بسم اللّه الرّحمن الرّحیم

برقراری ارتباط سریال AVR با

Hyper terminal-MATLAB-labVIEW

به همراه چندین پروژه عملی

ابوالفضل عابدی

abediid@yahoo.com

فهرست

چکیده ..... III.

فصل اول:مبانی ارتباط سریال

ارتباط سریال چیست؟ ۱

UART ۱

سطوح ولتاژ در RS_232 ..... ۲

نمای پورت RS_232 ..... ۴

اصول ارتباط سریال ۵

فصل دوم:ارتباط میکرو و Hyper terminal

مقدمه ۸

توصیف سخت افزار ساخته شده ۹

تصویر شماتیک و PCB ۱۰

تشریح قسمتهای مختلف سخت افزار ۱۱

برسی توابع برنامه بیسیک برای ارتباط باپورت سریال ۱۳

برنامه نوشته شده برای ارتباط با Hyper Terminal ۱۶

توضیح برنامه ۱۸

نحوه انجام تنظیمات Hyper Terminal در windows XP ۲۱

فصل سوم:ارتباط سریال با MATLAB

مقدمه ۲۴

پورت های سریال در MATLAB ۲۵

توابع و مشخصات نوشتن در پورت سریال ۲۶

توابع و مشخصات خواندن از پورت سریال ۲۷

پروژه ارتباط همزمان میکرو با MATLAB ۲۹

پروژه ارتباط غیر همزمان میرو با MATLAB ۳۱

فصل چهارم:ارتباط سریال با labVIEW

مقدمه ۳۵
طرح کلی پروژه ۳۶
Block diagram پروژه ارتباط سریال با labVIEW ۳۷
Front panel پروژه ارتباط سریال با labVIEW ۳۸
برسی قسمتهای مختلف برنامه نوشته شده در Block diagram ۳۹
برنامه میکرو کنترلر ۴۰

فهرست منابع و مراجع ۴۴

چکیده:

بشر هر روز شاهد پیشرفت علم الکترونیک و به دنبال آن علم کامپیوتر است.این گستردگی باعث آن شده که علم کامپیوتر دو زیر شاخه اصلی پیدا کند،یکی سخت افزار و دیگری نرم افزار.
مزیتهای کامپیوتر باعث جذب مهندسان و پژوهشگران به سمت خود شده است. طوری که هرجا نیاز به پردازش،نمایش،یا ذخیره اطلاعات دریافتی از محیط بود سریع از این ابزار قدرتمند کمک میگیرند.
از این رو دانستن روشهایی که باعث تبادل اطلاعات بین کامپیوتر و دنیای خارج میشود اهمیت بسیار دارد.یکی از این روشها استفاده از ارتباط سریال از نوع RS_232 است که به دلیل سادگی با گذشت سالهای زیادی از ابداع آن هنوز طرفداران زیادی دارد.
در این تحقیق ابتدا از معرفی و مشخصات این ارتباط سخن گفته شده و سپس ارتباط بین یک سخت افزار جانبی که وضیفه دریافت اطلاعات محیطی را دارد با چند نرم افزار برسی میشود.
نرم افزار اول Hyper Terminal است.مزیت این نرم افزار موجود بودن در windows XP است و نیازی به نصب ندارد.این مزیت باعث آن میشود که سخت افزار جانبی تهیه شده به سرعت با یک کامپیوتر جدید ارتباط بر قرار کند. مزیت دیگر این نرم افزار سادگی کار با آن است چون برای دریافت و ارسال اطلاعات دستور یا بلوکی ندارد و با باز کردن برنامه و چند تنظیم اولیه ارسال و دریافت به سادگی صورت میگیرد.
نرم افزار دیگری که ارتباط سریال با آن در این پروژه برسی شده MATLAB است.مزیت این نرم افزار قدرت پردازش بالا و داشتن توابع مختلف ریاضی برای کار با داده ها است.
نرم افزار آخر labVIEW است.مزیت این نرم افزار داشتن مشخصات گرافیکی بالا و زبان برنامه نویسی ساده خود است که به ما این امکان را میدهد به سادگی از ابزارهای گرافیکی استفاده کنیم.
ارتباط سریال با هر کدام از این نرم افزارها در یک فصل برسی شده است و برای هر کدام پروژه عملی قرار داده شده است.

فصل اول

مبانی ارتباط سریال

ارتباط سریال چیست؟

ارتباط سریال پروتکل سطح پایین مشترک برای ارتباط بین دو یا چند وسیله است که به صورت معمول یک وسیله کامپیوتر و وسیله دیگر می تواند یک مودم ، پرینتر ، کامپیوتر دیگر و یا یک وسیله علمـــی از قبیل اسیلوسکوپ و یا فانکشن ژنراتور باشد.

ارتباط سریال RS_232 یکی از عمومی ترین ارتباطات خارجی کامپیوتر تا چند سال اخیر بود ولی با توجه به نیاز بـه سرعتهای بالا تر در کاربردهای امروزی، پورتهای دیگر در حال جایگزین شدن با این پورت هستند.پورتهای جدیدی مثل USB و USB2 که توانایی ارسال سریع اطلاعات رادارند جایگزین مناسبی برای این پورت میباشند .

با این حال استفاده از این پورت نه تنها منسوخ نشده بلکه در بسیاری از موارد به دلایل اقتصادی و راحتـی کـار بـا آن نسبت به پورتهای دیگر ارجحیت دارد.

از نظر عملی ارتباط برقرار کردن با پورت سریال مشکل تر از پورت پارالل میباشد.برای کاربردی شدن پورت در اکثر موارد لازم است دستگاهی که به پورت سریال متصل میگردد،انتقال سـریال را بـه پارالـل تبـدیل کنـد کـه ایـن عمـل توسط

UART انجام میشود.

UART

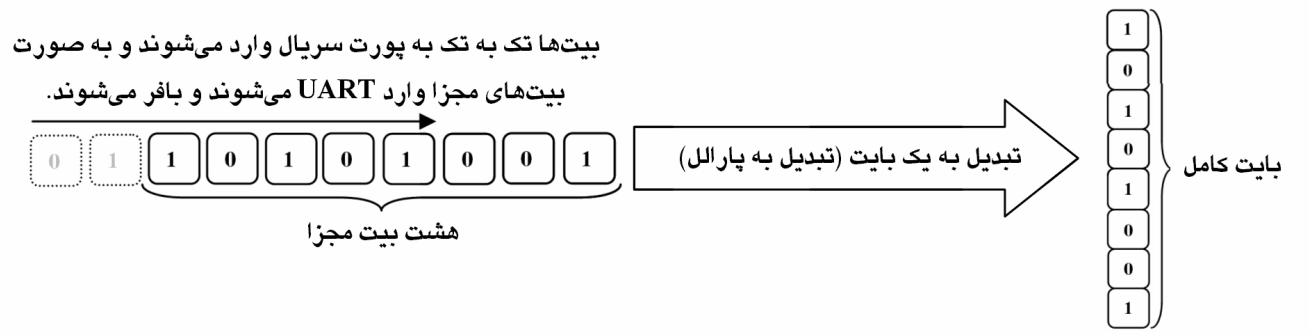
UART مخفف کلمات Univeral Asynchronous Reciever Transmiter به معنی فرستنده و گیرنـده جامع غیرهمزمان است. اطلاعات در پورت سریال به صورت بیت های پشت سر هم ارسال میشوند و میدانیم که بـرای کامپیوتر یک بیت به تنهایی معنی ندارد بلکه بایتها هستند که برای آنها قرار دادهای مختلف مثل کدهای Asciiو...

وجود دارد.

Uart بیتهای ورودی را به صورت پشت سر هم قرار میدهد و زمانی که تعـداد آنهـا بـه یـک بایـت رسـید در اختیـار سیستم عامل قرار میدهد.

در شکل صفحه بعد طرز کار uart را مشاهده میکنید.

نمایش کار کرد uart



سطوح ولتاژ در RS_232

در این استاندارد از دو سطح منطقی ۰و۱ استفاده میشود که به حالت ساکن سطح منطقی یک میباشد.جدول زیر این دوسطح را به همراه ولتاژ تعریف شده برای هر سطح نشان میدهد.

Data signals		
Level	Transmitter	Receiver
Logical 0	+5 V to +15 V	+3 V to +25 V
Logical 1	-5 V to -15 V	-3 V to -25 V
Undefined	-3 V to +3 V	

همان گونه که از جدول مشخص است سطوح ولتاژ با استاندارد تعریف شده TTL متفاوت است.

برای جبران این اختلاف و هم چنین بالا بردن امنیت از نویز و افزایش فاصله بین فرستنده و گیرنده از مدار مجتمع MAX232 میتوان استفاده کرد.این IC دارای دو بافر برای دریافت از RS232 و دو بافر برای ارسال داده دارد.

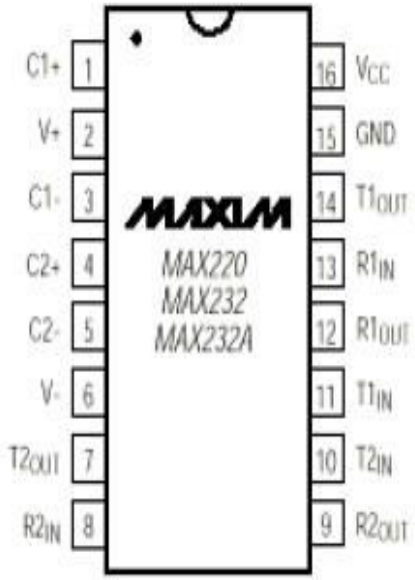
برگه اطلاعات این IC در فصل آخر قرار گرفته و در آنجا توضیحات کاملتری از نحوه کار آن وجود دارد.

در مدارات میکروکنترلری هم به دلیل اختلاف سطح ولتاژ با پروتکل RS_232 از این IC زیاد استفاده میشود.

اگر هدف فقط دریافت اطلاعات از پورت است و ارسال نیاز نیست میتوان از یک ترانزیستور نیز جهت تغییر سطح ولتاژ استفاده کرد ولی وقتی که هدف ارتباط دو سویه است به دلیل نیاز به ولتاژ منفی تر از زمین مدار، اهمیت این IC مشخص میشود.این IC با مدارات داخلی خود و استفاده از چند قطعه جانبی این ولتاؤ منفی را تولید میکند.

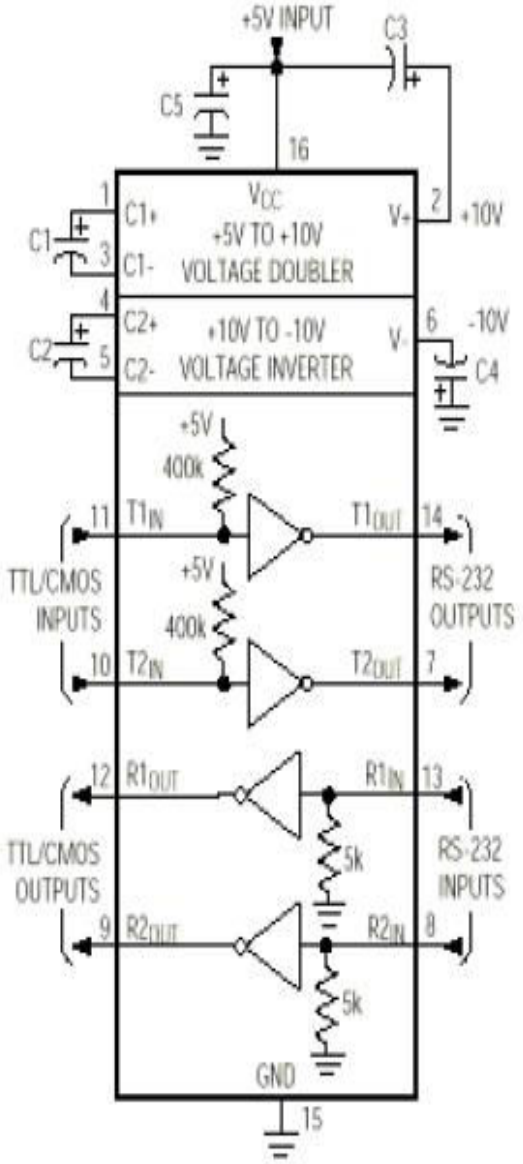
در شکل زیر این IC به همراه قطعات مورد نیاز جانبی نشان داده شده است.

نقشه ای سی MAX 232



DIP/SO

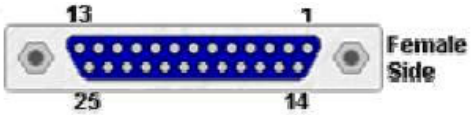
CAPACITANCE (µF)					
DEVICE	C1	C2	C3	C4	C5
MAX220	4.7	4.7	10	10	4.7
MAX232	1.0	1.0	1.0	1.0	1.0
MAX232A	0.1	0.1	0.1	0.1	0.1

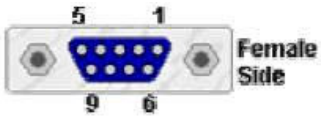


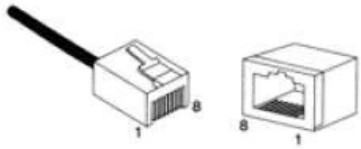
مدارات داخلی MAX TTL IC

نمای پورت RS_232

شماره پایه ها ، نام پایه ها و هم چنین کاربرد آنها در سه نوع پورت مختلف در ارتباط RS_232 در زیر نشان داده شده است.







Cannon 25			
Pin	Name	Direction	Description
1	SHIELD	---	Shield Ground
2	TXD	-->	Transmit Data
3	RXD	<--	Receive Data
4	RTS	-->	Request to Send
5	CTS	<--	Clear to Send
6	DSR	<--	Data Set Ready
7	GND	---	System Ground
8	CD	<--	Carrier Detect
9-19	N/C	-	-
20	DTR	-->	Data Terminal Ready
21	N/C	-	-
22	RI	<--	Ring Indicator
23-25	N/C	-	-

Cannon 9			
Pin	Name	Direction	Description
1	CD	<--	Carrier Detect
2	RXD	<--	Receive Data
3	TXD	-->	Transmit Data
4	DTR	-->	Data Terminal Ready
5	GND	---	System Ground
6	DSR	<--	Data Set Ready
7	RTS	-->	Request to Send
8	CTS	<--	Clear to Send
9	RI	<--	Ring Indicator

RJ-45			
Pin	Name	Direction	Description
1	RI	<--	Ring Indicator
2	CD	<--	Carrier Detect
3	DTR	-->	Data Terminal Ready
4	GND	---	System Ground
5	RxD	<--	Receive Data
6	TxD	-->	Transmit Data
7	CTS	<--	Clear to Send
8	RTS	-->	Request to Send

اصول ارتباط سریال:

Baud یا سرعت انتقال داده

سزعت انتقال داده در استاندارد RS-232 با Baud بیان میشود و میزان ارسال بیت در یک ثانیه را بیان میکند.

