم گردد. این تراشه یک مدار داخلی برای تشخیص تغذیه دارد و در صورت قطع تغذیه،به صورت خودکار تغذیه خود را از باطری پشتیبان تامین می‌کند.



شکل ۴-۷ بلوک دیاگرام داخلی DS1307

X1 و X2

این دو پایه باید به کریستال کوترتز 32.768KHz متصل شوند.

Vbat

این پایه،ورودی تغذیه پشتیبان است که معمولا به یک باطری ۳ ولت لیتیوم یا هر منبع انرژی دیگری با ولتاژی بین ۲,۵ تا ۳,۵ ولت متصل می‌شود. ممکن است سری کردن یک دیود با باطری و اتصال آن به این پایه عملکرد خوبی نداشته باشد.در صورتی که نیازی به تغذیه پشتیبان نباشد، این پایه باید زمین شود.در صورت قطع تغذیه تراشه، یک باطری 4.8mAhr می‌تواند اطلاعات داخلی تراشه را تا ۱۰ سال حفظ کند.

GND

این پایه به زمین متصل می‌شود.

SDA (Serial Data Input/Output)

این پایه،خط باس I2C است.پایه SDA،به صورت Open-Drain است و به یک مقاومت بالاکش خارجی نیاز دارد.

SCL (Serial Clock Input)

این پایه خط ورودی کلاک در باس I2C است که برای سنکرون کردن انتقال داده ها در ارتباط سریال به کار می‌رود. از آنجا که این پایه به صورت Open-Drain است به یک مقاومت بالاکش خارجی نیاز دارد.

SQW/OUT (Squar Wave Output)

زمانی که بیت SQWE برابر یک نوشته شود، این پایه یک موج مربعی را با فرکانس 8KHz،4KHz،1Hz یا 32KHz تولید می‌کند. این پایه به صورت Open-Drain است و به یک مقاموت بالاکش خارجی نیاز دارد.

VCC

این پایه منبع تغذیه اولیه تراشه است. محدوده قابل قبول تغذیه بین ۴,۵ تا ۵,۵ ولت است. در این حالت تراشه کاملا در دسترس است و می توان اطلاعات را بر روی آن نوشت یا از روی آن خواند ولی در صورتی که با افت ولتاژ Vcc،تراشه از تغذیه پشتیبا استفاده کند، اعمال خواندن و نوشتن غیر فعال می شوند اما ساعت و تقویم موجود در تراشه به کار خود ادامه می‌دهند. دقت کلاک این تراشه،به دقت کریستال به کار رفته بسنگی دارد. دما نیز می-تواند در فرکانس کریستال تغییر ایجاد کند.

۳-۳-۱. تقویم و ساعت

اطلاعات تقویم و ساعت، با خواندن رجیسترهای مربوطه در تراشه بدست می‌آیند و با نوشتن مقادیر مناسب در این رجیسترها،می توان آنها را تنظیم کرد. جدول زیر رجیسترهای RTC را نشان می‌دهد.

جدول(۴-۱) رجیسترهای DS1307

ADDRESS	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	FUNCTION	RANGE
00H	CH	10 Seconds			Seconds				Seconds	00–59
01H	0	10 Minutes			Minutes				Minutes	00–59
02H	0	12	10 Hour	10 Hour	Hours				Hours	1–12 +AM/PM 00–23
		24	PM/ AM							
03H	0	0	0	0	0	DAY			Day	01–07
04H	0	0	10 Date		Date				Date	01–31
05H	0	0	0	10 Month	Month				Month	01–12
06H	10 Year				Year				Year	00–99
07H	OUT	0	0	SQWE	0	0	RS1	RS0	Control	—
08H-3FH									RAM 56 x 8	00H–FFH

محتوای این رجیسترها به صورت BCD هستند.رجیستر مربوط به روز هفته در نیمه شب افزایش می یابد و مقادیر مربوط به روزهای هفته توسط کاربر تعریف می شوند و باید پشت سر هم باشد (به عنوان مثال اگر ۱ بیانگر یکشنبه باشد،۲ بیانگر دوشنبه است و ...). رجیسترهای RTC از آدرس 00h تا 07h قرار دارند و رجیسترهای RAM نیز در آدرس 08h تا 3Fh واقع شده اند.

۳-۳-۲. رجیستر کنترل DS1307

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
OUT	0	0	SQWE	0	0	RS1	RS0

شکل ۴-۸

این رجیستر برای کنترل عملکرد پایه SQW/OUT استفاده می‌شود.

بیت ۷ – OUT (Output Control)

در زمانی که موج مربعی خروجی پایه SQW/OUT غیر فعال باشد، این بیت سطح خروجی این پایه را کنترل می‌کند. اگر این بیت یک باشد سطح پایه SQW/OUT برابر یک و در غیر این صورت صفر خواهدشد.

بیت ۴ – SQWE (Square – Ware Enable)

زمانی که این بیت یک شود،خروجی موج مربعی در پایه SQW/OUT فعال می‌شود که در این صورت فرکانس تولید موج مربعی توسط بیتهای RS1 و RS0 تعیین می‌گردد.

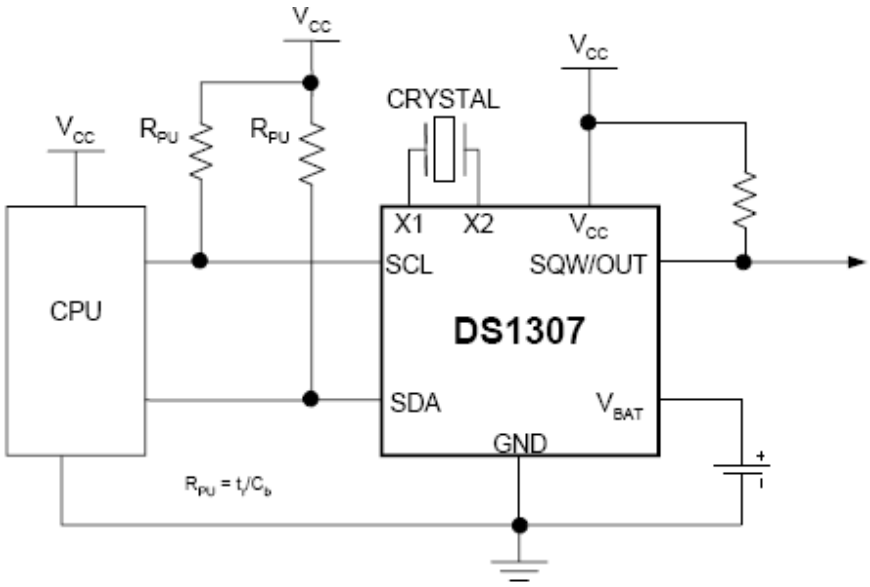
بیتهای ۰ و ۱ – RS1:0 (Rate Select)

در صورتی که موج مربعی خروجی در پایه SQW/OUT فعال شده باشد، این دو بیت به صورت جدول زیر فرکانس موج مربعی را مشخص می‌کند.

جدول(۴-۲) تنظیمات پایه SQW/OUT

RS1	RS0	SQW/OUT OUTPUT	SQWE	OUT
0	0	1Hz	1	X
0	1	4.096kHz	1	X
1	0	8.192kHz	1	X
1	1	32.768kHz	1	X
X	X	0	0	0
X	X	1	0	1

مدار نوعی ارتباط تراشه با میکروکنترلر به صورت زیر است:



شکل ۴-۹ نحوه اتصال تراشه به میکروکنترلر

۳-۳-۳. نرم افزار ارتباط با DS1307

این نرم افزار شامل دو بخش عمده است، یکی ارتباط با تراشه و خواندن و نوشتن اطلاعات و دیگری تبدیل تاریخ میلادی به شمسی است.

کامپایلر Codevision شامل توابع از پیش تعریف شده برای ارتباط با این تراشه از طریق باس I2C است. این توابع در فایل ds1307.h موجودند که برای استفاده از آنها ابتدا نام پورتی را که تراشه از طریق آن به میکروکنترلر متصل شده‌است را مشخص کرد سپس این فایل ds1307.h را به برنامه اضافه کرد.

```
/* the I2C bus is connected to PORTB */
/* the SDA signal is bit 3 */
/* the SCL signal is bit 4 */
#asm
        .equ __i2c_port=0x18
        .equ __sda_bit=3
        .equ __scl_bit=4
#endasm

/* now you can include the DS1307 Functions */
#include <ds1307.h>
```

توابع مورد استفاده به شرح زیراند:

Void rtc_init(unsigned char rs,unsigned sqwe,unsigned char out)

این تابع تنظیمات اولیه مربوط به تراشه را انجام می‌دهد.لازم است تا قبل از فراخوانی این تابع با استفاده از تابع i2c_init()، باس I2C مقداردهی اولیه شود.

Void rtc_get_time(unsigned char *hour,unsigned char *min,unsigned char *sec)

این تابع زمان جاری RTC را بر می گرداند.

Void rtc_set_time(unsigned char hour,unsigned char min,unsigned char sec)

این تابع زمان جاری RTC را تنظیم می‌کند.

Void rtc_get_date(unsigned char *date,unsigned char *month,unsigned char *year)

این تابع تاریخ جاری RTC را بر می‌گرداند.

Void rtc_set_date(unsigned char date,unsigned char month,unsigned char year)

این تابع تاریخ جاری RTC را تنظیم می‌کند.

