

www.MELEC.ir

پایان نامه کارشناسی مهندسی برق

گرایش الکترونیک

عنوان پروژه:

طراحی و ساخت سیستم مانیتورینگ دما و فشار داخل تایر

اتومبیل (TPMS)

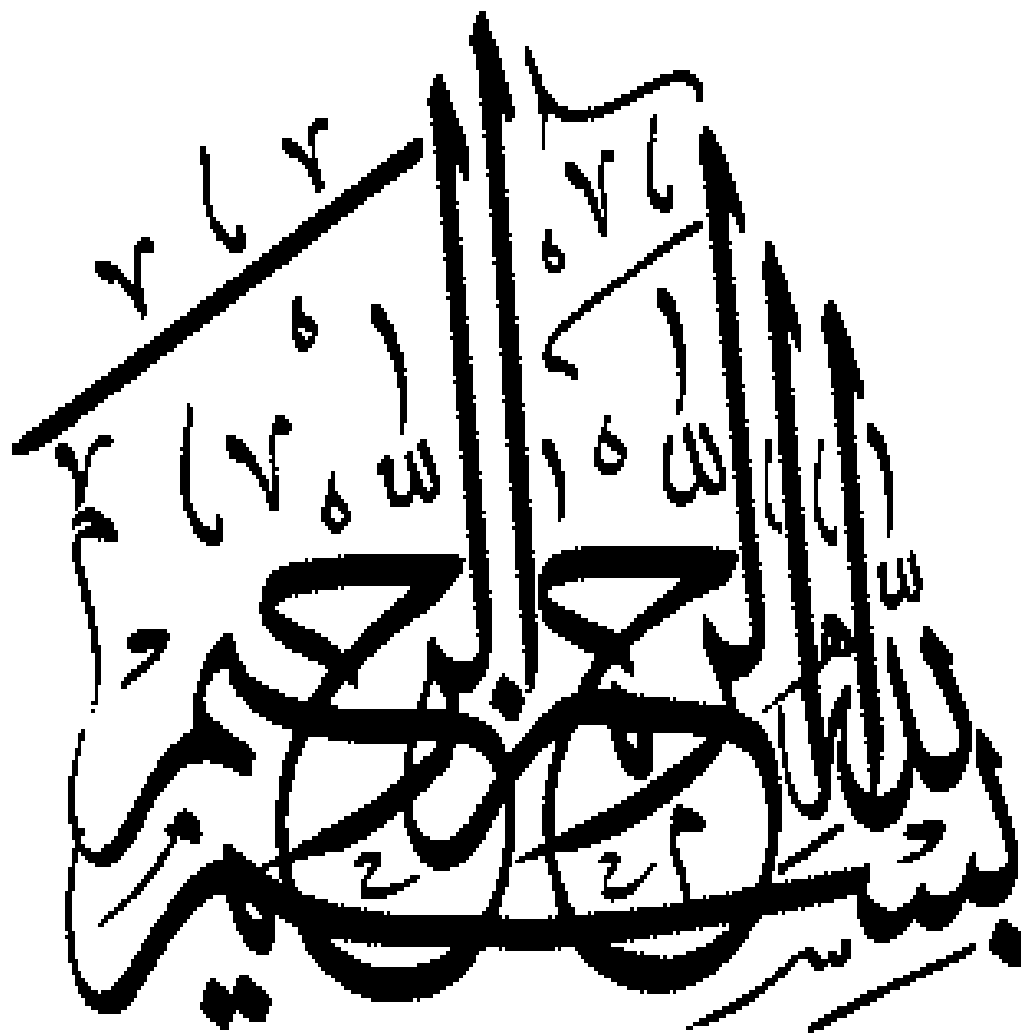
دانشجویان:

هادی آقابیگی - علیرضا سودی رضائی

بهار ۱۳۹۱

Melec.ir

www.MELEC.ir



Melec.ir

فهرست مطالب

صفحه	عنوان
۳	چکیده
۵	مقدمه

فصل اول : فشار هوا یا فشار جو

۷	۱-۱) فشار چیست؟
۷	۱-۲) واحدهای اندازه گیری فشار
۸	۱-۳) انواع فشار

فصل دوم : آشنایی با سنسور *TPMS* سری *MPXY8000*

۹	۲-۱) معرفی سنسور
۱۲	۲-۲) مدهای کاری
۱۴	۲-۳) عملکرد پایه های سنسور
۱۸	۲-۴) تنظیم آستانه خروجی
۲۰	۲-۵) خروجی سنسور فشار
۲۰	۲-۶) خروجی سنسور دما
۲۱	۲-۷) کاربردها

فصل سوم : آشنایی با بخش های نرم افزار و سخت افزار سیستم طراحی شده

۲۲	۳-۱) نحوه عملکرد <i>MCU</i> و ارتباط آن با سنسور
۲۵	۳-۲) روتین تقریب متوالی (<i>SAR</i>)

۲۷ نحوه دریافت و ارسال اطلاعات
۲۸ (۳-۴) بلوک دیاگرام کلی سیستم طراحی شده
۲۸ (۳-۵) مدار فرستنده
۲۸ (۳-۶) مدار گیرنده
۲۸ (۳-۷) نرم افزار های مربوطه

فصل چهارم : نتیجه گیری و پیشنهادات

۳۳ نتیجه گیری و پیشنهادات
----	------------------------------

فصل پنجم : مراجع و منابع

۳۴ مراجع و منابع
----	---------------------

فصل ششم : پیوست ها

۳۵ پیوست شماره (۱) : نقشه PCB مدار گیرنده
۳۶ پیوست شماره (۲) : نقشه PCB مدار فرستنده
۳۷ پیوست شماره (۳) : فهرست جداول
۳۸ پیوست شماره (۴) : فهرست شکل ها

در ابتدا بر خود لازم میدانیم که از زحمات و دلسوزی های همه اساتید محترم دانشکده فنی شهید شمسی
پور تهران صمیمانه تقدیر و تشکر کنیم. و نیز فرصت را مغتنم شمرده و از زحمات بی شائبه استاد
بزرگوارمان، جناب آقای مهندس یدالله رضازاده، که بی شک میتوان از ایشان به عنوان الگویی در علم و ادب
مشق گرفت، تشکر و قدردانی کنیم.

از خداوند منان طلب بهروزی و توفیق را برای همه این عزیزان خواستاریم.

هادی آقاییگی

علیرضا سودی رضایی

www.MELEC.ir

چکیده

بازبینی و نگهداری فشار لاستیک در اندازه استاندارد، امنیت راننده و سرنشینان خودرو را افزایش می دهد و نیز باعث بهبود کارائی وسیله نقلیه در جاده ها می باشد.

مسئله تنظیم نبودن فشار لاستیک های اتومبیل، به خصوص در مسافرت های طولانی همواره برای رانندگان و سرنشینان خودروها مشکل ساز بوده است. حتی در مواردی، تغییرات زیاد فشار تایر، باعث ترکیدگی لاستیک خودرو و در مواردی، مرگ سرنشینان خودرو شده است. بررسی به موقع فشار لاستیک های خودرو، علاوه بر کاهش مصرف سوخت، باعث بهبود عملکرد ترمز و کاهش مسافت خط ترمز و نیز باعث افزایش عمر لاستیک خودرو می گردد.

در سال های اخیر تلاشهای گسترده ای برای حل این مشکل انجام گرفته است و بسیاری از شرکت های بزرگ دنیا به طراحی و ساخت مدارات مجتمع مخصوص این کار (*Tire Pressure Monitoring System*) دست زده اند، که از موفقترین این شرکت ها میتوان شرکت *Motorola* را نام برد.

در این پژوهش ما توسط میکروکنترلر شرکت *Atmel* و سنسور شرکت *Motorola* و یک ماژول فرستنده و گیرنده باند *UHF* ، مداری را طراحی نموده ایم که در داخل تایر ماشین قرار می گیرد و در زمان روشن بودن اتومبیل، اطلاعات مربوط به فشار و دمای داخل تایر را بدست می آورد و از طریق فرستنده، کدهای حاصله را ارسال می کند. در سوی دیگر مداری را قرار داده ایم که به عنوان مدار *Master* در داخل کابین اتومبیل قرار می گیرد که اطلاعات فرستاده را دریافت نموده و پس از عملیتهای ریاضی که در داخل پردازنده آن انجام میشود مقدار حقیقی فشار و دمای هر کدام از تایرها، نمایش داده می شوند.

اگر فشار و دما از مقدار استاندارد تجاوز کنند و یا اینکه فشار از مقدار استاندارد تعریف شده کمتر شود، این سیستم به راننده هشدار می دهد که تا نسبت به تنظیم فشار لاستیک اقدامات لازم را انجام دهد.

در همایشی که در سال ۲۰۰۰ میلادی برگزار شد، تصمیم بر این گرفته شد که وسایل نقلیه مجهز به سیستم *TPMS* شوند. که در سال ۲۰۰۶، بیست درصد از خودروها به این سیستم مجهز شدند. این تعداد در سال ۲۰۰۷ به هفتاد درصد و در نهایت در سال ۲۰۰۸ کلیه وسایل نقلیه مجهز به سیستم *TPMS* شدند.

وظیفه یک *TPMS* اندازه گیری مداوم دما و فشار تایر اتومبیل می باشد.

به طور کلی دو روش کنترل فشار تایر خودرو وجود دارد که عبارتند از: *TPMS* مستقیم و *TPMS* غیر مستقیم.

در نوع مستقیم از یک سنسور که در چرخ نصب شده استفاده می شود که این سنسور داده ها را به کامپیوتر موجود در خودرو انتقال می دهد. این سنسور در نوع مستقیم، فشار هوای هر لاستیک را می سنجد و زمانی که فشار هوای داخل لاستیک، ۵٪ پائین تر از سطح تعریف شده کارخانه شود، سیگنال هشدار به کابین خودرو منتقل می شود که ممکن است آلامی به صدا درآید و یا اینکه چراغ هشدار مخصوص این سیستم که در داخل کابین تعبیه شده، روشن گردد.