طبق استانداردهای موجود حد اکثر سرعت برابر ۹۲۱۶۰۰ بیت بر ثانیه است.بهتر است برای طول کابل های بزرگ از نرخ ارسال کوچکتر استفاده شود.

جدول زیر برای چند نرخ ارسال،حد اکثر طول مجاز کابل را پیشنهاد میکند.

Baud rate [Bd]	Max length [ft]	Max length [m]
19 200	50	15
9 600	500	150
4 800	1 000	300
2 400	3 000	900

بیت PARITY

این بیت که بیت خطا نام دارد در صورت استفاده میتواند تاثیر نویز را بر سیستم ارسال و دریافت داده کم کند.

این بیت میتواند چهار وضعیت فعال داشته باشد. اولین مورد odd یا فرد است و این معنی را دارد که در صورتی مجموع یک های دریافتی به همراه بیت خطا، اگر عدد فردی باشد انتقال اطلاعات صحیح انجام گرفته.این قرار داد باید بین فرستنده و گیرنده یکسان باشد.روشن است که این بیت تنها میتواند در صورتی به ما کمک کند که نویز محیط آنقدر زیاد نباشد که در هر بار ارسال روی ۲ بیت یا بیشتر تا ثیر بگذارد.

نوع دیگر بیت خطا زوج یا even است که مشابه مورد گفته شده در بالا است.

دو نوع بیت خطای دیگر space و mark است.این دو نوع برای چک کردن خطا مفید نیستند.نوع space بیان میکند بیت خطا همیشه به صفر تغییر کند و نوع mark بیان میکند بیت خطا همیشه یک شود.

بیت شروع:

خطوط داده منطقی در زمان بیکار همیشه دارای یکی از دو حالت صفر یا یک هست در این استاندارد که وضعیت بیکار یک است بیت شروع صفر میشود یعنی اطلاعات پس از ارسال یک صفر فرستاده میشوند

بیت پایان

از این بیت برای هم زمان کردن فرستنده و گیرنده استفاده میشود.فاصله زمانی بین بیتهای شروع و پایان به میزان نرخ baud - تعداد بیتهای داده- استفاده از بیت خطا بستگی دارد و مستقل از داده ارسالی است.

بیت پایان همیشه یک است و اگر گیرنده در هنگام دریافت بیت پایان مقدار صفر را دریافت کرد خطایی رخ داده است.

تعداد بیتهای پایان قابل تنظیم است و مقدار 1و1.5و2 را میتوان برای آن انتخاب کرد.معمولا مقدار ۱ برای این بیت انتخاب میشود. انتخاب موارد دیگر باعث زیادتر شدن طول پیغام میشود.

Hand shaking

این لفظ بیانگر یک روش پر کاربرد برای همزمان سازی فرستنده و گیرنده است و معمولا در دو نوع سخت افزاری و نرم افزاری وجود دارد که در زیر برسی شده است.

روش سخت افزاری

کنترل جریان داده در روش سخت افزاری به این صورت است که دو سیم RTS/CTS علاوه بر دو سیم انتقال داده وجود دارند.زمانی که یکی از دو وسیله داده ای را جهت ارسال دارد سیگنال Request to send را فعال میکند سپس در آن طرف دستگاه دیگر با فعال کردن سیگنال Clear to send آمادگی خود را جهت دریافت داده اعلام میکند و سپس عملیات ارسال صورت میگیرد.

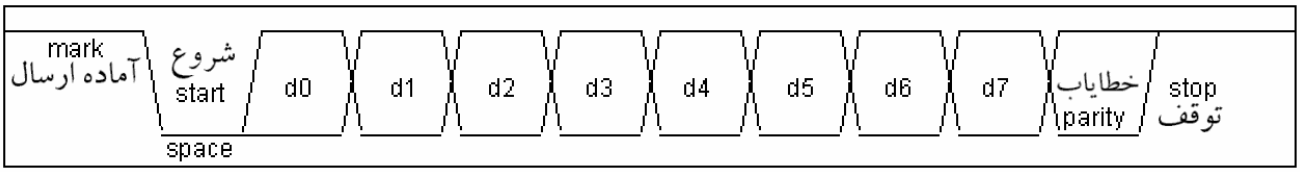
روش نرم افزاری

روش کنترل جریان نرم افزاری به نام Xon/Xoff شهرت دارد و شامل دو کاراکتر Xon و Xoff است.

Xon از کاراکتر ASCII 17 و Xoff از کاراکتر ASCII 19 استفاده میکند.مودم یک بافر کوچک برای در یافت داده ها در اختیار دارد که هر زمان این بافر پر شد با ارسال یک کاراکتر Xoff به فرستنده اطلاع میدهد که دیگر داده ای ارسال نکند .زمانی که بافر آن خالی شد با فرستادن یک کاراکتر Xon به فرستنده اعلام میکندکه ادامه داده ها را بفرستد.

امتیازی که این روش نسبت به روش سخت افزاری دارد این است که برای کنترل جریان داده نیازی به سیم های اضافی نیست،و ارتباط تنها با دو سیم RXD/TXD بر قرار میشود. عیب این روش کند بودن آن است.در پروژه های مربوط به میکرو کنترلر ها به دلیل محدودیت پایه ها و اجتناب از سیم کشی های زیاد از روش دوم استفاده میشود.

در شکل زیر یک مرحله ارسال داده را مشاهده میکنید.



فصل دوم

ارتباط میکرو و Hyper Terminal

مقدمه:

از آن جهت که در کاربردهای مهندسی لازم میشود که اطلاعات پارامتر های محیطی نظیر دما،فشار،شدت نور و ...را وارد کامپیوتر کنیم تا بتوان به کمک قدرت پردازش کامپیوتر و هم چنین ابزارهای گرافیکی مو جود در آن روند تغییرات آنها را مشاهده کرده و با دادن دستورات لازم از کامپیوتر به پروسه تغییراتی را درروند اجرای آن انجام داد، وجود ابزارهایی که میتوانند این ارتباط را برقرار کنند اهمیت زیادی دارد.

این تغییرات میتواند باز یا بسته شدن یک شیر تغییر وضعیت یک در، روشن یا خاموش شدن یک وسیله گرمایشی یا ... باشد.موارد گفته شده بیشتر در ضمینه مانیتورینگ و پروژه های کنترلی کاربرد دارد ولی گاهی مواقع اتصال یک طرفه است یعنی ما فقط اطلاعات را از بیرون وارد کامپیوتر کرده و از قدرت پردازش آن کمک میگیریم. این روش بیشتر در فعالیت های تحقیقاتی کار برد دارد.به این صورت که ما پس از وارد کردن اطلاعات به کامپیوتر از نرم افزارهای قدرتمند محاسباتی نظیر MATLAB برای پردازش داده ها،پیش بینی تغییرات آنها و هم چنین به کار بردن آنالیز های مختلف نظیر آنالیز فوریه روی سیگنالها کمک گرفت.

در این قسمت ارتباط دو طرفه بین میکرو avr و کامپیوتر از طریق نرم افزار Hyper Terminal برسی شده و یک برنامه به زبان بیسیک برای این میکروکنترلر نوشته شده است.

ار تباط میکرو و MATLAB در فصل بعد برسی میشود.

توصیف سخت افزار ساخته شده

قسمت اصلی سخت افزار تشکیل شده از یک میکرو AVR به شماره ATMEGA8 هست.این کامپیوتر کوچک اطلاعات را از سنسورها دریافت کرده و بعد از تبدیل به واحد اصلی خود از طریق ارتباط سریال برای کامپیوتر ارسال میکند. در زیر مشخصات این میکرو کنترلر را مشاهده میکنید.

ویژگی های این میکرو :

کارای بالا و توان مصرفی کم

دارای ۱۳۰ دستور که اکثر انها در یک سیکل اجرا میشوند

حداکثر کریستال مورد استفاده درatmega8 - ۱۶ مگاهرتزو atmega8L- ۸ (در این پروژه از نوع ATmega8استفاده شده است)

8k بایت حافظه فلش داخلی قابل برنامه ریزی

این حافظه میتواند تا ۱۰۰۰۰ بار نوشته و پاک شود (قابلیت پروگرم کردن تا ۱۰۰۰۰ بار)

512 بایت eeprom حافظه داخلی برای ذخیره اطلاعات

این حافظه میتواند تا ۱۰۰۰۰۰۰ بار نوشته و پاک شود

قفل برنامه داخل حافظه برای جلوگیری از خواندن ان

- -خصوصیات جانبی:

دارای دو تایمر کانتر با کانال pwm

۸ کانال مبدل انالوگ به دیجتال (در این پروژه از دو کانال آن استفاده شده است)

یک مقایسه کننده انالوگ داخلی

Usart قابل برنامه ریزی

Watchdog قابل برنامه ریزی با اسیلاتور داخلی

برای برنامه ریزی پروگرام کردن(داخل مدار) هنگامی که میکرو داخل مدار است با پروگرامر isp میتوانید میکرو را برنامه ریزی کنید

خصوصیات ویژه میکرو:

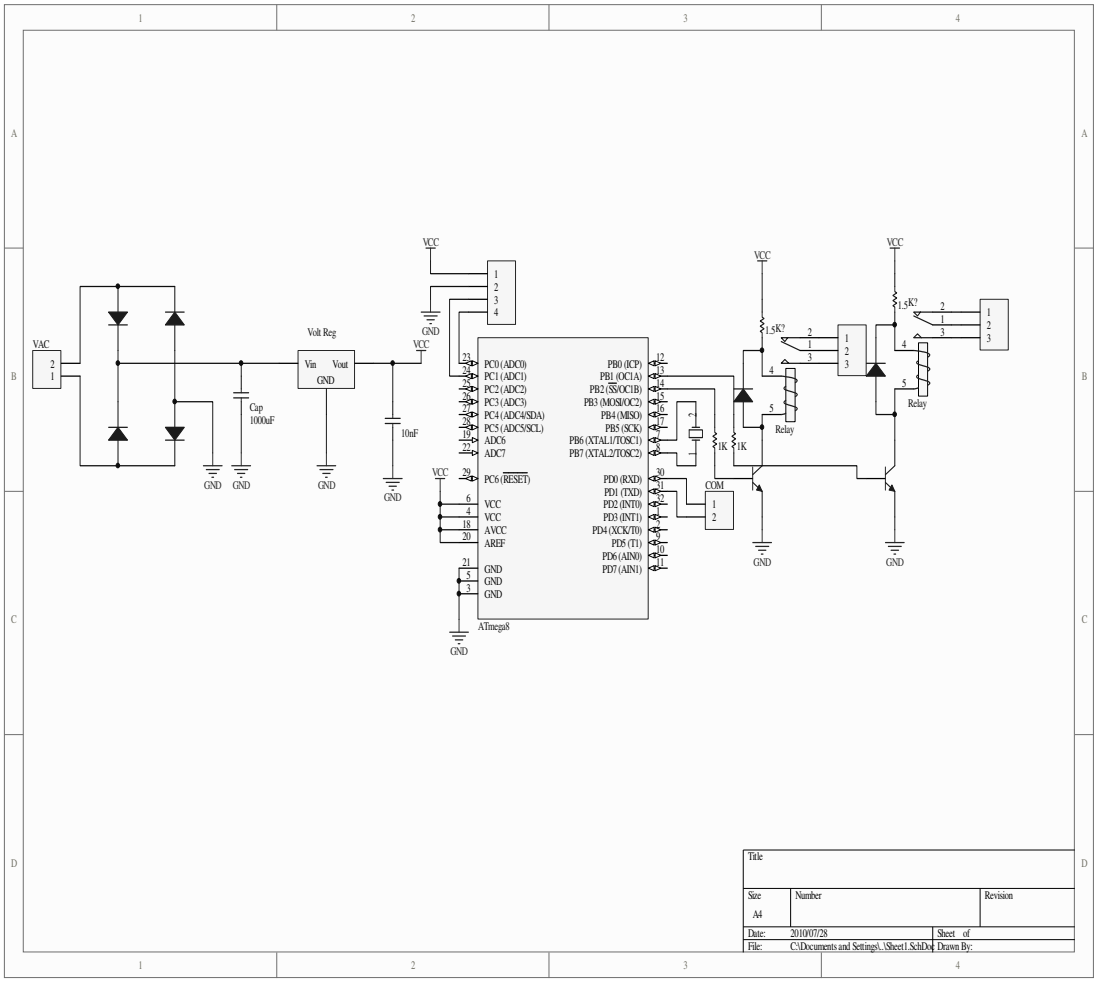
ریست شدن میکرو بعد از روشن شدن

دارای ۵ مد در حالت بیکاری برای مصرف کمتر انرژی و راندمان بیشتر

منبع وقفه داخلی و خارجی

دارای نوسان ساز داخلی کالیبره شده (حداکثر فرکانس این نوسان ساز ۸ مگا هرتز است)

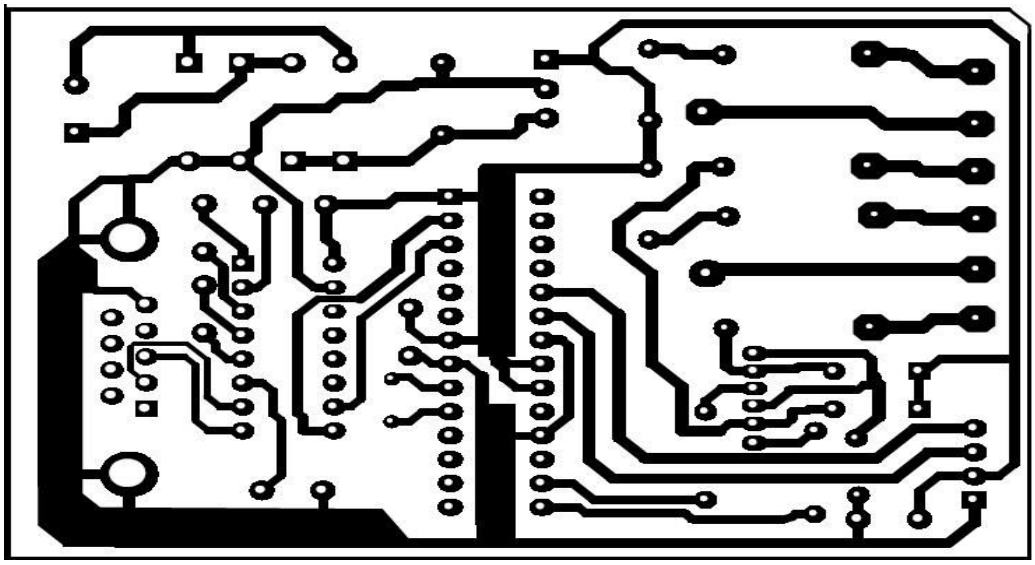
توضیحات بیشتر در datasheet میکرو که قسمتی از آن در فصل ۴ آورده شده قابل دسترس است.



