

سلام علیکم

hamid

کارت های هوشمند (SMARTCARD)



اتصال کارت تلفن به AVR و خواندن محتویات آن

تهیه کننده :

کاوه کیانمجد

kiyanmajd@gmail.com

kiyanmajd@iseee.ir



hamid

فهرست

۱	 مقدمه
۲	 فصل اول: آشنایی با Smartcard ها
۳	 ۱.۱ کارت هوشمند چیست؟
۳	 ۲.۱ تاریخچه کارت هوشمند
۴	 ۳.۱ بررسی ساختار
۴	 ۴.۱ انواع و دسته‌بندی کارت‌های هوشمند
۴	 ۱.۴.۱ دسته‌بندی بر اساس نحوه ارتباط با کارت خوان
۴	 ۱.۱.۴.۱ - کارت هوشمند تماسی (Smart Card Contact)
۵	 ۲.۱.۴.۱ - کارت هوشمند غیرتماسی (Contactless Smart Card)
۵	 ۳.۱.۴.۱ - کارت هوشمند ترکیبی (Dual-Interface Smart Card)
۶	 ۲.۴.۱ دسته‌بندی بر اساس نوع تراشه به کار رفته در کارت
۶	 ۱.۲.۴.۱ - کارت‌های دارای حافظه مدار مجتمع (Integrate Circuit (IC) Memory Cards)
۶	 ۲.۲.۴.۱ - کارت‌های دارای ریزپردازنده مدار مجتمع (Integrate Circuit (IC) Microprocessor Cards)
۸	 ۳.۲.۴.۱ - کارت‌های دارای حافظه نوری (Optical Memory Cards)
۸	 ۵.۱ برخی از کاربرد های مالی کارت های هوشمند :
۹	 ۱.۵.۱ Debit Card
۹	 ۲.۵.۱ Credit Card
۹	 ۳.۵.۱ Prepaid Card یا کارت های پیش پرداخت :
۱۰	 ۴.۵.۱ کیف پول الکترونیکی
۱۰	 ۵.۵.۱ کنسرسیوم ویزا و مستر (master card) :
۱۲	 فصل دوم: ارتباط میکرو با کارت تلفن

۱۳	۱.۲ مقدمه
۱۳	۲.۲ مشخصات فیزیکی کارت تلفن
۱۴	۳.۲ مشخصات الکترونیکی کارت تلفن
۱۵	۴.۲ حافظه کارت تلفن
۱۵	۵.۲ پروتکل ارتباطی کارت تلفن
۱۶	۱.۲.۵ ارتباط با کارت
۱۸	۶.۲ سخت افزار مورد نیاز برای ارتباط
۱۸	۱.۶.۲ شماتیک
۱۹	۲.۶.۲ سوکت اسمارت کارت
۱۹	۳.۶.۲ اتصالات پایه ها بر روی سوکت
۱۹	۴.۶.۲ نحوه استفاده از سوکت
۲۴	۷.۲ ارتباط با میکرو و برنامه نویسی
۲۴	۱.۷.۲ اتصالات سخت افزاری مورد نیاز
۲۴	۱.۱.۷.۲ سخت افزار با نمایشگر GLCD
۲۵	۲.۱.۷.۲ سخت افزار با نمایشگر LCD کاراکتری
۲۷	۲.۷.۲ برنامه نویسی
۲۷	۱.۲.۷.۲ برنامه نویسی با codevision
۳۹	۲.۲.۷.۲ برنامه نویسی با bascom
۴۹	۳.۷.۳ محاسبه مبلغ کارت تلفن

مقدمه

کارت هوشمند معمولاً کارتی از جنس PVC با ابعادی در حدود ۵/۵ در ۸/۵ سانتی‌متر است که بر روی آن یا در بین لایه‌های آن، تراشه‌های حافظه و ریزپردازنده برای ذخیره‌سازی داده‌ها و پردازش آنها قرار داده شده است. یک کارت هوشمند کامپیوتر کوچکی است که بر روی یک کارت پلاستیکی نصب شده است. قرار دادن یک تراشه در کارت به جای نوار مغناطیسی، آن را تبدیل به یک کارت هوشمند با کاربردهای گوناگون می‌نماید. این کارت‌ها به دلیل دارا بودن تراشه، قابلیت کنترل عملکرد را داشته و علاوه بر نگهداری اطلاعات شخصی و تجاری کاربر، امکان پردازش را نیز فراهم می‌نماید.



فصل اول:

آشنایی با Smartcard

۱.۱ کارت هوشمند چیست؟

کارت هوشمند کارتی پلاستیکی با ابعاد کارت‌های اعتباری (حدود ۵/۵ در ۸/۵ سانتی‌متر) است که بر روی آن یا در بین لایه‌های آن، تراشه‌های حافظه و ریزپردازنده برای ذخیره‌سازی داده‌ها و پردازش آنها قرار داده شده است. یک کارت هوشمند کامپیوتر کوچکی است که بر روی یک کارت پلاستیکی نصب شده است. این قبیل کارت‌ها به راحتی در جیب جای می‌گیرند و در کاربردهای مختلف مورد استفاده قرار می‌گیرند.



۲.۱ تاریخچه کارت هوشمند

اختراع کارت هوشمند را برای اولین بار فردی فرانسوی با نام رولاند مورنو در سال ۱۹۷۴ به ثبت رساند. از آن زمان به بعد، شرکت‌هایی نظیر Bull، Honeywell، Motorola در این زمینه به فعالیت پرداختند و در نتیجه فعالیت‌های آنها، در سال ۱۹۷۹ اولین کارت هوشمند ریزپردازنده‌ای ساخته شد. اولین استاندارد برای کارت هوشمند در سال ۱۹۸۶ و با عنوان ISO 789116/1 مطرح شد.

استفاده از کارت هوشمند در سطح ملی برای نخستین بار در فرانسه و در سال ۱۹۸۶ انجام گرفت. در این سال، شرکت مخابرات فرانسه برای اولین بار، به جای سکه در تلفن‌های عمومی از کارت هوشمند استفاده کرد که این اقدام سبب رفع بسیاری از مشکلات استفاده از تلفن‌های عمومی، سوءاستفاده‌ها و خرابکاری‌ها شد. پس از آن، از اوایل دهه ۹۰ میلادی، استفاده از کارت‌های هوشمند در کشورهای مختلف رواج پیدا کرد و به تدریج کاربردهای جدیدی برای آن پیدا شد.

۳.۱ بررسی ساختار

کارت هوشمند کارتی است که از یک ریزپردازنده و چیپ حافظه و یا فقط چیپ حافظه (بدون منطق برنامه‌پذیر) تشکیل شده است. کارت دارای ریزپردازنده می‌تواند اطلاعات روی کارت را اضافه، تغییر، حذف و مدیریت نماید، درحالی‌که کارت فقط دارای حافظه (مانند کارت‌های اعتباری تلفن)، می‌تواند فقط یک عملیات از پیش تعریف شده را قبول کند.

کارت‌های هوشمند برخلاف کارت‌های نوار مغناطیسی، می‌توانند کلیه توابع عملیاتی و اطلاعات مربوطه را در خود داشته باشند، بنابراین در زمان انجام تراکنش نیاز به ارتباط با بانک اطلاعاتی نخواهد داشت.

۴.۱ انواع و دسته‌بندی کارت‌های هوشمند

کارت‌های هوشمند را بر حسب موارد مختلفی دسته‌بندی می‌کنند؛ در ادامه به دو مورد آن می‌پردازیم:

در حال حاضر سه گروه (بر اساس نوع تراشه بکار رفته در آن، حافظه و ریزپردازنده) و سه گروه از لحاظ نحوه ارتباط با کارت خوانها موجود می‌باشند که از کارت‌های هوشمند در کاربردهای مختلف در دنیا و به صورت گسترده مورد استفاده قرار می‌گیرند:

۱.۴.۱ دسته‌بندی بر اساس نحوه ارتباط با کارت‌خوان

بر اساس این دسته‌بندی، کارت‌های هوشمند به سه گروه تقسیم می‌شوند:

۱.۴.۱-۱ کارت هوشمند تماسی (Smart Card Contact)

برای استفاده از این قبیل کارت‌ها، باید اتصال فیزیکی بین کارت و دستگاه کارت‌خوان برقرار گردد. داده‌های موجود بر روی کارت به صورت سریال به کارت‌خوان ارسال می‌شود و پس از پردازش، اطلاعات جدید از طریق همان پورت به روی کارت منتقل می‌شود. به عنوان نمونه، کارت‌های تلفن عمومی جزو این دسته محسوب می‌شوند. مشکل اصلی این قبیل کارت‌ها، خراب شدن کنتاکت‌های فلزی (محل‌های تماس) بر اثر عوامل خارجی نظیر ضربه و شرایط فیزیکی محیط است.

در شکل ۲ قسمت‌های موجود در کنتاکت‌های فلزی این نوع کارت به تصویر کشیده شده است.



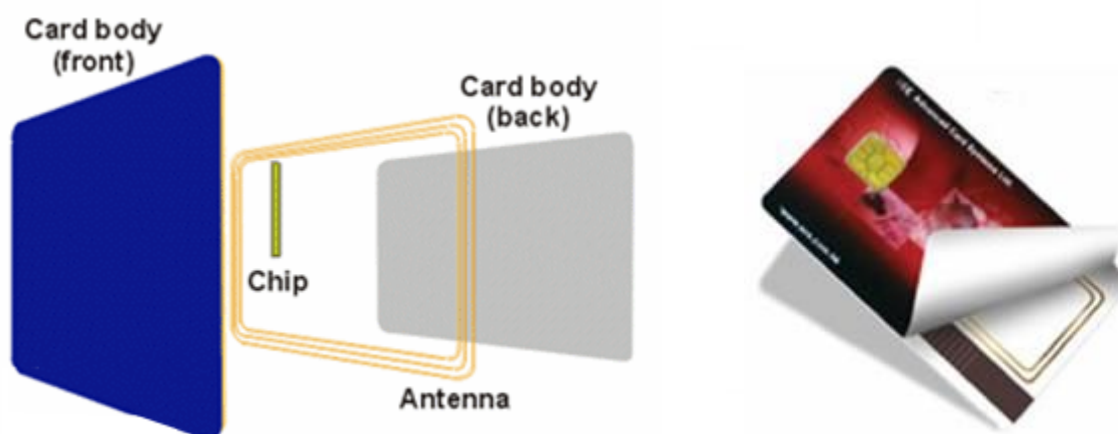
شکل ۱- شمایی از کارت هوشمند تماسی



شکل ۲ - قسمت‌های مختلف کنتاکت‌های کارت هوشمند تماسی

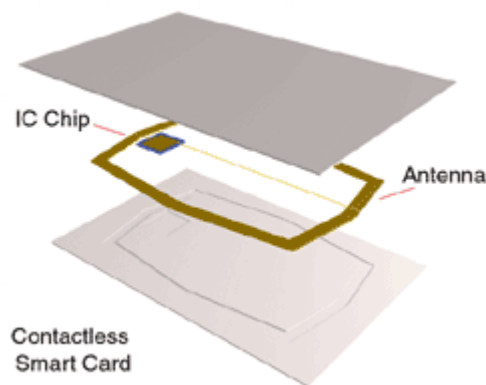
۲.۱.۴.۱- کارت هوشمند غیر تماسی (Contactless Smart Card)

در این نوع کارت هوشمند، ارتباط بین کارت و کارت‌خوان به صورت فیزیکی برقرار نمی‌شود؛ بلکه از طریق میدان‌های الکترومغناطیسی و یا امواج RF صورت می‌گیرد. برای برقراری ارتباط، آنتن مخصوصی بین تراشه‌های کارت قرار داده شده است که در فاصله‌های کم، تا حدود ۵۰ سانتیمتر، می‌تواند ارتباط ایجاد کند. کاربرد اصلی این قبیل کارت‌ها در مواردی است که عملیات مورد نظر باید سریع انجام گیرد، به عنوان نمونه می‌توان به کارت‌های مترو اشاره کرد. مزیت اصلی این قبیل کارت‌ها علاوه بر سهولت استفاده، عمر طولانی‌تر و ضریب ایمنی بالاتر آن است؛ زیرا در این نوع کارت، تراشه به همراه آنتن در میان لایه‌های تشکیل‌دهنده کارت قرار می‌گیرد.



۳.۱.۴.۱- کارت هوشمند ترکیبی (Dual-Interface Smart Card)

این نوع کارت ترکیبی از کارت‌های هوشمند تماسی و غیر تماسی است که با هر دو نوع دستگاه‌های کارت‌خوان سازگار است. از این نوع کارت‌ها برای ساخت کارت‌های چندمنظوره استفاده می‌شود.

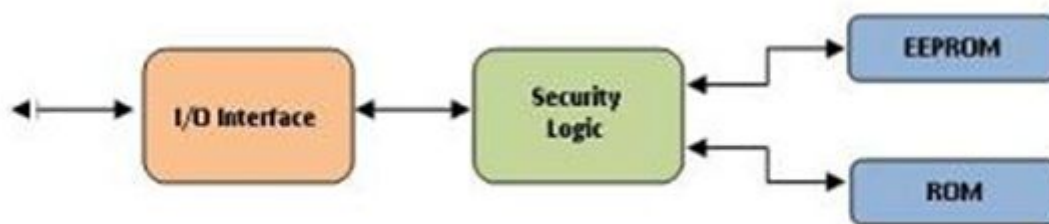


۲.۴.۱ دسته‌بندی بر اساس نوع تراشه به کار رفته در کارت

۱.۲.۴.۱-کارت‌های دارای حافظه مدار مجتمع (Integrate Circuit (IC Memory Cards)

کارت‌های حافظه مدار مجتمع می‌توانند ۱ تا ۴ کیلو بایت از داده را در خود نگهدارند، ولی هیچ پردازنده‌ای روی کارت برای عملیات روی داده ندارند. بنابراین این نوع کارت‌ها برای انجام پردازش وابسته به کارت‌خوان (که همچنین دستگاه گیرنده کارت نیز نامیده می‌شود) می‌باشند و برای کاربردهایی که در آن کارت یک عملیات ثابتی را انجام می‌دهد مناسب می‌باشند.