۳-۴. ارتباط با رایانه

از مهم ترین ویژگی های دستگاه های کنترل ورود و خروج ارتباط آنها با رایانه و تبادل اطلاعات است، از این رو در این سیستم نیز این امکان در نظر گرفته شده‌است تا به وسیله آن اطلاعات افراد و همچینین زمان ورود و خروج آنها که بر روی MMC ذخیره شده‌است به رایانه انتقال یابد. برای جلوگیری از دسترسی افراد غیر مسئول به اطلاعات دستگاه یک کلمه رمز در نظر گرفته شده‌است. تنها در صورتی که این کلمه رمز در ابتدای برقراری ارتباط، از سوی رایانه ارسال شود اطلاعات توسط میکروکنترلر از MMC خوانده شده و برای رایانه ارسال می‌شود.

۳-۴-۱. ارتباط سریال

مبادله‌ی بیت به بیت اطلاعات تنها از طریق یک کانال (البته به‌جز زمین) را ارتباط سریال می‌گویند. انواع ارتباط سریال به شرح زیر است:

- سنکرون (همزمان): یک سیگنال زمانی از طریق یک کانال مجزا همراه با پیام ارسال می‌شود (مثل استاندارد I2C و RS232).
- آسنکرون (غیرهمزمان): در این روش اطلاعات زمان بندی همراه با سیگنال ارسال می‌شود (مثل RS232، RS485).

از لحاظ تئوری تنها یک سیم برای انتقال اطلاعات سریال به صورت آسنکرون لازم است اما در واقعیت این مسئله عملی نیست. به عنوان مثال اگر یک بیت از اطلاعات بر اثر خطا یا نویز تغییر کند ممکن است کل اطلاعات بعد از آن یک بیت شیفت پیدا کند و پس از تفسیر و تبدیل به دیتای موازی کل اطلاعات مخدوش شود. بنابراین نیاز به استانداردهایی وجود دارد که امکان ارتباط قابل اطمینان را فراهم کند. یکی از این استانداردها RS232C است که در سال ۱۹۶۹ توسط موسسه EIA تعریف شد. اگرچه نام این استاندارد RS232C است اما معمولا به نام

RS232 شناخته می‌شود و مخفف Recommended Serial می باشد. این استاندارد معمولا در پورت سریال کامپیوترهای شخصی استفاده می‌شود.

پروتکل ارتباطی RS232 در لایه ی هفتم مدل OSI قرار می‌گیرد و موارد زیر در آن تعریف شده‌است:

- مشخصات الکتریکی سیگنال از قبیل سطح ولتاژ، نرخ سیگنال، زمان بندی و Slew Rate، سطح مقاومت ولتاژ، رفتار اتصال کوتاه، بیشینه ی جریان ظرفیتی و طول کابل
- مشخصات مکانیکی واسط از قبیل سوکت اتصال و تعریف پین ها
- عملکرد هر مدار در کانکتور واسط
- زیر مجموعه های استاندارد از مدارهای واسط برای آن کاربرد ارتباطی

همانطور که گفته شد این استاندارد در لایه های زیرین مدل OSI کار می‌کند و مشخصات لایه های بالا از جمله کدگذاری کاراکتر (مثلا ASCII) و همچنین قالب بندی کاراکتر (مثل بیت شروع و پایان یا بیت توازن) در قالب این استاندارد قرار نمی‌گیرد و معمولا بوسیله ی نرم افزار تعیین و تفسیر می شوند. این استاندارد سقف ماکزیمم نرخ بیت را ۲۰,۰۰۰ bps تعریف کرده است؛ در حالی که بسیاری از تجهیزات جانبی مثل مودم ها از این اندازه تجاوز کرده و با نرخ هایی بالاتر از این (مثل: ۳۸,۴۰۰ ، ۳۷,۶۰۰ و ۱۱۵,۲۰۰) با سطح ولتاژ RS232 کار می-کنند.

یکی از معایب استاندارد RS232 بر خلاف RS422 و RS485 این است که تنها برای ارتباط نظیر به نظیر (Peer to Peer) طراحی شده‌است و امکان شبکه شدن با دیگر ادوات را ندارد. این استاندارد برای فواصل تا حدود ۱۵ متر مناسب می‌باشد و برای فواصل بیشتر می توان از روش ۲۰mA loop استفاده کرد (مثل پروگرامر بعضی PLC های زیمنس). اما روش مناسب دیگر در محیط های صنعتی استفاده از RS485 می باشد که در فواصل زیاد

تا فرکانس ۱MHz نیز کار می‌کند. بنابراین RS232 برای ارتباطات کم دامنه مثل اتصال دو PC، PC با یک دستگاه یا دو دستگاه با یکدیگر استفاده می‌شود.

۳-۴-۲. ارتباط UART

ارتباط UART یکی از پروتکلهایی است که توسط انواع رایانه ها حمایت می‌شود و لذا برای برقراری ارتباط بین میکرو کنترلر و رایانه اغلب از این روش استفاده می‌شود.

بخش ارتباط سریال UART در میکرو کنترلرهای AVR قابلیت های متنوعی دارد که از جمله آن می توان به موارد زیر اشاره کرد:

- عملکرد Full Duplex
- عملکرد سنکرون و آسنکرون
- عمل به صورت Master و Slave در حالت سنکرون
- تولید کننده نرخ ارسال (Baudrate) دقیق
- حمایت از فریم های سریال ۵ – ۶ – ۷ – ۸ -۹ بیت داده و ۱ یا ۲ بیت توقف
- تولید Parity به صورت زوج یا فرد و امکان چک کردن سخت افزاری آن
- تشخیص خطاهای سرریز و نوع فریم
- فیلتر پایین گذر دیجیتال
- تولید سه وقفه مجزا برای اتمام TX ، خالی شدن رجیستر TX و اتمام RX
- کار در حالت ارتباط چند پردازنده

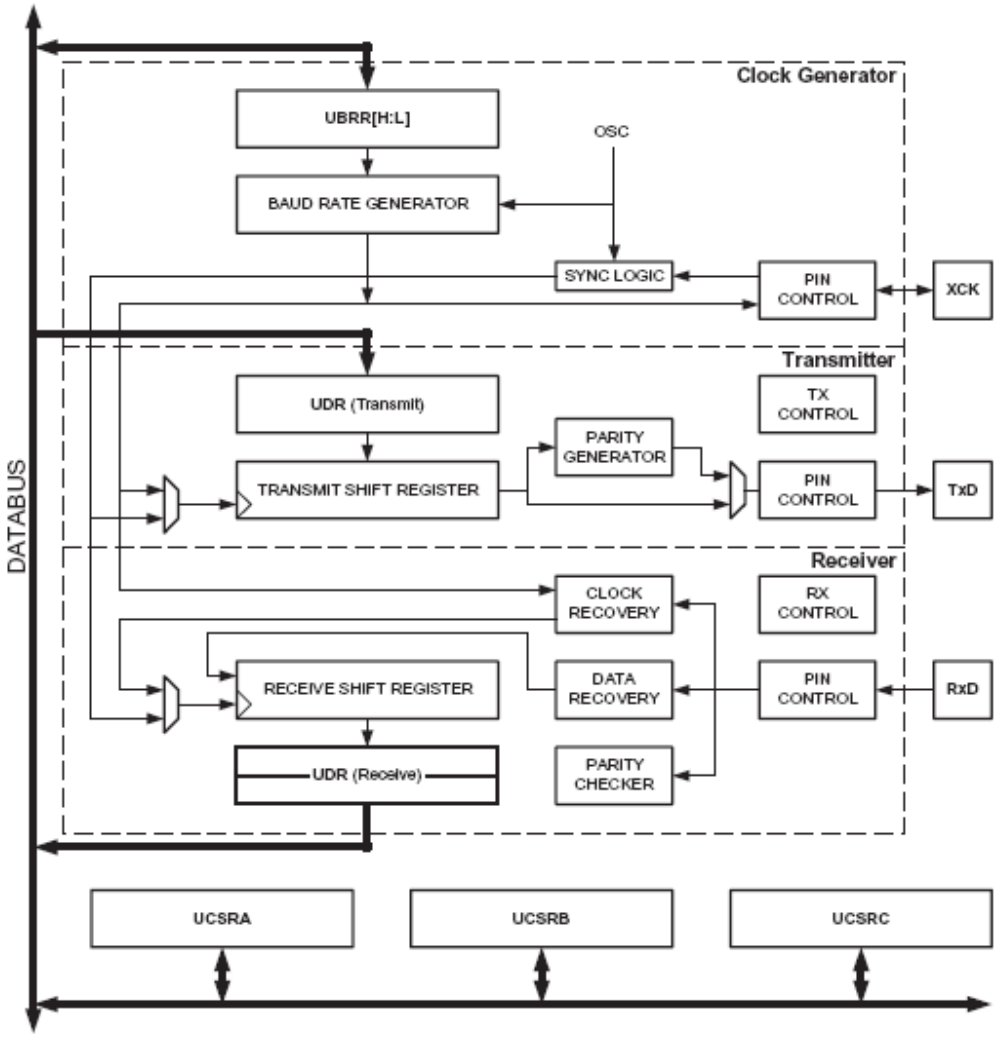
- امکان دوبرابر کردن سرعت در حالت آسنکرون

۳-۴-۳. سازگاری USART با UART در AVR

بعضی از انواع AVR تنها از ارتباط UART حمایت می‌کنند، به این معنی که ارتباط سریال تنها به صورت آسنکرون قابل انجام می‌باشد و انواع پیشرفته تر میکروکنترلرهای AVR می‌توانند به صورت سنکرون و آسنکرون ارتباط برقرار کنند. این دو نوع ارتباط سریال از نظر محل بیت ها در داخل رجیسترها، نحوه تولید نرخ ارسال، نحوه ارسال و دریافت اطلاعات و عملکرد بافر مربوط به ارسال اطلاعات کاملا مطابقت دارند و تنها عملکرد بافر مربوط به دریافت اطلاعات در ارتباط USART بهبود یافته است.