تصویر شماتیک پروژه

در مدار بالا قسمت تبدیل سطح ولتاژ کشیده نشده، این قسمت در فصل قبل برسی شده بود.

تصویر PCB در زیر قرار داده شده است



تصویر PCB پروژه

تشریح قسمتهای مختلف سخت افزار:

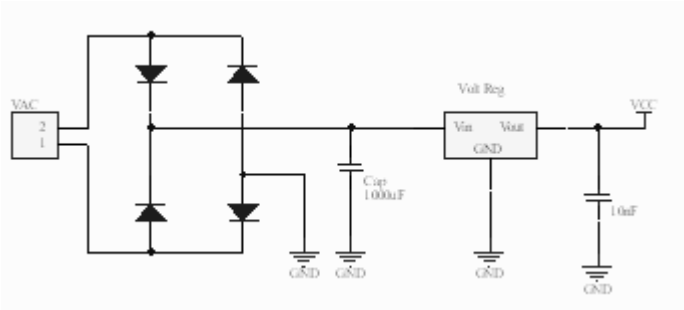
قسمت اول منبع تغذیه: این مدار از یک پل دیودی در ورودی و یک ic به شماره 7805و چند خازن تشکیل شده است. ابتدا سیگنال ورودی بعد از یک سو شدن توسط چهار دیود که در آرایش پل قرار گرفتند برای از بین بردن ریپل از یک خازن 1000uf استفاده شده است.سپس برای تثبیت ولتاژ روی ۵ ولت از یک ic استفاده شده.این قطعه که تغریبا در تمامی پروژه های دیجیتال قرار دارد دارای جریان خروجی حد اکثر نیم آمپر را داراست.در خروجی ic یک خازن عدسی برای از بین بردن نویز های فرکانس بالا که خازن اول موفق به حذف آنها نشده کمک گرفته شده. در صورت استفاده نکردن ار این خازن نویز های ورودی از منبع تغذیه باعث ریست شدن میکرو میشود .

استفاده از پل دیود در ورودی این امکان را به ما میدهد که علاوه بر موج های dc بتوان از سیگنالهای ac نیز برای تغذیه استفاده کرد.دامنه سیگنال ورودی باید حد اقل به اندازه ۷ ولت باشد تا در خروجی قسمت تغذیه ولتاژ ۵ ولت را داشت و مدار کار کند.

جریان ورودی مدار در حالتی که رله ها خاموش هستند از چندده میلی آمپر تجاوز نمیکند ولی با روشن شدن رله ها جریان حدود 200mA خواهد شد .جون جریان روشن شدن رله ها نیز از ic تثبیت کننده کشیده میشود بهتر است برای آن از یک خنک کننده الومینییمی استفاده کرد.

در حالت خاموش بودن رله ها با توجه به جریان مصرفی کم دستگاه میتواند روزها با یک باتری کار کند. میتوان برنامه ی میکرو را به صورتی نوشت که دستگاه با روشن شدن هر چند دقیقه اطلاعات سنسورها را در حافظه eprom خود ذخیره کند و با فشار دادن کلیدی این اطلاعات را برای کامپیوتر ارسال کند.

در شکل زیر قسمت تغذیه به صورت مجزا نشان داده شده است.



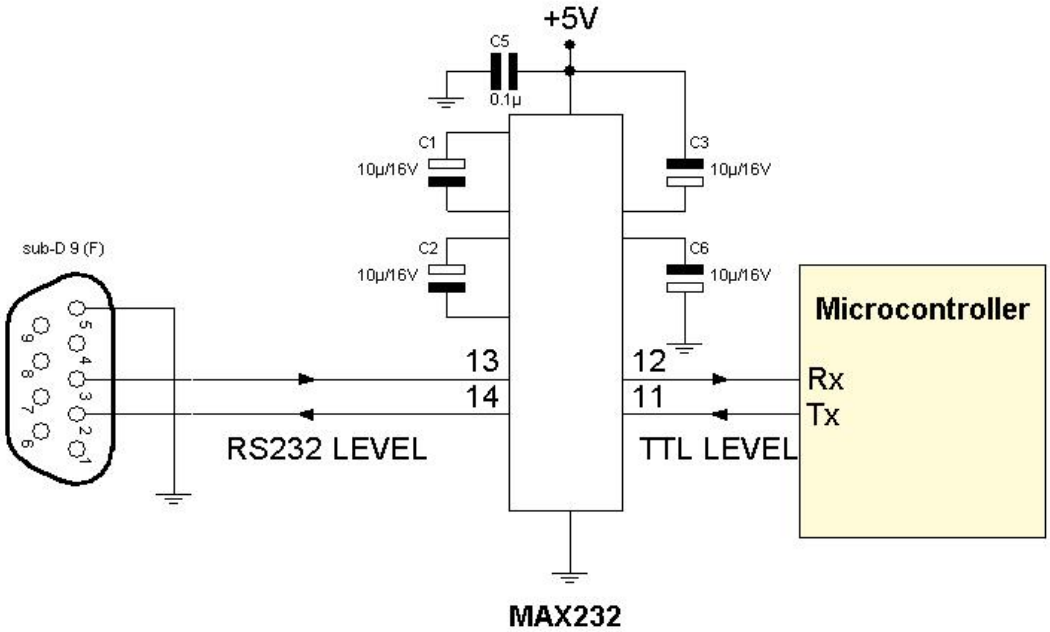
قسمت دوم رله ها : در این مدار از دو رله استفاده شده است که هر کدام دارای کنتاکت های باز و بیته است و هر دو کنتاکت توسط ترمینال در اختیار کاربر قرار داده شده است .

این رله ها از دو پایه پورت B فرمان میگیرند.چون جریان خروجی این پورتها کم است و نمیتوانند مستقیما رله ها را فعال کنند از دو ترانزیستور برای تقویت جریان استفاده شده است. این ترانزیستورها که به شماره bc 337 هستند با فعال شدن (یک شدن) پایه میکرو از طریق مقاومت یک کیلو روی بیس جریانی حدود 4mA به آنها تزریق میشود. با ورود جریان و اشباع ترانزیستور رله روشن میشود. چون ورودی رله ها از یک سیم پیچ تشکیل شده با روشن شدن رله و بر قراری جریان در سیم پیچ انرژی مغناطیسی در آن ذخیره میشود.این انرژی در هنگام خاموش شدن ترانزیستور به صورت ولتا ژ معکوس زیادی آزاد میشود و ممکن است باعش آسیب دیدن ترانزیستور شود لذا از یک دیود به شماره1N4148 که به صورت معکوس با رله موازی قرار گرفته شده .

قسمت سوم شامل ic به شماره max232 و یک پورت com:

این ic همان گونه که در فصل قبل گفته شد ولتاژ ۰ و۵ ولت خروجی میکرو را به ولتاژ های استاندارد پروتکل rs232 تبدیل میکند.و در واقع یک بافر ارتباط سریال با مدارات دیجیتال است.

در زیر این قسمت این IC به صورت مجزا نشان داده شده است.



تصویر-ئشط232- IC ومدارات مرتبط

برسی توابع برنامه بیسیک برای ارتباط با پورت سریال

در این قسمت ابتدا دستورات مربوط به ارتباط سریال در نرم افزار Bascom Avr برسی شده ودر آخر یک مثال آورده شده، سپس برنامه نوشته شده برای ارتباط سریال با hyper terminal به همراه توضیحات قرار داده شده است.

تعیین نرخ انتقال دیتا:

```
$BAUD=VAR
```

این دستور میزان انتقال دیتا در ثانیه را مشخص میکند وباید در هر دو وسیله ای که به هم متصل میشوند یکی باشد (در غیر این صورت ارتباط کار نمیکند) بهتر است نرخ انتقال دیتا در مناطق دارای نویز کمتر انتخاب شود.

دستور: PRINT

```
PRINT  VAR
```

توسط این دستور میتوان داده یا متغیری را به پورت سریال ارسال کرد .

دستور: PRINTBIN

```
PRINTBIN VAR
```

توسط این دستور متغیر VAR به باینر تبدیل شده سپس به پورت سریال ارسال میشود.

دستور: WAITKEY

```
VAR=WAITKEY()
```

این دستور تا زمانی که متغییر توسط دستگاه دیگر به پورت سریال ارسال شود منتظر میماند و پس از دریافت متغیر آن را در varقرار میدهد وبرنامه از خط بعد ادامه می یابد.

دستور: INKEY

```
VAR=INKEY()
```

این دستور مقدار اسکی کاراکتر دریافت شده از پورت سریال را برمیگرداند.

دستور: INPUTBIN

```
INPUTBIN  VAR
```

این دستور داده باینری را از پورت سریال میگیرد و در متغیر VAR قرار میدهد.

دستور: INPUTHEX

```
INPUTHEX  VAR
```

این دستور داده هگز رااز پورت سریال دریافت میکند و در متغیر VAR قرارمیدهد. مانند:
دو پایه txd و rxd میکرو نقش دریافت و ارسال داده را در حالت پیشفرض برعهده دارند ، با دستور زیر میتوان این دو پایه را به پایه های دلخواه تغییر داد:

```
Open "comx.y:$baud,8,n,1" For Output/input As #q
```

comx.y:نام پورت و پایه ای است که باید به عنوان txd یا rxd جدید عمل کند.

$baud:نرخ داده عبوری از پایه را نشان میدهد ، این مقدار باید با نرخ انتقال دیتا ی اصلی برابر باشد.

Output/input : پایه میتواند وردی داده (rxd)یا خروجی داده (txd)باشد.

Q : شماره کانال را مشخص میکند .

مانند:

```
Open "comd.1:19200,8,n,1" For Output As #1
Open "comd.0:19200,8,n,1" For Input As #2
```

در مورد بالا portd.1 به عنوان txd و portd.0 به عنوان rxd در نظر گرفته شده است ، همچنین نرخ انتقال داده برابر با ۱۹۲۰۰ است.

مثال برای برقراری ارتباط بین دو میکرو:

```
$regfile = "m32def.dat" : $crystal = 1000000
$baud = 9600
Config Portb = Input : Config Porta = Output
Dim A As Byte , Q As Byte
W:
Q = Pinb : Printbin Q
A = Inkey() : Porta = A
Goto W
end
```

میکرو ۲:

```
$regfile = "m32def.dat" : $crystal = 1000000
$baud = 9600
Config Portb = Input : Config Porta = Output
Dim A As Byte : Dim Q As Byte
W:
Q = Pinb : Printbin Q
A = Inkey() : Porta = A
Goto W
End
```

از انجا که ارتباط دو طرفه میباشد (هر دو میکرو دقیقا مانند هم هستند)در برنامه نرخ ارسال دو میکرو مشابه است .

درخط اول میکرو و کریستال معرفی شده است که میکرو مگا ۳۲ و کریستال مورد استفاده ۱۰ مگا هرتز میباشد.

در خط دوم نرخ انتقال دیتا مشخص گردیده است ، مقدار ان ۹۶۰۰ است . نرخ انتقال دیتا باید در هر دو میکرو یکسان باشد.

در خط سوم پورت b به عنوان ورودی (برای اتصال کلید) و پورت a به عنوان خروجی (برای اتصال led)معرفی شده اند.

در خط چهارم دو متغیر از جنس بایت برای ذخیره مقادیر معرفی شده است .

در خط پنجم شروع یک حلقه میباشد.

در خط ششم مقدار موجود بر روی پورت b در متغیر q ریخته میشود و سپس با دستور Printbin Q به پورت سریال فرستاده میشود.

در خط هفتم مقدار گرفته شده از پروت سریال در متغیر a ریخته میشود و بعد متغیر a بر روی پورت aریخته میشود.

خط هفتم پایان حلقه میباشد ، هنگامی که cpu میکرو به این خط رسید به برچسب w پرش میکند.

در حالتی که هیچ یک از کلید ها یک نشده اند ، مقدار q صفر دسیمال و &b00000000 باینری است ، حال اگر هر یک از کلید ها فشرده شود مقدار q تغیر میکند.

برنامه نوشته شده برای ارتباط با Hyper Terminal

```
'------------------------------------------------------------------
Title        : monororing and control temprecur with serial port '
Target       : Amega8 '
Program code  : BASCOM AVR '
: .Hardware req '
Use Hyperterminal or any other terminal program '
.to send the commands * '
'------------------------------------------------------------------
$regfile = "m8def.dat"  .۱
$crystal = 16000000  .۲
baud = 9600$  .۳
Config Adc = Single , Prescaler = Auto , Reference = Avcc  .۴
Start Adc  .۵

Dim A As Word , Sum As Word , N As Byte , Temp1 As Byte  .۶
Dim Temp2 As Byte,Max1 As Byte , Max2 As Byte  .۷
Config Portc = Input  .۸
Config Portb = Output  .۹
Wait 1 .۱۰

Reset Portb.1 .۱۱
Reset Portb.2

Print "---------------AVR projects ------------" .۱۲
Print "Serial Port monotoring "
Print "serial port settings:9600-8-N-1"
 Print"-----------------------------------------"
Wait 1
Input "insert max tempereature for sensor 1 " , Max1 .۱۳
Print Max1 .۱۴
Input "insert max tempereature for sensor 2 " , Max2 .۱۵
Print Max2 .۱۶
Startt : .۱۷
Sum = 0 .۱۸
For N = 1 To 10 .۱۹
A = Getadc(1)
Sum = Sum + A
```

```
Waitms 100
Next
Sum = Sum - 110 .۲۰
Sum = Sum / 16 .۲۱
Temp1 = Sum .۲۲
Sum = 0 .۲۳
For N = 1 To 10 .۲۴
A = Getadc(0)
Sum = Sum + A
Waitms 100
Next
Sum = Sum / 102 .۲۵
Temp2 = Sum .۲۶

Print "tempereature1= " ; Temp1 ; "      tempereature2= " ; Temp2 .۲۷
'------------------------------------------------------------------------

If Temp1 > Max1 Then
Set Portb.1
End If
If Temp2 > Max2 Then
Set Portb.2
End If

If Max1 > Temp1 Then
Reset Portb.1
End If
If Max2 > Temp2 Then
Reset Portb.2
End If

Wait 2
Goto Startt
End
```

توضیح برنامه:

نمای کلی برنامه این است که میکرو کنترلر ابتدا اطلاعات مربوط به حد اکثر مقدار مجاز سنسور ها را از کاربرگرفته و سپس هر یک ثانیه وضعیت سنسور ها را برای کامپیوتر ارسال میکند.و اگر مقدار خروجی سنسور از مقدار مجاز تعیین شده بیشتر شد رله ی مربوطه را فعال میکند و با تغییر وضعیت رله ها دستگاه کنترل کننده متصل شده به مدار برای بازگشت شرایط مطلوب تغییروضعیت میدهد.