این نوع کارت شامل واحدهای حافظه است که توسط یک سیستم امنیتی سخت‌افزاری محافظت می‌شود. در واحد حافظه ROM اطلاعات غیرقابل‌تغییر، نظیر شماره کارت و شماره دارنده کارت ذخیره می‌شود. از واحد حافظه EEPROM نیز برای نگهداری اطلاعاتی در طول زمان یا بر اساس نیاز کاربر تغییر می‌کنند، استفاده می‌شود، به عنوان مثال اطلاعات مربوط به اعتبار باقیمانده در کارت. از جمله کاربردهای این نوع کارت‌ها می‌توان به کارت تلفن همگانی، سیستم کنترل و شناسایی و مواردی از این قبیل اشاره کرد.



۲.۲.۴.۱-کارت‌های دارای ریزپردازنده مدار مجتمع (Integrate Circuit (IC Microprocessor

(Cards

کارت‌های ریزپردازنده (همچنین عموماً در صنعت بنام chip card نامبرده می‌شود) حافظه ذخیره‌سازی و امنیت بیشتری را نسبت به کارت‌های نوار مغناطیسی فعلی ارائه می‌کند. این نوع کارت‌ها همچنین می‌توانند داده روی

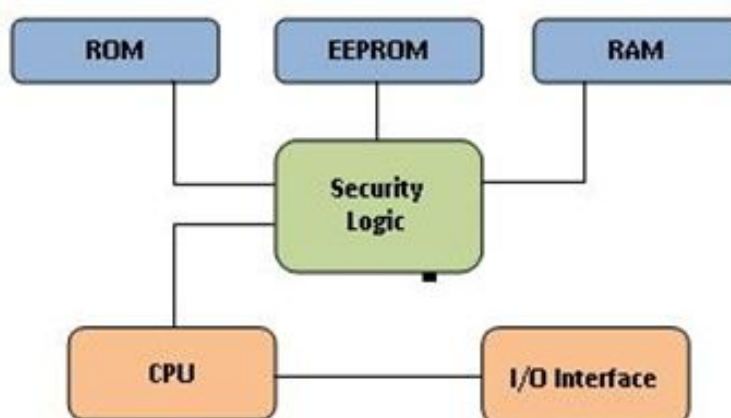
کارت را پردازش نمایند. نسل فعلی و تجاری این کارت‌ها دارای پردازنده ۸ بیتی، ۱۶ کیلوبایت حافظه فقط خواندنی و ۵۱۲ بایت حافظه دسترسی تصادفی (RAM) می‌باشند، که به آن‌ها قابلیت پردازشی معادل کامپیوترهای IBM-XT (البته با حافظه کمتر) را می‌دهد.

قیمت این کارت‌ها از کارت‌های نوع قبل بیشتر است و این کارت‌ها برای کاربردهای بسیار گوناگونی استفاده می‌شوند، بخصوص کاربردهایی که در خود رمزنگاری داشته و نیاز به مدیریت و محاسبات روی اعداد بزرگ را دارند. بنابراین چیپ کارت‌ها زیرساخت کارت‌هایی که ابزار شناسایی دیجیتال و امن را در خود دارند، می‌باشند (شکل ۱). برخی از کاربردهای این نوع کارت‌ها عبارتند از:

- کارت‌های تلفن‌های سلولار (SIM Cards)
- کارت‌های شناسایی
- کارت‌های امنیتی و دسترسی شبکه
- کارت‌های اعتباری و حاوی اطلاعات مالی

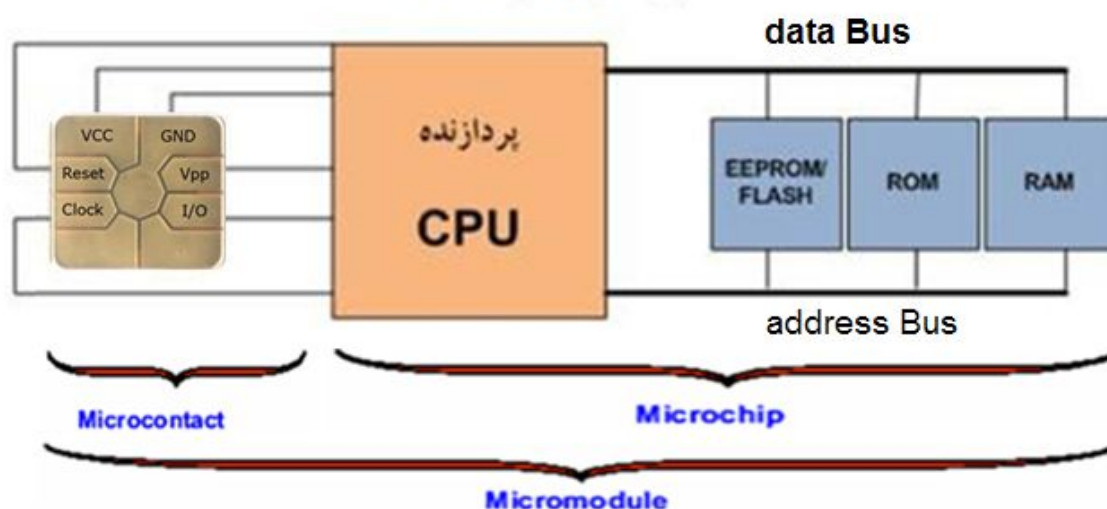
در ادامه به نقش هریک از واحدهای حافظه در این نوع کارت اشاره شده است:

- ✓ ROM: نگهداری سیستم‌عامل کارت هوشمند
- ✓ RAM: نگهداری موقت داده‌ها
- ✓ EEPROM: نگهداری برنامه کاربردی و داده‌های مرتبط با آن
- ✓ Security Logic: مهمترین قسمت‌های آن پردازنده امنیتی (Random Generator) و تولید کننده اعداد تصادفی (Random Generator) است.



واحد واسطه (Interface) این کارت ممکن است به یکی از صورت‌های تماسی، غیرتماسی و یا ترکیبی باشد که وظیفه برقراری ارتباط با محیط خارج از کارت را برعهده دارد. در شکل (۲) نحوه ارتباط یک واحد واسطه تماسی با CPU و واحدهای حافظه نمایش داده شده است.

ماژول کارت هوشمند



ارتباط ماژول ها در کارت هوشمند با واسطه ی تماسی

۳.۲.۴.۱-کارت های دارای حافظه نوری (Optical Memory Cards)

این نوع کارت ها شبیه یک دیسک فشرده است که در بالای کارت چسبانده شده است. کارت های حافظه نوری می توانند تا 4MB اطلاعات ذخیره کنند، ولی یکبار قابل نوشتن می باشند و داده قابل ویرایش یا حذف نمی باشد. این نوع کارت ها برای کاربردهایی که نیاز به ثبت سوابق است بسیار مناسب می باشند، مانند پرونده های پزشکی، کارت های رانندگی و یا سوابق مسافرتی. در حال حاضر این نوع کارت ها هیچ پردازنده ای ندارند (هرچند که بزودی انواع دارای حافظه نیز عرضه خواهد شد). با وجود اینکه این نوع کارت ها از نظر قیمت با کارت های تراشه قابل رقابت می باشد، اما کارت خوان ها از پروتکل های غیر استاندارد استفاده می کنند و گران می باشند.

۵.۱ برخی از کاربرد های مالی کارت های هوشمند :

هر کاربری در اینترنت برای دریافت خدمات و خرید محصول، نیاز به پرداخت اینترنتی خواهد داشت. برای پرداخت هم پنج راه پیش رو دارد:

۱- کارت های حساب (Debit Card)

۲- کارت های اعتباری (Credit Card)

۳- کارت های پیش پرداخت شده (Prepaid Card)

۴- کیف پول الکترونیکی

۵- کنسرسیوم ویزا و مستر (master card)

۱.۵.۱ Debit Card

مشابه کارت های بانکی ما هستند (کارت بانک ملی و ملت و ...) از این جهت که این کارت ها به حساب بانکی متصل هستند و صاحب کارت فقط به میزان موجودی حساب بانکیش حق برداشت از کارت را خواهد داشت؛ برداشت می تواند از طریق دستگاه POS – ATM و یا درگاه پرداخت اینترنتی انجام شود.

البته ممکن است بانک صادر کننده کارت محدودیت هایی تعریف بکند و به شما اجازه برداشت تمام موجودی کارت

را ندهد برای نمونه کارت های عمده بانک های ایران حق برداشت دویست هزار تومان پول نقد از دستگاه ATM را به شما می دهند و یا اجازه انتقال وجه سه میلیون تومانی در روز از دستگاه ATM را به کاربران خود می دهند (فرقی نمی کند حتی اگر شما چند ده میلیون تومان در حساب بانکی خود سپرده داشته باشید)



۲.۵.۱ Credit Card

مشابه کارت های طرح حکمت ما هستند از این جهت که تمام ویژگی های Debit Card را دارند ولی صاحب حساب می تواند به میزان اعتبارش نزد بانک از کارتش برداشت داشته باشد. به عبارتی صاحب کارت نیاز ندارد برای برداشت در حسابش موجودی داشته باشد و خواهد توانست خریدها و برداشت های خود از کارت را پایان ماه به بانک پرداخت نماید.



۳.۵.۱ Prepaid Card یا کارت های پیش پرداخت:

مشابه کارت هدیه بانک های ایران هستند. یعنی شما یک کارت با مبلغ مشخصی را می خرید و از آن امکان برداشت معادل موجودی کارت را خواهید داشت. تفاوت عمده آن با دو نوع دیگر، متصل نبودن این کارت ها به حساب بانکی است.



در حقیقت prepaid card ها هیچ چیزی نیستند جز اسکناس های صفر و یک شده. از آنجایی که همه کاربران اینترنت حساب بانکی ندارند (مثلا افراد زیر ۱۸ سال) و همه بانک های دنیا عضو کنسرسیوم ویزا و مستر نیستند و ... در عمل عده زیادی امکان پرداخت وجه در خریدهای اینترنتی را از طریق حساب بانکیشان را ندارند که این نوع کارت می تواند در این موارد به عنوان جایگزین عمل کند.

ظهور ده ها پول الکترونیک با حوزه فعالیت های منطقه ای و بین المللی شکل جدید از امکان پرداخت در اینترنت را فراهم کرده است. cashu و onecard در خاورمیانه - الی پی در چین - یوکش در اروپا، والی در هلند، لیبرتی رزرو و پرفکت مانی و ... از نوع پول های اینترنتی پیش پرداخت شده هستند.

۴.۵.۱ کیف پول الکترونیکی

کیف پول الکترونیکی از انواع کارت های الکترونیکی پول است که در بسیاری از کشورها رایج است. با استفاده از این کارت می توان کرایه تاکسی را حتی تا خردترین ارقام پرداخت کرد. تفاوت دیگری که این نوع کارت مثلا با کارت خرید دارد این است که این کارت رمز ندارد و اگر کسی آن را پیدا کند دقیقا مانند کیف پول واقعی بدون هیچ زحمتی می تواند از آن برداشت کند. این کارت که یکی از رایج ترین کارت ها در کشورهای توسعه یافته است، باعث حذف پول کاغذی از جامعه و بالطبع صرفه جویی بسیار در این راستا است.



۵.۵.۱ کنسرسیوم ویزا و مستر (master card) :

ویزا و مستر مانند شبکه شتاب ایران هستند که البته به صورت بین المللی فعالیت می کنند. کارتی که عضو شبکه شتاب باشد به شما اجازه می دهد از تمام دستگاه های ATM و POS ها و درگاه های اینترنتی متصل به

شتاب پرداخت یا برداشت انجام دهید. اما کارت هایی که عضو شبکه شتاب نیستند تنها در دستگاه های ATM و POS های همان بانک قابل استفاده هستند. ویزا و مستر هم همین قابلیت را فراهم می کنند. کارتی که به شما توسط بانک داده می شود (چه Debit و چه Credit) اگر دارای لوگوی ویزا و مستر باشد؛ کافی است به دنبال ATM , POS , online payment gateway , هایی بگردید که لوگوی ویزا یا مستر روی آن درج شده باشد. به عبارتی شما محدود به استفاده از روش های پرداخت و برداشت ارئه شده از بانک صادر کننده کارت خود نیستید. در این شبکه تبدیل ارز نیز صورت می گیرد و می توانید با کارت صادر شده از MKB بانک مجارستان در بنگلادش، تاکا از دستگاه ATM برداشت نمایید.



لوگوی ویزا و مستر و...



البته در کشور ما پس از انقلاب و به دلیل تحریم های یک جانبه امکان تبادل وجه در بانک ها از طریق کارت هایی نظیر master card میسر نمی باشد . این امر خود سبب ایجاد اختلال در برخی امور شده است . به عنوان نمونه گردشگران خارجی نمی توانند هزینه های خود را از طریق کارت های بین المللی بپردازند و مجبورند برای سفر به ایران تبدیل ارز انجام دهند و پول نقد حمل کنند.