نگاه اجمالی به ارتباط UART

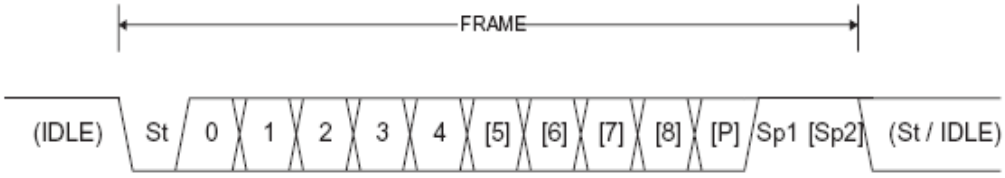
واحد ارتباط سریال، یک تولید کننده کلاک داخلی را شامل می‌شود که ار آن در حالت آسنکرون و سنکرون در مد Master استفاده می‌کند. در این واحد یک کانتر وجود دارد که همیشه به صورت کاهشی می‌شمارد. این کانتر کلاک خود را به صورت مستقیم از کلاک سیستم می گیرد.هر یار که محتوای این کانتر به صفر می‌رسد یک کلاک تولید می‌شود و محتوای کانتر از رجیستر UBRR (USART Baud Rate Register) بار می‌شود، سپس بسته به مد عملکرد ارتباط سریال، کلاک تولید شده بر ۲، ۸ یا ۱۶ تقسیم می‌شود که قسمتهای مختلف ار آن استفاده می‌کنند.



شکل ۴-۱۰ بلوک دیاگرام ارتباط سریال

یک فریم سریال از یک کاراکتر داده با بیتهای همزمان ساز (بیتهای شروع و توقف) و یک بیت Parity اختیاری برای تشخیص خطا تشکیل شده‌است. USART هر ۳۰ ترکیب معتبر از فریمهای زیر را حمایت می‌کند:

- یک بیت شروع
- ۵، ۶، ۷، ۸ یا ۹ بیت داده
- بیت Parity زوج یا فرد یا هیچکدام
- یک یا دو بیت توقف



St	Start bit, always low.
(n)	Data bits (0 to 8).
P	Parity bit. Can be odd or even.
Sp	Stop bit, always high.
IDLE	No transfers on the communication line (RxD or TxD). An IDLE line must be high.

شکل ۴-۱۱ فرمت فریم ارتباط سریال

هر فریم با بیت شروع آغاز می‌شود و بعد از آن بیتهای داده، از کم ارزش ترین بیت ارسال می‌شوند. بعد ار بیتهای داده بیت parity و در آخر بیت‌های توقف فرستاده می‌گردند.

قبل از برقراری ارتباط سریال باید UART تنظیم شود. این مرحله شامل تنظیم نرخ ارسال، تعیین نوع فریم و فعال کردن قابلیت ارسال و یا دریافت است.

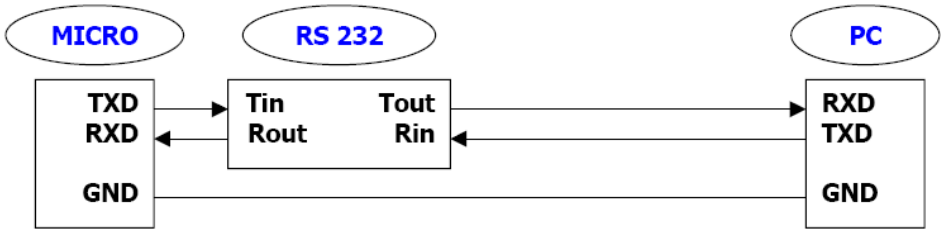
واحد تولید نرخ ارسال با استفاده از کلاک میکروکنترلر، نرخ های مختلف را تولید میکند. نکته مهم در ارتباط سریال این است که تولید نرخ های ارسال مختلف به کمک نوسانگرها و کریستالهایی با فرکانس استاندارد میتواند خطایی به همراه داشته باشد و از آنجاییکه در بعضی موارد این خطا بیش از حد بزرگ است، باید در هنگام طراحی به این نکته توجه نمود.

بنابر این اگر قصد استفاده از ارتباط سریال با کامپیوتر را داریم باید از کریستال هایی با فرکانس بالا استفاده کنیم که در این پروژه از کریستال 16MHZ استفاده شده‌است تا نرخ ارسال 56000bps بدون خطا ایجاد شود.

۳-۴-۴. سخت افزار لازم برای ارتباط با رایانه

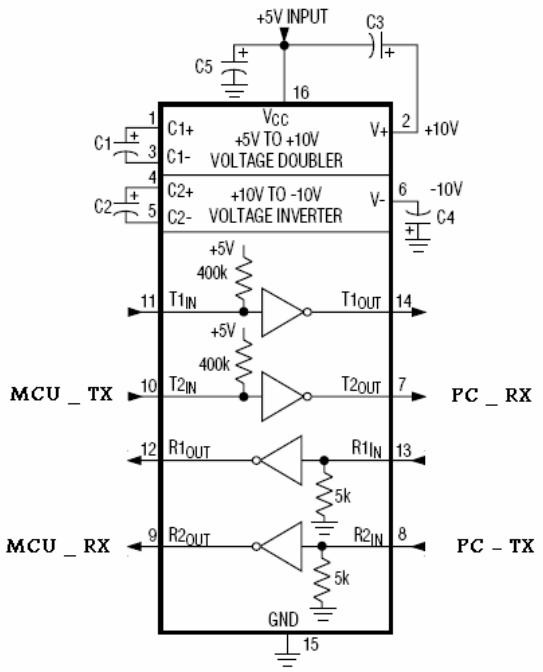
با توجه به اینکه برای برقراری ارتباط با رایانه ناچاریم از استاندارد RS232 استفاده کنیم، لازم است به نحوی بتوانیم سطوح ولتاژ TTL ایجادشده توسط میکروکنترلر و استاندارد RS232 را به یکدیگر تبدیل کنیم. در استاندارد RS232 سطح منطقی صفر ولتاژی بین ۳ تا ۲۵ ولت و سطح منطقی یک ولتاژی بین ۳- تا ۲۵- ولت دارند.

برای تبدیل سطوح ولتاژ TTL و استاندارد RS232 روش های مختلفی وجود دارد، اما به عنوان یکی از ساده ترین راه‌ها، ما از یک تراشه MAX232 استفاده کرده‌ایم.



شکل ۴-۱۲ شمای کلی ارتباط با رایانه به کمک تذاشه MAX232

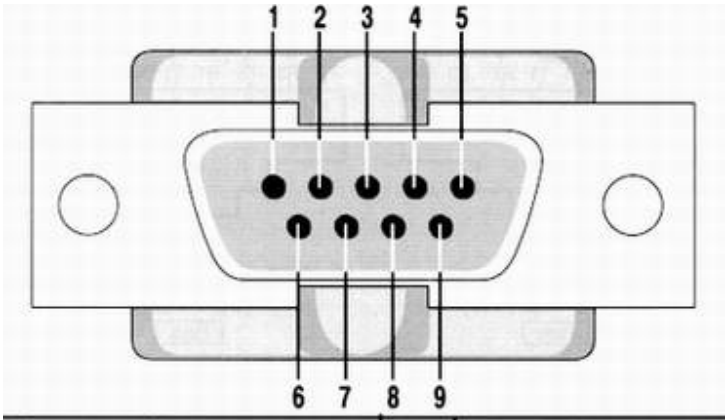
اما مدار عملی برای راه اندازی تراشه MAX232 به صورت زیر است، تمام خازن ها یک میکروفارادی هستند.



شکل ۴-۱۳ مدار راه اندازی تراشه MAX232

۳-۴-۵. توضیحاتی راجع به پورت سریال رایانه

DTE سرنام Data Terminal Equipment است و DCE مخفف Equipment Data Communications. این عبارات برای نشان دادن پین های کانکتور یک دستگاه و جهت ارتباطی سیگنال در پین ها می باشند. معمولا کامپیوتر یک DTE می باشد و دستگاه های دیگر یک DCE و شما می توانید در ادامه ی توضیحات DTE را یک کامپیوتر و DCE را یک دستگاه دیگر فرض کنید. در این استاندارد DTE از یک کانکتور ۹ پین Female (یا ۲۵ پین) و DCE از یک کانکتور ۹ پین Male (یا ۲۵ پین) استفاده می‌کند. بنابراین می توان یک DCE را مستقیما به یک DTE کرد اما در حالتهای غیر از این باید از کابل های Null Modem استفاده کرد که در ادامه به آن خواهم پرداخت. اگرچه استاندارد RS232 برای کانکتور ۲۵ پین نیز تعریف شده‌است اما ما در اینجا بحثمان را بر روی کانکتور ۹ پین متمرکز می‌کنیم چون اغلب PC ها از این کانکتور استفاده می‌کنند و DB9 نام دارد.



Pin	Signal	Pin	Signal
1	Data Carrier Detect	6	Data Set Ready
2	Received Data	7	Request to Send
3	Transmitted Data	8	Clear to Send
4	Data Terminal Ready	9	Ring Indicator
5	Signal Ground		

شکل ۴-۱۴ شمای یک کانکتور سریال مادگی و نام پین‌ها

در زیر مشخصات پین های کانکتور ۹ پین استاندارد RS232 را از DTE ملاحظه می کنید:

پین ۱ و ۹

این پین ها به ترتیب Carrier Detect و Indicator Ring یا CD و RI نام دارند و مربوط به مودم می شوند که اولی برای تشخیص حامل بر روی خط و دومی برای اعلان یک تماس تلفنی به مودم شماست و معمولا در طراحی های کنترلری به ندرت مورد استفاده قرار می گیرند.