در زیر برنامه از ابتدا و به صورت خط به خط توضیح داده شده.

۱. در این خط شماره میکرو به کامپایلر معرفی شده است.
۲. فرکانس کریستال متصل شده به مدار معرفی شده که ۱۶ مگاهرتز است.این مقدار حد اکثر قابل استفاده برای این میکرو کنترلر است.
۳. در این خط نرخ ارسال داده در ارتباط سریال بیان شده که در دو دستگاه باید یکی باشد.
۴. در این خط مبدل آنالوگ به دیجیتال پیکر بندی شده است.برای استفاده از دستور getadc در مد single پیکر بندی شده و هم چنین انتخاب تقسیم فرکانسی برای ورودی مبدل در اختیار کامپایلر قرار داده شده تا با توجه به کریستال میکرو مقدار مناسبی را انتخاب کند.برای ولتاژ مرجع مبدل نیز از ولتاژ روی پایه avcc استفاده شده.این پایه در سخت افزار به ۵ ولت متصل شده است پس میتوان با توجه به ده بیتی بودن مبدل گفت با ازای خروجی ۵ ولت در سنسورها عدد ۱۰۲۳ از مبدل خوانده میشود.
۵. در خط بعد برای روشن شدن و برقراری تغذیه داخلی مبدل از این دستور استفاده شده است.
۶. در این خط چند متغییر معرفی شده که در طول برنامه از آنها استفاده شده است.
۷. در این خط نیز مانند خط بالا چند متغییر تعریف شده که دو متغییر max1 وmax2 قرار است حد اکثر مجاز دما را در خود ذخیره کنند و متغییر های temp1و temp2 دمای فعلی را در خود ذخیره میکنند.چون دما قرار نیست از صد درجه بالاتر برود و خروجی سنسورها نیز حد اکثر برای این دما حساب شده این متغییر ها از نوع بایت تعریف شده اند.
۸. در این خط portc که به عنوان ورودی مبدل آنالوگ به دیجیتال است به عنوان ورودی تعریف شده است.دو سنسور به پایه های اول و دوم این مبدل متصل شده اند.
۹. این دستور پایه هایی که دو رله به آنها متصل است را به عنوان خروجی تعریف میکند.
۱۰. این دستور یک ثانیه تا خیر ایجاد میکند.این تاخیر باعث میشود بعد از متصل شدن تغذیه دستگاه فورا ارسال اطلاعات صورت نگیرد.
۱۱. این دو خط رله ها را خاموش میکنند تا وضعیت اولیه آنها خاموش باشد.

۱۲. در این چند خط اطلاعاتی در مورد پروژه برای کامپیوتر ارسای میشود از قبیل نام پروژه و تنظیمات مربوط به ارتباط سریال .

۱۳. در این خط متنی برای گرفتن عدد مربوط به حد اکثر دما در جایی که سنسور اول قرار دارد ارسال شده و جواب دریافتی در متغییر max1 ذخیره میشود.

۱۴. در این خط برای اطمینان از انتقال صحیح عدد دریافتی مجدد نمایش داده میشود.

۱۵. در این خط متنی برای گرفتن عدد مربوط به حد اکثر دما در جایی که سنسوردوم قرار دارد ارسال شده و جواب دریافتی در متغییر max2 ذخیره میشود.

۱۶. در این خط برای اطمینان از انتقال صحیح عدد دریافتی مجدد نمایش داده میشود.

۱۷. در این خط یک لیبل قرار داده شده که با اتمام یک سیکل برنامه مجدد از این خط دنبال شود.

۱۸. در این قسمت مقدار متغییر sum که از نوع word است صفر میشود کاربرد ابن متغییر در خط بعد دیده میشود.

۱۹. در این خط و دستورات موجود در داخل حلقه for به تعداد ده مرتبه با فاصله زمانی ۱۰۰ میلی ثانیه مقدار خروجی سنسور اندازه گیری شده و هر بار در متغییر sum جمع میشود.دلیل گرفتن ده نمونه بالا بردن دقت نمونه برداری است. جواب نهایی از معدی این ده نمونه محاصبه شده است.

۲۰. در این خط مقدار offset خروجی سنسور اول از آن کم شده. با آزمایش مشخص شد که در دمای صفر درجه خروجی مبدل عدد ۱۱ را نشان میدهد و با توجه به جمع شدن ۱۰ مقدار از این خروجی عدد ۱۱۰ از مقدار sum کم شد تا در دمای صفر درجه مقدار آن صفر شود.

۲۱. در این خط طبق نتیجه آزمایش که بیان میکرد به ازای هر ده درجه دما ۱۶واحد به عدد خروجی مبدل افزوده میشود مقدار sum که حاصل ده بار جمع کردن خروجی مبدل بود بر ۱۶ تقسیم شد.

۲۲. در این خط دمای محیط اول که در متغییر sum قرار دارد به متغییر temp1 فرستاده میشود که برای ارسال از آن استفاده شود.

۲۳. در این خط مجدد مقدار sum صفر شد تا برای گرفتن مقادیر سنسور جدید آماده باشد.

۲۴. در این خط با استفاده از یک for عدد تولید شده توسط خروجی مبدل ده بار با هم جمع میشود و حاصل از میانگین این ده بار جمع محاسبه میشود

۲۵. چون در ورودی مبدل دوم از یک پتانسیومتر استفاده شده مقدار offset نداریم و چون میخواهیم وقتی که ولوم در نهایت خود است عدد ۱۰۰ را به عنوان دما داشته باشیم عدد نهایی را بر ۱۰۲ تقسیم میکنیم.

۲۶. در این قسمت دمای نهایی مبدل دوم در متغییر temp2 قرار میگیرد.این متغییر میتوانست حذف شود و مقدار sum نمایش داده شود ولی برای ساده تر شدن فهم برنامه استفاده شده.

۲۷. در این خط دو مقدار دمای بدست آمده برای نمایش در کامپیوتر به پورت سریال ارسال شده اند.

در خطهای بعدی دو مقدار بدست آمده برای دما با مقادیر حد اکثر تعیین شده مقایسه میشوند و در صورت بزرگتر بودن از حد مجاز رلهی مربوطه فعا میشود.سپس برنامه با دو ثانیه تاخیر به لیبل start پرش میکند تا عملیات از سر گرفته شود.مجموع تاخیر های برنامه برابر ۴ ثانیه میشود یعنی هر چهار ثانیه عمل ارسال داده انجام میشود.

دلیل استفاده نکردن از محدوده دمایی به عنوان هیسترزیس که معمولا در مدارات کنترل دما استفاده میشود وجود متغییر ها به صورت byte است.چون این متغییر ها اعشار ندارند و فقط مقادیر صحیح دارند خود یک هیسترزیس را برای ما ایجاد میکند.

در قسمت بعد تنظیمات مربوط به قسمت کامپیوتری پروژه قرار داده شده است.

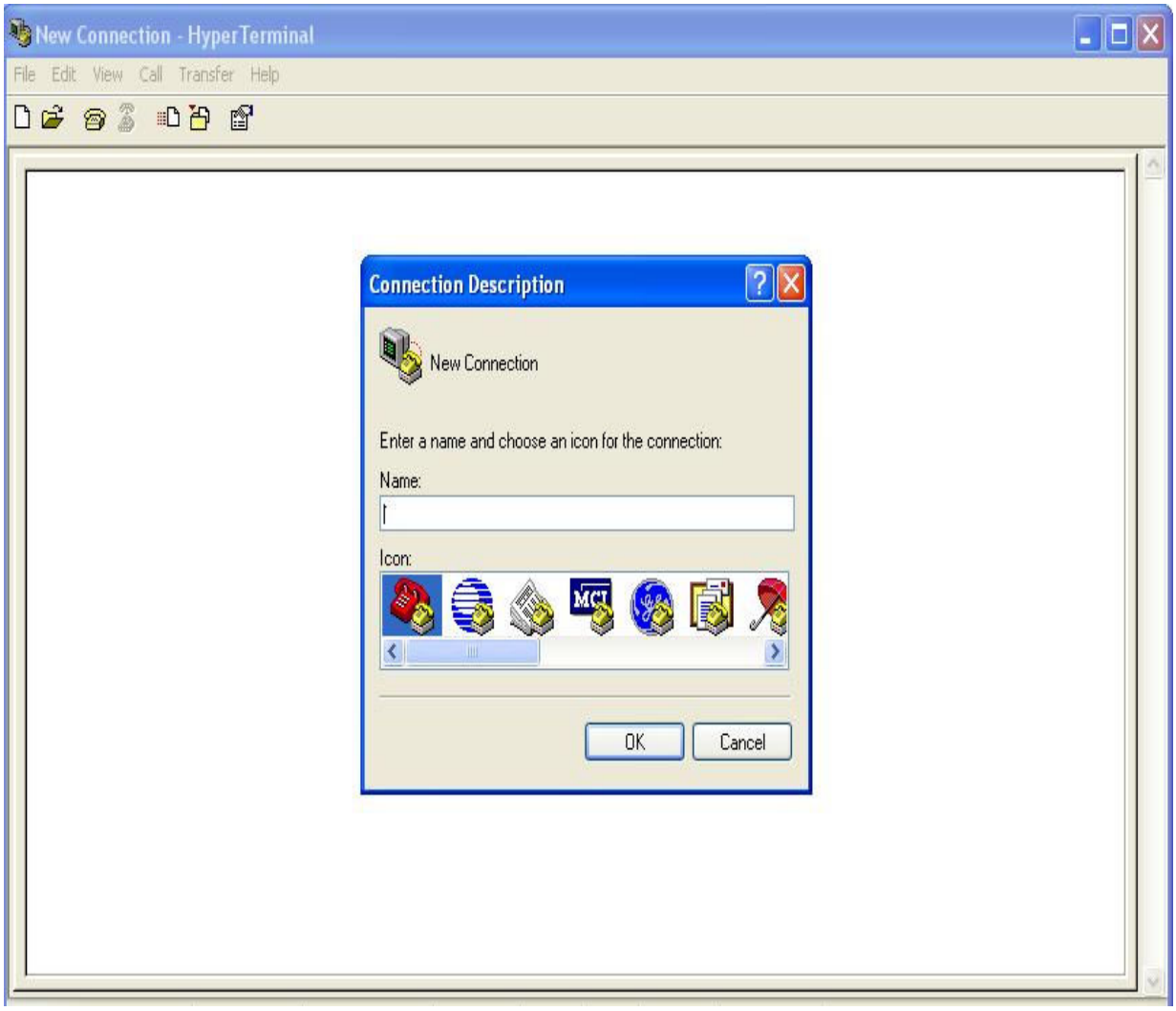
نحوه انجام تنظیمات Hyper Terminal در windows XP

این برنامه که به صورت رایگان در ویندوز XP قرار دارد برای بر قراری ارتباط با پورت com و همچنین مودم میباشد.

از مسیر زیر میتوانید این بر نامه را باز کنید.

All programs>accessories>communications>Hyper terminal

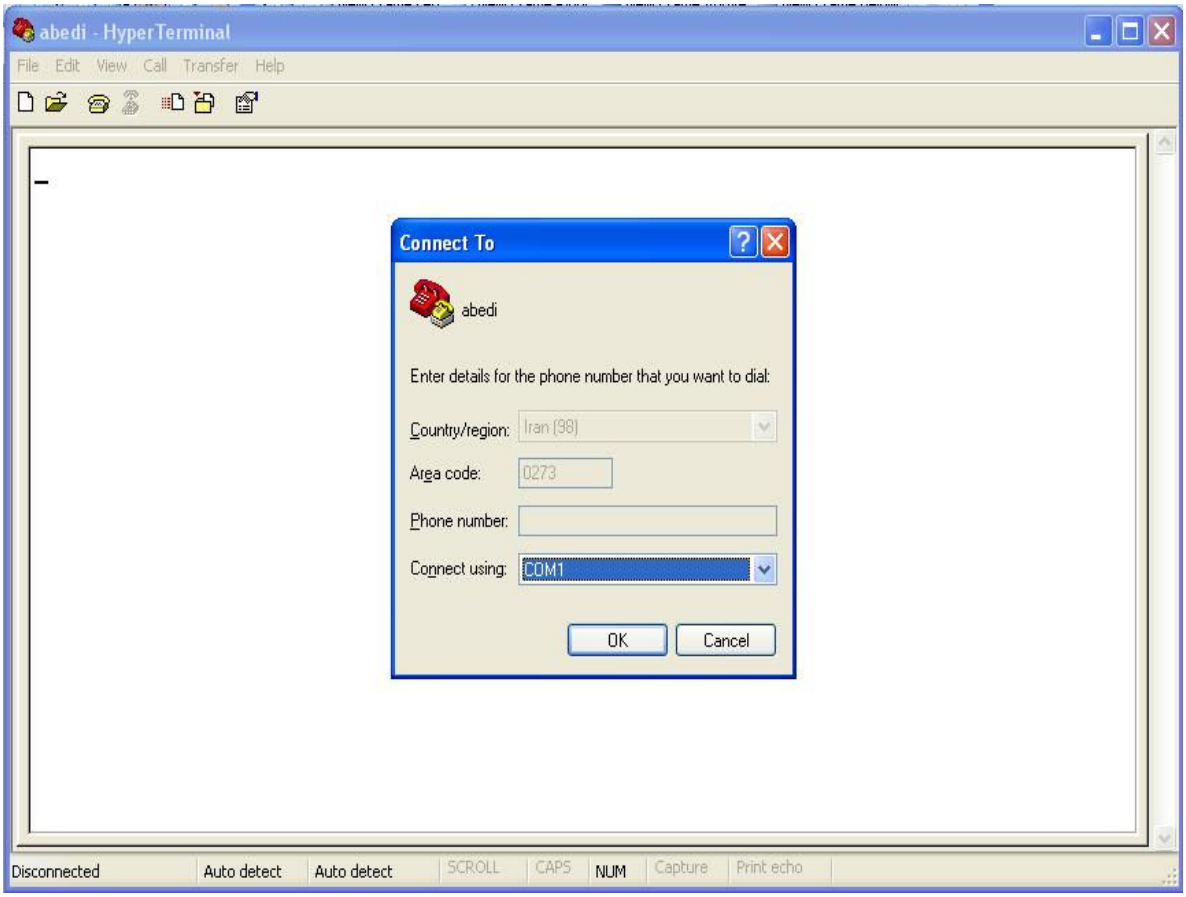
بعد از اجرای برنامه پنجره زیر نمایش داده میشود