فصل دوم:

ارتباط میکرو با کارت تلفن

۱.۲ مقدمه

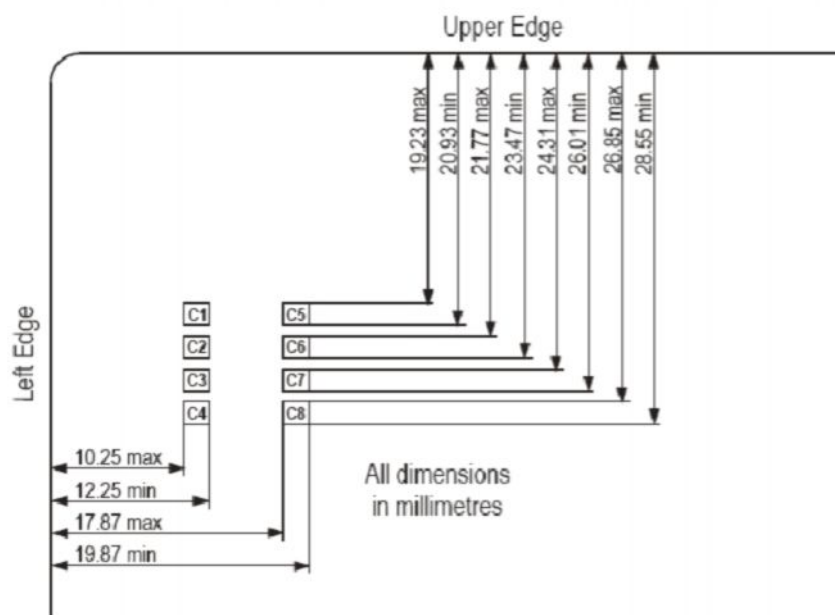
در این فصل شما را نحوه ی اتصال کارت های تلفن به میکروکنترلر avr و نحوه خواندن محتویات کارت و برنامه نویسی با هر دو کامپایلر bascom و codevision آشنا خواهیم.

۲.۲ مشخصات فیزیکی کارت تلفن

کارت تلفن ها نیز مانند سایر اسمارت کارت ها از استاندارد ISO7816 برای ابعاد کارت و محلی قرار گیری آیسی حافظه استفاده می کنند. در شکل زیر می توانید نمونه ای از کارت تلفن را مشاهده کنید که استاندارد فوق در آن رعایت شده.



نکته: تمامی مقادیر مشخص شده برای ابعاد کار بر حسب واحد میلی متر (mm) می باشد.
ابعاد و محل قرار گرفتن پایه های آیسی بر روی کارت (شکل زیر شامل کلیه اسمارت کارت ها می باشد)



اتصالات c4 و c8 در کارت‌های تلفن مورد استفاده قرار نمی‌گیرد در نتیجه بر روی آن هم تعبیه نمی‌شود.

۳.۲ مشخصات الکترونیکی کارت تلفن

۱- V_{cc} : این پایه، پایه‌ی مثبت تغذیه‌ی آیسی حافظه می‌باشد و باید به +5 ولت متصل شود.

۲- GND : پایه، پایه‌ی منفی تغذیه‌ی آیسی حافظه می‌باشد و باید به 0 ولت (زمین مدار) متصل شود.

۳- VPP : این پایه جهت برنامه‌ریزی حافظه (EEPROM) با ولتاژ بالا به کار می‌رود، در این فصل چون فقط قصد خواندن از کارت را داریم از این پایه استفاده نخواهیم کرد.

۴- $clock$: برای ارتباط آیسی حافظه با دنیای بیرون نیاز به سیگنال پالس دارد که از این پایه کلاک مورد نیاز خود را تامین می‌کند.

۵- $reset$: از این پایه برای ریست کارت استفاده می‌شود.

۶- I/O : داده‌ها از این پایه به صورت سریال وارد یا خارج می‌شوند. (به صورت نیمه دوطرفه^۱، یعنی در آن واحد

این پایه یا می‌تواند ورودی باشد یا خروجی.)



^۱ half-duplex

۴.۲ حافظه کارت تلفن

کارت تلفن ها معمولاً دارای ۶۴ بیت حافظه eeprom می باشند (یعنی ۵۱۲ بیت).

بیت ۰ تا ۳۱	11011000	00101010	11111111	11001010
بیت ۳۲ تا ۶۳	00101110	11101000	01001100	11000000
بیت ۶۴ تا ۹۵	00000000	00000000	00000011	00000111
بیت ۹۶ تا ۱۲۷	00001111	11111111	11111111	11111111
بیت ۱۲۸ تا ۱۵۹	11111111	11111111	11111111	11111111
بیت ۱۶۰ تا ۱۹۱	11111111	11111111	11111111	11111111
بیت ۱۹۱ تا ۲۲۳	11111111	11111111	11111111	11111111
بیت ۲۲۴ تا ۲۵۵	11111111	11111111	11111111	11111111
بیت ۲۵۶ تا ۲۸۷	11111111	11111111	11111111	11111111
بیت ۲۸۸ تا ۳۱۹	11111111	11111111	11111111	11111111
بیت ۳۲۰ تا ۳۵۱	10101010	00100100	11011011	11101111
بیت ۳۵۲ تا ۳۸۳	00010000	10011010	00101000	01111010
بیت ۳۸۴ تا ۴۱۵	11111111	11111111	11111111	11111111
بیت ۴۱۶ تا ۴۴۷	11111111	11111111	11111111	11111111
بیت ۴۴۸ تا ۴۷۹	11111111	11111111	11111111	11111111
بیت ۴۸۰ تا ۵۱۱	11111111	11111111	11111111	11111111

چهار بیت اول مربوط به کدهای مخصوص کشور و شرکت سازنده و چهار بیت دوم مربوط به سریال کارت می باشد در واقع همان عددی است که بر روی کارت حک شده البته به صورت هگزا. بیت های ۹ و ۱۰ و ۱۱ و ۱۲ و ۱۳ حاوی اعتبار کارت می باشند.

۵.۲ پروتکل ارتباطی کارت تلفن

برای برقراری ارتباط با کارت تلفن به سه عدد از پایه های آن نیاز داریم که عبارتند از CLK ، Res و I/O. به پایه های CLK ، Res پایه های کنترلی و به پایه I/O پایه دیتا گفته می شود.



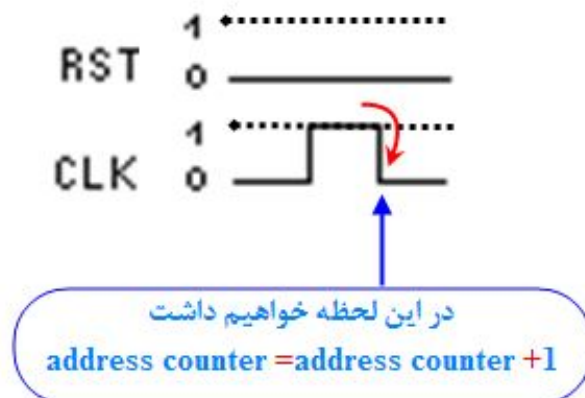
۱.۲.۵ ارتباط با کارت

: Reset

ابتدا پایه‌ی ریست به حالت بالا (+۵) رفته و سپس پایه clock در وضعیت بالا قرار می‌گیرد در این حالت مقدار شمارنده ادرس، ریست شده و به ۰ تبدیل می‌شود و به ازای هر پالس ساعت مقدار address counter (شمارنده ادرس حافظه) یک واحد افزایش می‌یابد البته تا مادامی که پایه ریست در وضعیت پایین قرار دارد.

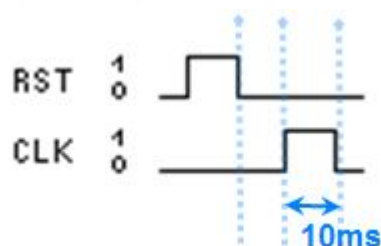


بعد از ریست و پایین آمدن پایه ریست و clock، اولین بیت در پایه i/o ظاهر می‌شود و به ترتیب با هر بالا رفتن clock مقدار شمارنده افزایش یافته و با پایین آمدن آن مقدار بیت بعدی در i/o قرار می‌گیرد این بدین معنا است که برای خواندن بیت x باید پایه clock به تعداد n بار در وضعیت بالا و سپس پایین قرار بگیرد تا شمارنده ادرس حافظه برابر با مقدار x شود در نتیجه خروجی i/o بلافاصله بعد از پایین آمدن ساعت برابر با مقدار بیت x می‌شود.



: Write

بالا بردن پایه ریست (+5) در حالی که ساعت در وضعیت پایین (0) قرار دارد، به معنای غیر فعال کردن شمارنده آدرس می‌باشد در این حالت بعد از پایین آمدن ریست clock، بمدت حداقل ۱۰ میلی ثانیه در وضع بالا قرار می‌گیرد و مقدار



بیتی که شمارنده آدرس به آن اشاره می‌کند به صفر تبدیل می‌شود یا به عبارتی عمل Write انجام می‌شود و بعد از پایین آمدن ساعت مقدار بیت جدید که ۰ می‌باشد در i/o قرار می‌گیرد و همچنین شمارنده آدرس دوباره فعال می‌شود و با پالس ساعت بعدی بطور معمول یک واحد افزایش خواهد یافت.

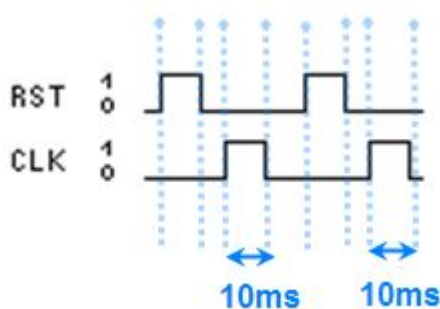
: unit counter

معمولا ۴ بیت (در بعضی موارد ۵ بیت) به اختیار کارت اختصاص می‌یابد که شامل بایتهای ۱۰ و ۱۱ و ۱۲ و ۱۳ (در حالت ۵ بیتی بیت نهم را نیز شامل می‌شود) است برای کاهش یک واحد از اعتبار، یکی از بیت‌های بیت ۱۳ از مقدار ۱ به ۰ تبدیل می‌شود در نتیجه بعد از ۸ بار کاهش اعتبار، مقدار بیت ۱۳ برابر ۰ می‌شود و برای کاهش اعتبار کارت باید یک بیت از بیت قبلی (۱۲) به ۰ تبدیل شود و در مقابل ۸ بیت بیت ۱۳ به ۱ مبدل شود به زبان ساده هر یک واحد (بیت و یا خانه) از بیت ۱۲ برابر با ۸ واحد از بیت ۱۳ و همینطور هر واحد از بیت ۱۱ برابر با ۸ واحد بیت ۱۲ به این ترتیب بیت ۱۳ برابر ۸ واحد و بیت ۱۲ برابر ۸*۸ واحد و بیت ۱۱ برابر ۸*۸*۸ واحد و بیت ۱۰ برابر ۸*۸*۸*۸ واحد که مجموعا ۴۶۸۰ واحد حاصل می‌شود

همانطور که گفته شد بعد از ۸ بار کاهش اعتبار، همه خانه های (بیت های) ۱۳ برابر ۰ می شود در این شرایط با استفاده از WRITECARRY یک بیت از بیت مورد نظر (۱۲) را پاک (WRITE) می کنیم و در مقابل ۸ بیت بیت بعدی (۱۳) به ۱ تبدیل می شوند و اکنون می توان یک واحد (بیت) از بیت ۱۳ را بمنظور کاهش اعتبار به ۰ تبدیل کرد (WRITE) و این در مورد بایتهای ۱۱ و ۱۰ نیز صادق است.

: WriteCarry

در حالی که ساعت در وضعیت پایین قرار دارد ریست بالا می رود که باعث غیر فعال شدن شمارشگر می شود بعد از پایین آمدن ریست، ساعت بمدت ۱۰ میلی ثانیه بالا می رود و خانه (بیت) مورد نظر ۰ می شود با پایین آمدن

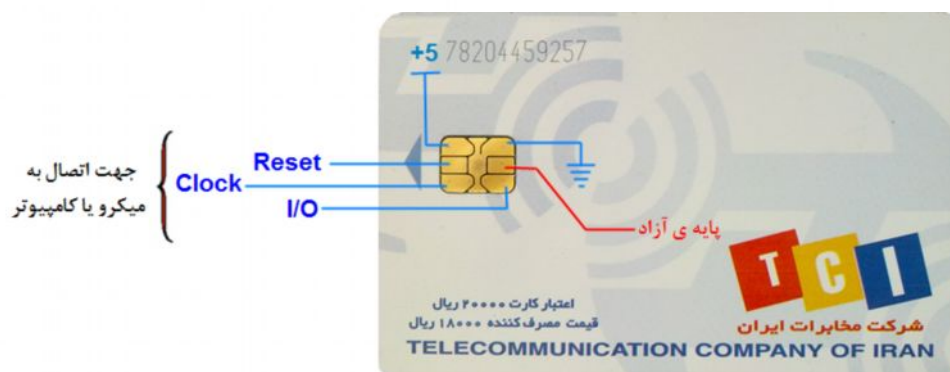


ساعت، ۰ در i/o ظاهر می شود (پایه i/o در وضعیت پایین قرار می گیرد) و مجدداً شمارشگر فعال می شود اما قبل از پالس بعدی ساعت ریست بالا می رود و این به معنی درخواست WriteCarry از کارت و همچنین غیر فعال کردن شمارشگر ادرس می باشد، در نتیجه بعد از پایین آمدن ریست پایه ساعت بمدت ۱۰ میلی ثانیه در وضعیت بالا باقی می ماند تا کارت بتواند بیت مورد نظر را به FF\$ تبدیل کند (در همه خانه های آن مقدار ۱ را قرار دهد)

۶.۲ سخت افزار مورد نیاز برای ارتباط

۱.۶.۲ شماتیک

سه پایه reset، clock و i/o قابل اتصال به پورت دلخواه از میکرو.



۲.۶.۲ سوکت اسمارت کارت

برای کارت تلفن هم از همین سوکت استفاده می‌شود، زیرا همان طور که گفته شد کارت تلفن هم از استاندارد ISO7816 استفاده می‌کند (در اینجا از لحاظ ابعاد و محل قرار گیری اتصالات).