پین های ۲و۳

این پین ها به ترتیب Received Data و Data Transmitted یا RD و TD نام دارند و به ترتیب جهت دریافت داده ها یک از یک DCE و ارسال داده ها از یک DTE به یک DCE به کار می روند. این نامگذاری‌ها ممکن است فریبنده باشند چون پین TD در یک DCE جهت دریافت داده‌ها به کار گرفته می‌شود. خط TD در زمان معطلی توسط DTE در حالت مارک نگه داشته می‌شود.

پین های ۷و۸

ابتدا کمی در مورد مفهوم Flow Control توضیح می‌دهیم. فرض کنید که یک DTE را به یک DCE متصل کرده‌ایم، از آنجایی که معمولا سرعت DTE (کامپیوتر) از سرعت یک DCE (مثلا یک مودم یا یک میکروکنترلر) بیشتر می‌باشد. اگر مکانیسمی جهت کنترل انتقال اطلاعات وجود نداشته باشد، DCE قادر نخواهد بود همه اطلاعات ارسالی از DTE را دریافت کند. این مکانیسم را Control Flow یعنی کنترل جریان اطلاعات می‌گویند.

پین های ۷ و ۸ به ترتیب Request to Send و Send Clear to یا RTS و CTS می‌نامند. فرض کنید که DTE می خواهد یک کاراکتر را برای DCE ارسال کند، در این حالت DTE از طریق پین RTS یک منطقی ارسال می‌کند و در صورتیکه DCE قادر به دریافت اطلاعات باشد یک سیگنال پاسخ از طریق پین CTS ارسال می‌کند و در غیر اینصورت خط در حالت صفر نگه داشته می‌شود و DTE اجازه ندارد اطلاعات را ارسال کند. به این روش Handshaking یا دست دادن نیز می‌گویند.

پین های ۴و۶

واضح است که DCE به علت سرعت کمتر برای ارسال نیاز به کسب مجوز از DTE ندارد اما در صورت لزوم پین های Ready Data Set و Ready Data Terminal به ترتیب نقشی مشابه RTS و CTS را ایفا می‌کنند (به ترتب خطوط ۴ و ۶)

پین ۵

زمین سیگنال می باشد.

۳-۵. نرم افزار میکروکنترلر برای ارتباط با رایانه

برای برقراری ارتباط بین رایانه و میکرو کنترلر،ابتدا رایانه یک رشته ۴ تایی را به عنوان رمز برقراری ارتباط به میکروکنترلر ارسال می‌کند، با ارسال این رشته وقفه دریافت اطلاعات از طریق پورت سریال در میکروکنترلر فعال می‌شود، سپس کدهای مربوط به این وقفه اجرا می شوند که در آن صحت رمز ارسالی ار رایانه بررسی می‌شود و در صورت صحت آن، میکروکنترلر وارد حالت ارتباط با رایانه می‌شود.

در حالت ارتباط با رایانه، میکروکنترلر از ابتدای حافظه MMC تا محل کنونی، اطلاعات مربوط به ID ،تاریخ و ساعت را می‌خواند و به رایانه ارسال می‌کند.

۳-۶. LCD کاراکتری

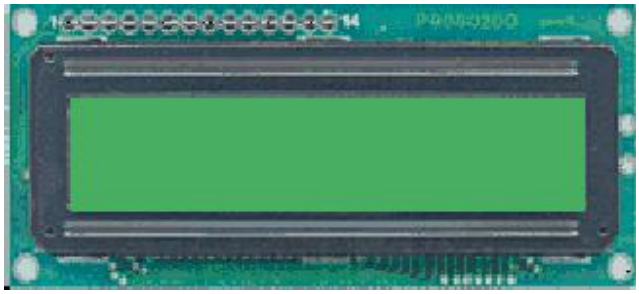
در بیشتر مدارهای میکروکنترلری ما نیاز به نمایش اطلاعات برای کاربر داریم، برای این کار راه های مختلفی وجود دارد. از جمله استفاده از چند LED، استفاده از 7segment، استفاده از LCD و ...،اما LCD ها به علت داشتن قابلیتهای بیشتر نمایش، ابعاد مختلف و برنامه ریزی آسان اغلب مورد توجه قرار می گیرند.

LCD ها شامل انواع کاراکتری و گرافیکی هستند که در اینجا نوع کاراکتری آن را معرفی و طرز استفاده آن را شرح می‌دهیم.

LCD های کاراکتری نیز خود از نظر نحوه تبادل اطلاعات به دو دسته سریال و موازی تقسیم می شوند، از مزایای نوع سریال استفاده کمتر از پین های میکروکنترلر است.

از نظر اندازه، این LCD ها دارای تنوع مختلفی هستند از جمله 16x2،32x2،40x2،16x4 و ... که منظور از این اعداد تعداد کاراکترهای قابل نمایش در یک سطر و تعداد سطرهای آن است، به عنوان مثال نوع 16x2 دارای ۲ سطر است و در هر سطر تعداد ۱۶ کاراکتر را نمایش می‌دهد.

ما در این پروژه از نوع 16x2 که بسیار پرکاربرد است استفاده کرده‌ایم که وظیفه آن نمایش اطلاعات برای کاربر و سوپروایزر است. در حالت عادی و وقتی سیستم منتظر ورود کاربر است اطلاعات مربوط به تاریخ و ساعت روی LCD نمایش داده می‌شود. اگر خطایی در سیستم رخ دهد اطلاعات مربوط به این خطا روی LCD نمایش داده می‌شود، همچنین سوپروایزر می‌تواند به کمک صفحه کلید و با وارد کردن رمز عبور به قسمت تنظیمات دستگاه وارد شود که اطلاعات مربوط به این تنظیمات روی LCD نمایش داده می‌شود.



شکل ۴-۱۵ یک نمونه LCD از نوع 16x2

این نوع LCD ها بسته به اینکه دارای نور پشت زمینه باشند یا نه دارای ۱۴ یا ۱۶ پایه هستندکه در جدول زیر این پایه ها معرفی شده اند.

جدول (۴-۳)

توضیح	I/O	سمبول	پایه
زمین منبع تغذیه	---	VSS	1
ولتاژ ۵ ولت منبع تغذیه	---	VDD	2
ولتاژ کنترل شدت نور صفحه	---	VEE	3
اگر RS=0 باشد رجیستر دستور انتخاب می‌شود اگر RS=1 باشد رجیستر داده انتخاب می‌شود	I	RS	4
R/W=0 برای نوشتن اطلاعات R/W=0 برای نوشتن اطلاعات	---	R/W	5
فعال ساز	I	E	6
بیت ۰ باس داده	I/O	D0	7
بیت ۱ باس داده	I/O	D1	8
بیت ۲ باس داده	I/O	D2	9
بیت ۳ باس داده	I/O	D3	10
بیت ۴ باس داده	I/O	D4	11
بیت ۵ باس داده	I/O	D5	12
بیت ۶ باس داده	I/O	D6	13

بیت ۷ باس داده	I/O	D7	14
آنود LED پشت زمینه LCD	---	BLA	15
کاتود LED پشت زمینه LCD	---	BLK	16

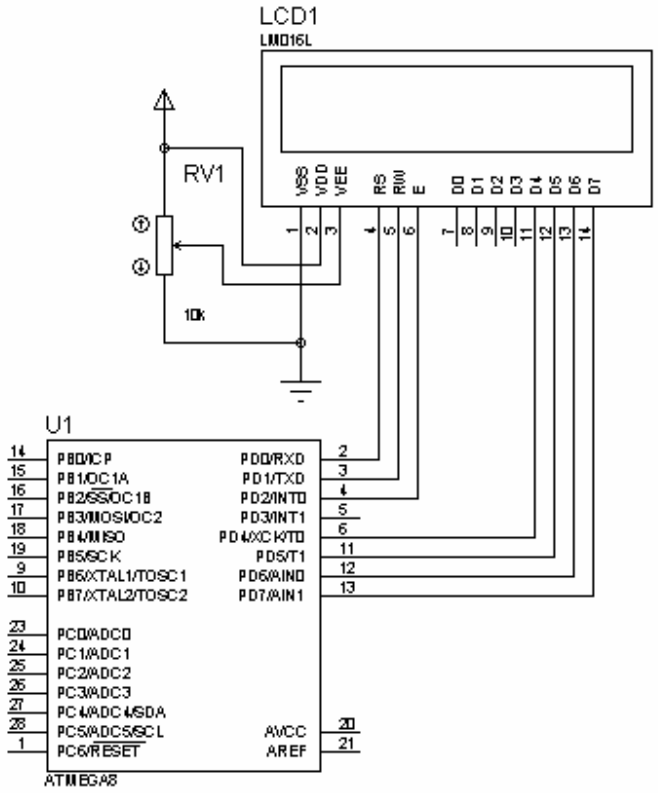
۳-۶-۱. توضیح مختصری راجع به پایه ها

- **پایه RS**: در داخل LCD دو رجیستر وجود دارد،که توسط پایه RS انتخاب می شوند. اگر RS=0 شود رجیستر دستور (Instruction Register) انتخاب می‌شود تا فرمانهای مانند پیکره بندی LCD،پاک کردن آن،جابجایی مکان نما و ... برای LCD ارسال شود. در صورتی که اگر RS=1 شود، رجیستر داده (Data Register)انتخاب می‌شود، تا کاربر بتواند اطلاعاتی را که می خواهد روی LCD بنویسد برای LCD ارسال کند.
- **پایه R/W**: به کمک این پایه کاربر مشخص می‌کند که می‌خواهد اطلاعات را روی LCD بنویسد یا از روی آن بخواند. اگر این پایه یک شود اطلاعات از روی LCD خوانده می‌شود و در صورتی که صفر شود اطلاعات روی آن نوشته می‌شود.
- **پایه E:** اگر در این پایه پالسی از یک به صفر قرار داده شود،اطلاعاتی که روی پایه های D0 تا D7 قرار دارد درون یکی از رجیسترهایی که توسط پایه RS (Register Select) مشخص می‌شود، جای می گیرد. حداقل زمانی که این پایه باید صفر باشد ۴۵۰ نانو ثانیه است.