صفحه اول Hyper Terminal

با گذاشتن یک نام دلخواه okرا بزنید

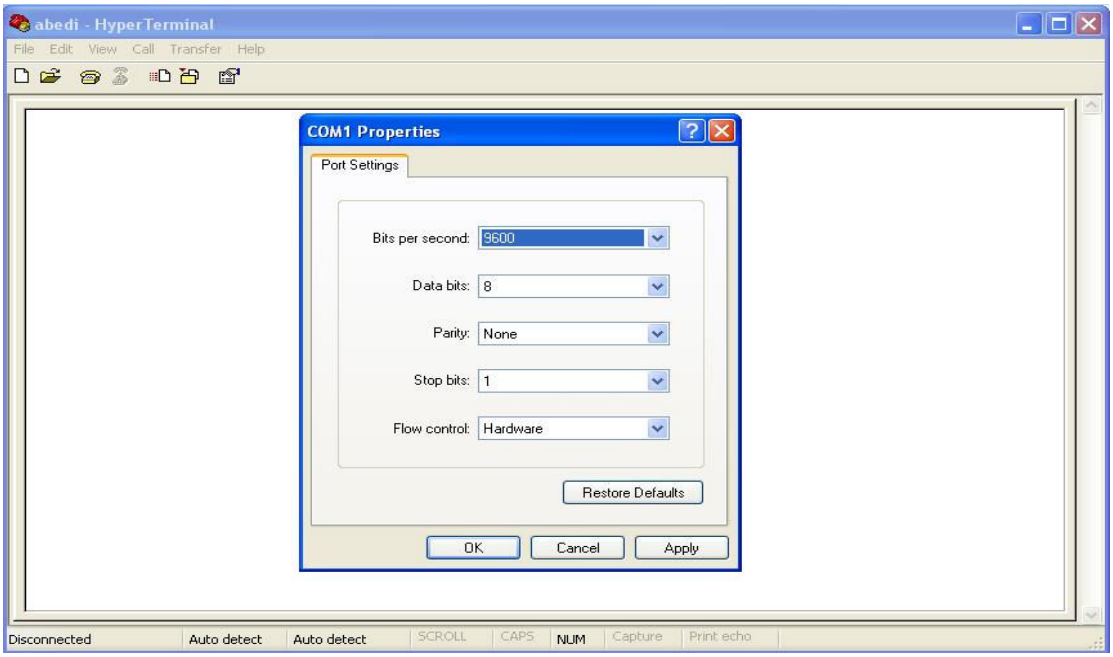
سپس پنجره ای مانند شکل بعد نمایش داده خواهد شد.



تنظیم شماره پورت

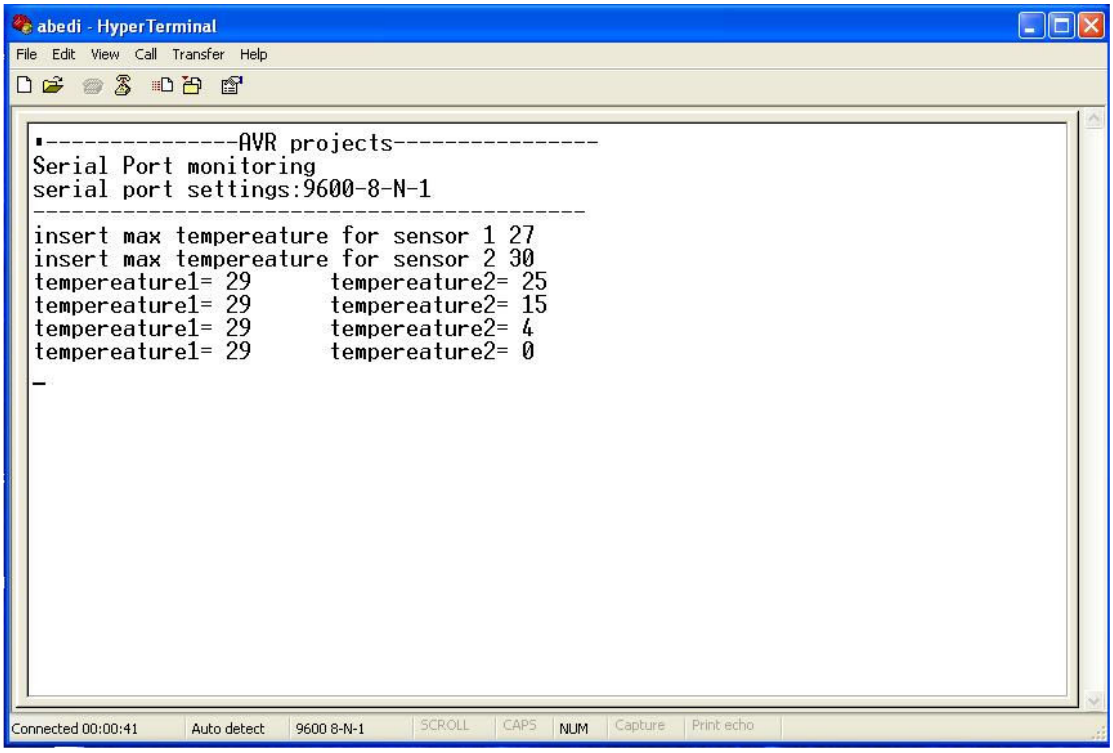
در پنجره فوق شماره پورتی را که میخواهیم با آن ارتباط بر قرار کنیم را انتخاب میکنیم.

بعد از آن پنجره ای مانند شکل زیر نمایش داده میشود.



صفحه مربوط به تنظیمات ارتباط سریال

در قسمت اول باید نرخ ارسال داده را مشخص کرد که باید با عدد انتخاب شده در برنامه میکرو یکی باشد.۰ در گزینه بعدی تعداد بیتهای ارسالی در هر ارسال مشخص میشود و باید با تنظیمات انجام شده در میکرو یکیان باشد.در قسمت های بعد نیز باید وجود یا عدم وجود بیت خطاو صفر یا یک بودن بیت انتهای ارسال و هم چنین کنترل سخت افزار را مشخص کرد.



صفحه ارسال و دریافت اطلاعات

بعد از روشن کردن سخت افزار میکرو از شما میخواهد مقدار حد اکثر برای هر سنسور را مشخص کنید که با مشخص کردن این مقادیر چنانچه دما از مقدار مشخص شده بیشتر باشد رله مربوطه روشن شده و سیستم خنک کننده روشن میشود.

فصل سوم

ارتباط سریال با MATLAB

مقدمه

بدون شک MATLAB یکی از قوی ترین نرم افزارهای مهندسی است. نرم افزاری که علاوه بر مهندسها خیلی از ریاضی دانها، پزشکها و ... را به سوی خود جذب کرد.شناخت این نرم افزار به دلیل سادگی کار با آن و داشتن توابع آماده و دستورات ریاضی بسیار اهمیت دارد.

در کاربردهای علمی بعضی از مواقع لازم میشود که سیگنالهای محیطی را مورد پردازش قرار داد تغییرات آنها را برسی کرد یا آنها راروی نموداری رسم کرده و با هم مقایسه کرد و از همه مهمتر اعمال ریاضی روی آنها انجام داد مثل گرفتن تبدیل فوریه از داده ها و بدست آوردن طیف فرکانسی آنها .در این گونه مواقع یک دستگاه جانبی وظیفه دریافت اطلاعات را بر عهده دارد.پس از جمع آوری داده ها آنها را باید به MATLAB انتقال داد . از این رو سخت افزار ساخته شده که در فصل قبل برسی شد میتواند به عنوان یک نمونه گیر عمل کند به این صورت که میکروکنترلر اطلاعات را گرفته آنها را ذخیره کرده سپس در مواقعی خاص آنها را از طریق ارتباط سریال به کامپیوتر ارصال میکند یا این که اطلاعات را به صورت هم زمان بعد از گرفتن از سنسور به کامپیوتر میفرستد.در این فصل برای هر دو مورد برنامه نوشته شده و مورد برسی قرار گرفته است.

در ابتدا این فصل دستورات MATLAB در ارتباط با پورت سریال برسی شده و بعد از آن به پروژه های عملی پرداخته شده است.

پورتهای سریال در MATLAB :

در رابطه ارتباط با پورتهای سریال MATLAB برنامه ها و دستورات متعدد و موثری دارد .این دستورات رابطه با پورت سریال را خیلی راحت می کند. قبل از معرفی دستورات مشخصات ارتباطی که توسط HELP این نرم افزار نوشته شده مورد برسی قرار میگیرد.

BaudRate

سرعتی که بیتها فرستاده می شوند را مشخص می کند .

BaudRate را بر حسب بیت بر ثانیه پیکربندی می شود.بیت های ارسالی شامل بیت شروع ، بیت های داده ، بیت توازن ،(اگر به کار روند) و بیت های توقف هستند.هر چند اگر بیت های داده ذخیره شوند .

توجه: برای اینکه بتوان خواندن و نوشتن موفقی داشت هم کامپیوتر و هم دستگاه جانبی باید در یک باود تنظیم شوند .در محیطهای پر نویز و یا مواقعی که طول کابل زیاد است بهتر است از baud های کمتر استفاده شود.

Parity

نوع مقابله توازن را مشخص می کند. که خالی ، زوج ، فرد ،علامت و یا فاصله است .

دستور serial :

شئ پورت سریال را ایجاد می کند. مثلا می توان نوشت :

```
s = serial('COM1');
```

که شئ s را ایجاد می کند.این دستور باید در ابتدای شروع کار با پورت نوشته شود و مشخص میکند ما قصد ارتباط با کدام پورت سریال را داریم.

دستور fopen:

شئ پورت سریال را به وسیله متصل می کند. یعنی در حقیقت پورت را باز می کند.تا قبل از استفاده از این دستور داده های ارسال شده به پورت قابل خواندن نیستند ولی بعد از اجرای این دستور بافر پورت فعال شده و داده ها در آن ذخیره میشوندو تا زمانی که از دستور خواندن پورت استفاده نشود داده ها در بافر قرار میگیرند .با داشتن این مزیت دیگر نگران این نیستسم که ممکن است داده ای به پورت ارسال شود و ما در آن هنگام پورت را نخوانیم و داده از بین برود.

دستور fclose :

شئ پورت سریال را از وسیله جدا می کند.

databits

Databits=value

این دستور تعداد بیت های ارسالی را مشخص میکند که میتواند هر کدام از مقادیر8-7-6-5 باشد

توابع و مشخصات نوشتن در پورت سریال

توابع نوشتن

Fprintf

این تابع برای نوشتن متن در پورت سریال استفاده میشود .

در دستور مقابل cmd متنی است که قرار است ارسال شود

```
fprintf(obj,'cmd')
```

fwrite

این تابع برای ارسال باینری در پورت سریال استفاده میشود.به کمک این تابع میتوان انواع متغییرهای تا ۶۴ بیتی را فرستاد

```
fwrite(obj,A)
```

stopasync

با این دستور میتوان عملیات خواندن غیر هم زمان و نوشتن غیر هم زمان که دو دستور بالا بودند را غیر فعال کرد

```
stopasync(obj)
```

مشخصات نوشتن :

:BytesToOutput

شامل تعداد بایت هایی است که در بافر خروجی جریان دارند.با باز کردن پورت و نوشتن دستور fopen(obj) مقدار پیش فرض صفر میشود و بید ار آن میتوان این مقدار را تغییر داد. مقدار آن نباید در ارتباط هم زمان میکرو و وسیله جانبی از صفر تغییر کند .

:OutputBufferSize

سایز بافر خروجی را مشخص می کند . مقدار پیش فرض بافر خروجی نیز ۵۱۲ عدد از نوع متغییر double است.

:Timeout

حد اکثر زمان برای اینکه عملکرد خواندن و یا نوشتن کامل شود را مشخص می کند .مقدار پیش فرض این زمان ۱۰ ثانیه است و میتوان آن را تغییر داد. از زمانی که عمل خواندن یا نوشتن انجام میگیرد یک تایمر فعال شده و اگر مقدار این تایمر به عدد timeout رسید و هنوز عمل ارسال و دریافت صورت نگرفته بود برنامه پیغام خطا میدهد.در دستور fscanf(obj) اگر بافر خالی باشد بعد از مدت زمان timeout داده ای به پورت ارسای نشود تابع [] را برمیگرداند و پیغام خطای زیر را میدهد.

.Warning: A timeout occurred before the Terminator was reached

:TransferStatus

اگر یک عملکرد خواندن و یا نوشتن آسنکرون در پروسه باشد را نمایش می دهد.در عالتی که خواندن و نوشتنی صورت نمی گیرد مقدار برگشتی idle است و در مواقعی که یک عمل نوشتن آسنکرون صورت میگیرد مقدار برگشتی write است در حالت خواندن آسنکرون مقذار برگشتی read خواهد بود .

:ValuesSent

تعداد مقادیر ارسال شده توسط پورت سریال را نشان میدهد.با باز کردن پورت یعنی استفاده از دستور fopen(obj) مقدار این متغییر صفر میشود .