در روش *TPMS* غیر مستقیم، اصولاً فشار تایر اندازه گیری نمی شود؛ بلکه سرعت چرخش هرکدام از چرخ های خودرو مانیتور می شود و نظر به اینکه اگر فشار یک چرخ کم شود، قطر موثر آن نیز کاهش می یابد؛ در نتیجه سرعت چرخش آن چرخ (که فشار کمتری دارد) نسبت به سایر چرخ ها زیاد می شود و از این طریق، آلام هشدار کم بودن فشار، به صدا درخواهد آمد و راننده را از این موضوع با اطلاع می سازد. البته لازم به ذکر است که سیستم های غیر مستقیم، کارایی بالایی ندارند. چون در صورتیکه تمام لاستیک ها به یک مقدار کاهش فشار داشته باشند، دیگر این سیستم قادر به تشخیص این مشکل نبوده و عملاً هشدار نمی دهد.

در سوی دیگر، سیستم های *TPMS* مستقیم می توانند فشار همه لاستیک ها را به طور دقیق مانیتور کنند.

سیستم های مستقیم دارای یک گیرنده *UHF* در داخل کابین خودرو و چهار ماژول (در خودروهای سواری)، در داخل هر یک از چرخ های خودرو هستند. هر یک از این ماژول ها دارای یک سنسور فشار و یک فرستنده *UHF* هستند که به واحد کنترل (میکرو کنترلر) ماژول متصل می باشند.

قطعات بکار رفته در *TPMS* توانایی کار تا دمای حداکثر ۱۲۵ درجه سلسیوس را دارند و با جریان بسیار کمی کار می کنند، بطوریکه تا حدود ده سال نیازی به تعویض باتری ندارند.

روشی که ما در پروژه خود بکار گرفته ایم، روش *TPMS* مستقیم می باشد که عملکرد کلی آن، به اختصار توضیح داده شد. و در مراحل بعدی، نحوه عملکرد سنسور مخصوص *TPMS* و همچنین عملکرد برنامه بصورت فلو چارت و مدار طراحی شده توسط نرم افزار *Altium Designer DXP* توضیح داده شده است.

لازم به ذکر است به دلیل آنکه امروزه، زبان های برنامه نویسی فراوانی وجود دارد و به علت این تنوع، کامپایلرهای گوناگونی نیز جهت کد نویسی وجود دارند و نیز هر روز بر تعداد این کامپایلرها اضافه می شود، و دستورات هر کامپایلری، تقریباً با کامپایلرهای دیگر متفاوت است، سعی بر آن نهادیم که از یک زبان خاص جهت توضیح توابع بکار رفته شده در بدنه اصلی برنامه، استفاده نکنیم و جهت عمومی شدن این توضیحات، از روش ترسیم فلوچارت استفاده کرده ایم.

فصل اول: فشار هوا یا فشار جو:

(۱-۱) فشار چیست؟

فشار کمیتی اسکالر است و برابر با نیروئی است که در هر نقطه بر حسب وزن ستونی از هوا که در بالای آن نقطه وجود دارد، بر سطح وارد می‌شود. این فشار در سطح آبها و دریاهای آزاد در حدود ۱ کیلوگرم بر سانتی‌متر مربع (1 kg/cm^2) است. با افزایش ارتفاع از سطح زمین، به دلیل کاسته شدن ارتفاع ستون هوای قرار گرفته در بالای سطح، فشار جو کاهش می‌یابد.

$$p = \frac{F}{A} \quad \text{یا} \quad p = \frac{dF_n}{dA}$$

فشار هوای گرم در ظرف در بسته بیشتر است زیرا با افزایش دما، جنبش مولکولی افزایش یافته و برخوردها با دیواره ظرف بیشتر می‌شود. (باید به این نکته توجه داشت که این حالت در ظرفی در بسته و در حجم ثابت است) اما فشار هوا وقتی دما بیشتر شود، کاهش می‌یابد. زیرا با افزایش دما تراکم مولکولی کم و تعداد ضربات کاهش می‌یابد.

(۲) واحدهای اندازه گیری فشار:

واحدهای مختلفی برای اندازه گیری فشار هوا وجود دارد، اما واحد فشار در سیستم بین المللی یکاها (SI) پاسکال می‌باشد و با نماد (pa) نشان می‌دهند. لازم به ذکر است که فشار هوای سطح دریا برابر با 100 Kpa می‌باشد. در جدول شماره (۱) چند واحد مهم فشار و رابطه آنها با یکدیگر آمده است:

	Pascal	Bar	Pound Per Square Inch
	pa	bar	psi
1 pa	1	10^{-5}	145.04×10^{-6}
1 bar	10^5	1	145037744
1 psi	6.845×10^3	68.948×10^{-3}	1

جدول شماره (۱): رابطه واحدهای مختلف فشار

(۳-۱) انواع فشار:

فشار بر دو گونه نسبی و مطلق می باشد:

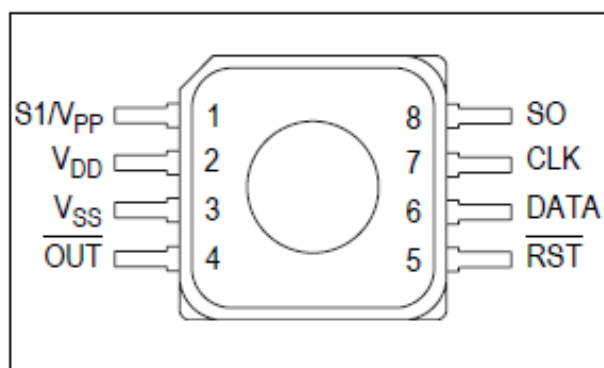
در اندازه گیری فشار، هرگاه مبدا اندازه گیری صفر باشد، این فشار را فشار مطلق گویند (psiA) که در آن، A معرف Absolute (مطلق) می باشد و اگر مبدا اندازه گیری، فشار هوای محلی باشد، این فشار را فشار نسبی (psiG) گویند که در آن، G معرف Gauge (پیمانه) می باشد. بدیهی است که پیمانه های اندازه گیری مرسوم باید فشار نسبی را اندازه گیری کنند.

www.MELEC.ir

فصل دوم: آشنایی با سنسور *TPMS* سری *MPXY8000*

(۱-۲) معرفی:

سنسور نیمه هادی سری *MPXY8000* یک سنسور هشت پین، جهت مونیتورینگ فشار میباشد که از یک المان ظرفیت خازنی جهت اندازه گیری فشار، یک المان جهت اندازه گیری دما و یک مدار واسط (با قابلیت *Wake – Up*) که همگی آنها بر روی یک *Chip* با پکیج *SSOP* قرار گرفته اند، که شامل یک فیلتر حفاظت میانی می باشند. شکل ظاهری و پایه های این سنسور در شکل (۱) نشان داده شده است.



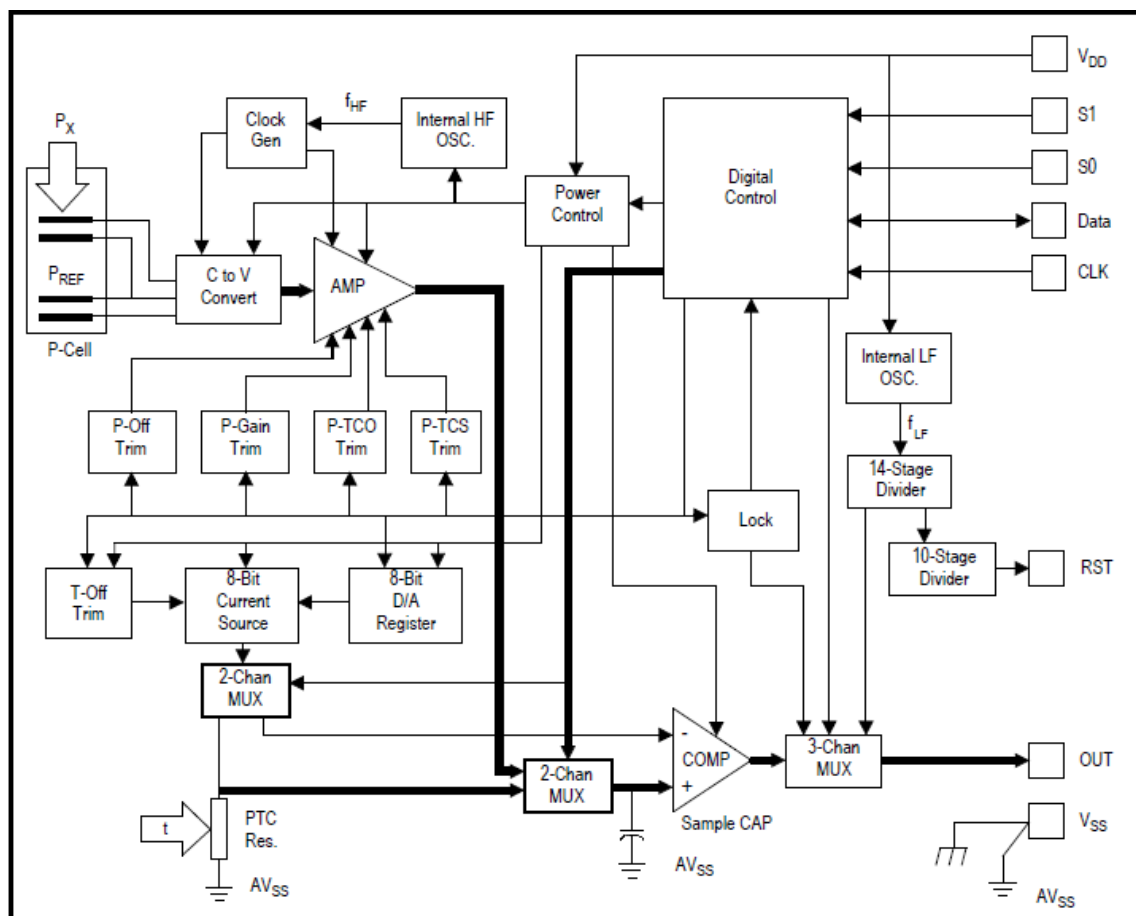
شکل (۱): شکل ظاهری و پایه های این سنسور *MPXY8020A*

بلوک دیاگرام سنسور سری *MPXY8000* در شکل (۲) نشان داده شده است.