۳-۶-۲. نحوه اتصال LCD به میکروکنترلر

با اینکه این نوع LCD های کاراکتری دارای ۸ پایه برای تبادل اطلاعات هستند (ارتباط ۸ بیتی) اما دارای قابلیت ارتباط با ۴ پایه را نیز دارند (ارتباط ۴ بیتی) که این امر باعث صرفه جویی در استفاده از پایه های میکروکنترلر می‌شود.

کتابخانه موجود برای LCD کاراکتری در نرم افزار Codevision نیز برای ارتباط ۴ بیتی نوشته شده‌است.



شکل ۴-۱۶ نحوه اتصال LCD به میکروکنترلر

البته بهتر است که پایه های آزاد LCD توسط ۴ عدد مقاومت به زمین متصل شوند.

۳-۶-۳. برنامه نویسی برای LCD

مشخص‌کردن پورتی که LCD به آن متصل است و همچنین اضافه کردن کتابخانه LCD.این قسمتها باید در ابتدای برنامه اضافه شوند.

خطی از برنامه که بین دو عبارت #asm و #endasm آمده است یک دستور اسمبلی است و مشخص کننده پورتی است که LCD به آن متصل است، به عنوان مثال در شکل زیر LCD به پورت D متصل شده‌است و عدد 0x12 مشخص کننده آدرس این پورت است.

```
// Alphanumeric LCD Module functions
#asm
   .equ __lcd_port=0x12 ;PORTD
#endasm
#include <lcd.h>
```

جدول(۴-۴)

نام پورت	آدرس پورت
PORTA	**0x1b**
PORTB	**0x18**
PORTC	**0x15**
PORTD	**0x12**

اگر بخواهیم از پورت دیگری استفاده کنیم آدرس آن را از جدول مربوط به آدرس رجیسترها در دیتاشیت میکروکنترلر مورد نظر می‌یابیم.

پیکره بندی LCD

برای پیکره بندی LCD در تابع main برنامه و قبل از عبارت While(1) این عبارت را اضافه می‌کنیم:

```
// LCD module initialization
lcd_init(16);
```

عدد ۱۶ به معنای این است که LCD دارای ۱۶ کاراکتر در هر سطر است.

توابع ارتباط با LCD در Codevision

کتابخانه نوشته شده در Codevision برای ارتباط با LCD مدلهای زیر را پشتیبانی می‌کند:

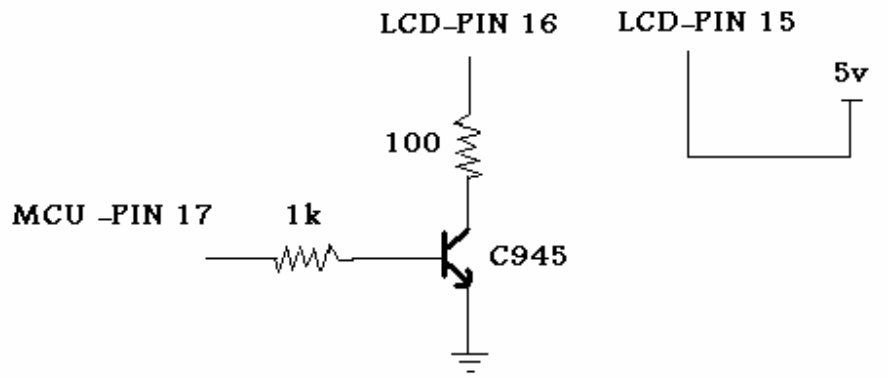
1x8, 2x12, 3x12, 1x16, 2x16, 2x20, 4x20, 2x24 , 2x40

جدول(۴-۵)

unsigned char lcd_init(unsigned char lcd_columns)
این تابع LCD را پیکره بندی می‌کند، صفحه نمایش را پاک کرده و مکان نما را در سطر صفر و ستون صفر قرار می‌دهد، مقدار lcd_columns تعداد کاراکترهایی را که LCD در یک سطر نمایش می‌دهد، مشخص می-کند. این تابع در صورتی که LCD وجود داشته باشد مقدار ۱ را بر میگرداند.
void lcd_clear(void)
این تابع صفحه نمایش را پاک کرده و مکان نما را در سطر صفر و ستون صفر قرار می‌دهد.
void lcd_gotoxy(unsigned char x, unsigned char y)
این تابع مکان نما را به ستون x و سطر y منتقل می‌کند. توجه کنید که شماره سطرها و ستون ها از صفر آغاز می‌شود.
void lcd_putchar(char c)
این تابع کاراکتر مشخص شده با متغییر c را در موقعیت فعلی مکان نما نمایش می‌دهد.
void lcd_puts(char *str)
این تابع رشته str ،که در حافظه SRAM وجود دارد را در موقعیت فعلی مکان نما نمایش می‌دهد.
void lcd_putsf(char flash *str)
این تابع رشته str ،که در حافظه FLASH وجود دارد را در موقعیت فعلی مکان نما نمایش می‌دهد.

۳-۶-۴. کنترل نور پشت زمینه LCD

در مواقعی که به LCD نیازی نیست، میکروکنترلر نور پشت زمینه آن را خاموش می‌کند تا از اتلاف توان جلوگیری کند. برای این کار میتوان از مدار شکل (۴-۱۷) استفاده کرد.

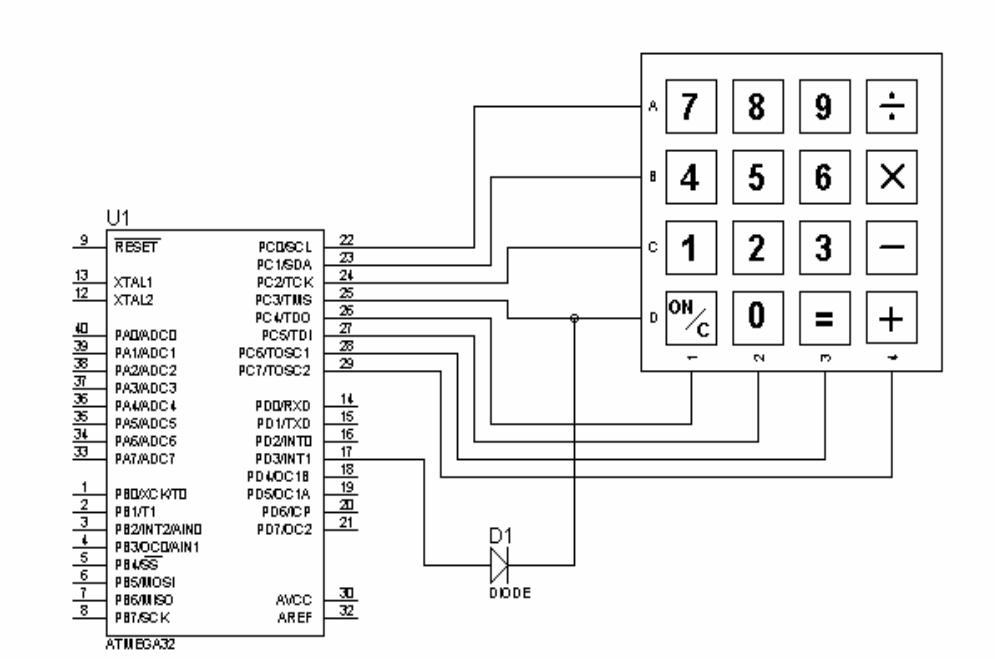


شکل ۴-۱۷ اتصال LCD به میکروکنترلر برای کنترل نور پشت زمینه

میکروکنترلر به کمک PORTD.3 (پایه ۱۷) نور پشت زمینه خاموش روشن می‌کند. برای این کار این پایه به صورت خروجی تعریف شده‌است، تا با صفر و یک کردن این پایه نور پشت زمینه LCD را کنترل شود.

۳-۷. صفحه کلید ماتریسی

یکی از ساده ترین راه های ارتباط کاربر با دستگاه های الکترونیکی صفحه کلید های ماتریسی هستند. استفاده از این صفحه کلیدها باعث صرفه جویی در پایه های میکروکنترلر می شود. نحوه اتصال صفحه کلید ماتریسی به میکروکنترلر در شکل زیر نمایش داده شده‌است.



شکل ۴-۱۸ نحوه اتصال صفحه کلید به میکروکنترلر

در یک صفحه کلید ماتریسی برای تشخیص کلید فشرده شده،میکروکنترلر تمامی ردیف ها را با صفر کردن پایه های خروجی مربوطه زمین می‌کند و سپس ستونها را می‌خواند. اگر مقدار خوانده شده از ستونها ۱۱۱۱ باشد یعنی کلید فشرده نشده است، ولی در صورتی که یکی از آنها صفر باشد، این بدین معنی است که در ستون صفر شده یک کلید فشرده شده‌است، در این موقع میکروکنترلر برای تشخیص کلید از بالاترین ردیف شروع می‌کند، به این ترتیب که میکروکنترلر تنها ردیف بالا را صفر می‌کند و مقدار ستون را می‌خواند. اگر مقدار ستون صفر نبود، به این معنی است که کلید فشار داده شده، در این ردیف قرار نداشته است. در این صورت میکروکنترلر سراغ ردیف بعدی می رود و این کار را آنقدر تکرار می‌کند تا کلید فشرده شده را بیابد.