توابع و مشخصات خواندن از پورت سریال

توابع خواندن از پورت سریال

:Fgetl

این تابع یک خط داده ارسالی را بر میگرداند

```
tline = fgetl(obj)
```

fgetl

این دستور همانند دستور بالا میباشد با این تفاوت که علامت پایان متن را نیز به همراه متن بر میگرداند.

```
tline = fgets(obj)
```

fread

این دستور عدد ارسالی به پورت را برمیگرداند.

```
A = fread(obj)
```

fscanf

این دستور برای خواندن از پورت است و اطلاعات روع پورت را ابتدا به متن تبدیل کرده و سپس برمیگرداند.

```
A = fscanf(obj)
```

مشخصات خواندن:

:BytesAvailable

تعداد بایت های قابل دسترسی در بافر ورودی را نمایش می دهد .در واقع همان اطلاعات دریافت شده ای که هنوز خوانده نشدند.بعد از باز کردن پورت این مقدار صفر میشود.توجه شود که این دستور فقط در ارتباط asynchron خروجی دارد و در ارتباط سنکرون همواره خروجی آن صفر است.

:InputBufferSize

سایز بافر ورودی را برحسب بایت مشخص می کند.و مقدار پیش فرض آن ۵۱۲ است.

:Timeout

این دستور برای قسمت خواندن و نوشتن مشترک است و در قسمت قبل برسی شد.

:TransferStatus

اگر یک عملکرد خواندن و یا نوشتن آسنکرون در پروسه باشد را نمایش می دهد.در عالتی که خواندن و نوشتنی صورت نمی گیرد مقدار برگشتی `idle` است و در مواقعی که یک عمل نوشتن آسنکرون صورت میگیرد مقدار برگشتی `write` است در حالت خواندن آسنکرون مقذار برگشتی `read` خواهد بود .

:ValuesReceived

مجموع تعداد مقادیر خوانده شده از وسیله را نمایش می دهد .خروجی این تابع از نوع عدد و مقدار پیش فرض آن صفر است.

پروژه ارتباط همزمان میکرو با MATLAB

در این پروژه ازسخت افزار معرفی شده در فصل قبل استفاده شده است.سخت افزار هر چند میلی ثانیه اطلاعات یک سنسور را خوانده و از طریق پورت سریال ارسال میکند .

در آن طرف دیگر MATLAB این داده ها را در یک بردار ذخیره کرده و بعد از اتمام کار آنها را روی یک نمودار نشان میدهد .برای داشتن تغییرات در حد وسیع به جای استفاده از یک سنسورر به عنوان ورودی مبدل آنالوگ به دیجیتال از یک مقاومت متغییر استفاده شده است .به این صورت که میکرو صد بار وضعیت این مقاومت را برسی کرده و عددی بین صفر تا ۱۰۲۴ را به آن نسبت میدهد و آن عدد را برای MATLAB ارسال میکند.

در ادامه ابتدا برنامه نوشته شدهبرای میکرو کنترل قرار دارد و بعد از آن برنامه MATLAB و در آخر نموداری که تغییرات این پتانسیومتر را نسبت به زمان نشان میدهد.

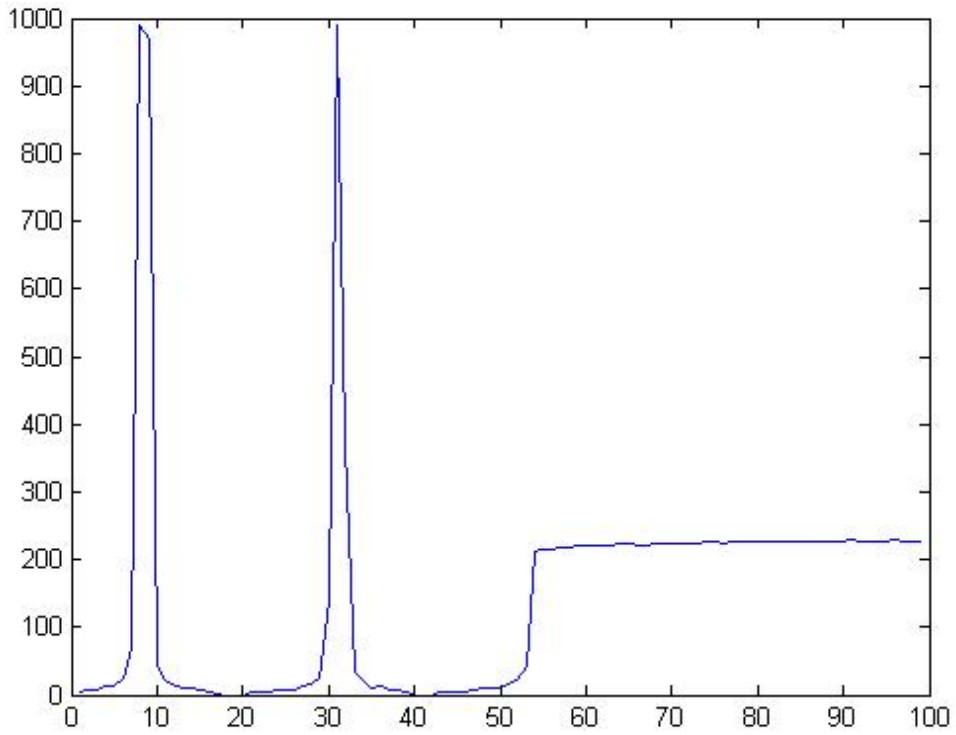
```
$regfile = "m8def.dat "
$crystal = 16000000
$baud = 9600
Config Adc = Single , Prescaler = Auto , Reference = Avcc
Start Adc
Dim A As Word , N As Byte
Wait 2
For N = 1 To 100
A = Getadc(0)
Waitms 100
Print A
Next
Wait 100

End
```

برنامه بسیار ساده است ابتدا میکرو را معرفی کرده که از نوع ATmega8 هست سپس شماره کریستال و نرخ ارسال داده.بعد از آن پیکر بندی مبدل آنالوگ به دیجیتال و شروع به کار آن.تعریف دو متغییر که یکی برای ذخیره داده خروجی مبدل و دیگری شمارش تعداد ارسالها .در ادامه هم به کمک یک for صد بار ارسال صورت میگیرد.بعد از راه اندازی دستگاه پتانسیومتر چرخیده شده و تغییرات آن ارسال شده.این ارسالها با فاصله زمانی ۱۰۰ میلی ثانیه صورت گرفته اند پس میتوان گفت عمل نمونه گیری و ارسال ۱۰ ثانیه طول خواهد کشید .

در ادامه برنامه matlab برای دریافت این داده ها و رسم آنها نوشته شده است این برنامه میتواند در mfile یا محیط command window نوشته شود.

```
clc
s = serial('COM1')
fopen(s)
for n=1:1:100
n
out= fscanf(s);
c=str2num(out);
a(n)=c
end
fclose(s)
x=1:1:100;
plot(x,a)
```

در برنامه نوشته شده در بالا خط اول صفحه نمایش command window را پاک میکند.سپس com1 به عنوان ارتباط سریال معرفی میشود در خط بعد پورت باز شده و آماده دریافت اطلاعات است .بعد از آن به کمک یک for داده های دریافت شده تک تک گرفته شده و چون این داده ها از نوع رشته در متغییر out هستند ابتدا به صورت عدد تبدیل شده و یکی یکی در بردار a قرار میگیرند.بعد از تکمیل این بردار پورت بسته شده و بردار جدیدی که صد عنصر دارد معرفی میشود این بردار قرار است محور x را برای ما ایجاد کند.در خط آخر داده های دریافتی که در a قرار گرفتند روی یک نمودار رسم میشوند.یک نمودار بدست آمده از این روش در زیر نمایش داده شده است.



منحنی تغییرات پتانسیومترنمایش داده شده توسط MATLAB

پروژه ارتباط غیر هم زمان میکرو و MATLAB

در این پروژه قصد داریم اطلاعات ورودی سنسور را ابتدا در حافظه EEPROM میکرو ذخیره کنیم سپس اطلاعات بدست آمده که با نمونه گیری حاصل شده را به MATLAB انتقال داده و پردازش کنیم. دلیل استفاده از حافظه EEPROM این است که با قطع جریان برق یا همان تغذیه مدار اطلاعات از بین نمیروند.و میتوان دستگاه را با یک باتری در محل مورد نظر قرار داد و بعد از در یافت اطلاعات به کامپیوتر متصل کرد.

برای این منظور از سخت افزار پروژه قبل استفاده شده.برای آزمایش از یک لیوان محتوی آب در دمای حدود صد درجه سانتی گراد که در محیطی با دمای شش درجه سانتی گراد قرار گرفته استفاده شده .از سنسور دمای موجود در سخت افزار برای اندازه گیری دمای آب کمک گرفته شده.برنامه طوری نوشته شده که اطلاعات سنسور را هر یک ثانیه در حافظه EEPROM ذخیره کند.

میکرو استفاده شده دارای ۵۱۲ بایت حافظه EEPROM است و ما حد اکثر میتوانیم ۵۱۲ بار نمونه گیری انجام دهیم .در صورت نیاز به تعداد نمونه گیری های بیشتر میتوانیم از میکرو های دیگری که دارای حافظه EEPROM بیشتری هستند استفاده کرد .یا از IC های حافظه خارجی یا در نهایت از کارت های حافظه MMC کمک گرفت .

در این پروژه نمونه برداری تعداد ۱۳۰ بار در مدت ۱۳۰ ثانیه صورت گرفت و سپس اطلاعات از طریق ارتباط سریال برای MATLAB ارسال شد.

برنامه میکرو به این صورت است که پس از روشن شدن اگر کلید شماره یک که به پورت D7 متصل است فشرده شده باشد عمل نمونه گیری و ذخیره اطلاعات روی EEPROM شروع خواهد شد برای ارسال اطلاعات نیز با روشن شدن دستگاه در صورت فشردن کلید دوم که به پایه D6 متصل است عمل ارسال صورت میگیرد.

برای بیشتر شدن دقت نمونه برداری برای هر داده ده بار نمونه گیری انجام شده و نتیجه نهایی از میانگین آنها نتیجه شده است.

در صفحه بعد برنامه میکرو قرار داده شده است .

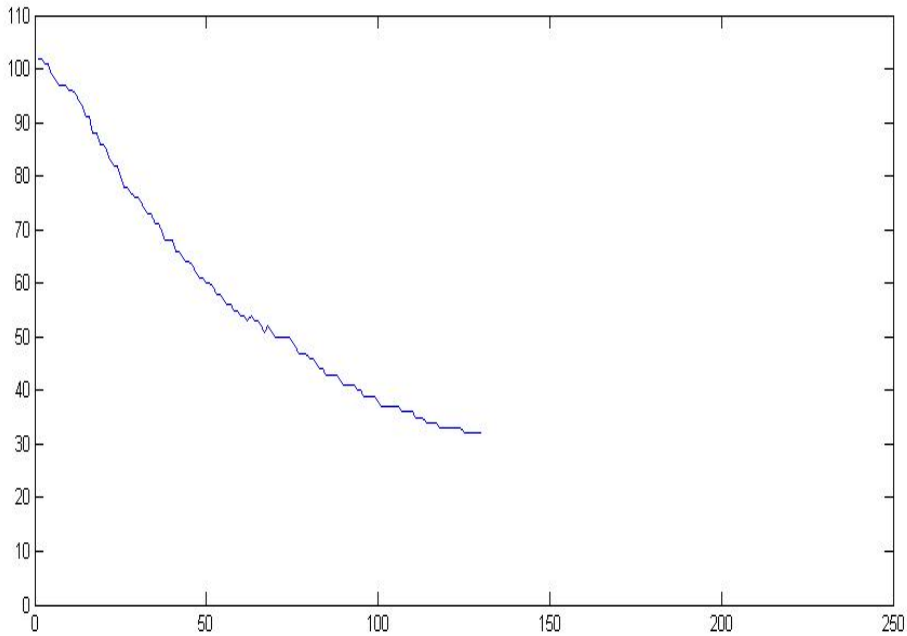
```
$ regfile = "m8def.dat"
$ crystal = 16000000     'give here the value of the X-tal you use in Hertz
$    baud = 9600

    Config Adc = Single , Prescaler = Auto , Reference = Avcc
    Start Adc

   Dim A As Word , Sum As Word , N As Byte , M As Word , B As Byte
    Config Portd = Input
    Config Portc = Input
    Config Portb = Output
    Wait 1
Do
   If Pind.7 = 1 Then
    For M = 0 To 511
      A = Getadc(1)
      Sum = Sum + A
      Waitms 100
      Next
     Sum = Sum - 110
     Sum = Sum / 16
     B = Sum
    Writeeeprom B , M
   End If

   If Pind.6 = 1 Then
    For M = 1 To 511
    Readeeprom B , M
    Print B
    Waitms 10
   Next

  End If

Loop

End
```

برای گرفتن اطلاعات از پورت سریال توسط نرم افزار MATLAB برنامه زیر نوشته شده است.

```
clc
s = serial('COM1')
fopen(s)
for n=1:1:130
    n
 out= fscanf(s);
 c=str2num(out);
 a(n)=c
end
fclose(s)
x=1:1:130;
plot(x,a)
```

این برنامه پس از دریافت اطلاعات آنها را روی نمودار رسم میکند.

پس از دریافت داده ها نمودار رسم شده که بیانگر تغییرات دما نسبت به زمان است در زیر ترسیم شده است.