سنسور فشار یک مبدل (*Transducer*) خازنی است و جهت *Surface Micromachining* ساخته شده است. سنسور دما از یک مقاومت تشعشعی (*Diffused*) ساخته شده است و نیز مدار واسط بر روی یکی از کعبه های سنسور قرار گرفته است که در ساخت آن از یک سیلیکون استاندارد با تکنولوژی *CMOS* استفاده شده است.

پردازش سیگنال فشار با یک مبدل خازنی به ولتاژ که در ادامه آن یک تقویت کننده خازنی سوئیچ شده قرار دارد، آغاز می شود. این تقویت کننده دارای افست و گین متوازن قابل تنظیم می باشد. افست و گین بوسیله مقادیر کالیبراسیون که توسط کارخانه سازنده در *EEPROM Trim Register* ذخیره شده

است، کالیبره می شود. همچنین این تقویت کننده دارای مدار جبران ساز دمائی (*Temperatuer Compensation*) برای حساسیت و افست است که توسط کارخانه سازنده تنظیم شده و در *EEPROM Trim Register* ذخیره شده است.



شکل (۲): بلوک دیاگرام داخلی سنسور MPXY8020A

فشار توسط یک مقایسه گر ولتاژ مونیتور می گردد که مقدار فشار را با یک آستانه λ (Threshold) بیتی که توسط ورودی سریال قابل تنظیم است، مقایسه می شود.

با تنظیم نمودن آستانه و بازبینی وضعیت پایه OUT ، وسیله خارجی میتواند چک کند که آیا یک آستانه کم فشار حاصل شده است و یا اینکه تبدیل آنالوگ به دیجیتال (A/D) بالاتر از هشت بیت صورت گرفته است.

همانگونه که در بلوک دیاگرام سنسور پیداست، چون خازن نمونه گیر به پایه غیرمعکوسگر مقایسه کننده ولتاژ متصل شده است و پایه معکوسگر مقایسه کننده به شیفتر رجیستر D/A وصل است؛ در صورتیکه مقدار ولتاژ ذخیره شده در خازن بیشتر از مقدار آنالوگ - متناظر با کد دیجیتال موجود در شیفتر رجیستر D/A - باشد، خروجی پایه OUT برابر با یک منطقی خواهد شد و در صورتیکه مقدار ولتاژ ذخیره شده در خازن کمتر از مقدار آنالوگ - متناظر با کد دیجیتال موجود در شیفتر رجیستر D/A - باشد، خروجی پایه OUT برابر با صفر منطقی می شود.

دما توسط یک مقاومت تشعشی (*Diffused*) با ضریب حرارتی مثبت اندازه گیری می شود و پس از آن به یک منبع جریان اعمال می شود که در نتیجه آن یک ولتاژی ایجاد خواهد شد. مقدار این ولتاژ در دمای اطاق توسط کارخانه سازنده در *EEPROM Trim Register* کالیبره شده است.

یک *Multiplexer* دو کاناله میتواند تعیین کند که سیگنال فشار یا سیگنال دما به خازن نمونه گیری (که این مقدار توسط مقایسه گر ولتاژ با میزان آستانه متغیر مقایسه می شود و بدین صورت مقدار سیگنال روی خازن نمونه گیری مونیتور می شود) برسد تا یک خروجی دیجیتال برای دما حاصل شود.

یک اسیلاتور داخلی فرکانس پائین و کم توان با فرکانس 5.4 KHz به همراه یک تقسیم کننده ۱۴ طبقه ($2^{14} = 16384$)، یک پالس متناوب را برای پین OUT مهیا می کنند. این پالس میتواند جهت Wake – Up یک *MCU* خارجی مورد استفاده قرار گیرد که بدین صورت *Interface* سنسور شروع به کار می کند.

یک مقسم ۱۰ طبقه دیگر ($2^{10} = 1024$)، هر ۵۲ دقیقه، یک پالسی را تولید میکند که برای *Reset* کردن *MCU* خارجی مورد استفاده قرار گیرد.

توان مصرفی را میتوان توسط مدهای کاری مختلف که بوسیله پین های خارجی قطعه انتخاب می گردند، کنترل کرد.

این قطعه دارای چند مد کاری مختلف است و این مدهای کاری وابسته به ولتاژ اعمالی به پین های S_0 و S_1 است. در جدول شماره (۲) مدهای کاری این قطعه آمده است.

S_1	S_0	مد کاری	مدار فعال شده				شمارنده
			سیستم اندازه گیری فشار	سیستم اندازه گیری دما	مقایسه گر A/D خروجی	اسیلاتور فرکانس پائین	دیتای سریال
0	0	Standby/Reset	خاموش	خاموش	خاموش	روشن	فعال
0	1	اندازه گیری فشار	روشن	خاموش	خاموش	روشن	Reset
1	0	اندازه گیری دما	خاموش	روشن	خاموش	روشن	Reset
1	1	Output Read	خاموش	خاموش	روشن	روشن	فعال

جدول شماره (۲): مدهای کاری سنسور MPXY8020A

در تمامی مدهای لیست شده در بالا، تا زمانیکه منبع تغذیه مدار به پین V_{DD} متصل باشد، کانال مالتی پلکسرها، رجیستر D/A ، اسیلاتور فرکانس پائین ($LF Oscillator$) و مقسم های پالس خروجی، توان مصرف خواهند کرد.

زمانیکه فقط پین S_0 برابر با یک منطقی باشد، مدار اندازه گیری فشار فعال می شود و سیگنال خروجی فشار توسط یک مالتی پلکسر به خازن نمونه گیر متصل می شود. هنگامیکه پین S_0 به وضعیت صفر منطقی برود، مالتی پلکسر خاموش خواهد شد و این کار برای این است که سیگنال روی خازن نمونه گیر، قبل از خاموش شدن مدار اندازه گیری فشار، در خازن نمونه گیر ذخیره شود.

هنگامیکه فقط پین S_1 برابر با یک منطقی باشد، مدار اندازه گیری دما فعال می شود و سیگنال خروجی دما توسط یک مالتی پلکسر به خازن نمونه گیر متصل می شود. هنگامیکه پین S_1 به وضعیت صفر منطقی برود، مالتی پلکسر خاموش خواهد شد و این کار برای این است که سیگنال روی خازن نمونه گیر، قبل از خاموش شدن مدار اندازه گیری دما، در خازن نمونه گیر ذخیره شود.

تذکر ۱) صرفنظر از اینکه مدار اندازه گیری فشار یا دما فعال باشد، باید تمام بیت های پاک کننده *EEPROM*، همواره فعال باشند.

تذکر ۲) اگر ولتاژ روی پین S_1 از مقدار 2.5 برابر V_{DD} تجاوز کند، قطعه به مد *Trim/Test* خواهد رفت.

تذکر ۳) اگر برای کاهش مصرف جریان عنصر، ولتاژ منبع تغذیه V_{DD} خاموش شود، باید برای جلوگیری از روشن شدن قطعه، تمام بیت های ورودی را در وضعیت صفر منطقی قرار داد.

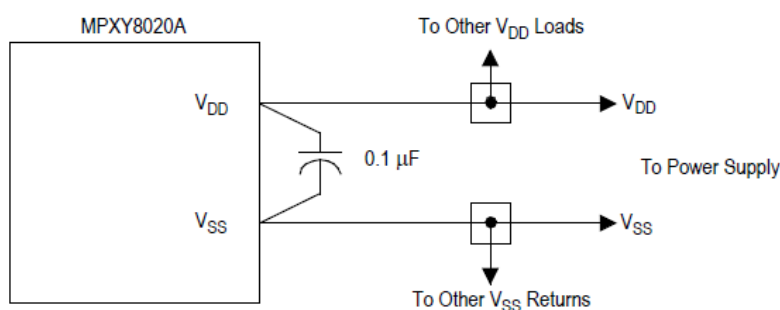
مادامیکه تغذیه خاموش است، اگر یکی از پین های ورودی یک منطقی باشند، ممکن است به واسطه یک دیود محافظ *ESD*، قطعه روشن شود. در چنین حالتی ولتاژ موثر V_{DD} در حدود 0.3 Volt کمتر از ولتاژ اعمالی به پین ورودی است و تمام جریان قطعه از ورودی قطعه گرفته می شود.

(۲-۳) عملکرد پایه های قطعه:

وظایف هر یک از پایه های قطعه را شرح ذیل می باشند:

پایه های V_{DD} و V_{SS} :

بواسطه پین های V_{DD} و V_{SS} توان برای IC کنترل، مهیا می شود. V_{DD} پلاریته مثبت تغذیه و V_{SS} زمین آنالوگ و دیجیتال می باشد. IC کنترل بوسیله یک منبع تغذیه مجزا عمل می کند. بنابراین سیم های رابط منبع تغذیه باید به پین های V_{DD} و V_{SS} متصل شوند. نحوه اتصال منبع تغذیه به قطعه در شکل (۳) آمده است.



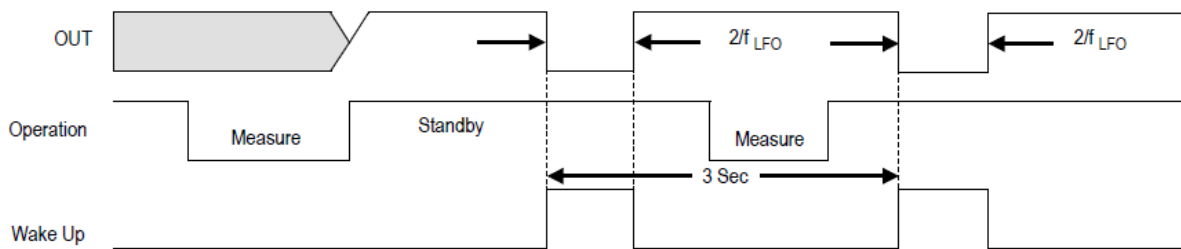
شکل (۳): نحوه اتصال منبع تغذیه به سنسور MPXY8020A

پایه OUT:

این پایه یک سیگنال دیجیتال را مهیا می کند که این سیگنال به ولتاژ اعمال شده به مقایسه گر ولتاژ و سطح آستانه منتقل شده به یک رجیستر ۸ بیتی از یک وسیله خارجی بستگی دارد.