برای اینکه میکروکنترلر به صورت پیوسته ستون ها را نخواند از وقفه خارجی استفاده شده است، برای اینکار سطر چهارم صفحه کلید ماتریسی با یک دیود به پایه وقفه خارجی میکروکنترلر متصل شده است و هرگاه کلیدی از این سطر (این سطر شامل کلید Enter است) فشرده شود، وقفه فعال می‌شود و میکروکنترلر ستونها را قرائت می‌کند تا کلید فشرده شده را بیابد.

۴. سخت‌افزار کارت

یکی از نکات مهم در طراحی کارت، انتخاب یک میکروکنترلر کم مصرف و در عین حال مناسب، از لحاظ امکانات می‌باشد؛ در طراحی یک کارت الکترونیکی که به منبع تغذیه (باتری) مجزا در خود احتیاج دارد، طول عمر باتری فاکتور مهمی محسوب می‌شود. به همین دلیل از میکروکنترلر mega8L در کارت استفاده شد[۱]. علت انتخاب mega8 نوع "L"، بازه ولتاژ عملیاتی آن است که می‌تواند از ۲,۷ تا ۵,۵ ولت باشد که به راحتی توسط یک باتری ساعتی 3v قابل تامین است.

۵. برنامه اصلی دستگاه کارت خوان

برنامه ای که درون حافظه میکروکنترلر دستگاه کارت خوان قرار داده شده‌است شامل بخش های زیر است:

- پیکره بندی دستگاه کارت خوان
- وقفه تایمر
- وقفه خارجی
- وقفه UART
- مبدل تاریخ میلادی به شمسی
- حلقه بی پایان

دستگاه کارت خوان دارای حالتهای (State) مختلفی شامل حالت انتظار برای کارت،حالت تایید کارت و حالت ارتباط با رایانه است که همیشه حالت فعلی دستگاه با متغییر rd_state مشخص می‌شود.

۵-۱. پیکره بندی دستگاه کارت خوان

۱- البته از میکروکنترلرهای سری Tiny ، Tiny45 و Tiny85 جهت استفاده در کارت مناسب تر بودند، اما در دسترس نبودند.

این قسمت از برنامه میکرو کنترلر که درون تابع reader_initialize قرار گرفته بعد از راه اندازی دستگاه کارت خوان (اتصال تغذیه به آن) اولین بخشی از برنامه است که اجرا می‌شود، این بخش قسمت های مختلف دستگاه را پیکره بندی می‌کند که این قسمت ها شامل موارد زیر است:

- پورتها
- تایمر
- وقفه خارجی
- ارتباط سریال (UART)
- ارتباط SPI
- LCD
- مولد عدد تصادفی
- پورتهای ارتباطی با دیود مادون قرمز
- صفحه کلید ماتریسی

همچنین دراین قسمت از برنامه،زیر برنامه self_test اجرا می‌شود که به تست قسمتهای مختلف مدار می پردازد و در صورتی که خطایی وجود داشته باشد،مانع از راه اندازی دستگاه می‌شود و خطای مربوطه را روی نمایشگر نمایش می‌دهد. در انتها این قسمت از برنامه به متغییر rd_state که مشخص کننده حالت دستگاه کارت خوان است،مقدار اولیه (حالت انتظار برای یافتن کارت) را اختصاص می‌دهد. سپس وقفه عمومی میکرو کنترلر فعال می‌شود تا در صورت سرریز شدن تایمر و یا لبه پایین رونده در پایه INT0 برنامه های مربوط به این دو وقفه اجرا شود.

۵-۲. وقفه تایمر

این وقفه که به کمک تایمر صفر میکروکنترلر ایجاد می‌شود، برای به وجود آوردن زمانی برابر ۱۰ میلی‌ثانیه است، که برای پروتکل ارتباطی مادون قرمز بین دستگاه کار خوان و کارت استفاده می‌شود. با قرار دادن یک دستور شرطی بر روی متغییر rd_state در ابتدای کدهای مربوط به این وقفه تنها زمانی تمام کدهای درون وقفه اجرا می شوند که دستگاه کارت خوان در حالت انتظار برای کارت و یا حالت تایید کارت باشد.

۵-۳. وقفه خارجی

این وقفه هنگامی رخ می‌دهد که لبه پایین رونده یک پالس به پایه INT0 اعمال شود.این وقفه به منظور تشخیص فشرده شدن کلیدی از صفحه کلید ماتریسی است. هنگامیکه کلید Enter بر روی صفحه کلید ماتریسی فشرده شود این وقفه اجرا می‌شود. سپس زیر برنامه password_check اجرا می‌شود که از کاربر درخواست رمز برای ورود به قسمت تنظیمات دستگاه را می‌کند. اگر رمز اشتباه وارد شود پیغامی مبنی بر اشتباه بودن رمز بر روی نمایشگر نمایش داده می‌شود و دستگاه به حالت انتظار برای کارت باز می‌گردد، اما اگر رمز به صورت صحیح وارد شود زیر برنامه setting اجرا می‌شود.

با اجرای زیر برنامه setting بر روی نمایشگر عبارات راهنمایی نمایش داده می‌شود که بر طبق آن کاربر مطلع می‌شود که کلید F1 برای ورود به قسمت تنظیمات ساعت، کلید F2 تنظیمات تاریخ و F3 خروج از قسمت تنظیمات است.

۵-۴. وقفه UART

برای اینکه میکروکنترلر همیشه آماده ارتباط با رایانه باشد،برای شروع ارتباط دیتگاه با رایانه از وقفه UART استفاده شده‌است، بدین صورت که رایانه ابتدا رمزی را به دستگاه ارسال می‌کند و در صورت صحت این رمز، دستگاه به حالت ارتباط با رایانه وارد می‌شود و اطلاعات را به رایانه ارسال می‌کند. رشته ۴ تایی رمز دستگاه در متغیر interface_pass ذخیره می‌شود.

۵-۵. تبدیل تاریخ میلادی به شمسی

تبدیل تاریخ میلادی به شمسی بسیار راحتتر از تبدیل تاریخ شمسی به میلادی است .برای نوشتن این الگوریتم به اختلاف روزهای میان اولین روز سال میلادی و اولین روز سال شمسی نیاز داریم که این اختلاف روز(در صورتی که سال کبیسه باشد یا نباشد "79"روز است.)

برای تشخیص کبیسه بودن یا نبودن سال ازروش زیر استفاده میکنیم:

اگر سال داده شده بر"۱۰۰و۴۰۰ بخشپذیر باشد" یا بر" ۱۰۰ بخشپذیر نباشد بر ۴ بخشپذیر باشد "آنگاه سال کبیسه است، درغیراینصورت سال کبیسه نیست.با توجه به کبیسه بودن یا کبیسه نبودن سال مشخص می‌کنیم که در کدامین روز سال میلادی قرار داریم. دو حالت پیش می‌آید:

- روزی که در آن قرار داریم از ۷۹ بیشتر است. به این معنی است که در ماههای بعد از فروردین قرار داریم. حال باید مشخص کنیم که در ۶ ماه اول سال شمسی قرار داریم یا در ۶ ماه دوم سال قرار داریم، برای اینکار ابتدا ۷۹ روز از تعداد روزها کم می‌کنیم تا در اول فروردین قرار بگیریم حال اگر تعداد روزها از "۱۸۶"(۳۱*۶) کمتر باشد یعنی در ۶ ماه اول سال شمسی قرار داریم در غیر اینصورت در ۶ ماه دوم قرار داریم.

اگر در ۶ ماه اول سال قرار گرفته باشیم :تعداد روزها را بر "۳۱" تقسیم می‌کنیم (۶ ماه اول در سال شمسی ۳۱ روزه است). اگر باقیمانده این تقسیم صفر شد خارج قسمت تقسیم برابر با ماه شمسی می‌شود و روز شمسی برابر با ۳۱ می‌شود. اگر باقیمانده صفر نشود ماه شمسی برابر با خارج قسمت باضافه یک می‌شود و روز شمسی همان باقیمانده است.

اگر در ۶ ماه دوم سال قرار گرفته باشیم "۱۸۶" روز از تعداد روزها کم می‌کنیم و آن را بر "۳۰" تقسیم می‌کنیم. اگرباقیمانده این تقسیم صفر شد خارج قسمت تقسیم باضافه"۶" برابر با ماه شمسی می‌شود و روز شمسی برابر با ۳۰ می‌شود. اگر باقیمانده صفر نشود ماه شمسی برابر با خارج

قسمت باضافه"۷" می‌شود و روز شمسی همان باقیمانده است. سال شمسی از تفاضل سال میلادی با"۶۲۱" بدست می‌آید.

روزی که در آن قرار داریم کمتر از"۷۹"است که این به این معنی است که درروزهایی بین اولین روزسال میلادی تا اولین روز شمسی (ماههای دی ،بهمن و اسفند)قرار داریم. اختلاف روز بین اولین روز سال میلادی داده شده و اولین روز دی ماه در سال شمسی را در نظر می گیریم که این اختلاف برای سال کبیسه"۱۱" و برای غیر کبیسه ۱۰ است. دقت کنید که در این الگوریتم برای مشخص کردن این اختلاف باید سال قبل از سال داده شده را در نظر بگیریم زیرا سال قبل بر روی اولین روز سال میلادی تاثیر می گذارد. اختلاف روز با تعداد روز محاسبه شده جمع می‌کنیم، آن را بر"۳۰" تقسیم می‌کنیم (۳ ماه آخر سال شمسی ۳۰ روزه است).

اگر باقیمانده این تقسیم صفر شود خارج قسمت تقسیم باضافه"۹" برابر با ماه شمسی می‌شود و روز شمسی برابر با"۳۰" می‌شود.