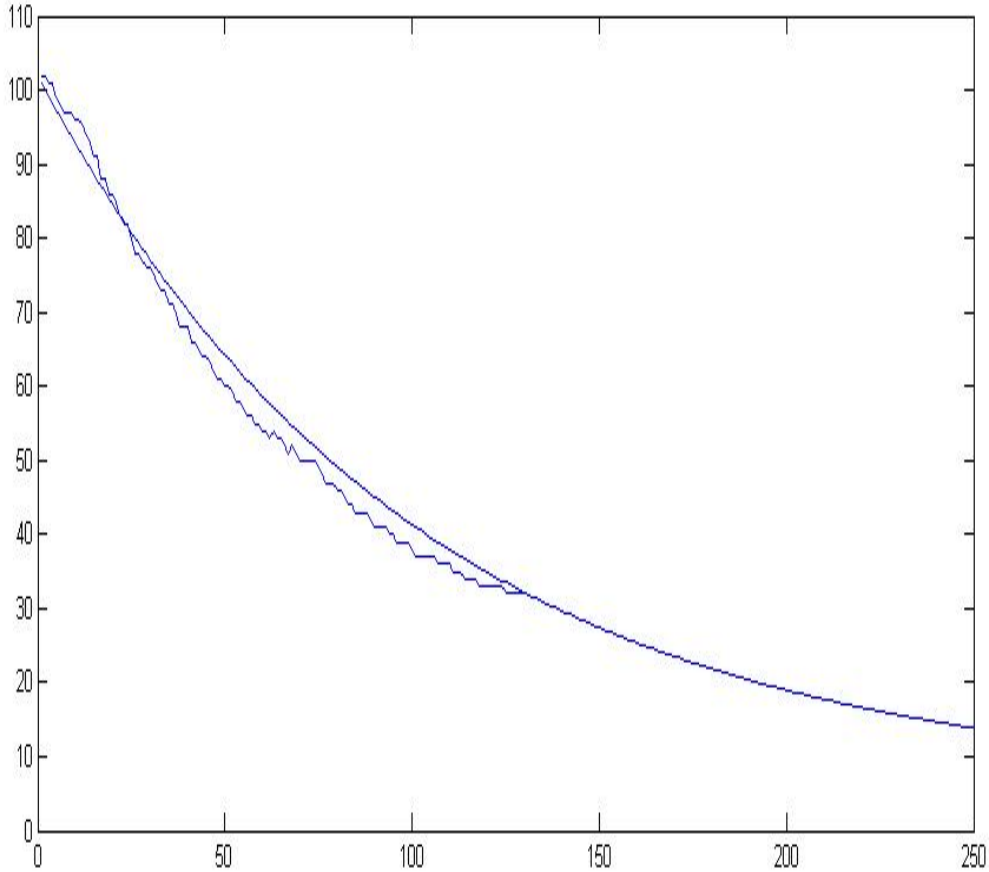


منحنی تغییرات دما نمایش داده شده توسط MATLAB

انتقال این داده ها به محیط MATLAB مارا قادر میسازد تا با استفاده از روش هـای ریاضی رفتار بعدی سیستم را پیش بینی کنیم.

در نمودار صفحه قبل طبق قانون تبرید نیوتن میتوان تغییرات دمای جسم را بـا جمـع یـک عـدد ثابـت و جمله ای نمایی به صورت تابعی از زمان تعیین کرد. عدد ثابت همان دمای نهایی جسم است که در ایـن مورد ۶ درجه سانتی گراد است. عامل نمایی به صورت C * EXP(K*t) خواهد بود.

با استفاده از دمای اولیه C=96 خواهد شد با چند بار سعس و خطا و رسم نمودار مقدار K عدد 0.01 بدست خواهد آمد.از فرمول حاصل شده میتوان برای پیش بینی دماهای بعـدی کمـک گرفـت. در زیـر دو نمودار با هم نمایش داده شده است ،که نشان میدهد دقت تقریب به چه اندازه است.



منحنی تغییرات دما نمایش داده شده توسط MATLAB وتقریب نمایی آن

فصل چهارم

ارتباط سریال با labVIEW

مقدمه

نـرم افـزار labVIEW یکـی از قدرتمنـد تـرین نـرم افزارهـا در زمینـه کنتـرل و monitoring است.خصوصیتی که این نـرم افـزار را از بقیـه نـرم افزارهـای دیگـر متمـایز میکنـد زبـان برنامـه نویـسی گرافیکی آن است که باعث سادگی کار با آن شده است علاوه بر آن وجود توابع مختلف و گونـاگون و قابلیت برقراری ارتباط با نرم افزارهای مهم مهندسـی نظیـر MATLAB و وجـود سـخت افزارهـای مرتبط با این نرم افزار به صورت کارتهای I/O از ویژگی های دیگر این نرم افزار است .

این قابلیت ها باعث آن میشود که به جای استفاده از PLC های گران قیمت و صفحات نمایش HMI از یک کامپیوتر با قیمت خیلی پایین تر و کاربردهای وسیع تر در پروژه های صنعتی استفاده کرد.

در این فصل به برسی یک پروژه که از سـخت افـزار معرفـی شـده در فـصل دوم اسـتفاده شـدهپرداخته میـشود.ایـن پروژهـدر مـورد ارتبـاط بـین میکـرو کنترلـر و labVIEW اسـت بـه ایـن صـورت کـه میکروکنترل اطلاعات دو سنسور را برای نرم افزار ارسال میکند و نـرم افـزار ایـن داده هـارا بـه کمـک ابزارهای گرافیکی قدرتمند خود به نمایش میگذارد و وضعیت ابـن داده هـا را بـسته بـه مقـدار مطلـوب تعیین شده نمایش میدهد.

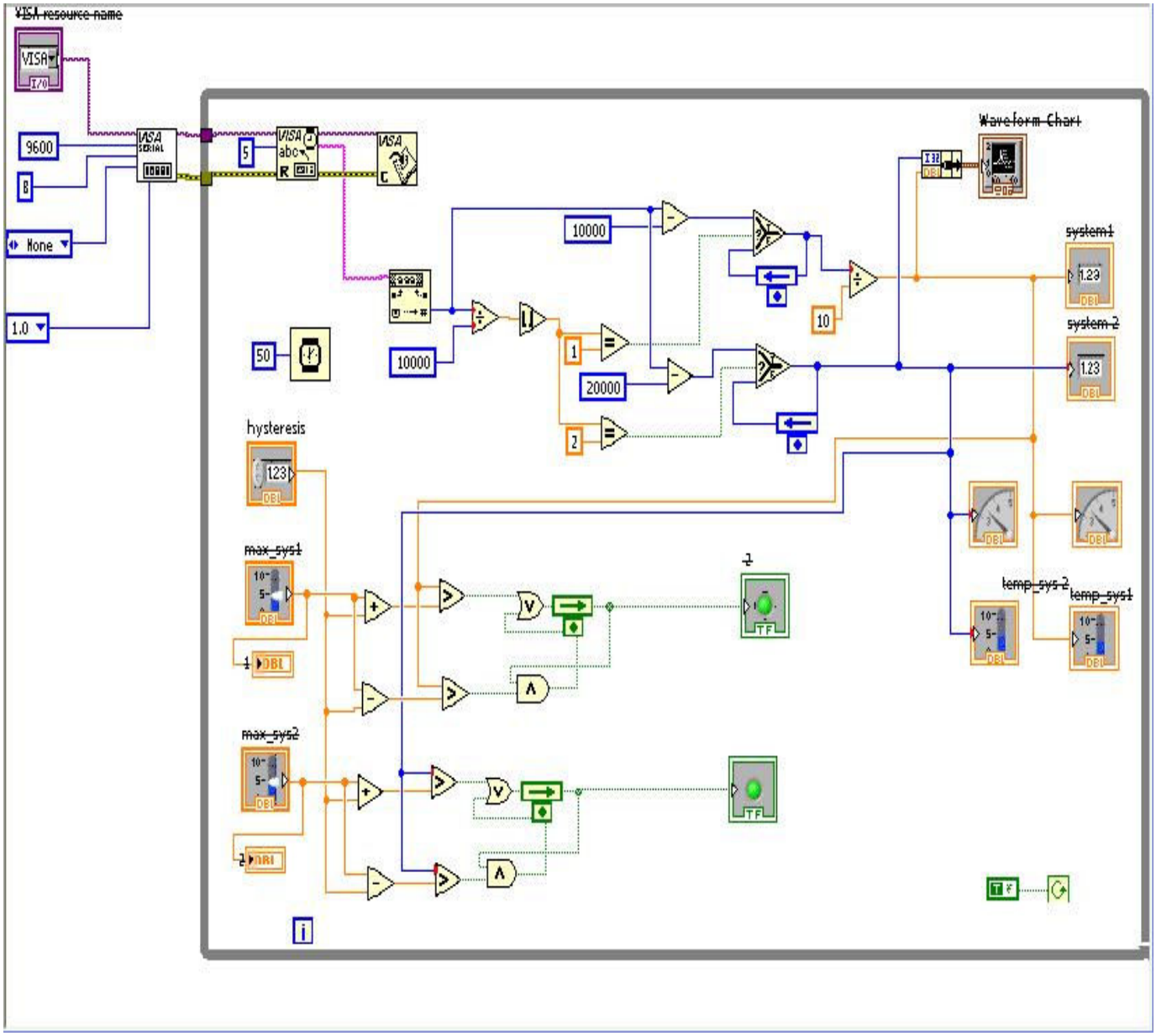
در این فصل به دلیل آشنایی با مشخصات پورت سریال در فـصل هـای قبـل از ابتـدا بـه توضـیح برنامـه پرداخته میشود.

طرح کلی پروژه:

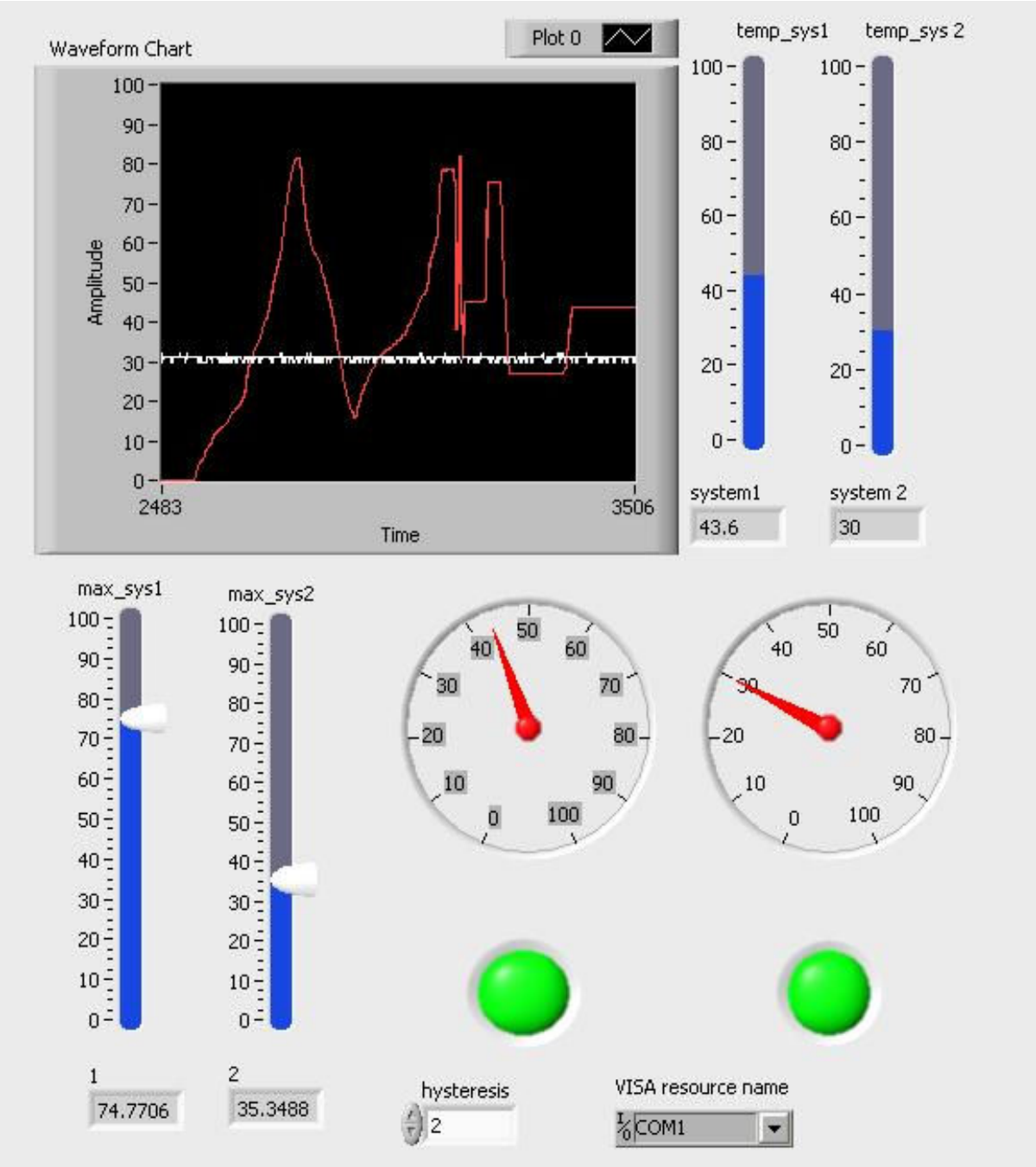
طرح کلی این برنامه به این صورت است که میکرو هر ۵۰ میلی ثانیه اطلاعات یک سنسور را از طریق ارتباط سریال میفرستد و پس از دریافت توسط کامپیوتر labVIEW آن را نمایش میدهد.
در سخت افزار موجود از یک عدد سنسور دما به شماره LM35D و یک عدد پتانسومتر استفاده شده است که هر کدام به یک کانال مبدل آنالوگ به دیجیتال به عنوان سنسور متصل هستند و در توضیحات گاهی از هردو آنها به عنوان سنسور یاد شده.
دلیل استفاده از یک پتانسیومتر به جای سنسور این است که محدوده تغییرات وسیعی را در زمان کوتاه بتوان ایجاد کرد ،این تغییرات زیاد در ساده تر شدن تست پروژه کمک میکند و طبیعی است که در کاربردهای عملی از یک سنسور به جای این قطعه استفاده میشود.
عدد گرفته شده از مبدل آنالوگ به دیجیتال در صورتی که مربوط به کانالی است که به آن سنسور دما متصل است پس از تبدیل به مقدار دما اندازه گیری شده ارسال میشود ولی عدد گرفته شده از کانالی که به آن پتانسیومتر متصل است مستقیما ارسال میشود که با توجه به ده بیتی بودن مبدل عدد خروجی آن میتواند بین صفر تا ۱۰۲۳ تغییر کند.
چون در هر بار فقط اطلاعات یک سنسور ارسال میشود باید به گونه ای مشخص شود که داده مورد نظر برای کدام سنسور است برای این کار روشهای زیادی را میتوان به کار برد. روشی که در این پروژه استفاده شده به این صورت است که عدد بدست آمده قبل از ارسال در صورتی که مربوط به سنسور اول باشد ثابت ۱۰۰۰۰ با آن جمع شده و سپس ارسال میشودو در صورتی که مربوط به سنسور دوم باشد عدد ۲۰۰۰۰ با آن جمع میشود.labVIEW بعداز دریافت عدد از پورت سریال در صورتی که رقم پنجم یک بود آنرا به عنوان خروجی سنسور اول و اگر رقم پنجم عدد دریافتی دو بود به عنوان خروجی سنسور دوم در نظر میگیرد.
labVIEW پس از دریافت و تفکیک اطلاعات آن را روی نمایشگرهای مختلف نشان میدهد و با عددی که توسط کاربر به عنوان حد اکثر مقدار مجاز مشخص شده مقایسه میکند و در صورت بیشتر شدن از این مقدار LED داخل برنامه به صورت قرمز رنگ تغییر وضعیت میدهد.
درمدارات کنترل ،معمولا درعمل مقایسه با حد اکثر مجاز، عددی به عنوان هیسترزیس استفاده میشود.در این برنامه کاربر این عدد را نیز میتواند به مقدار دلخواه تغییر دهد.
میتوان برنامه را طوری بهبود داد که هم زمان با روشن شدن LED قرمز درون labVIEW دستوری از برنامه به سخت افزار فرستاده شود که وضعیت رله ها را تغییر دهد ولی در این پروژه این قسمت برسی نشده است.