زمانیکه قطعه در مد *Standby* قرار گیرد، این پایه یک منطقی خواهد بود و اگر یک *Overflow* از مقسم فرکانس آشکار شود، این پین به سطح صفر منطقی کلاک خواهد شد. این امر به پین OUT این

توانائی را خواهد داد که یک وسیله خارجی نظیر *MCU* را *Wake - Up* کند. شکل (۴) کلاک خروجی پین *OUT* را نمایش می دهد.

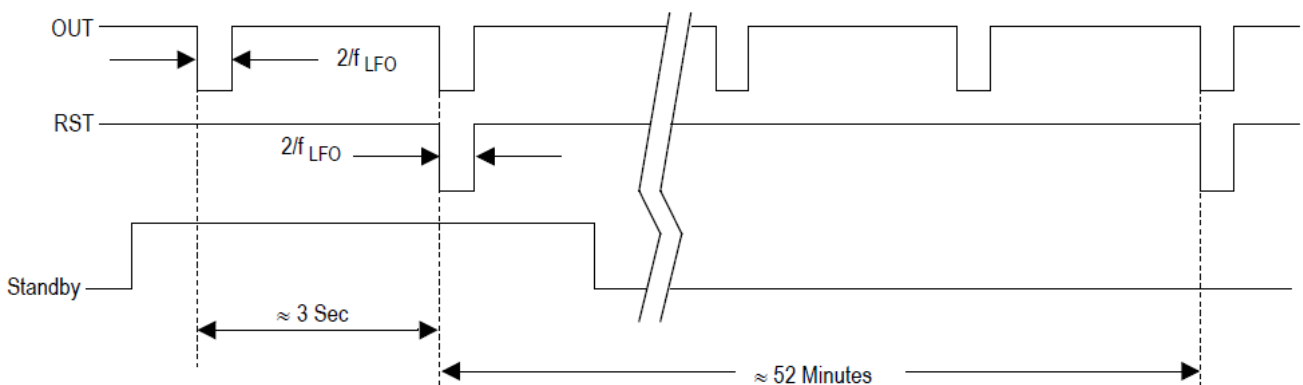


شکل (۴) : کلاک خروجی پایه *OUT* در سنسور *MPXY8020A*

پایه \overline{RST} :

این پایه معمولاً یک منطقی است و اگر یک *Overflow* از مقسم فرکانس آشکار شود، این پین به سطح صفر منطقی کلاک خواهد شد. این امر به پین \overline{RST} این توانائی را خواهد داد که یک وسیله خارجی نظیر *MCU* را *Reset* کند. این پالس صرفنظر از مد کاری قطعه، تقریباً هر ۵۲ دقیقه یکبار بر روی پین \overline{RST} ظاهر می شود. پالس برای مدت دو سیکل اسیلاتور فرکانس پائین به طول می کشد که در شکل (۵) نشان داده شده است.

باتوجه به اینکه پین \overline{RST} نیز مانند پین *OUT* توسط رشته ای تقسیم کننده یکسان کلاک شده است، هر



شکل (۵) : شکل موج پایه \overline{RST}

۵۲ دقیقه که پین \overline{RST} پالس می دهد، یک پالس نیز روی پین OUT خواهد آمد.

www.MELEC.ir

پایه S_0 :

پین S_0 جهت انتخاب مد کاری قطعه که در جدول (۲) آمده است، بکار می رود. پین S_0 دارای یک اشمیتتریگر داخلی، به عنوان بخشی از ورودی این پین است که برای بهبود مصونیت در برابر نویز می باشد. پین S_0 دارای یک عنصر پائین کشنده (*Pull – Down*) داخلی است تا زمانیکه پین بدون اتصال می باشد، به سطح صفر منطقی برود.

پایه S_1 :

پین S_1 جهت انتخاب مد کاری قطعه که در جدول (۲) آمده است، بکار می رود. پین S_1 دارای یک اشمیتتریگر داخلی، به عنوان بخشی از ورودی این پین است که برای بهبود مصونیت در برابر نویز می باشد. این پین دارای یک عنصر پائین کشنده (*Pull – Down*) داخلی است تا زمانیکه پین بدون اتصال می باشد، به سطح صفر منطقی برود.

این پایه همچنین به منظور فعال سازی *Trim* و تست قطعه نیز بکار گرفته می شود. همچنین بیشترین ولتاژ پروگرامینگ V_{PP} برای *EEPROM Trim Register* بواسطه این پایه تامین می گردد.

پایه DATA:

وظیفه پین DATA ، *Serial Data In* یا *SDI* برای تنظیم آستانه مقایسه گر ولتاژ است.

پین DATA دارای یک اشمیت تریگر داخلی، به عنوان بخشی از ورودی این پین است که برای بهبود مصونیت در برابر نویز می باشد. این پین دارای یک عنصر پائین کشنده (*Pull – Down*) داخلی است تا زمانیکه پین بدون اتصال می باشد، به سطح صفر منطقی برود.

پایه CLK :

پین CLK ، کلاکی را جهت *Load* کردن و انتقال دیتا به پین DATA ، فراهم می سازد.

اطلاعات روی پین DATA با لبه بالارونده سیگنال پین CLK به یک شیفت رجیستر کلاک می شوند. در هشتمین لبه پائین رونده پین CLK ، اطلاعات به رجیستر *D/A* منتقل می شوند. این پروتکل شاید توسط *SPI* یا *SIOP Serial I/O* که در برخی از *MCU* ها یافت می شود، اجرا شود.

پین CLK دارای یک اشمیت تریگر داخلی، به عنوان بخشی از ورودی این پین است که برای بهبود مصونیت در برابر نویز می باشد. این پین دارای یک عنصر پائین کشنده (*Pull – Down*) داخلی است تا زمانیکه پین بدون اتصال می باشد، به سطح صفر منطقی برود.

وضعیت پین OUT توسط یک مقایسه گر ولتاژ تعیین می شود که حالت خروجی آن به سطح ولتاژ ورودی روی خازن نمونه گیر و سطح یک ولتاژ آستانه هشت بیتی قابل تنظیم، وابسته است.

مادامیکه پین CLK کلاک می کند، آستانه توسط انتقال بیت‌های دیتا - بوسیله پین DATA - به رجیستر دیجیتال به آنالوگ (ADR)، قابل تنظیم می باشد. *Timing* این اطلاعات در شکل (۶) نشان داده شده است.

با لبه بالارونده کلاک پین CLK، دیتا به *Serial Shift Register* منتقل می شوند. در لبه پائین رونده هشتمین کلاک، اطلاعات درون *Serial Shift Register* در رجیستر *ADR* موازی لچ می شود. هر موقع که V_{DD} در مدار باشد رجیستر نیز *Power Up* خواهد ماند. ابتدا *MSB* اطلاعات سریال به پین DATA کلاک می شود. این ترتیب انتخاب آستانه در جدول (۳) نشان داده شده است.

Function		Bit Weight	Data Bit
	LSB	1	D ₀
		2	D ₁
		4	D ₂
Voltage Comparator Threshold Adjust (8bit)		8	D ₃
		16	D ₄
		32	D ₅
		64	D ₆
	MSB	128	D ₇

جدول شماره (۳): ترتیب انتخاب بیت‌های آستانه

یک تبدیل آنالوگ به دیجیتال (A/D) با هشت سطح آستانه مختلف، از طریق الگوریتم تقریب متوالی (*Successive Approximation*) حاصل می شود. و همچنین پین OUT میتواند در برخی سطوح خاص، جهت *Trip* شدن تنظیم شود.

ولتاژ روی خازن نمونه گیر میتواند به مدت کافی جهت یک تبدیل هشت بیتی، حفظ شود.

اگر مدت زمان بین دو *Read*، طولانی تر از زمان نگهداری تعیین شده (t_{SH}) باشد، ممکن است برای اندازه گیری جدید نیاز به *Refresh* باشد.

شمارنده ای که تعداد پالس های اعمالی به قطعه را مشخص می کند، هنگامیکه قطعه به مد اندازه گیری دما یا مد اندازه گیری فشار برود، *Reset* خواهد شد. این امر روشی برای *Reset* شمارش اطلاعات انتقال یافته را فراهم می سازد و آن در صورتی است که *Clock Stream*، بتدریج در طول یک انتقال از بین رود.

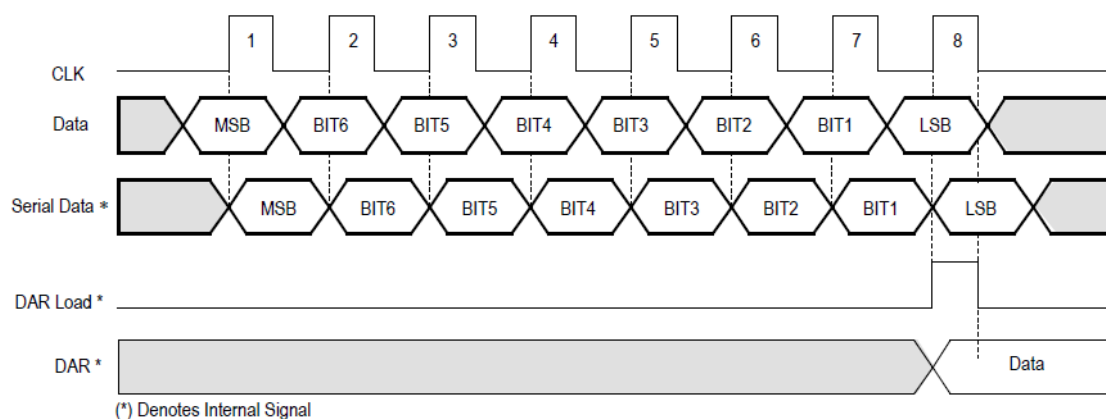
در هر دو مد اندازه گیری، پین های DATA و CLK نباید *Reset* شوند تا نویز دما و فشار اندازه گیری شده کاهش پیدا کند.