اگر باقیمانده صفر نشود ماه شمسی برابر با خارج قسمت باضافه"۱۰"می‌شود و روز شمسی همان باقیمانده است. در این حالت سال شمسی از تفاضل سال میلادی با"۶۲۲" بدست می‌آید (زیرا در سال قبل قرار داریم).

۵-۶. حلقه بی پایان

این قسمت از برنامه در واقع اصلی ترین قسمت برنامه میکروکنترلر می باشد زیرا پس از راه اندازی میکروکنترلر این قسمت از برنامه است دائما اجرا می‌شود. با اجرای این حلقه ابتدا زیر برنامه main_display اجرا می‌شود که بر روی نمایشگر ساعت و تاریخ دیتگاه را نمایش می‌دهد، همچینین پیغامی مبنی بر آماده بودن دستگاه برای تایید کارت بر روی نمایشگر نمایش داده می‌شود.

بقیه کدهای مربوط به این حلقه بی پایان درون یک عبارت switch قرار گرفته است که بر روی متغییر rd_state که نشان‌دهنده حالت دستگاه است عمل switch انجام می‌شود.

چون دستگاه در ابتدا و به صورت پیش فرض در حالت انتظار برای کارت است با اجرای switch به قسمت مربوط به انتظار برای کارت وارد می‌شود. در ابتدای کدهای مربوط به این قسمت یک دستور شرطی به صورت زیر:

if(ledcomm_client_present())

قرار دارد و در صورتی کدهای درون این دستور شرطی اجرا می‌شود که کارتی در اطراف دستگاه وجود داشته باشد. سپس عباراتی مبنی بر یافتن یک کارت بر روی نمایشگر نمایش داده می‌شود. کلیدی از صفحه کلید ماتریسی است رونده لبه پایین رونده دستگاه کارت خوان در حالت انتظار برای کارت و یا ارتباط با کارت باشد.

در ادامه با کمک الگوریتم تولید عدد تصادفی یک عدد تصادفی ایجاد می‌شود و پس از رمزگذاری توسط الگوریتم رمز TEA توسط ارتباط مادون قرمز به کارت ارسال می‌شود و دستگاه منظتر دریافت پاسخ از سوی کارت می‌شود. در این حالت LED قرمز خاموش و LED زرد روشن می‌شود. در انتهای این قسمت دستگاه به حالت تایید کارت وارد می‌شود.

در این قسمت دستگاه پس از دریافت اطلاعات از کارت آن را به کمک الگوریتم TEA رمز نگاری می‌کند و در صورت صحت اطلاعات کارت را مجاز محسوب می‌کند و پیغامی را مبنی بر آن بر روی نمایشگر نمایش می‌دهد، سپس اطلاعات مربوط به کارت (ID)، تاریخ و ساعت کنونی دستگاه بر روی MMC ذخیره می‌شود. در این حالت LED زرد خاموش و LED سبز روشن می‌شود.

اما اگر اطلاعاتی که از کارت باز می گردد صحیح نباشد پیغامی مبنی بر آن بر روی نمایشگر نمایش داده می-شود و وضعیت LED ها به همان صورت قبلی می‌ماند. در انتها حالت دستگاه به حالت انتظار برای کارت تغییر می-کند.

آخرین مورد درون switch مربوط به حالت ارتباط با رایانه است. البته دستگاه تنها هنگامی که رمز ارسالی از رایانه صحیح باشد وارد این حالت می‌شود که دستورات مربوط به آن درون وقفه خارجی میکروکنترلر قرار دارد. در

ابتدای کدهای مربوط به ارتباط با رایانه وقفه عمومی میکروکنترلر غیر فعال می‌شود تا در هنگام ارتباط با رایانه کد هیچ وقفه ای اجرا نگردد.سپس بر روی نمایشگر عبارت ارتباط با رایانه نمایش داده می‌شود.

در ادامه میکروکنترلر، از سکتور صفر MMC تا سکتوری فعلی اطلاعات را می‌خواند و از طریق پورت سریال به رایانه منتقل می‌کند. در پایان پیغامی مبنی بر اتمام ارسال اطلاعات به رایانه فرستاده می‌شود، همچنین بر روی نمایشگر نیز این پیغام نمایش داده می‌شود و میکرو کنترلر به حالت انتظار برای کارت باز می‌گردد.

۶. برنامه اصلی کارت

برنامه ای که درون حافظه میکروکنترلر کارت قرار داده شده‌است شامل بخش های زیر است:

- پیکره بندی کارت
- وقفه تایمر
- وقفه خارجی
- حلقه بی پایان

دستگاه کارت خوان دارای حالتهای (State) مختلفی شامل حالت خواب،حالت انتظار برای پالس سنکرون سازی و حالت دریافت اطلاعات از کارت خوان و حالت پاسخ دادن به کارت خوان است که همیشه حالت فعلی دستگاه با متغییر مشخص می‌شود.

۶–۱. پیکره بندی کارت

این قسمت از برنامه میکرو کنترلر که درون تابع key_initialize قرار گرفته بعد از راه اندازی کارت (اتصال تغذیه به آن) اولین بخشی از برنامه است که اجرا می‌شود، این بخش قسمت‌های مختلف دستگاه را پیکره بندی می‌کند که این قسمت ها شامل موارد زیر است:

- پورتها
- تایمر
- وقفه خارجی
- ارتباط سریال (UART) برای عیب یابی
- پورتهای ارتباطی با دیود مادون قرمز
- حالت خواب (sleep)

در انتها این قسمت از برنامه به متغییر s که مشخص کننده حالت کارت است،مقدار اولیه (حالت خواب) را اختصاص می‌دهد. سپس وقفه عمومی میکرو کنترلر فعال می‌شود تا در صورت سرریز شدن تایمر و یا لبه پایین رونده در پایه INT1 برنامه های مربوط به این دو وقفه اجرا شود.

۶–۲. وقفه تایمر

این وقفه که به کمک تایمر ۲ میکروکنترلر ایجاد می‌شود، برای به وجود آوردن زمانی برابر ۱۰ میلی‌ثانیه است،که برای پروتکل ارتباطی مادون قرمز بین دستگاه کار خوان و کارت استفاده می‌شود.

۶-۳. وقفه خارجی

این وقفه هنگامی رخ می‌دهد که لبه پایین رونده یک پالس به پایه INT1 اعمال شود.این وقفه به منظور تشخیص فشرده شدن کلید روی کارت است. هنگامیکه این کلید فشرده می‌شود کارت از حالت خواب خارج می‌شود و شروع به فرستادن پالس از طریق دیود مادون قرمز می‌کند تا با دستگاه کارت خوان ارتباط برقرار کند (کاربر این کلید را هنگامی‌که کارت را جلوی دستگاه کارت خوان قرار داده‌است فشار می‌دهد).

۶-۴. حلقه بی پایان

این قسمت از برنامه در واقع اصلی ترین قسمت برنامه میکروکنترلر می باشد زیرا پس از راه اندازی میکروکنترلر این قسمت از برنامه است که دائما اجرا می‌شود. کدهای مربوط به این حلقه بی پایان درون یک عبارت switch قرار گرفته است که بر روی متغییر s که نشان‌دهنده حالت کارت است عمل switch انجام می‌شود.

چون دستگاه در ابتدا و به صورت پیش فرض در حالت خواب است با اجرای switch به قسمت مربوط به خواب وارد می‌شود. در این حالت پورت مربوط به دیود مادون قرمز به حالت بالاکش (pull-up) در می‌آید و میکروکنترلر به حالت خواب (sleep) می‌رود. در این حالت میکروکنترلر توان کمتری را از باتری دریافت می‌کند در نتیجه طول عمر باتری بیشتر می‌شود.

به محض اینکه کلید روی کارت توسط کاربر فشرده شود میکرو کنترلر از حالت خواب خارج می‌شود و پس از پیکره بندی دوباره دیود مادون قرمز وارد حالت انتظار برای پالس سنکرون سازی می‌شود. پس از اینکه کارت با کارت خوان سنکرون گردید وارد حالت دریافت اطلاعات از کارت خوان می‌شود. در این حالت کارت اطلاعات را از دستگاه کارت خوان دریافت می‌کند و وارد حالت پاسخ به دستگاه کارت خوان می‌شود. در این حالت کارت اطلاعاتی را که در حات قبلی از کارت خوان دیافت کرده‌است را به کمک الگوریتم TEA رمز نگاری می‌کند. سپس پاسخ مناسب را به همراه اطلاعات خود (ID) به کارت خوان ارسال می‌کند و در انتها به حالت خواب می‌رود.