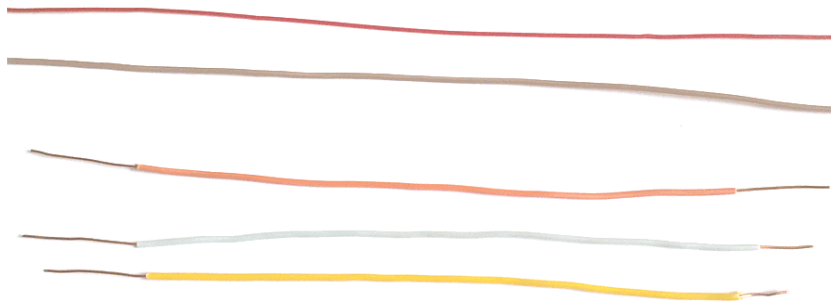
۳.۶.۲ اتصالات پایه ها بر روی سوکت



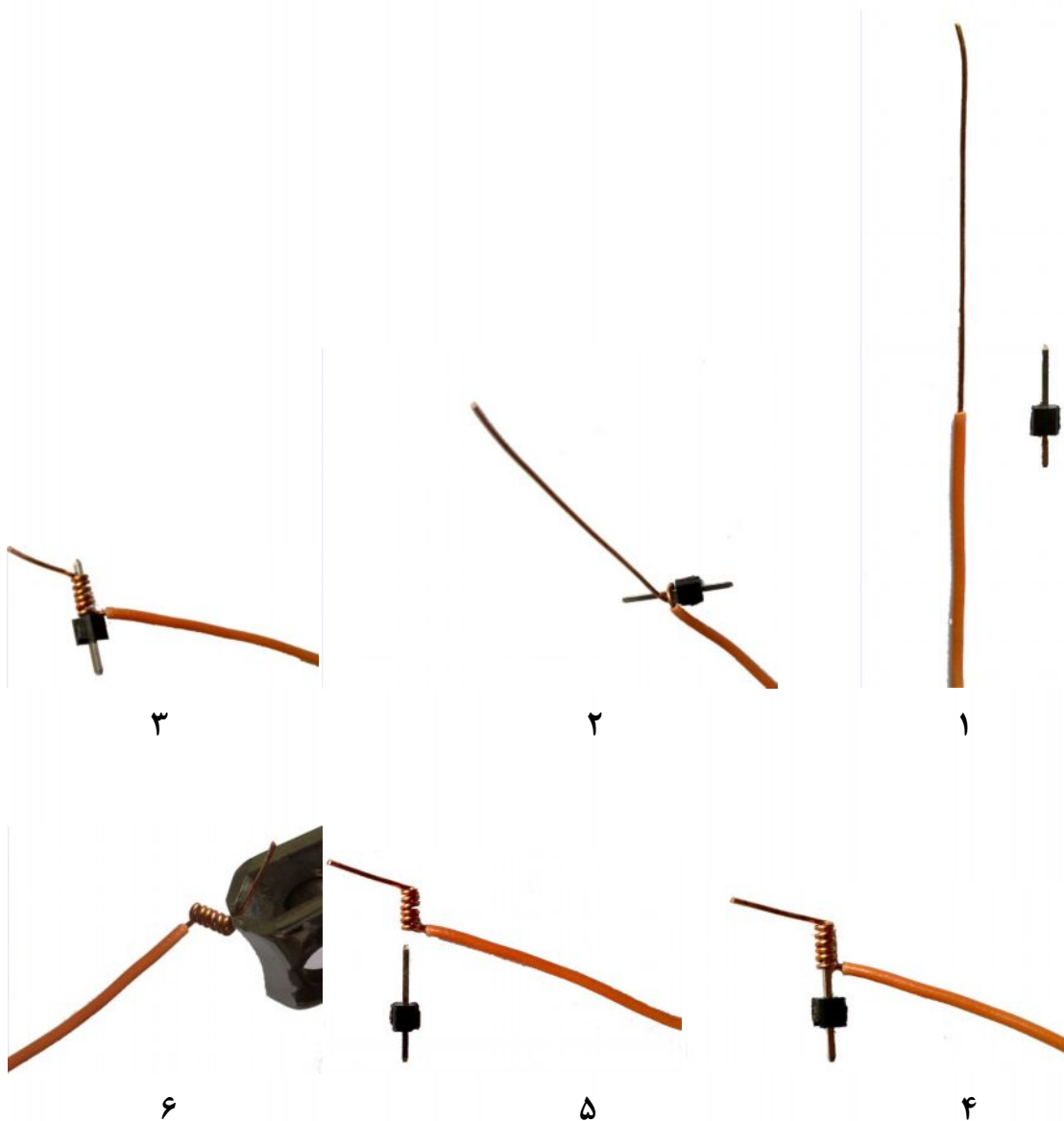
۴.۶.۲ نحوه استفاده از سوکت

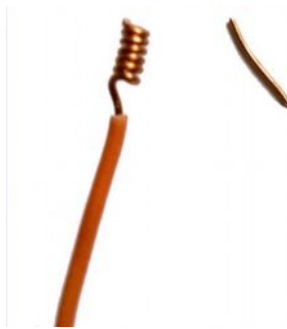
دو را پیش رو خواهید داشت:

۱. باید سوکت را بر روی فیبر سوار کنید، که می‌توانید از برد آموزشی موجود در سایت که شامل GLCD و تاج نیز می‌باشد استفاده کنید.
 ۲. از آنجایی که این سوکت را نمی‌توان بر روی بردبرد جای داد از ترفند زیر برای اتصال به بردبرد استفاده کنید.
- ابتدا ۵ تکه سیم تهیه کرده سپس از دو سر آنها را لخت کنید.



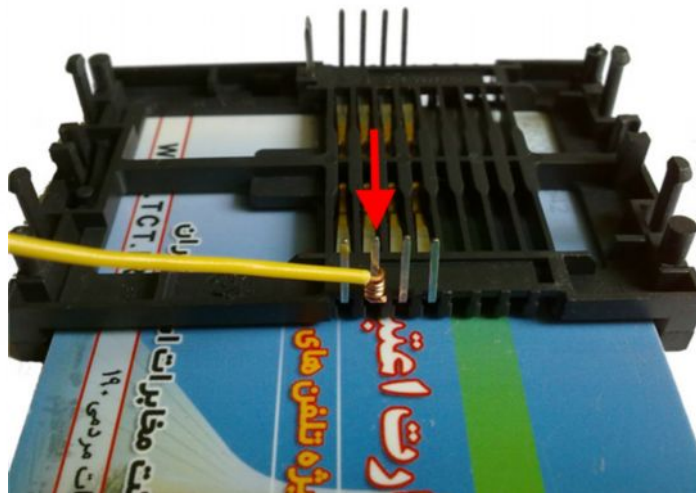
سپس یک پایه از پین حدر را جدا کرده و مراحل را مطابق تصاویر دنبال کنید. (از یک میله مناسب نیز میتوانید استفاده کنید.)





۷

مطابق شکل زیر سیم ها را در محل مربوطه قرار دهید.



از این مرحله به بعد به دو روش می‌توانید ادامه دهید.

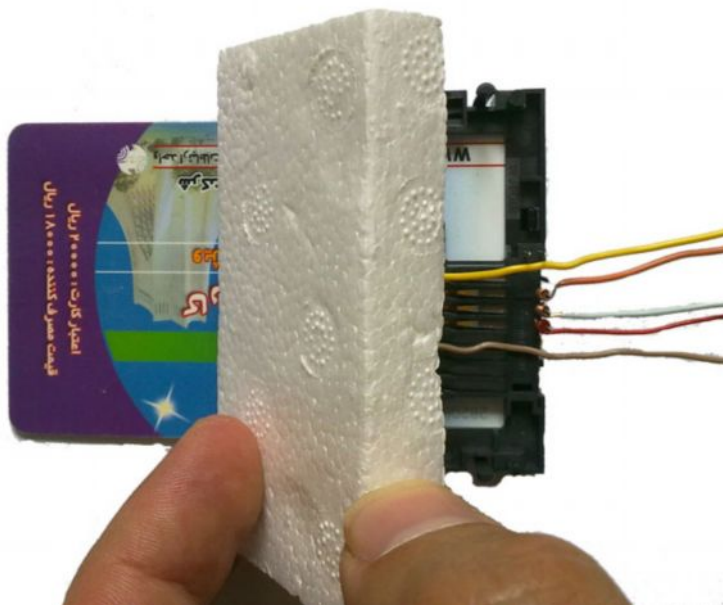
۱. به صورت زیر سیم ها را به اطراف سوکت محکم کنید.



۲۱

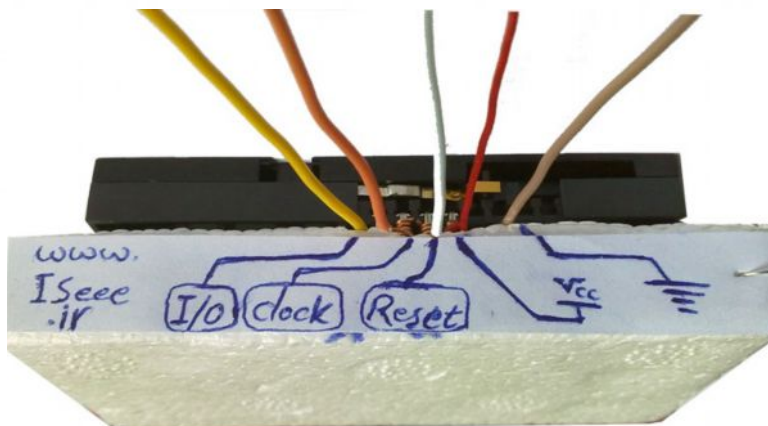


۲. ابتدا یک تکه یونولیت را به اندازه‌ی سوکت برش داده و سپس مطابق اشکال زیر سوکت را بروی آن قرار داده، حال فشار دهید تا پایه های سوکت در یونولیت فرو روند.





در این حالت می‌توانید نام پایه‌ها را نیز بر روی یونولیت مشخص کنید. (مطابق شکل زیر)

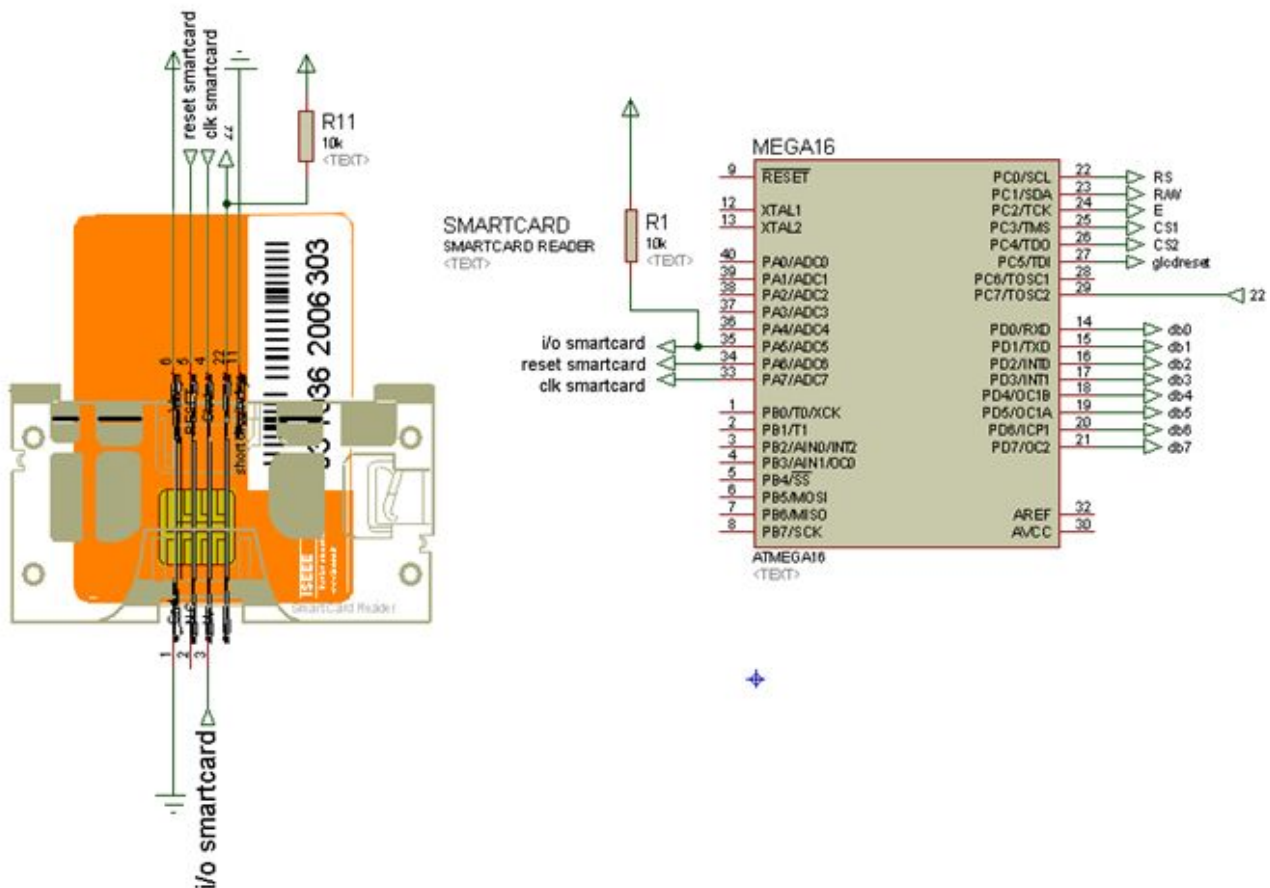


اکنون می‌توانید با استفاده از این سیم‌ها کارت تلفن را بر روی بردبرد تست نمایید.

۷.۲ ارتباط با میکرو و برنامه نویسی

۱.۷.۲ اتصالات سخت افزاری مورد نیاز

۱.۱.۷.۲ سخت افزار با نمایشگر GLCD



در سخت افزار فوق اتصالات به قرار زیر می باشد:

الف. اتصالات GLCD

پورت D به باس دیتای GLCD اختصاص یافته و ۶ پین از پایه های پورت C به باس کنترلی GLCD متصل شده.