هر تغییری در محتویات رجیستر مستلزم آنست که عنصر در مد *Standby* یا مد *Out Put Read* قرار گرفته باشد.

در پی *Power Up* شدن قطعه، مقادیر شمارنده *Serial Bit* و وضعیت رجیستر *DAR*، تعریف نشده هستند.

هنگامیکه پین های S_0 یا S_1/V_{PP} در سطح یک منطقی باشند و به سطح صفر منطقی برگردند شمارنده *Serial Bit*، میتواند *Reset* شود.

با پائین نگه داشتن سطح منطقی پین DATA تا زمانیکه پالس پین CLK به هشت کلاک برسد، میتوان رجیستر را *Reset* نمود.



شکل (۶) : بارگذاری رجیستر DAR

(۵-۴) خروجی سنسور فشار:

کانال فشار، خروجی مدار اندازه گیری آنالوگ را با ولتاژ مرجع D/A مقایسه می کند.

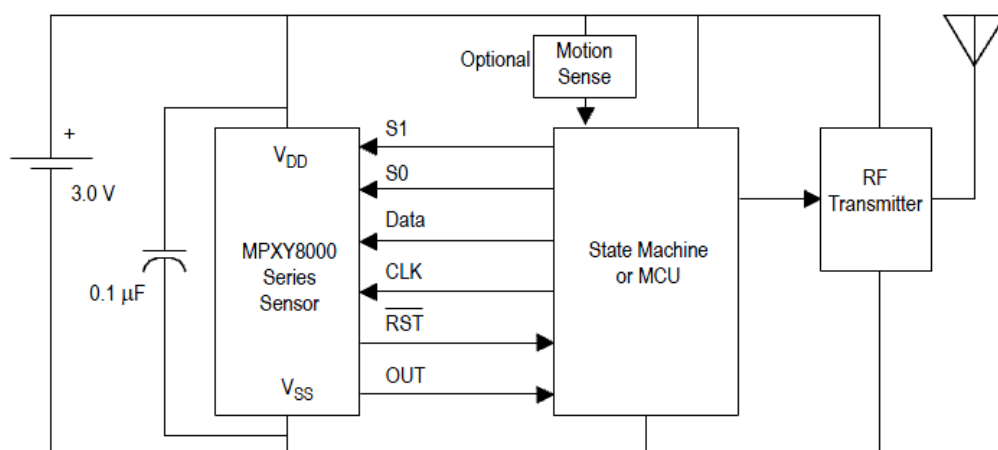
قطعه در دو مقدار نامی مختلف کالیبره شده است که این دو مقدار به انتخاب کالیبراسیون بستگی دارند.

(۶-۲) خروجی سنسور دما:

کانال دما، خروجی یک مقاومت با ضریب حرارتی مثبت (PTC) که از یک منبع جریان سوئیچ شده حاصل می شود را مقایسه می کند. منبع جریان فقط زمانی که کانال دما انتخاب شده باشد، فعال است.

(۷-۲) کاربردها:

پیشنهاد می شود از نمونه کاربردی شکل (۷) استفاده شود



شکل (۷) : یک نمونه مدار پیشنهادی سنسور MPXY8020A

فصل سوّم : آشنایی با بخش های نرم افزار و سخت افزارِ سیستم طراحی شده

(۱-۳) نحوه عملکرد *MCU* و ارتباط با سنسور *TPMS*:

همانگونه که در بخش معرفی عملکرد سنسور *MPXY8020A* توضیح داده شد، این سنسور، یک چیپ هشت پایه می باشد که جهت راه اندازی آن، لازم است که به آن دیتای تعریف شده داده شود و در طرف دیگر، باید خروجی آن را، به ازای ورودی های داده شده، چک کنیم.

در این پروژه، *MCU* و سنسور *TPMS* از طریق پروتکل ارتباطی *SPI* با یکدیگر ارتباط دارند و بدین منظور، پایه های *DATA* و *CLK* در سنسور، به ترتیب به پایه های *MISO* و *SCK* در *MCU* متصل می گردند.

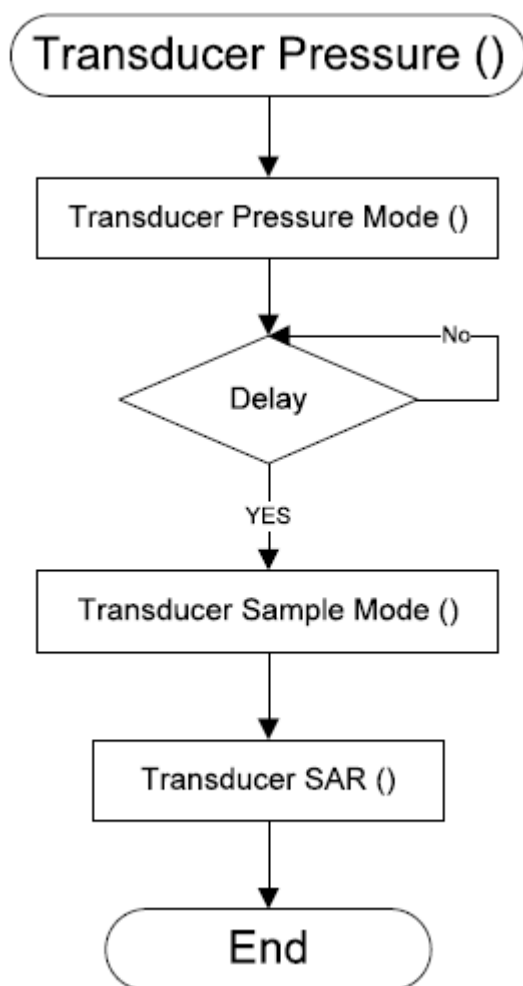
پالس هایی همزمانی دو عنصر در *MCU* (که به عنوان *Master* عمل میکند)، تولید می شود بر روی پایه *CLK* قرار می گیرند. و دیتایی که باید در رجیستر *DAR* قرار گیرد، از طریق پایه *MISO* به سنسور میرسد.

جهت تعیین مدهای اندازه گیری سنسور که از طریق *MCU* انتخاب می گردند، از توابع *Transducer Pressure* و *Transducer Temperature* استفاده می شود که از طریق این توابع می توانیم سنسور را در هر مودی اندازه گیری که نیاز باشد، قرار دهیم.

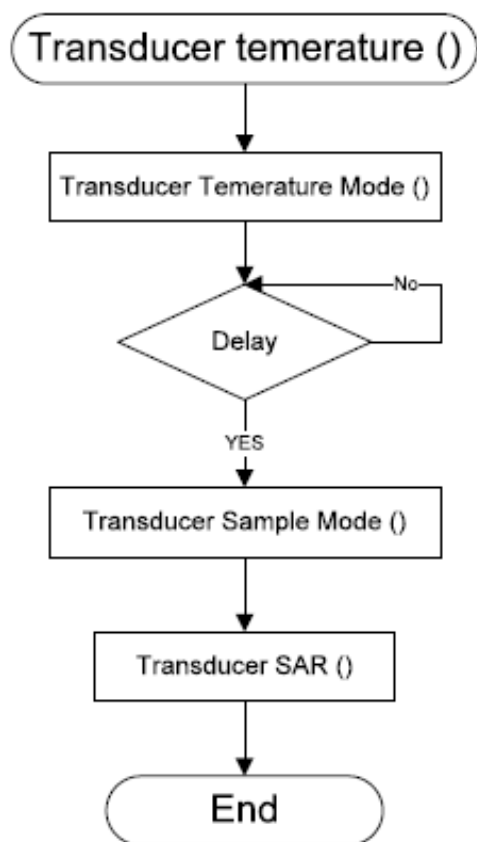
فلوچارت توابع *Transducer Pressure* و *Transducer Temperature* در صفحه بعد آمده است.

لازم به ذکر است که پس قرار گرفتن سنسور در مُد اندازه گیری موردنظر، باید آن را در مُد *Output Read* قرار دهیم تا خروجی سنسور را مانیتور کنیم. برای این کار از تابع *Read Transducer* که در برنامه نویسی *MCU* باید لحاظ شود، استفاده می کنیم

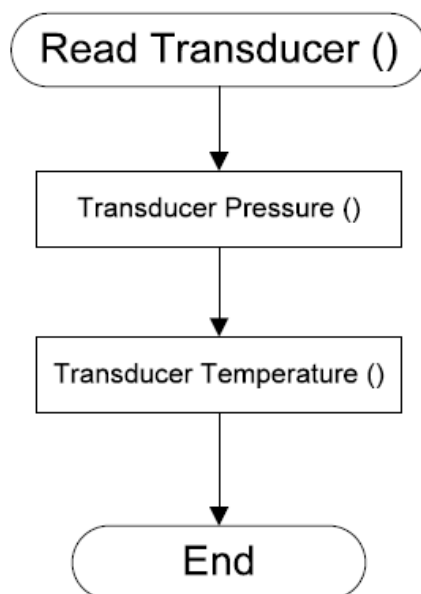
اصلی ترین تابعی که وظیفه انجام این کار را به عهده دارد، تابع *SAR* می باشد.



شکل (۸): فلوچارت فلوچارت تابع *Transducer Pressure*



شکل (۹): فلوچارت فلوچارت تابع *Transducer Temperature*



شکل (۱۰): فلوچارت تابع *Read Transducer*

(۲ - ۳) روتین تقریب متوالی (SAR):

برای محاسبه نرم افزاری فشار و دمای لاستیک خودرو توسط سنسور *MPXY8020A*، از روتین تقریب متوالی (*Successive Approximation Routine*) استفاده می کنیم. در ذیل، شرح عملکرد این روتین بیان شده است.

به دلیل اینکه عدد دیجیتال خروجی متناظر با فشار یا دمای اندازه گیری شده یک عدد هشت بیتی است، پس در ابتدا، حلقه ای با شرط تکرار هشت بار، تعریف می کنیم و در آن متغیرهای تخمین (*Estimate*)، وزن (*Weight*)، حدس (*Guess*) و خروجی (*Output*) را تعریف می کنیم.