برنامه نوشته شده در قسمت block diagram در زیر نمایش داده شده است درصفحه بعد نمایش صفحه front panel قرار دارد و در ادامه به برسی هر قسمت از برنامه به صورت مجزا پرداخته شده. سپس برنامه میکرو کنترلر برسی میشود.

بلوک دیاگرام برنامه ارتباط سریال با labVIEW

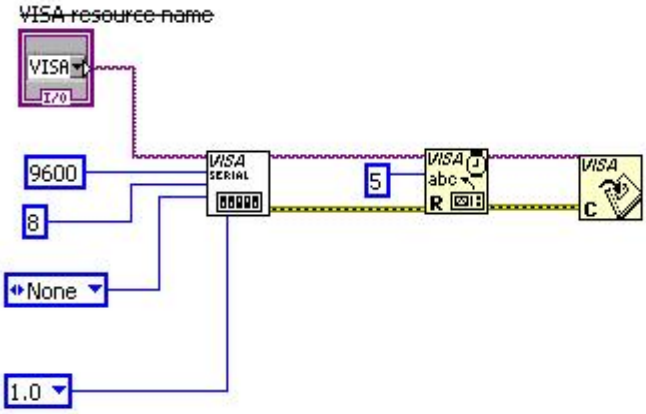


در زیر قسمت front panel برنامه صفحه قبل نشان داده شده است.



برسی قسمتهای مختلف برنامه نوشته شده در block diagram

قسمت اول: برسی بلوکهای پیکر بندی پورت و دریافت اطلاعات



این قسمت که دربالا نمایش داده شده در ابتدا از یک بلوک به نام
VISA configure serial portاستفاده شده است.

ازاین بلوک برای پیکر بندی ارتباط سریال استفاده میشود.در قسمت resource name که در بالا از نوع کنترلی قرار داده شده میتوان شماره پورت سریالی که به سخت افزار خارجی متصل است را وارد کرد.

تنظیمات دیگر این بلوک شامل نرخ ارسال داده تعداد بیتهای ارسالی زمان time out و... رامیتوان تغییر داد و در صورت تغییر ندادن از مقدار پیش فرض استفاده میشود.

بلوک بعدی که مربوط به خواندن از پورت است در ادامه قرار دارد. دراین بلوک یک ورودی به عنوان VISA resource name وجود دارد که میتوان از خروجی بلوک پیکر بندی برای آن استفاده کرد.

تمامی بلوکهای مربوط به ارتباط سریال دارای این ورودی هستند وهمین قسمت را در خروجی خود برای ارتباط با بلوک بعدی دارند.این مزیت به ما این امکان را میدهد که بلوکها را پشت سر هم ببندیم.

پایه دیگری که در ورودی و خروجی هر بلوک سریال وجود دارد به نام err است.وقتی که اجرای برنامه به بلوکی میرسد و در اجرای آن بلوک مشکلی به وجود می آید آن مشکل به صورت پیغامی نمایش داده میشود و توسط اتصال سیم های err شماره خطا به بلوک بعد منتقل میشود حال اگر در ورودی بلوکی err وجود داشت معلوم میشود در اجرای بلوکهای قبلی مشکلی به وجود آمده.

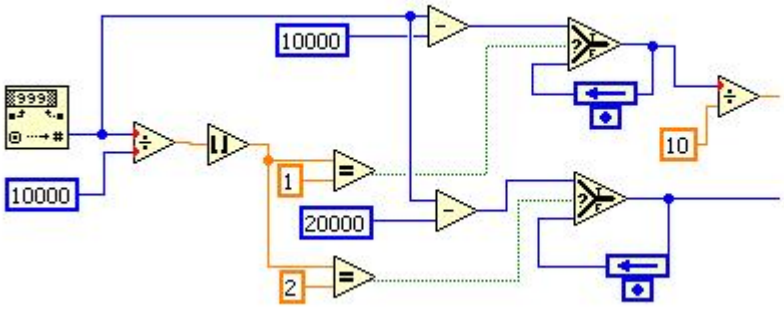
ورودی دیگر این بلوک طول بایتهایی است که باید بخواند و در پروژه حاضر با توجه به ارسال اعداد پنج رقمی برای داده ها، این مقدار برابر پنج در نظر گرفته شده است.

در خروجی این بلوک که به نام read buffer است اطلاعات خوانده شده قرار دارد.این اطلاعات به صورت string است.

خروجی دیگر این بلوک به نام return count است که تعداد بایتهای خوانده شده را به صورت عدد به ما میدهد.

در آخرین بلوک این بخش که بعد ازعمل خواندن اجرا میشود پورت سریال بسته میشودو تا اجرای مجدد بلوک خواندن ،اطلاعات دریافت نمیشود.

قسمت دوم:تبدیل و تفکیک اطلاعات دریافتی



نام اولین بلوک این قسمت که در بالا نمایش داده شده decimal string to number است.

همان گونه که در بخش قبل گفته شد خروجی بلوک مربوط به خواندن از نوع string هست.

برای انجام عملیات ریاضی و همچنین نمایش اطلاعات باید این داده ها را به عدد تبدیل کرد. این بلوک این کار را برای ما انجام میدهد. بلوک دارای یک ورودی default است ودر صورتی که اطلاعات ورودی کدهای مربوط به اعداد نباشند این عدد به عنوان خروجی ارسال میشود.عدد پیش فرض صفر است و در این پروژه از همین عدد استفاده شده است.

برنامه با ید به گونه ای باشد که وقتی خطایی در دریافت رخ میدهد عدد خروجی این بلوک یعنی صفر ذاذه ورودی حساب نشود وبه صورت خود کار حذف شود.با روش به کار برده شده برای ارسال دادهای سنسورها که در قبل گفته شد این عمل به خوبی انجام میپذیرد.

در صورتی که داده ارسالی صحیح باشد عددی بین ۱۰۰۰۰ تا ۱۱۰۲۳یا بین ۲۰۰۰۰ تا حدود ۲۰۱۰۰ است.اگر عدد دریافتی درخروجی بلوک در بازه اول باشد مربوط به پتانسیومتر است و اگردر بازه دوم

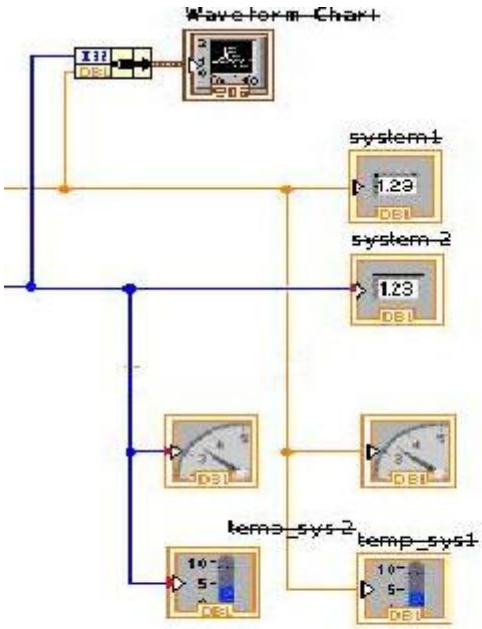
باشد مربوط به سنسور دما است که باید مستقیماً نمایش داده شود و اگر هم صفربود حتماً خطایی رخ داده و نباید نمایش داده شود.

برای تفکیک این وضعیت ها ابتدا عدد دریافتی را بر ۱۰۰۰۰ تقسیم شده.عدد حاصل پس از روند شدن ۱یا ۲ یا صفر میتواند باشد.در مقایسه انجام گرفته اگر عدد ۱ بود بلوکی که select نام دارد حاصل عدد دریافتی منهای ۱۰۰۰۰ را در خروجی میفرستد یعنی دقیقا همان عدد خروجی مبدل آنالوگ به دیجیتال میکرو که بین ۰ تا ۱۰۲۳ است.عدد حاصل برای نمایش بر ۱۰ تقسیم شده تا بازه ای تقریباً بین صفر تا صد ایجاد کند.

اگر شرط بلوک select اول برقرار نبود به این مفهوم است که عدد دریافتی یا مربوط به سنسور دوم است یا در روند دریافت اشکالی رخ داده، این بلوک وضعیت قبلی خود را که به ورودی false فیدبک شده را حفظ میکند و همان عدد قبلی را برای نمایش میفرستد.

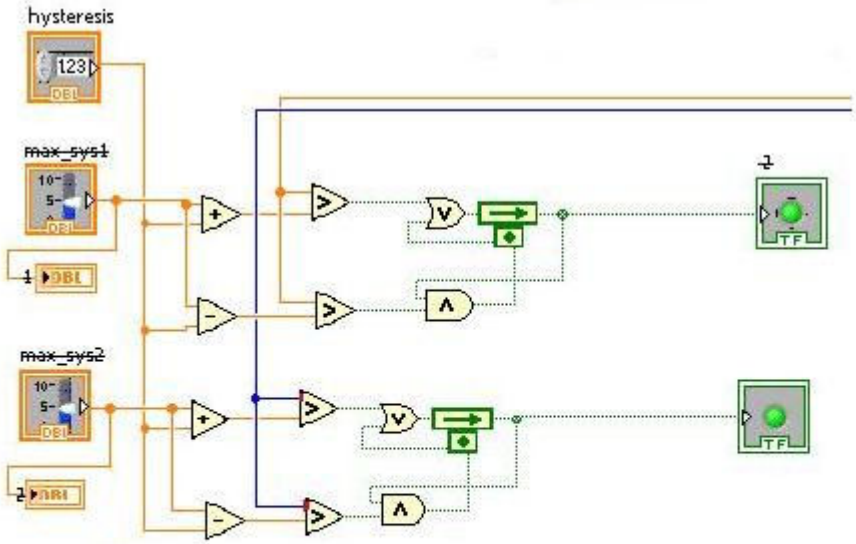
شبیه این عملیات برای بلوک select دوم که مربوط به سنسور دوم است هم انجام شده که واضح است و نیاز به توضیح ندارد.

قسمت سوم:نمایشگرها



این قسمت در بالا نمایش داده شده،اطلاعات خروجی قسمت قبل که به دو قسمت تفکیک شده بودند و هر کدام مربوط به یک سنسور است و بازه های آنها به صورت مورد نظر تبدیل شدند برای نمایش روی یک waveform chart ابتدا توسط بلوک bundle ترکیب شده و سپس نمایش داده میشوند.علاوه بر این چند بلوک دیگر که اطلاعات را به صورت عددی،عقربه ای و ستونی نمایش میدهند وجود دارد.

قسمت چهارم:مقایسه مقدار سنسورها با مقدار مجاز



این قسمت در بالا نمایش داده شده است.اصلی ترین قسمت آن بخش هیسترزیس است.

در صورت نداشتن این خصوصیت این قسمت تنها با یک مقایسه کننده قابل ساخت بود.

این قسمت این گونه ساخته شده که مقدار مجاز تعیین شده توسط کار بر ابتدا یک بار با مقدار مشخص شده برای هیسترزیس جمع شده وبار دیگر تفریق میشود.وضعیت خروجی باید به گونه ای باشد که وقتی ورودی این قسمت بین این دو حد باشند خروجی وضعیت قبلی خود را حفظ کند زمانی که بزرگتر از آنها باشد خروجی فعال و وقتی کوچکتر باشد خروجی غیر فعال باشد.

چون اگرورودی بالاترازمجموع حد مجاز و هیسترزیس بود باید حتما خروجی فعال باشد با یک گیت or به خروجی متصل شده.حالت دیگر روشن بودن خروجی این است که وضعیت قبل روشن بوده و مقدار دما از حد پایین گفته شده در بالا بیشتر باشد.این دو شرط با یک گیت and برسی شده اند و خروجی این گیت به گیت or داده شده است.

این آرایش برای هردو ورودی انجام شده است.

برای این که تمام چهار مرحله گفته شده در بالا مرتب اجرا شوند همه ی بلوکها در یک حلقه بینهایت قرار داده شده اند و با توجه به این که اطلاعات هر ۵۰ میلی ثانیه ارسال میشود در این حلقه از یک تاخیر به همین میزان استفاده شده است.

یک نکته مهم در ارتباط با این پروژه این است که حتماً driver برنامه labVIEW به نام Measurement & Automation Explorer نصب شده باشد.در غیر این صورت پورتهای سریال قابل دسترسی نیستند.

برنامه میکرو کنترلر:

```
$regfile = "m8def.dat"
$crystal = 16000000
$baud = 9600
 Config Adc = Single , Prescaler = Auto , Reference = Avcc
 Start Adc

 Dim A As Word , Sum As Word , N As Byte
 Config Portc = Input
 Wait 1

 Do
Sum = 0
For N = 1 To 10
  A = Getadc(0)
  Sum = Sum + A
  Waitms 5
Next
   Sum = Sum / 10
   Sum = Sum + 10000
    Print Sum
Sum = 0
For N = 1 To 10
  A = Getadc(1)
  Sum = Sum + A
  Waitms 5
Next
  Sum = Sum - 110
  Sum = Sum / 16
  Sum = Sum + 20000
  Print Sum

 Loop

End
```

فهرست منابع و مراجع :

نرم افزارها:

BASCOM-AVR HELP
MATLAB HELP
LabVIEW HELP

مقالات:

پایانه سریال در چند کلمه - فرشید سفید گران
پورت سریال- مشخصات و کاربردها – مرتضی ظفری

سایتها :

www.ir-micro.com
www.matlabseven.blogfa.com
و...