ب. اتصالات کارت تلفن

پین PA5 متصل به i/o کارت

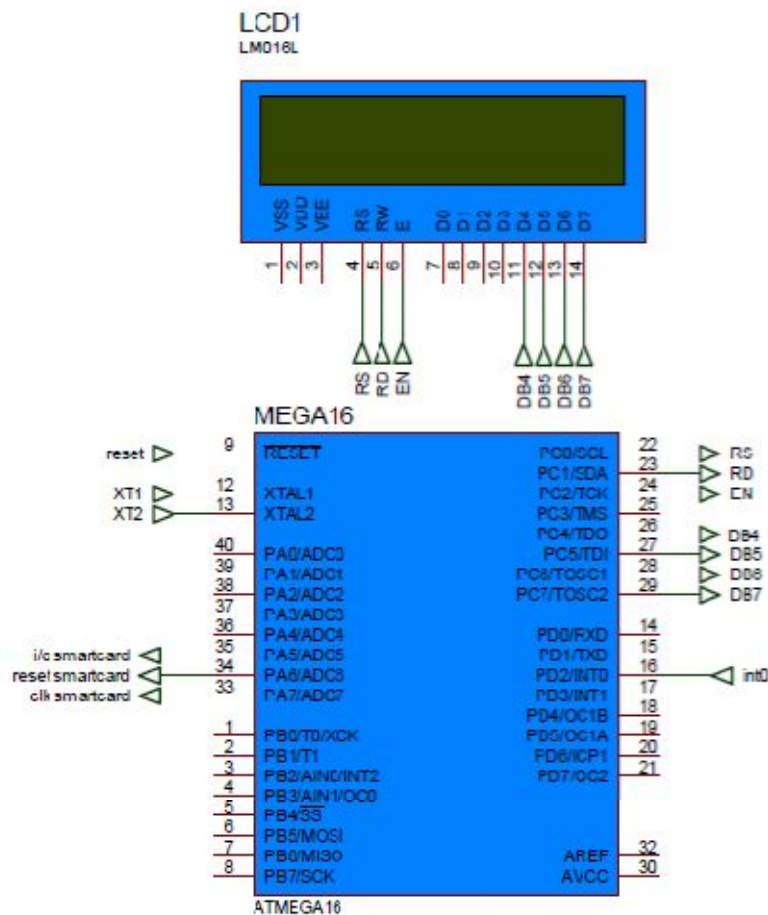
پین PA6 متصل به ریست کارت

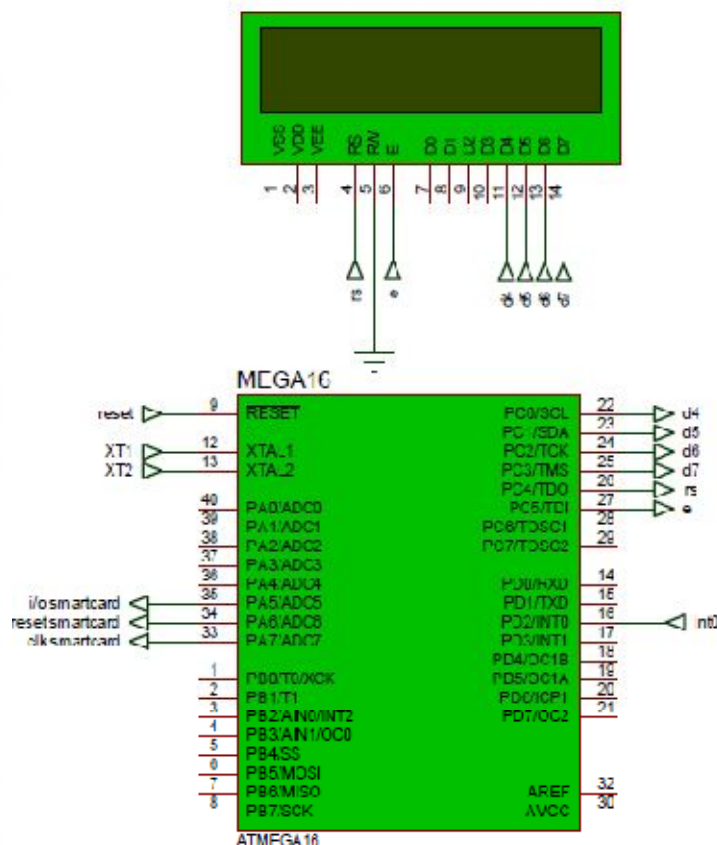
پین PA7 متصل به کلاک کارت

پین PC7 متصل به کلید تعبیه شده بر روی سوکت جهت تشخیص ورود کارت به سوکت.

توجه مهم: پایه i/o کارت اوپن کلکتور می باشد پس حتما از مقاومت پول آپ استفاده شود، می توانید مقاومت پول آپ داخلی پایه ای از میکرو که به i/o متصل می باشد را فعال کنید یا که مثل ما از مقاومت بیرونی جهت پول آپ استفاده کنید.

۲.۱.۷.۲ سخت افزار با نمایشگر LCD کاراکتری





در سخت افزار فوق اتصالات به قرار زیر می باشد:

الف. اتصالات LCD

شماتیک اول (رنگ آبی) برای codevision می باشد و شماتیک دوم (رنگ سبز) برای bascom که در هر دو lcd به پورت C متصل می باشد.
تذکر: برای استفاده از پورت C برای اتصال به LCD حتما باید JTAG را غیر فعال کنید. (می توانید نحوه ی غیر فعال کردن JTAG را با جستجو در google پیدا کنید.)
ب. اتصالات کات تلفن

دقیقا مشابه حالت قبلی می باشد یعنی:

پین PA5 متصل به i/o کارت

پین PA6 متصل به ریست کارت

پین PA7 متصل به کلاک کارت

پین PC7 متصل به کلید تعبیه شده بر روی سوکت جهت تشخیص ورود کارت به سوکت.

۲.۷.۲ برنامه نویسی

در این بخش افراد به دو دسته تقسیم می شوند، گروه اول فقط با codevision کار کرده اند و گروه دوم با bascom، که در ادامه برای هر دو گروه توضیحات لازم داده شده است. توضیحات و مثال ها آورده شده در ادامه برای هر دو کامپایلر کاملاً یکسان بوده پس دوستانی که به هر دو کامپایلر تسلط دارند مطالعه یکی از آنها کفایت می کند.

۱.۲.۷.۲ برنامه نویسی با codevision

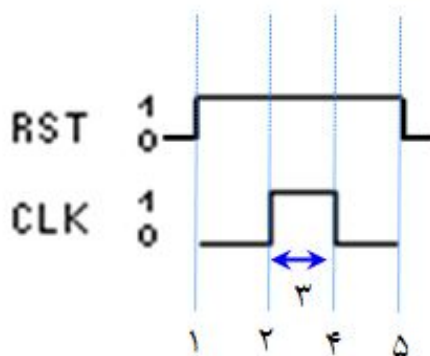
مرحله اول:

قبل از هر چیز پایه های متصل به کارت را نامگذاری می کنیم تا بتوانید به راحتی کارت را به پایه های دلخواه خود متصل کنیم.

```
#define Io_telecard PINA.5
#define Reset_telecard PORTA.6
#define Clk_telecard PORTA.7
#define Check_insert_telecard PIN.7
```

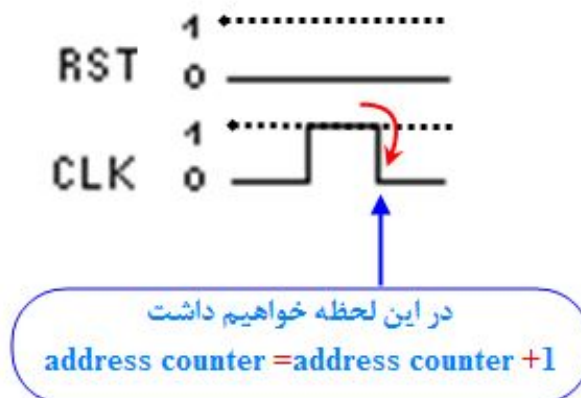
مرحله دوم:

برای خواندن از کارت همان طور که گفته شد ابتدا باید کارت ریست شود. یعنی باید مطابق شکل زیر عمل کنیم



زیر برنامه ریست کارت:

```
void Resettelecard()
{
    Reset_telecard=1;
    Clk_telecard=1;
    delay_ms(50);
    Clk_telecard=0;
    Reset_telecard=0;
}
```



مرحله سوم:

حال نوبت به خواندن از کارت می رسد، که باید با هر بار آمدن پالس کلاک از سطح بالا به پایین باید بیت قرار گرفته بر روی خروجی i/o کارت، با میکرو خوانده و در متغیری ذخیره کنیم. برای این کار ابتدا یک متغیر آرایه‌ای از نوع بایت تعریف می‌کنیم. چون ۶۴ بایت داریم پس باید آرایه‌ای ما ۶۴ تایی باشد و از آنجایی که هر بایت ۸ بیتی می باشد پس نوع بایت بهترین انتخاب برای نوع آرایه‌ای ما می باشد. یعنی خواهیم داشت.

`char array[64];`

در زبان C دستوری برای دسترسی به تک تک بیت های یک متغیر موجود نمی باشد، و برای این کار یا باید از عملگرهای بولی مثل OR و شیفت استفاده شود یا مطابق زیر عمل شود.
 با دستور زیر می‌توان بیت مورد نظر از متغیری را تبدیل به یک کرد:

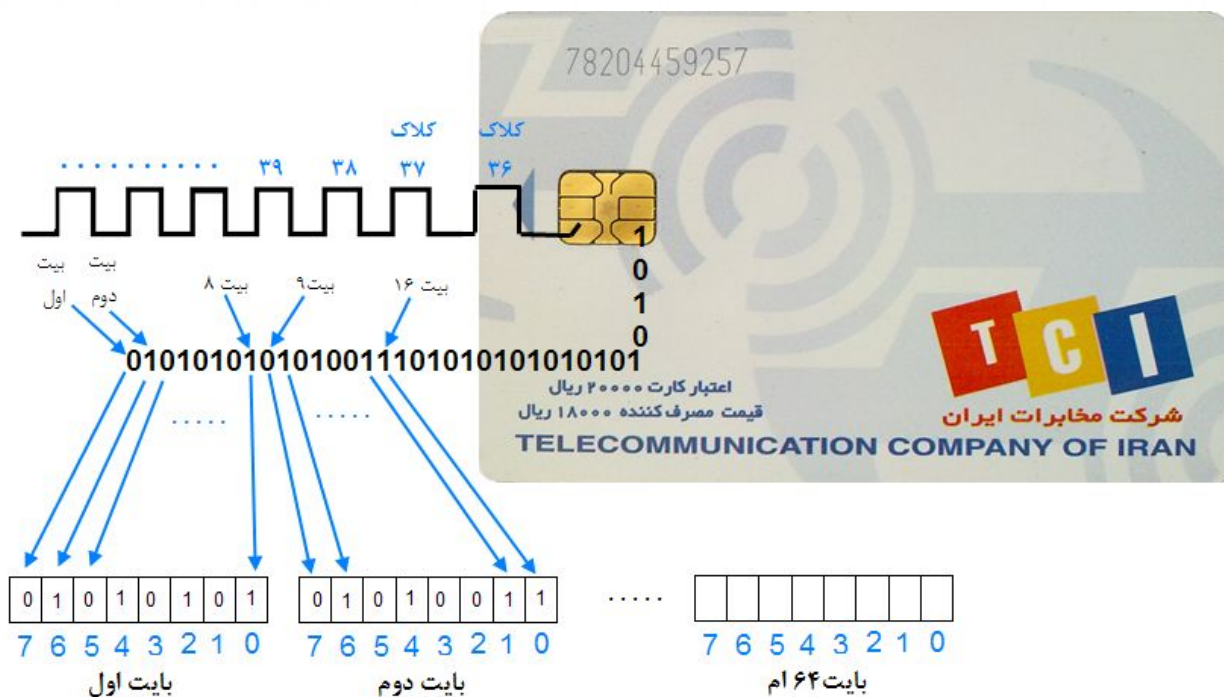
`variable |= 1 << num_bit;`

یعنی اگر `num_bit` برابر با ۳ شود، بیت چهارم (و نه بیت سوم) از متغیر `variable` چه صفر باشد چه یک تبدیل به یک می‌شود.

برای صفر کردن بیتی از متغیر نیز از دستور زیر استفاده می‌کنیم.

`variable &= ~(1 << num_bit);`

یعنی اگر `num_bit` برابر با ۰ شود، بیت صفرم از متغیر `variable` چه صفر باشد چه یک تبدیل به صفر می‌شود.



حال از دو حلقه تو در تو برای ذخیره بیتها استفاده می کنیم، که متغیر حلقه‌ی داخلی وظیفه‌ی شمارش بیت های بایت مورد نظر را بر عهده دارد و حلقه بیرونی وظیفه کنترل بایت‌ها را بر عهده دارد.

```
for (i=0;i<64;i++)
{
    for (j=7;0<=j;j--)
    {
        if(Io_telecard==1){
            array[i]|=1<<j;
        }
        else{
            array[i]&=~(1<<j);
        }
        Clk_telecard=1;
        delay_us(100);//Wait 10us
        Clk_telecard=0;
    }
}
```

در ابتدا $i=0$ می‌باشد پس با وارد شدن به حلقه داخلی با تعداد تکرار ۸ بار (از ۷ تا ۰) ۸ پالس ایجاد شده و بیت قرار گرفته در i/o بعد از هر پالس در خانه‌ی i ام از بایت اول ذخیره می‌شود. این عمل با استفاده از دستور شرطی if صورت می‌گیرد به این صورت که با چک کردن i/o تصمیم می‌گیرد که در بیت مورد نظر از متغیر مربوطه مقدار ۱ قرار دهد یا مقدار صفر.