وضعیت پایه OUT سنسور (یک یا صفر بودن این پایه) در متغیر *Output* قرار داده می شود و متغیر *Weight* متغیری است که دارای مقدار اولیه *0b 10000000* بوده و پس از انجام هر *loop* از حلقه، یک واحد به سمت راست شیفت داده می شود. متغیر *Estimate*، متغیری است که در پایان *loop* هشتم حلقه (یعنی آخرین *loop* اجرایی حلقه)، حاوی مقدار فشار یا دمای اندازه گیری شده می باشد.

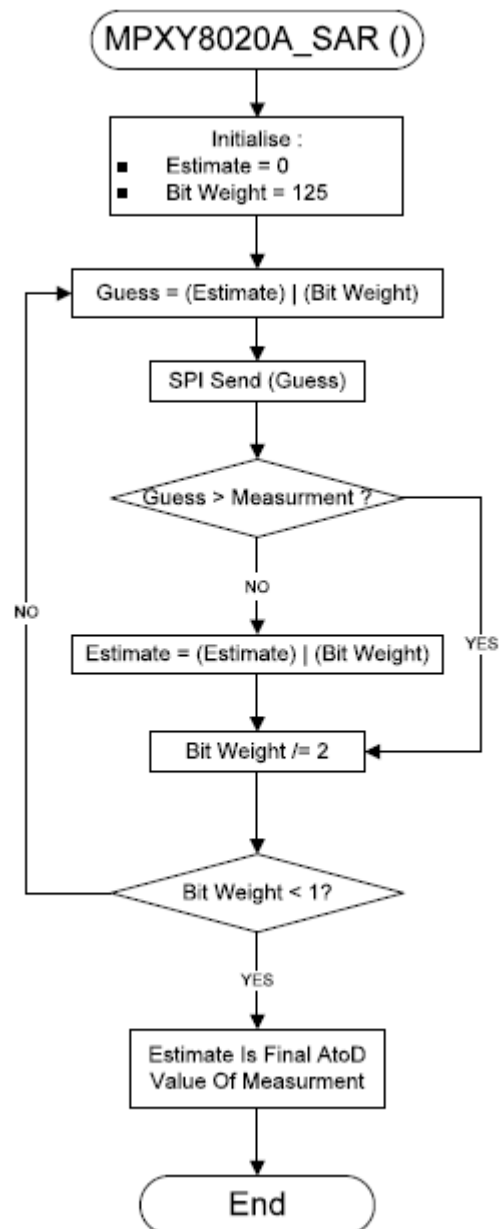
در اولین حلقه اجرایی ($n = 7$)، متغیر *Guess* از عمل *OR* بیتی بین دو متغیر *Estimate* و *Weight* مقداردهی می شود. سپس توسط واحد *SPI* موجود در *MCU*، عدد ذخیره شده در متغیر *Guess* را به پایه DATA در سنسور منتقل می کنیم. مقدار *Guess* انتقال داده شده به رجیستر دیجیتال به آنالوگ (*DAR*) در سنسور وارد می شود.

بعد از اتمام این مرحله، وضعیت پایه OUT سنسور را بررسی می کنیم. اگر این پایه در وضعیت یک منطقی قرار گیرد، بیانگر آنست که حدس ما (*Guess*) درست بوده و باید متغیر *Estimate* را برابر متغیر *Guess* قرار دهیم و پس از پایان این حلقه اجرا شده، مقدار موجود در متغیر *Weight* را نصف حالت قبلی می کنیم.

دومین حلقه ($n = 6$) شروع به اجرا می کند و روتین بالا دوباره تکرار می گردد. در صورتیکه در این حلقه، وضعیت پایه OUT صفر منطقی باشد، هیچ عملی بر روی متغیرهای حلقه انجام نمی گیرد و تنها متغیر *Weight*، دوباره نصف حالت قبلی می گردد.

نهایتاً تا انجام آخرین حلقه ($n = 0$) این روتین دائماً تکرار می گردد و پس از اتمام این حلقه، متغیر *Estimate* با عددی پر می گردد که بسته به وضعیتی که تعریف کرده ایم، فشار یا دمای اندازه گیری شده می باشد. فلوچارت روتین (*SAR*) بصورت زیر می باشد:

www.MELEC.ir



شكل (١١) : فلوچارت روتين (SAR)

(۳-۳) نحوه دریافت و ارسال اطلاعات:

بدلیل اینکه سیستم اندازه گیری فشار باد تایر، باید در درون لاستیک خودرو قرار گیرد، لذا نیاز است که برای ارتباط بین گیرنده و فرستنده از ارتباط بی سیم (*Wireless*) استفاده نمائیم.

در این مدل طراحی شده، از ماژول های آماده *FSK* استفاده شده است که در گیرنده از ماژول $HM - R$ و در فرستنده از ماژول $HM - T$ استفاده شده است.

فرکانس کاری ارتباط، 433 MHz و نرخ ارسال و دریافت داده ها 4800 bps می باشد.

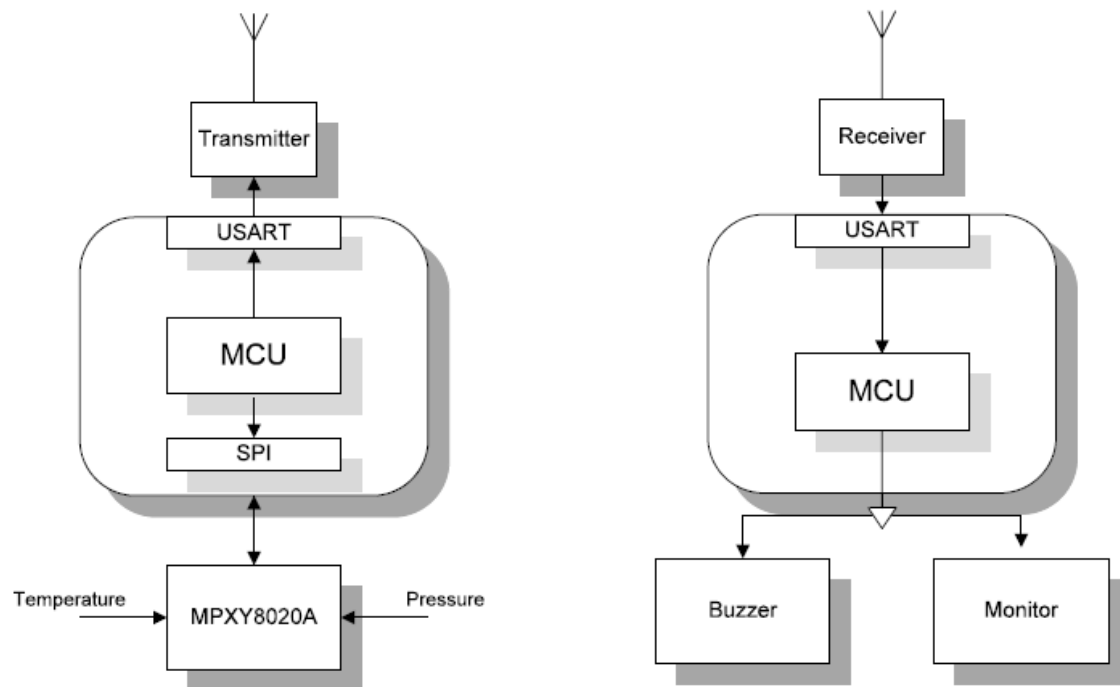
اطلاعات مربوط به فشار و دمای اندازه گیری شده در فرستنده، به *MCU* می رسد و بدلیل آنکه برای ارسال این اطلاعات، باید از یک پروتکل استاندارد استفاده کنیم، لذا بهترین روش برای ارسال این اطلاعات، پروتکل *USART* می باشد. که در آن، اطلاعات بسته بندی (*Capsule*) شده و در یک قالب (*Frame*) ده بیتی، که تشکیل شده است از یک بیت شروع، یک بیت پایان و هشت بیت دیتای مورد نظر، به پایه *RXD* منتقل شده و بعد از آن به پایه *DATA*، در ماژول فرستنده منتقل می شود و فرستنده، این قالب را به گیرنده ارسال می کند.

همانگونه که پایه *DATA* ماژول فرستنده به پایه *RXD* در *MCU* متصل شده است، باید پایه *DATA* ماژول گیرنده نیز به پایه *TXD* در *MCU* متصل شود و از این طریق یک ارتباط غیرهمزمان (*Synchronous*) توسط رابط *USART* بین فرستنده و گیرنده برقرار می گردد و دیتا به راحتی ارسال می شود.

www.MELEC.ir

(۳ - ۴) بلوک دیاگرام کلی سیستم طراحی شده:

طبق توضیحات داده شده در قسمت های مختلف، اگر بخواهیم یک جمع بندی کلی در رابطه با سیستم *TPMS* داشته باشیم و خلاصه سیستم طراحی شده را به صورت بلوک دیاگرام رسم کنیم، مدار فرستنده و گیرنده به صورت زیر خواهند بود:



شکل (۱۲) فلوچارت سیستم گیرنده

شکل (۱۳) فلوچارت سیستم فرستنده

(۵ ۳) مدار فرستنده:

همانگونه که در مدار شکل (۱۴) مشاهده می شود، *MCU* بکار رفته در این مدار، *AT Mega 8 L* می باشد که به دلیل توان تلفاتی کم و نیز ولتاژ کاری پائین باعث حفظ انرژی باتری می شود.