۷. نرم افزار گزارش اطلاعات ورود و خروج

یکی از قابلیت های این سیستم امکان گزارش گیری از اطلاعات ثبت شده در کارت حافظه دستگاه کارت خوان می باشد. بدین منظور نرم افزاری طراحی شده‌است که پس از برقراری ارتباط فیزیکی بین پورت سریال کامپیوتر و دستگاه کارت خوان، قادر به دریافت اطلاعات ورود و خروج خواهدبود. این نرم – افزار در محیط دات نت ۲ طراحی و به زبان سی شارپ نوشته شده‌است. در ادامه، این نرم افزار که SecureReporter نام دارد از جهات مختلف بررسی شده‌است:

۷-۱-۱. مشخصات فنی

از دید فنی و تکنیکی، نرم افزار SecureReporter فاقد پیچیدگی درخور توجهی است. همانطور که در بالا گفته شد، این نرم افزار درمحیط DotNet 2.0 طراحی شده‌است و به زبان شی گرای سی شارپ می‌باشد. گزارش های دریافتی از کارت خوان، در پایگاه داده ای ذخیره می‌شود. این پایگاه داده ها از نوع Microsoft Access می باشد. پایگاه داده تنها دارای یک جدول است و رابطه ای نمی باشد. جدول از سه ستون ID, Date, Time تشکیل شده که به ترتیب (از چپ به راست) ID کارت (فرد) ، تاریخ ورود و خروج فرد و ساعت ورود و خروج را در خود نگه می‌دارند. هر رکورد به محض دریافت از پورت سریال در پایگاه داده ثبت می‌شود. به منظور نمایش گزارش از داده های موجود در پایگاه داده، از مجموعه CrystalReports (از ابزار های محیط دات نت) استفاده شده‌است. این نرم افزار دارای امکانات ویژه جهت طراحی و تهیه گزارش می باشد. این ابزار مانند سایر ابزار های محیط دات نت، دارای یک سری قالب های از پیش تعیین شده و کامپوننت های مختلف می باشد که به کمک آنها دیگر نیازی به انجام کارهای تکراری و وقت گیر نیست و به برنامه نویس این امکان را می‌دهد که بیشتر بر روی اهداف اصلی برنامه متمرکز شود. همچنین استفاده از اینگونه ابزار ها باعث سازگاری بیشتر برنامه با استاندارد های رایج می‌شود. در محیط دات نت ۲، کلاس پورت سریال نیز افزوده شده که کار با پورت سریال کامپیوتر را راحت تر می‌کند. در نسخه های قبلی دات نت برای کار با پورت سریال، نیاز به استفاده از فایل mscomm.dll و متغیرها و توابع موجود در آن بود که پیچیدگی هایی را موجب می‌شد.

۷-۱-۲. مشخصات کاربری

برنامه SecureLED ظاهری ساده دارد و کارکردن با آن بسیار راحت است. برای دریافت اطلاعات از پورت سریال کافیست از منوی فایل، گزینه Get Report را انتخاب کرده و در پنجره ظاهر شده رمز ارتباط را وارد کنید. پس از فشردن دکمه OK، برنامه آماده دریافت داده ها می‌شود.

از قابلیت های این برنامه می توان به موارد زیر اشاره کرد:

- تغییر تنظیمات پورت سریال (نام پورت، سرعت ارتباط و ...)
- امکان گزارش گیری از داده های موجود در بانک اطلاعاتی بر اساس ID، تاریخ، ساعت و به صورت تلفیقی
- تبدیل گزارش به فرمت‌های مختلف (PDF, Doc, ...) و امکان چاپ آنها
- جستجوی یک داده خاص در گزارش
- حذف داده‌ها از بانک اطلاعاتی برنامه (مثلاً داده های قدیمی)

۸. نتایج و جمع‌بندی

۸-۱. سرعت

سرعت انتقال اطلاعات در این سیستم ۲۵۰ بیت بر ثانیه است. البته می‌توان با کم کردن دقت به سرعت بیشتری دست پیدا کرد. اما در این پروژه با توجه به حجم داده مبادله شونده، سرعت فوق بسیار مناسب بوده و نیاز به کاستن دقت نیست.

۸–۲. دقت

دقت به دست آمده بر حسب موفقیت در تبادل صحیح پیام‌ها ۹۰ درصد است که برای هر بیت دقتی معادل ۹۹ درصد خواهد بود.

۸–۳. امنیت

به لحاظ امنیت فیزیکی باید توجه کرد که بدنه طوری طراحی شود که دسترسی به اجزای درونی کارت‌خوان امکان‌پذیر نباشد. از نظر نرم‌افزاری نیز الگوریتم رمزنگاری مورداستفاده (TEA) بسیار امن می‌باشد. همچنین به دلیل رد و بدل نشدن کلید مشترک، احتمال سرقت آن بسیار پایین و تقریباً غیر ممکن است. با تنظیم LOCK BIT های میکروکنترلر می‌توان مانع هر گونه خواندن اطلاعات آن شد.

۸–۴. تداخل با نور محیط

این سیستم با امواج موجود در محیط هیچ مشکلی نداشته ودر شرایط مختلف تست شده و عملکرد صحیحی از خود نشان داده است.

۸–۵. سهولت استفاده

رابط کاربری این سیستم بسیار ساده است و فقط یک دکمه است که باید توسط کاربر فشرده شود. تنها مورد دردسرساز قرار دادن کارت در مقابل کارت‌خوان و در محل مناسب است. البته این مشکل در صورت طراحی مناسب بدنه تا حد زیادی برطرف می‌شود.

با توجه به نتایج به‌دست‌آمده مشاهده می‌شود که این سیستم می‌تواند جایگزین مناسبی برای تکنولوژی‌های مشابه باشد. همچنین هزینه تولید این نوع سیستم امنیتی در مقایسه با سایر تکنولوژی‌ها از جمله RFID و یا کارت‌های مغناطیسی بسیار پایین است. ضمن اینکه به نسبت آنها از امنیت بیشتری برخوردار می‌باشد.

۹. مراجع

[Sep99] Sepp A. Virtanen , “On Communications Protocols and their Characteristics Relevant to Designing Protocol Processing Hardware”, 1999

[Nia01] Nizamettin AYDIN, “Communication Protocols for IDEAS”, 2001

[Dav] David A. Johnson, “Optical Through-the-Air Communications Handbook”,

http://www.resurgentsoftware.com/perfect led light.html

http://www.nongnu.org/avr-libc/user-manual/modules.html

http://www.atmel.com/dyn resources/prod_documents/doc2503.pdf

http://www.avrfreaks.net/

http://winavr.scienceprog.com/

http://hackaday.com/

http://www.atmel.com/dyn resources/prod_documents/doc2543.pdf

http://www-s.ti.com/sc/ds/tps7250.pdf

http://www.cl.cam.ac.uk/ftp/users/djw3/ tea.psMERL

http://www.merl.com/publications/TR2003-035

http://www.circuitcellar.com/avr2004/DA3616.html

http://msdn.microsoft.com/

http://office.microsoft.com/enus/access/FX100646921033.aspx?CTT=96&Origin=CL10057004103 3

http://www.codeproject.com/csharp/SerialCommunication.asp

http://www.imagineeringezine.com/files/air-bk2.html

پیوست ۱– الگوریتم TEA

Tiny Encryption Algorithm، یک الگوریتم رمزنگاری بلوکی است که ویژگی متمایزکننده آن از سایرین، سادگی الگوریتم و سهولت پیاده سازی آن می‌باشد. این الگوریتم درسال ۱۹۹۴، توسط دیوید ویلر و راجر نیدهم ابداع شد.

این الگوریتم، فاقد هرگونه حق مالکیت انحصاری است و هرکس می‌تواند از آن به طور رایگان استفاده کند (البته حقوق مالکیت انحصاری،حتی در صورت وجود هم در کشور ما رعایت نمی‌شوند).

مشخصات الگوریتم

الگوریتم TEA، بلوک های داده ۶۴ بیتی را با استفاده از یک کلید ۱۲۸ بیتی رمز می‌کند. این الگوریتم دارای ساختاری تکراریست که معمولا" در ۶۴ مرحله انجام می‌شود. جدول کلید مورد استفاده در این الگوریتم بسیار ساده است و همه بخشهای کلید را به یک طریق در هر مرحله ترکیب می‌کند. به منظور جلوگیری از حملاتی که بر مبنای تشابه مراحل صورت می گیرد، مضارب مختلفی از "ثابت جادویی"، در هر مرحله به کار گرفته می‌شود. ثابت جادویی برابر است با ۲۶۵۴۴۳۵۷۶۹ که از فرمول Φ ⁄ (2^{32}) بدست می‌آید و Φ نیزدر ریاضیات، "نسبت طلایی" است که مقدار آن ...1.6180339887498948482 است.

اندازه خیلی کوچک الگوریتم TEA

آن را به عنوان یک گزینه مناسب در سیستم‌هایی که محدودیت زیادی در توان پردازشی یا حافظه RAM دارند، مطرح ساخته است. قطعه کد زیر که به زبان C است نحوه رمزنگاری و رمزگشایی در این الگوریتم را نشان می‌دهد:

```
void encrypt(unsigned long* v, unsigned long* k) {
     unsigned long v0=v[0], v1=v[1], sum=0, i;           /* set up */
     unsigned long delta=0x9e3779b9;                     /* a key schedule
constant */
     unsigned long k0=k[0], k1=k[1], k2=k[2], k3=k[3];   /* cache key */
     for (i=0; i < 32; i++) {                            /* basic cycle start */
         sum += delta;
```

```
        v0 += ((v1<<4) + k0) ^ (v1 + sum) ^ ((v1>>5) + k1);
        v1 += ((v0<<4) + k2) ^ (v0 + sum) ^ ((v0>>5) + k3);   /* end cycle */
    }
    v[0]=v0; v[1]=v1;
}
--------------------------------------------------------------------------
void decrypt(unsigned long* v, unsigned long* k) {
    unsigned long v0=v[0], v1=v[1], sum=0xC6EF3720, i;  /* set up */
    unsigned long delta=0x9e3779b9;                     /* a key schedule
constant */
    unsigned long k0=k[0], k1=k[1], k2=k[2], k3=k[3];   /* cache key */
    for(i=0; i<32; i++) {                               /* basic cycle start */
        v1 -= ((v0<<4) + k2) ^ (v0 + sum) ^ ((v0>>5) + k3);
        v0 -= ((v1<<4) + k0) ^ (v1 + sum) ^ ((v1>>5) + k1);
        sum -= delta;                                   /* end cycle */
    }
    v[0]=v0; v[1]=v1;
}
--------------------------------------------------------------------------
```

V داده ایست که باید رمز شود، K کلید رمز و delta ثابت جادویی است.