حال بعد از ذخیره ۸ بیت اول باید بیت های بعدی در بایت دوم ذخیره شوند که این امر توسط حلقه‌ی بیرونی انجام می‌شود یعنی با خارج شدن از حلقه داخلی، حلقه بیرونی به i یک عدد اضافه می‌کند و مقدار i برابر با ۱ می‌شود، در این حالت برنامه دوباره وارد حلقه‌ی داخلی شده و ۸ بیت را خوانده و در خانه های بایت دوم ذخیره می‌کند. این مراحل ۶۴ بار تکرار می‌شود تا تمامی بیت های حافظه کارت خوانده شود.

تذکر: در codevision شماره‌ی متغیر آرایه از صفر شروع می‌شود. array(4) شامل ۴ متغیر از شماره‌های ۰ تا ۳ می‌باشد

ولی شماره بیت‌های متغیر آرایه مانند سایر متغیرها از صفر شروع می‌شود، یعنی در خانه اول متغیر از نوع بایت، بیت صفرم قرار دارد و در خانه هشتم، بیت ۷ تم قرار دارد. (به همین دلیل حلقه داخلی از ۰ تا ۷ می‌باشد)
زیر برنامه خواندن از کارت تلفن:

```
void Readtelecard()
{
    signed char i,j;
    DDRA.5=0; //Config Porta.5 = Input
    DDRA.6=1; //Config Porta.6 = Output
    DDRA.7=1; //Config Porta.7 = Output
    for (i=0;i<64;i++)
    {
        for (j=7;0<=j;j--)
        {
            if(Io_telecard==1){
                array[i]=1<<j;
            }
            else{
                array[i]&=~(1<<j);
            }
            Clk_telecard=1;
            delay_us(100); //Wait 10us
            Clk_telecard=0;
        }
    }
}
```

تا این مرحله حافظه خوانده شد و در داخل متغیر array در دسترس می‌باشد. می‌توانید متناسب با پروژه خود از آن استفاده کنید.

پروژه ۱\ (codevision): (پیوست مقاله می‌باشد).

خواندن محتوای کارت تلفن و نمایش ۸ بایت اول بر روی GLCD همان طور که در شکل مشاهده می‌کنید مقادیر دسیمال بایت‌ها بر روی LCD نشان داده شده‌اند.



```
#include <mega32.h>
#include<delay.h>
#include <ks0108.h>
#include <stdlib.h>
#define ADC_VREF_TYPE 0x20
#define Io_telecard PINA.5
#define Reset_telecard PORTA.6
#define Clk_telecard PORTA.7
#define Check_insert_telecard PIN.7
void Showresult();
void Resettelecard();
void Readtelecard();
void init();

unsigned int read_adc(unsigned char adc_input)
{
  ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
  // Delay needed for the stabilization of the ADC input voltage
  delay_us(10);
  // Start the AD conversion
  ADCSRA|=0x40;
  // Wait for the AD conversion to complete
  while ((ADCSRA & 0x10)==0);
  ADCSRA|=0x10;
  return ADCH;
}

char array[64],stemp[4];
void init();
char i;
signed int x,y;

void main(void)
```

```

{

    // Micro initialize
    init();
    // Graphic Lcd initialize
    glcd_Init();

    while (1)
    {
        /*******x coordinate*****
        PORTA=0b00001;
        DDRA=0b00000101;
        delay_ms(20);
        x=read_adc(1);
        /******* y coordinate *****
        PORTA=0b00000010;
        DDRA=0b00001010;
        delay_ms(20);
        y=read_adc(0);
        /*******
        if (x!=0 && y!=0)
        {
            glcd_Clear();
            Resetelecard();
            Readtelecard();
            Showresult();
        }
    };
}

void init()
{
    // Declare your local variables here

    // Input/Output Ports initialization
    // Port A initialization
    // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
    // State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
    PORTA=0x20;
    DDRA=0xC0;

    // Port B initialization
    // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
    // State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
    PORTB=0x00;
    DDRB=0x00;

    // Port C initialization
    // Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out Func1=Out Func0=Out
    // State7=0 State6=0 State5=0 State4=0 State3=0 State2=0 State1=0 State0=0
    PORTC=0x20;
    DDRC=0x7F;

    // Port D initialization
    // Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out Func1=Out Func0=Out
    // State7=0 State6=0 State5=0 State4=0 State3=0 State2=0 State1=0 State0=0

```

```

PORTD=0x00;
DDRD=0xFF;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=FFh
// OC0 output: Disconnected
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer 1 Stopped
// Mode: Normal top=FFFFh
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer 1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer 2 Stopped
// Mode: Normal top=FFh
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
MCUCR=0x00;
MCUCSR=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;

// ADC initialization
// ADC Clock frequency: 125.000 kHz

```

```
// ADC Voltage Reference: AREF pin
// ADC Auto Trigger Source: None
ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0x83;
}
```

```
//*****
```

```
void Showresult()
{
    unsigned char strbyte[] = "byte",si[2];
    glcd_Clear();
    for(i=0;i<8;i++)
    {
        itoa(array[i],stemp);
        itoa(i,si);
        glcd_Printf(i,0,strbyte);
        glcd_Printf(i,26,si);
        glcd_Putchar(i,34,'=');
        glcd_Printf(i,40,stemp);
    }
    delay_ms(500);
}
```

```
//*****
```

```
void Resettelecard()
{
```

```
    Reset_telecard=1;
    Clk_telecard=1;
    delay_ms(50);
    Clk_telecard=0;
    Reset_telecard=0;
}
```

```
void Readtelecard()
```

```
{
    signed char i,j;
    DDRA.5=0; //Config Porta.5 = Input
    DDRA.6=1; //Config Porta.6 = Output
    DDRA.7=1; //Config Porta.7 = Output
    for (i=0;i<64;i++)
    {
        for (j=7;0<=j;j--)
        {
            if(Io_telecard==1){
                array[i]=1<<j;
            }
            else{
                array[i]&=~(1<<j);
            }
            Clk_telecard=1;
            delay_us(100); //Wait 10us
            Clk_telecard=0;
        }
    }
}
```

پروژه ۲ (codevision): (پیوست مقاله می باشد).

همان پروژه قبلی است با این تفاوت که مقدار نمایش داده شده بر روی LCD بر حسب هگز می باشد.
پس در برنامه قبلی کافی است زیر برنامه Showresult به صورت زیر تغییر یابد.

```
void Showresult()
{
    unsigned char strbyte[] = "byte",si[2];
    glcd_Clear();
    for(i=0;i<8;i++)
    {
        //itoa(array[i],stemp);
        sprintf(stemp,"0x%2X",array[i]);
        itoa(i,si);
        glcd_Printf(i,0,strbyte);
        glcd_Printf(i,26,si);
        glcd_Putchar(i,34,'=');
        glcd_Printf(i,40,stemp);
    }
    delay_ms(500);
}
```



پروژه ۳ (codevision): (پیوست مقاله می باشد).

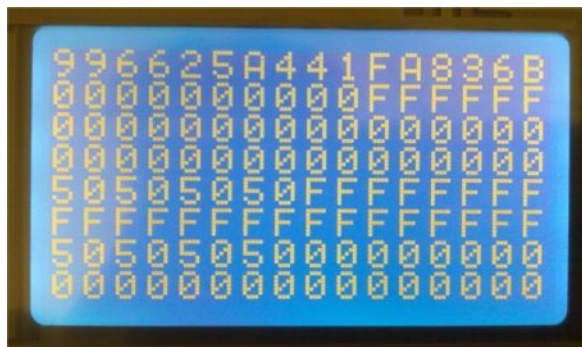
همان پروژه قبلی با نمایش کل محتویات بر روی LCD، پس برنامه این پروژه نیز فقط در زیر برنامه Showresult با دو برنامه قبل متفاوت است:

```
void Showresult()
{
    char n,j ;
    n=0;
    glcd_Clear();
```

```

for(i=0;i<8;i++)
{
    for(j=1;j<=128;j=j+16)
    {
        sprintf(stemp,"%02X",array[n]);
        glcd_Printf(i,j,stemp);
        n++;
    }
}
delay_ms(500);
}

```



پروژه ۴ (codevision): (پیوست مقاله می باشد).

خواندن محتوای کارت تلفن و نمایش ۸ بایت اول بر روی LCD کاراکتری ۱۶ در ۲.

```

#include <mega32.h>

// Alphanumeric LCD Module functions
#asm
.equ __lcd_port=0x15 ;PORTC
#endasm
#include <lcd.h>
#include <delay.h>
#include <ks0108.h>
#include <stdlib.h>
#define ADC_VREF_TYPE 0x20
#define Io_telecard PINA.5
#define Reset_telecard PORTA.6
#define Clk_telecard PORTA.7
#define Check_insert_telecard PIND.7
void Showresult();
void Resettelecard();
void Readtelecard();

// External Interrupt 0 service routine
interrupt [EXT_INT0] void ext_int0_isr(void)
{
    // Place your code here
    lcd_clear();
    Resettelecard();
    Readtelecard();
    Showresult();
}
// Declare your global variables here

```

```

char array[64];
char i;
bit flag;

void main(void)
{
PORTA=0x20;
DDRA=0x20;

PORTB=0x00;
DDRB=0x00;

PORTC=0x00;
DDRC=0x00;

PORTD=0x00;
DDRD=0x00;

// External Interrupt(s) initialization
// INT0: On
// INT0 Mode: Falling Edge
// INT1: Off
// INT2: Off
GICR|=0x40;
MCUCR=0x02;
MCUCSR=0x00;
GIFR=0x40;

// LCD module initialization
lcd_init(16);
while (1)
{
// Place your code here
lcd_clear();
lcd_putsf("telecard reader#");//Lcd "telecard reader#"
delay_ms(500);//Waitms 500
if (flag == 0 ){
    lcd_gotoxy(0,1);//Locate 2 , 1
    lcd_putsf("<press the key!>");//Lcd "<press the key!>"
    delay_ms(500);
}
flag=~flag;
};
}

//*****
void Showresult()
{
    unsigned char si[3];
    lcd_clear();
    for(i=0;i<4;i++){
        itoa(array[i],si);
        lcd_puts(si);
    }
}

```

```

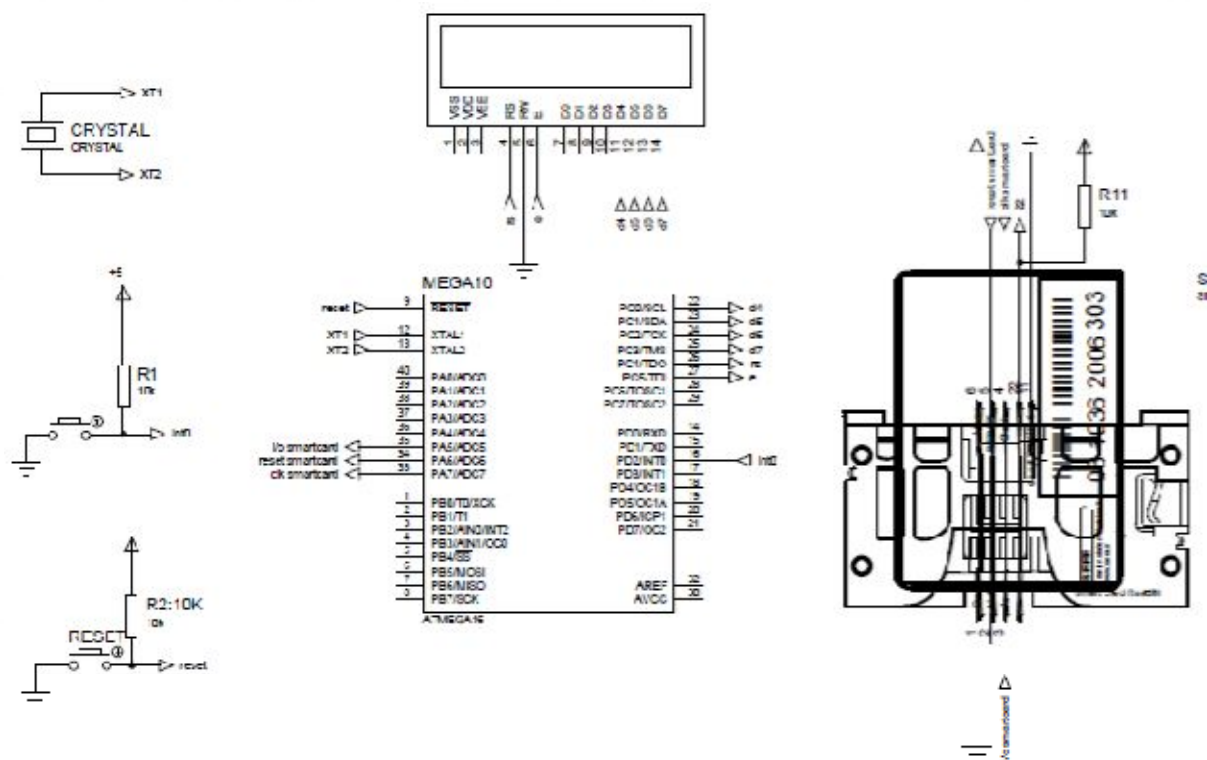
    lcd_gotoxy(0,1);//Locate 2 , 1
    for(i=4;i<8;i++){
        itoa(array[i],si);
        lcd_puts(si);
    }
    delay_ms(500);
}

//*****
void Resettelecard()
{

    Reset_telecard=1;
    Clk_telecard=1;
    delay_ms(50);
    Clk_telecard=0;
    Reset_telecard=0;
}

void Readtelecard()
{
    signed char i,j;
    DDRA.5=0; //Config Porta.5 = Input
    DDRA.6=1; //Config Porta.6 = Output
    DDRA.7=1; //Config Porta.7 = Output
    for (i=0;i<64;i++)
    {
        for (j=7;0<=j;j--)
        {
            if(Io_telecard==1){
                array[i]=1<<j;
            }
            else{
                array[i]&=~(1<<j);
            }
            Clk_telecard=1;
            delay_us(100);//Wait 10us
            Clk_telecard=0;
        }
    }
}

```



با زدن کلید متصل به اینترپت، میکرو شروع به خواندن و سپس نمایش ۸ بایت اول حافظه کارت می کند.