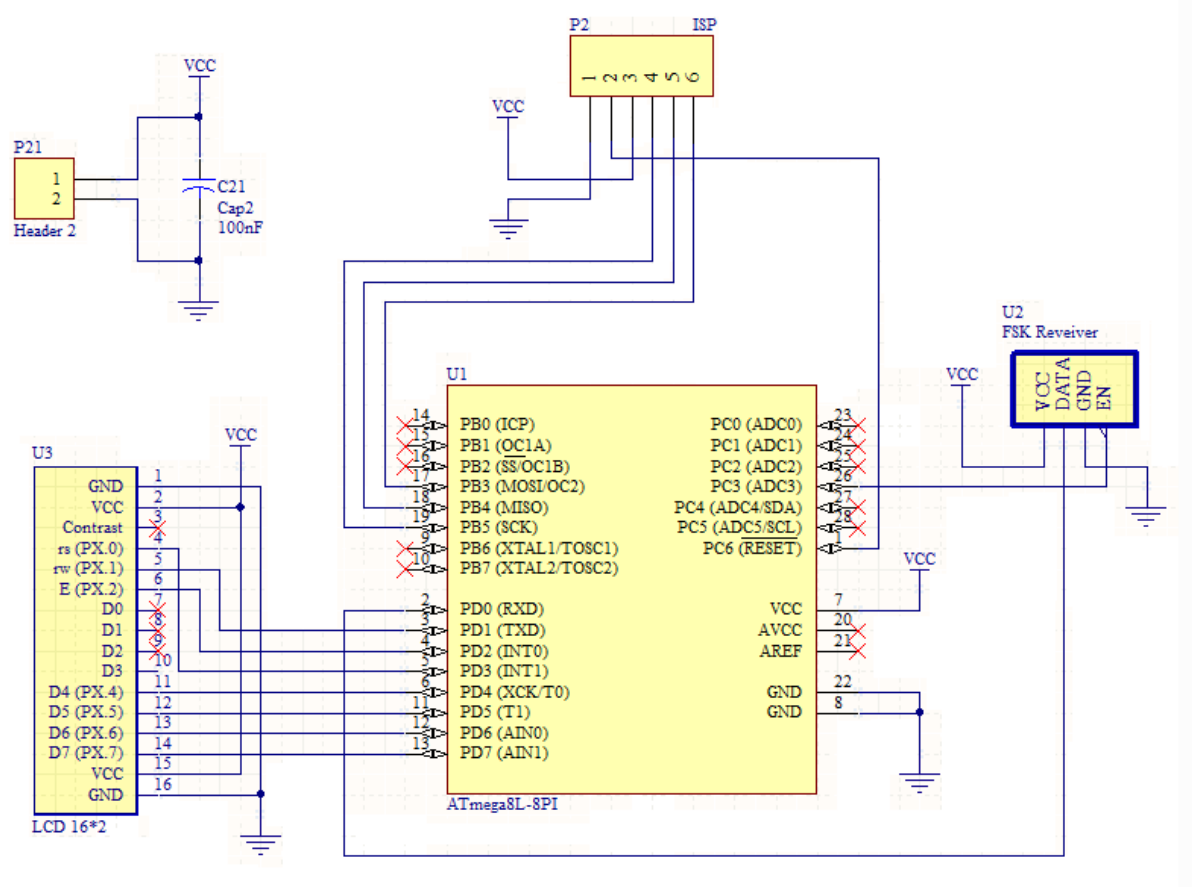
همچنین نحوه اتصال پایه های سنسور *MPXY8020A* به میکروکنترلر، قابل مشاهده می باشد.

دلیل اتصال پایه های *DATA* به *MOSI* و پایه *CLK* به *SCK* در میکروکنترلر، جهت اجرای پروتکل *SPI* می باشد.

پایه های S_0 و S_1 به عنوان ورودی های تعیین کننده مدهای کاری سنسور هستند و با یک و صفر کردن آنها توسط میکروکنترلر، میتوان بین مدهای کاری سنسور سوئیچ نمود.

پایه *OUT* جهت بازبینی وضعیت خروجی سنسور است.

همانطور که مشاهده می کنید، پایه *DATA* در ماژول فرستنده، به پایه *PD.1 (TXD)* متصل شده و وظیفه این پایه، ارسال اطلاعات برطبق پروتکل *USART* می باشد.

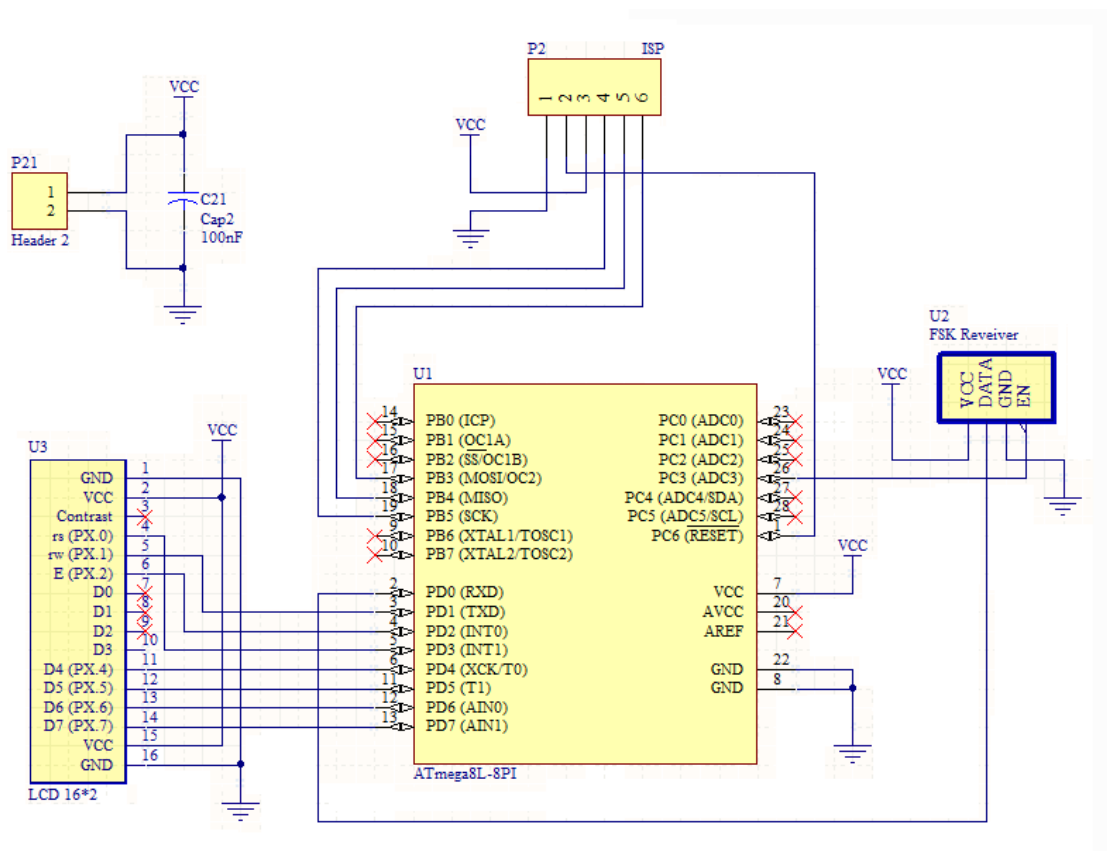


شکل (۱۴) شماتیک مدار فرستنده

(۳-۶) مدار گیرنده:

در مدار شکل (۱۵) مدار گیرنده را مشاهده می کنید. بدلیل آنکه مدار گیرنده در داخل کابین خودرو قرار می گیرد و نیازی به استفاده از باتری نمی باشد، لذا تأکیدی بر استفاده از میکروکنترلرهای کم مصرف نمی باشد. *MCU* بکار رفته در مدار گیرنده، *AT Mega 8* می باشد.

LCD به کار رفته در این مدار که به جهت نمایش مقدار فشار یا دمای لاستیک خودرو است، بر روی پایه های *PD.1* تا *PD.7* قرار گرفته است. به جهت استفاده از پروتکل *USART* در فرستنده، باید از همین پروتکل نیز در گیرنده، جهت آشکارسازی اطلاعات دریافتی، استفاده نمود که به این دلیل پایه *DATA* در مازول گیرنده، به پایه *PD.0 (RXD)* متصل شده و وظیفه این پایه، دریافت اطلاعات برطبق پروتکل *USART* می باشد.



شکل (۱۵) شماتیک مدار گیرنده

همانطور که در شکل های شماره (۱۵) و شماره (۱۶) آمده است؛ برای جلوگیری از اتلاف زمان در پروگرام کردن برنامه ها و نیز رفع مشکلات نرم افزاری که مستلزم تغییر در برنامه نویسی میکروکنترلر است، در هر دو مدار فرستنده و گیرنده، سوکت مخصوص *ISP (In Ststem Programing)* جهت پروگرام کردن کدهای برنامه نویسی در میکروکنترلرها پیش بینی شده است.

(۷-۳) نرم افزارهای مربوطه:

در این پروژه ای که با آن آشنا شدید، میتوان از نرم افزارهای مختلفی جهت انجام پروژه استفاده کرد و دلیل آنکه در آخرین قسمت از توضیحات معرفی پروژه، به معرفی نرم افزارهای بکار برده شده پرداختیم اینست که؛ این قسمت به میزان تسلط طراح بر نرم افزارهای مورد نیاز، بستگی دارد و بسته به میزان تسلط طراح، این نرم افزار ها نیز تغییر می کنند.

به دلیل آنکه در این پروژه از میکروکنترلرهای شرکت *Amel* استفاده شده است و نیز عمومی بودن کارائی های بالای زبان *C*، از برنامه نویسی به این زبان استفاده شده است.

جهت برنامه نویسی به زبان *C* میتوان از *Compiler* های مختلفی از قبیل *AVR Studio* ، *Codevision AVR* و ... استفاده نمود. در این پروژه از نرم افزار *Codevision AVR* جهت کُد نویسی میکروکنترلرها استفاده شده است.

جهت تسهیل در بکار گیری توابع اصلی، تمامی آنها در بخش های قبلی، بصورت فلوچارت بیان شد است و به کسانی که در آینده بخواهند این پروژه را تکمیل کنند، می توانند این توابع را در کامپایلرهای دیگر، و یا حتی به زبانهای دیگر برنامه نویسی، کُد نویسی کنند.

برای ترسیم شماتیک مدارها و همچنین طراحی بُرد مدار چاپی (*PCB*) آنها نیز میتوان از نرم افزار گوناگونی از قبیل *Protel 99 SE* ، *PSPICE* ، *Proteus* ، *Altium Designer* و ... استفاده نمود. در این پروژه از نرم افزار *Altium Designer* جهت ترسیم شماتیک و *PCB* استفاده شده است. در پیوست شماره (۱)، شکل *PCB* مدارهای فرستنده و گیرنده آمده است.