۲.۲.۷.۲ برنامه نویسی با bascom

مرحله اول:

قبل از هر چیز پایه های متصل به کارت را نامگذاری می کنیم تا بتوانید به راحتی کارت را به پایه های دلخواه خود متصل کنیم.

Io_telecard Alias Pina.5

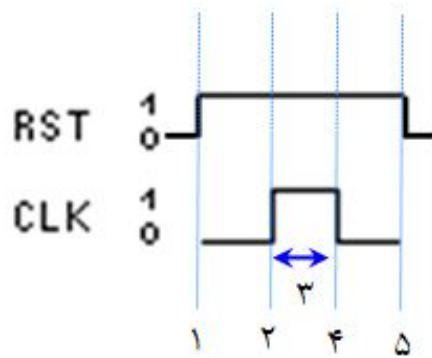
Reset_telecard Alias Porta.6

Clk_telecard Alias Porta.7

Check_insert_telecard Alias Pinc.7

مرحله دوم:

برای خواندن از کارت همان طور که گفته شد ابتدا باید کارت ریست شود. یعنی باید مطابق شکل زیر عمل کنیم

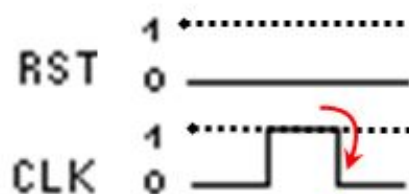


زیر برنامه ریست کارت:

Resettelecard:

Set Reset_telecard	مرحله ۱
Set Clk_telecard	مرحله ۲
Waitms 10	مرحله ۳
Reset Clk_telecard	مرحله ۴
Reset Reset_telecard	مرحله ۵

Return



در این لحظه خواهیم داشت
 $\text{address counter} = \text{address counter} + 1$

مرحله سوم:

حال نوبت به خواندن از کارت می رسد، که باید با هر بار آمدن پالس کلاک از سطح بالا به پایین بیت قرار گرفته بر روی خروجی i/o کارت، با میکرو خوانده و در متغیری ذخیره کنیم. برای این کار ابتدا یک متغیر آرایه‌ای از نوع بایت تعریف می‌کنیم. چون ۶۴ بایت داریم پس باید آرایه‌ی ما ۶۴ تایی باشد و از آنجایی که هر بایت ۸بیتی می باشد پس نوع بایت بهترین انتخاب برای نوع آرایه‌ی ما می باشد. یعنی خواهیم داشت.

Dim Array(64) As Byte

برای دسترسی به تک تک بیت‌های آرایه باید مطابق زیر عمل کنیم.

Array(i).j

i: بیت مورد نظر

j: بیت مورد نظر از بیت i

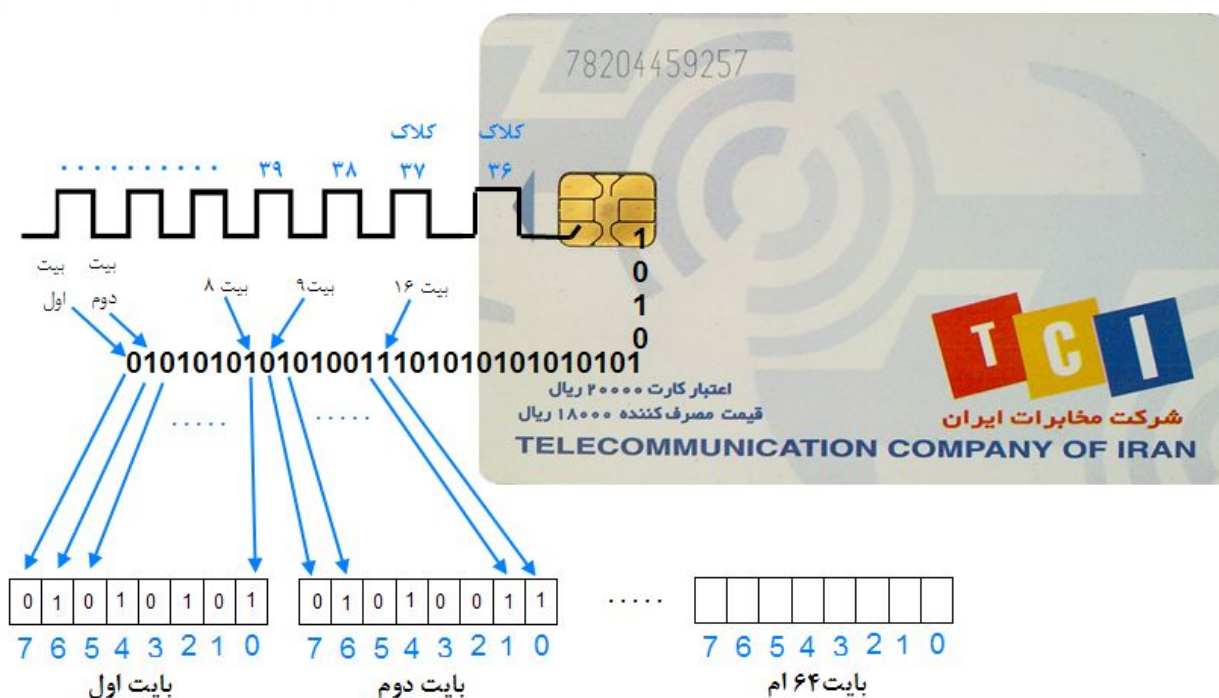
به طور مثال اگر بخواهیم بیت ۵ از بیت ۳۳ را صفر کنیم، خواهیم داشت:

Array(33).5=0

یا اگر بخواهیم بیت ۳ از بیت ۱۰ را در یک متغیر دیگر ذخیره کنیم، خواهیم داشت:

Dim temp as bit

Temp=Array(10).3



حال از دو حلقه تو در تو برای ذخیره بیت‌ها استفاده می‌کنیم، که متغیر حلقه‌ی داخلی وظیفه‌ی شمارش بیت‌های بایت مورد نظر را بر عهده دارد و حلقه بیرونی وظیفه کنترل بایت‌ها را بر عهده دارد.

For I = 1 To 64

For J = 7 To 0 Step -1

Set Clk_telecard

Waitus 100

Reset Clk_telecard

Array(i).j = Io_telecard

Next

Next

در ابتدا $i=1$ می‌باشد پس با وارد شدن به حلقه داخلی با تعداد تکرار ۸ بار (از ۰ تا ۷) ۸ پالس ایجاد شده و بیت قرار گرفته در i/o بعد از هر پالس در خانه‌ی ژام از بایت اول ذخیره می‌شود. حال بعد از ذخیره ۸ بیت اول باید بیت‌های بعدی در بایت دوم ذخیره شوند که این امر توسط حلقه‌ی بیرونی انجام می‌شود یعنی با خارج شدن از حلقه داخلی، حلقه بیرونی به i یک عدد اضافه می‌کند و مقدار i برابر با ۲ می‌شود، در این حالت برنامه دوباره وارد حلقه‌ی داخلی شده و ۸ بیت را خوانده و در خانه‌های بایت دوم ذخیره می‌کند.

این مراحل ۶۴ بار تکرار می‌شود تا تمامی بیت‌های حافظه کارت خوانده شود.

تذکره: در `bascom` شماره‌ی متغیر آرایه از یک شروع می‌شود. `array(4)` شامل ۴ متغیر از شماره‌های ۱ تا ۴ می‌باشد ولی شماره بیت‌های متغیر آرایه مانند سایر متغیرها از صفر شروع می‌شود، یعنی در خانه اول متغیر از نوع بایت، بیت صفرم قرار دارد و در خانه هشتم، بیت ۷م قرار دارد. (به همین دلیل حلقه داخلی از ۰ تا ۷ می‌باشد) زیر برنامه خواندن از کارت تلفن:

Readtelecard:

Config Porta.5 = Input

Config Porta.6 = Output

Config Porta.7 = Output

Dim I As Byte

Dim J As Byte

For I = 1 To 64

For J = 7 To 0 Step -1

`Array(i).j = Io_telecard`

Set Clk_telecard

کمتر از این مقدار هم می‌توانید استفاده کنید // 100 **Waitus**

Reset Clk_telecard

Next

Next

Return

تا این مرحله حافظه خوانده شد و در داخل متغیر `array` در دسترس می‌باشد. می‌توانید متناسب با پروژه خود از آن استفاده کنید.

پروژه ۱) `(bascom)`:

خواندن محتوای کارت تلفن و نمایش ۸ بایت اول بر روی `GLCD`. همان طور که در شکل مشاهده می‌کنید مقادیر دسیمال بایت‌ها بر روی `LCD` نشان داده شده‌اند.



```

$regfile = "m32def.dat"
$crystal = 8000000
$lib "glcdKS108.lib"
#include "font8x8.font"
*****

'Declare your variables here
Dim X_coordinate As Word
Dim Y_coordinate As Word
Dim Array(64) As Byte
*****

'Config Adc
Config Adc = Single , Prescaler = Auto , Reference = Avcc
*****

Config Porta.5 = Input
Config Porta.6 = Output
Config Porta.7 = Output
Config Portc.7 = Input

Io_telecard Alias Pina.5
Reset_telecard Alias Porta.6
Clk_telecard Alias Porta.7
Check_insert_telecard Alias Pinc.7
*****

'config glcd
Config Graphlcd = 128 * 64sed , Dataport = Portd , Controlport = Portc , Ce1 = 4 , Ce2 = 3 , Cd = 0 , Rd = 1 , Enable = 2 ,
Reset = 5
SetFont Font8x8

Do
Gosub Readtouch
If X_coordinate <> 0 And Y_coordinate <> 0 Then
    Gosub Resettelecard

```

```

    Gosub Readtelecard
    Gosub Showresult
End If

```

```

Loop
End                                'end program

```

Readtouch:

```

    While X_coordinate = 0 Or Y_coordinate = 0      ' montazer lams shodan safheye touch

*****'
*****' y coordinate*****
    Porta = &B01100001
    Ddra = &B11100101
    Y_coordinate = Getadc(1(
*****' x coordinate*****
    Porta = &B11000010
    Ddra = &B11101010
    X_coordinate = Getadc(0(
*****'

    Wend

    Waitms 400

*****'
*****' y coordinate*****
    Porta = &B01100001
    Ddra = &B11100101
    Y_coordinate = Getadc(1(
*****' x coordinate*****
    Porta = &B11000010
    Ddra = &B11101010
    X_coordinate = Getadc(0(
*****'

```

```

Return

```

Readtelecard:

```

    Config Porta.5 = Input
    Config Porta.6 = Output
    Config Porta.7 = Output
    Dim I As Byte
    Dim J As Byte

    For I = 1 To 64
        For J = 7 To 0 Step -1
            Array(i).j = Io_telecard
            Set Clk_telecard
            Waitus 10
            Reset Clk_telecard
        Next
    Next
Return

```

Showresult:

```

Cls
Lcdat 1 , 1 , "byte1=" ; Array(1)
Lcdat 2 , 1 , "byte2=" ; Array(2)
Lcdat 3 , 1 , "byte3=" ; Array(3)
Lcdat 4 , 1 , "byte4=" ; Array(4)
Lcdat 5 , 1 , "byte5=" ; Array(5)
Lcdat 6 , 1 , "byte6=" ; Array(6)
Lcdat 7 , 1 , "byte7=" ; Array(7)
Lcdat 8 , 1 , "byte8=" ; Array(8)
Wait 1
Return

```

Resettelecard:

```

Set Reset_telecard
Set Clk_telecard
Waitms 10
Reset Clk_telecard
Reset Reset_telecard
Return

```

پروژه ۲ (BASCOM):

همان پروژه قبلی است با این تفاوت که مقدار نمایش داده شده بر روی LCD بر حسب هگز می باشد. پس در بر نامه قبلی کافی است زیر برنامه Showresult به صورت زیر تغییر یابد.

Showresult:

```

Cls
Lcdat 1 , 1 , "byte1=" ; Hex(array(1))
Lcdat 2 , 1 , "byte2=" ; Hex(array(2))
Lcdat 3 , 1 , "byte3=" ; Hex(array(3))
Lcdat 4 , 1 , "byte4=" ; Hex(array(4))
Lcdat 5 , 1 , "byte5=" ; Hex(array(5))
Lcdat 6 , 1 , "byte6=" ; Hex(array(6))
Lcdat 7 , 1 , "byte7=" ; Hex(array(7))
Lcdat 8 , 1 , "byte8=" ; Hex(array(8))
Wait 1
Return

```



پروژه ۳ (BASCOM):

همان پروژه قبلی با نمایش کل محتویات بر روی LCD، پس برنامه این پروژه نیز فقط در زیر برنامه Showresult با دو برنامه قبل متفاوت است:

*****'

Showresult:

Cls

Dim N As Byte

For I = 1 To 8

For J = 1 To 128 Step 16

N = N + 1

Lcdat I, J, Hex(array(n))

Next

Next

N = 0

Wait 1

Return

*****'



پروژه ۴ (BASCOM) :

خواندن محتوای کارت تلفن و نمایش ۸ بیت اول بر روی LCD کاراکتری ۱۶در۲.

```
$regfile = "m16def.dat"
$crystal = 8000000

Config Lcd = 16 * 2
Config Lcdpin = Pin , Db4 = Pinc.0 , Db5 = Pinc.1 , Db6 = Pinc.2 , Db7 = Pinc.3 , Rs = Pinc.4 , E = Pinc.5
Enable Interrupts
Config Int0 = Falling
Enable Int0
On Int0 Key1
*****'
'Declare your variables here
Dim Array(64) As Byte
Dim Flag As Bit

Config Porta.5 = Input
Config Porta.6 = Output
Config Porta.7 = Output
Config Portc.7 = Input

Io_telecard Alias Pina.5
Reset_telecard Alias Porta.6
Clk_telecard Alias Porta.7
Check_insert_telecard Alias Pinc.7

'start program
Do
Cls
Lcd "telecard reader"#
Waitms 500
If Flag = 0 Then
    Locate 2 , 1
    Lcd "<press the key"<!
    Waitms 500
End If
Toggle Flag
Loop
End 'end program
*****'

Key1:
Do
    Gosub Resetelecard
    Gosub Readtelecard
    Gosub Showresult
Loop
*****'

'read telecard's eeprom:read all of bit's & save to 64byte arraye
Readtelecard:
    Config Porta.5 = Input
    Config Porta.6 = Output
```

```

Config Porta.7 = Output
Dim I As Byte
Dim J As Byte

```

```

For I = 1 To 64
  For J = 7 To 0 Step -1
    Array(i).j = Io_telecard
    Set Clk_telecard
    Waitms 10
    Reset Clk_telecard
  Next
Next
Return

```

Showresult:

Cls

Lcd Array(1) ; Array(2) ; Array(3) ; Array(4)

Lowerline

Lcd Array(5) ; Array(6) ; Array(7) ; Array(8)

Waitms 500

Return

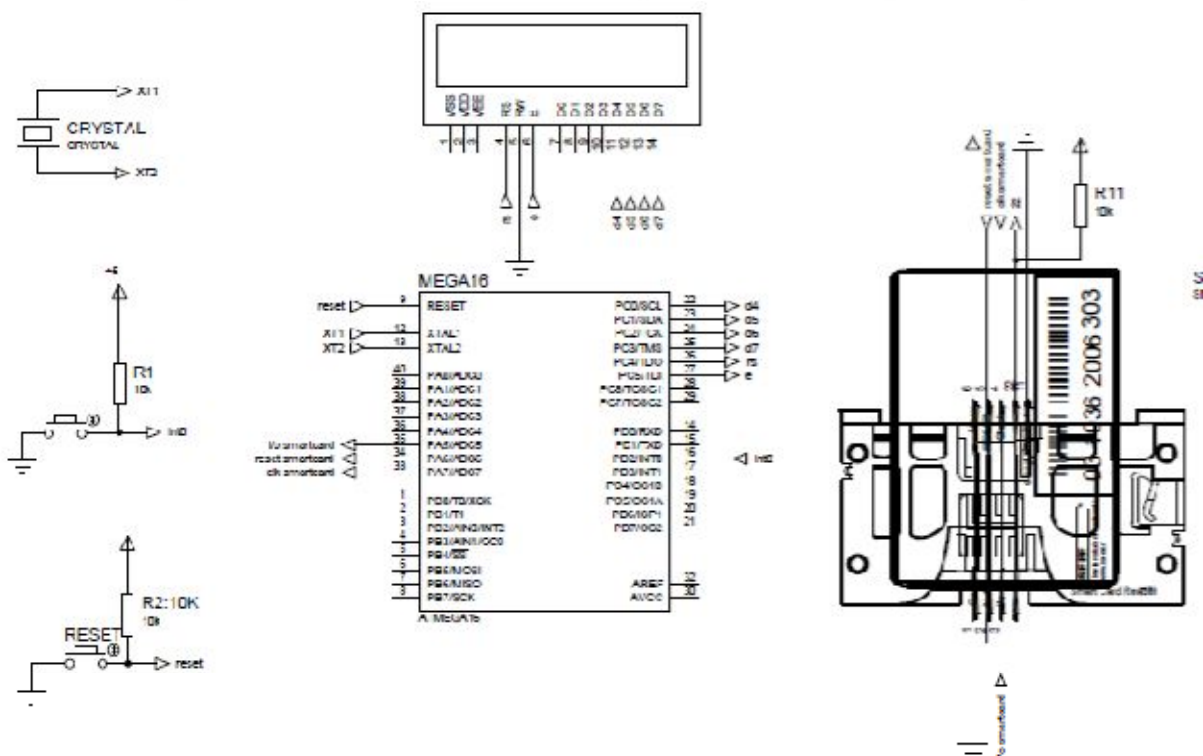
Resettelecard:

```

Set Reset_telecard
Set Clk_telecard
Waitms 100
Reset Clk_telecard
Reset Reset_telecard

```

Return



با زدن کلید متصل به اینترپت، میکرو شروع به خواندن و سپس نمایش ۸ بایت اول حافظه کارت می کند.

۳.۷.۳ محاسبه مبلغ کارت تلفن

همان طور در قسمت ۵.۲ گفته شد:

"معمولا ۴ بایت به اعتبار کارت اختصاص می یابد که شامل بایتهای ۱۰ و ۱۱ و ۱۲ و ۱۳ است برای کاهش یک واحد از اعتبار، یکی از بیتهای بایت ۱۳ از مقدار 1 به 0 تبدیل می شود در نتیجه بعد از ۸ بار کاهش اعتبار، مقدار بایت ۱۳ برابر 0 می شود و برای کاهش اعتبار کارت باید یک بیت از بایت قبلی (۱۲) به 0 تبدیل شود و در مقابل ۸ بیت بایت ۱۳ به ۱ مبدل شود به زبان ساده هر یک واحد (بیت و یا خانه) از بایت ۱۲ برابر با ۸ واحد از بایت ۱۳ و همینطور هر واحد از بایت ۱۱ برابر با ۸ واحد بایت ۱۲ به این ترتیب بایت ۱۳ برابر ۸ واحد و بایت ۱۲ برابر ۸*۸ واحد و بایت ۱۱ برابر ۸*۸*۸ واحد و بایت 10 برابر ۸*۸*۸*۸ واحد که مجموعا ۴۶۸۰ واحد حاصل می شود

همانطور که گفته شد بعد از ۸ بار کاهش اعتبار، همه خانه های (بیتهای) بایت 13 برابر 0 می شود در این شرایط با استفاده از WRITECARRY یک بیت از بایت مورد نظر (۱۲) را پاک (WRITE) می کنیم و در مقابل ۸ بیت بایت بعدی (۱۳) به ۱ تبدیل می شوند و اکنون می توان یک واحد (بیت) از بایت ۱۳ را بمنظور کاهش اعتبار به 0 تبدیل کرد (WRITE) و این در مورد بایتهای ۱۱ و ۱۰ نیز صادق است."

فرض کنید بایت های ۱۰ تا ۱۳ خوانده شده از کارت به قرار زیر باشد. (بایت ۹م رو در نظر نگیرید)

	Byte 9	Byte 10	Byte 11	Byte 12	Byte 13
۱ مقدار هگز	00	07	0F	BF	7F
۲ مقدار باینری	0000 0000	0000 0111	0000 1111	1011 1111	0111 1111
۳ تعداد بیت هایی که برابر یک می باشد	0	3	4	7	7
۴ میزان ارزش هر بیت	$8^4=4096$	$8^3=512$	$8^2=64$	$8^1=8$	$8^0=1$
۵ ارزش بایت مربوطه	$0*4096=0$	$3*512=1536$	$4*64=256$	$7*8=56$	$7*1=7$
مجموع	1855				

همان طور که می بینید بیت ها با قرار گرفتن در بایت های مختلف ارزش های مختلفی می گیرند مثلا هر بیت در بایت ۱۰ برابر با ۵۱۲ واحد ارزش می باشد در حالی که هر بیت از بایت ۱۲ تنها دارای ۸ واحد ارزش می باشد. توجه داشته باشید که تعداد بیت های 1 موجود در هر یک از بایت ها میزان ارزش آن بایت را مشخص می کند نه محل قرار گرفتن بیت ها در آن، به طور مثال در بایت ۱۱ چون چهار بیت یک داریم و هر بیت ارزشی برابر ۶۴ واحد دارد پس بایت ۱۱ در کل $۶۴*۴$ یعنی ۲۵۶ واحد ارزش دارد.

حال با توجه به مقدمات فوق محاسبه مبلغ موجود در کارت را با برنامه نویسی پیاده می کنیم.

ابتدا باید تعداد بیت‌های ۱ هر یک از بایت‌ها شمرده شود (محاسبه سطر سوم جدول)، برای این منظور یک متغیر از نوع آرایه چهارتایی با نام Sum_array_bit تعریف می‌کنیم، سپس از دو حلقه تو در تو به فرم زیر استفاده کرده تا تعداد بیت‌ها را در متغیر مربوطه ذخیره کنیم

```
For I = 10 To 13
  Sum_array_bit(I - 9) = 0
  For J = 0 To 7
    If Array(i).j = 1 Then Incr Sum_array_bit(I - 9)
  Next
Next
```

سپس با دستورات زیر سطرهای ۴ و ۵ جدول را پیاده کرده تا مجموع ارزش کارت محاسبه شود.

```
Temp = Sum_array_bit(1) * 512
Total_value = Total_value + Temp
Temp = Sum_array_bit(2) * 64
Total_value = Total_value + Temp
Temp = Sum_array_bit(3) * 8
Total_value = Total_value + Temp
Temp = Sum_array_bit(4) * 1
Total_value = Total_value + Temp
```

حال با یک تناسب ساده می‌توانید مقدار محاسبه شده که بدون واحد می‌باشد را به واحد‌های ریال یا تومان تبدیل کنید. به این صورت که از تعدادی کارت استفاده می‌کنیم که از قبل از مبلغ آنها آگاهی داریم (با استفاده از بجه-های تلفن)، سپس اعداد بدست آمده از خواندن کارتها را با مبلغ واقعی آنها مقایسه نموده تا رابطه‌ی بین آنها را به دست آوریم.

به طور مثال ما یک کارت حاوی ۱۱۵۰ ریال و یکی حاوی ۴۰۰۰ ریال را خواندیم، مقدار خوانده شده از کارت اول ۲۳ بدست آمد و برای کارت دوم عدد ۸۰ خوانده شد.

با تقسیم مبالغ کارت بر عدد خوانده شده از آنها هر دو به عدد ۵۰ منجر شد. این یعنی برای محاسبه مبلغ از روی عدد خوانده شده کفایت عدد خوانده شده را در ۵۰ ضرب کنیم. یعنی:

```
Total_value = Total_value * 50
```

پس یک زیر برنامه به صورت زیر برای محاسبه مبلغ می‌نویسیم.

```
Calculate_value:
  Total_value = 0
  For I = 10 To 13
    Sum_array_bit(i - 9) = 0
    For J = 0 To 7
      If Array(i).j = 1 Then Incr Sum_array_bit(i - 9)
    Next
  Next
  Temp = Sum_array_bit(1) * 512
  Total_value = Total_value + Temp
  Temp = Sum_array_bit(2) * 64
  Total_value = Total_value + Temp
  Temp = Sum_array_bit(3) * 8
  Total_value = Total_value + Temp
  Temp = Sum_array_bit(4) * 1
  Total_value = Total_value + Temp

  Total_value = Total_value * 50
```

Return

برای اجتناب از افزایش تعداد صفحات مقاله تنها زیر برنامه آورده شده، سورس کامل برنامه پیوست مقاله می باشد.

توجه: (سورس ندارد) می توانید همین روال را با codevision هم انجام دهید. به طور مثال برای شمارش بیتها دو راه دارید، اول این که بایت مربوطه را عملگرهای بیتی (بولی) مورد تحلیل قرار داده تا تعداد بیتها مشخص شود، دوم این که به صورت زیر عمل کنید:

```
for (i = 10 ; 13<I;i++)
{
    Sum_array_bit[I -9] = 0;
    for (j = 0 ; 7>=j;j++)
    {
        If (Array[i]&(1<<j)) Sum_array_bit[i -9]++;
    }
}
```

برای محاسبات هم می تونید در ساده ترین حالت از فرم زیر استفاده کنید:

Total_value=(Sum_array_bit[1]*512+ Sum_array_bit[2]*64+ Sum_array_bit[3]*8+ Sum_array_bit[4])*50

حالا به سلیقه ی خودتان می توانید از این مقاله در پروژه هایی مثل قفل رمز و کنترل عبور و مرور و یا حتی سیستم حضور و غیاب برای ادارات نیز استفاده کنید.

توجه: پروژه ی قفل رمزی با کارت تلفن به زبان basic توسط آقای حسنی پژو ساخته شده (جهت استفاده برای درب ورودی منزل)، و در سایت iseee در دسترس می باشد.



از علی:

در گذشته مرا برادری بود که در راه خدا برادریم می نمود

خردی دنیا در دیده اش وی را در چشم من بزرگ می داشت ، و شکم بر او سلطه ای نداشت ، پس آنچه نمی یافت آرزو نمی کرد و آنچه را می یافت فراوان به کار نمی برد . بیشتر روزهایش را خاموش می ماند ، و اگر سخن می گفت گویندگان را از سخن می ماند و تشنگی پرسندگان را فرو می نشاند . افتاده بود و در دیده ها ناتوان ، و به هنگام کار چون شیر بیشه و مار بیابان . تا نزد قاضی نمی رفت حجت نمی آورد و کسی را که عذری داشت . سرزنش نمی نمود ، تا عذرش را می شنود . از درد شکوه نمی نمود مگر آنگاه که بهبود یافته بود . آنچه را می کرد می گفت و بدانچه نمی کرد دهان نمی گشود . اگر با او جدال می کردند خاموشی می گزید و اگر در گفتار بر او پیروز می شدند ، در خاموشی مغلوب نمی گردید . بر آنچه می شنود حریصتر بود تا آنچه گوید ، و گاهی که او را دو کار پیش می آمد می نگریست که کدام به خواهش نفس نزدیکتر است تا راه مخالف آن را پوید.

بر شما باد چنین خصلتها را یافتن و در به دست آوردنش بر یکدیگر پیشی گرفتن . و اگر نتوانستید ، بدانید که اندک را به دست آوردن بهتر تا همه را واگذارن .

نهج البلاغه

موفق و پیروز باشید!

گروه کیان مهر

www.iseee.ir