فصل چهارم : نتیجه گیری و پیشنهادات

در این پروژه برای ساخت مدار حسگر فشار و دما، از یک چیپ (سنسوری) که مربوط به کمپانی Motorola می باشد و نیز جهت انجام پردازش اطلاعات و کنترل سیستم از یک MCU که مربوط به کمپانی Atmel می باشد استفاده شده است. نظر به اینکه این دو مدار مجتمع ساخت دو کمپانی مختلف بوده و نیز MCU بکار رفته فاقد فرستنده بوده است، در انجام پروژه، به مشکلات بسیاری برخورد نمودیم که پیشنهاد می شود جهت سهولت در برقراری ارتباط بی سیم و نیز تطبیق بین دو چیپ بکار گرفته شده، از MCU هایی که مربوط به کمپانی Motorola که دارای فرستنده و گیرنده داخلی هستند استفاده شود.

www.MELEC.ir

فصل پنجم : مراجع و منابع

[1] Motorola MPXY8020A Tire Pressure Monitoring Sensor

[2] Atmel AT Mega8

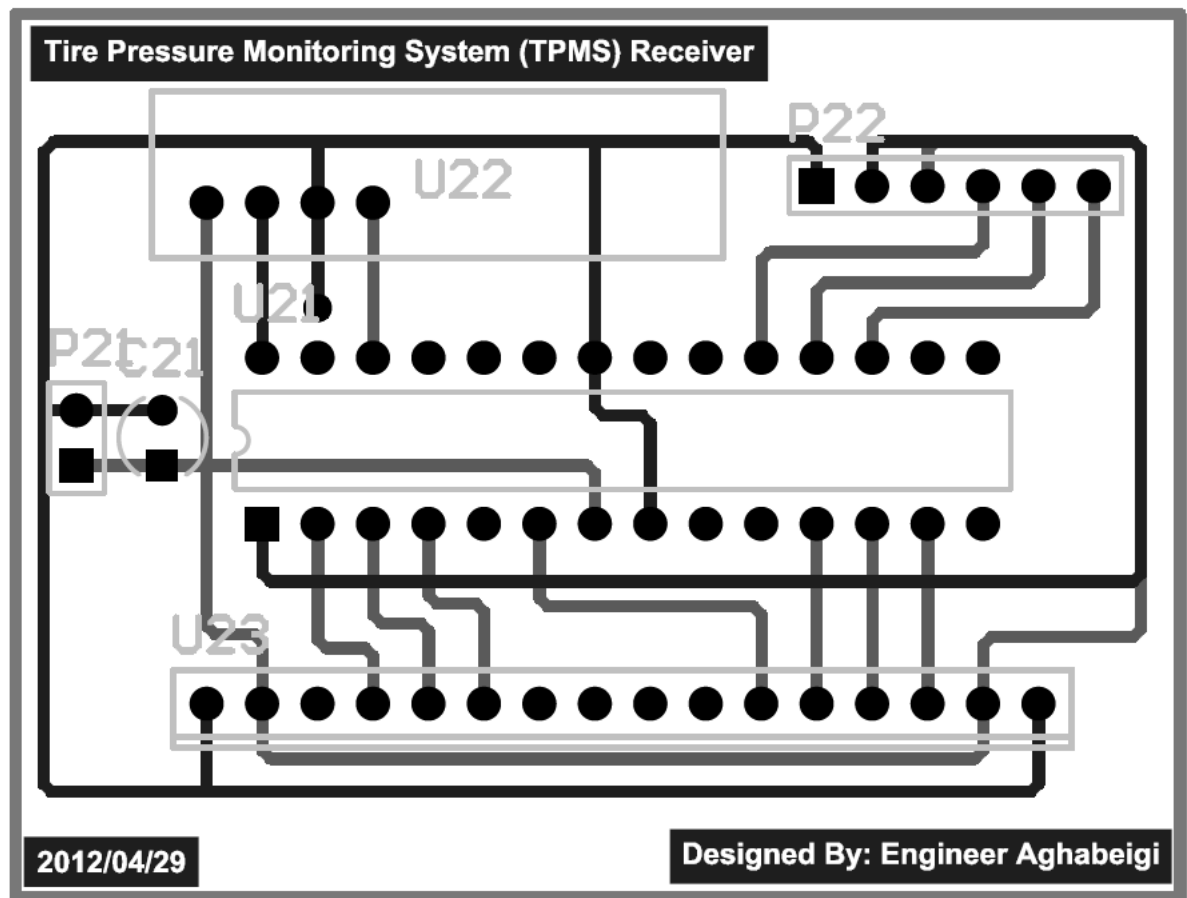
[3] Hoper RF Electronic HM – T

[4] Hoper RF Electronic HM – R

[۵] فرزاد مظاهریان، “مرجع کامل میکروکنترلرهای AVR”، انتشارات نص ، ۱۳۸۸

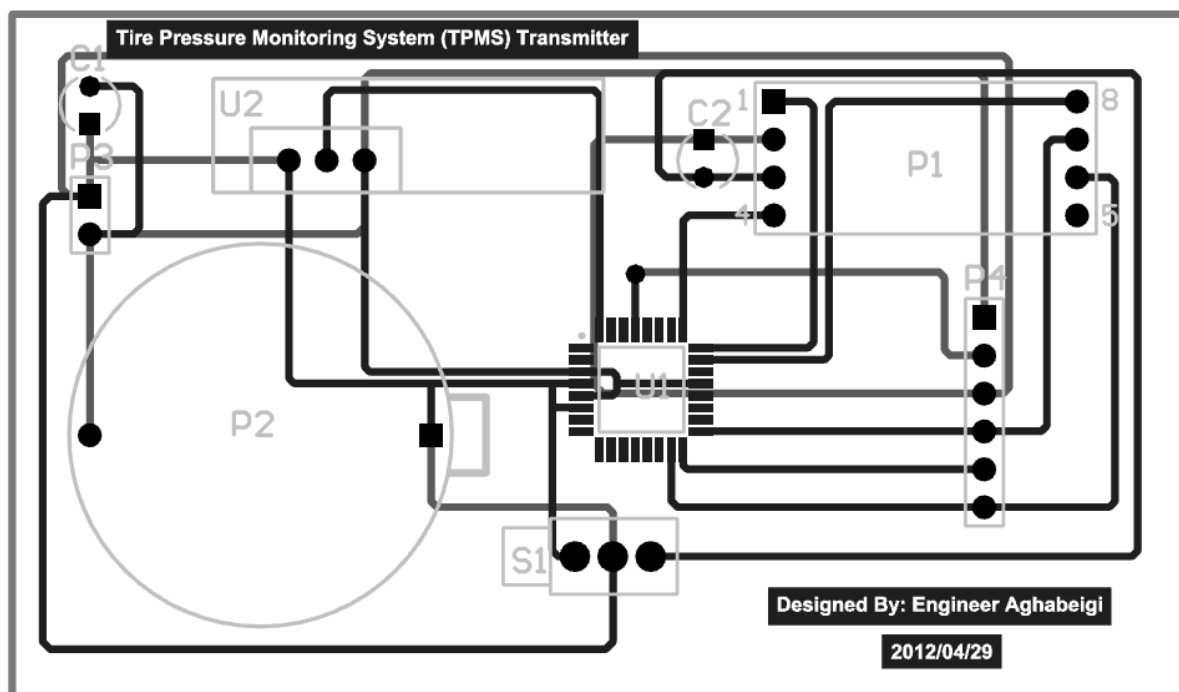
فصل ششم : پیوست ها

پیوست شماره (۱): نقشه PCB مدار گیرنده



شکل (۱۶) نقشه PCB مدار گیرنده

پیوست شماره (۲): نقشه PCB مدار فرستنده



شکل (۱۷) نقشه PCB مدار فرستنده

پیوست شماره (۳): فهرست جداول

صفحه ۵	جدول شماره (۱): رابطه واحدهای مختلف فشار
صفحه ۱۰	جدول شماره (۲): مدهای کاری سنسور <i>MPXY8020A</i>
صفحه ۱۶	جدول شماره (۳): ترتیب انتخاب بیت‌های آستانه

پیوست شماره (۴): فهرست شکل ها

صفحه ۷	شکل شماره (۱): شکل ظاهری و پایه های این سنسور <i>MPXY8020A</i>
صفحه ۸	شکل شماره (۲): بلوک دیاگرام داخلی سنسور <i>MPXY8020A</i>
صفحه ۱۲	شکل شماره (۳): نحوه اتصال منبع تغذیه به سنسور <i>MPXY8020A</i>
صفحه ۱۳	شکل شماره (۴): کلاک خروجی پایه OUT در سنسور <i>MPXY8020A</i>
صفحه ۱۳	شکل شماره (۵): شکل موج پایه \overline{RST}
صفحه ۱۸	شکل شماره (۶): بارگذاری رجیستر DAR
صفحه ۱۹	شکل شماره (۷): یک نمونه مدار پیشنهادی سنسور <i>MPXY8020A</i>
صفحه ۲۱	شکل شماره (۸): فلوجارت فلوچارت تابع <i>Transducer Pressure</i>
صفحه ۲۲	شکل شماره (۹): فلوجارت فلوچارت تابع <i>Transducer Temperature</i>
صفحه ۲۲	شکل شماره (۱۰): فلوجارت تابع <i>Read Transducer</i>
صفحه ۲۴	شکل شماره (۱۱): فلوجارت روتین (<i>SAR</i>)
صفحه ۲۶	شکل شماره (۱۲): فلوجارت سیستم گیرنده
صفحه ۲۶	شکل شماره (۱۳): فلوجارت سیستم فرستنده
صفحه ۲۸	شکل شماره (۱۴): شماتیک مدار فرستنده
صفحه ۲۹	شکل شماره (۱۵): شماتیک مدار گیرنده
صفحه ۳۲	شکل شماره (۱۶): نقشه PCB مدار گیرنده

www.MELEC.ir