AVR SIMPLE OS

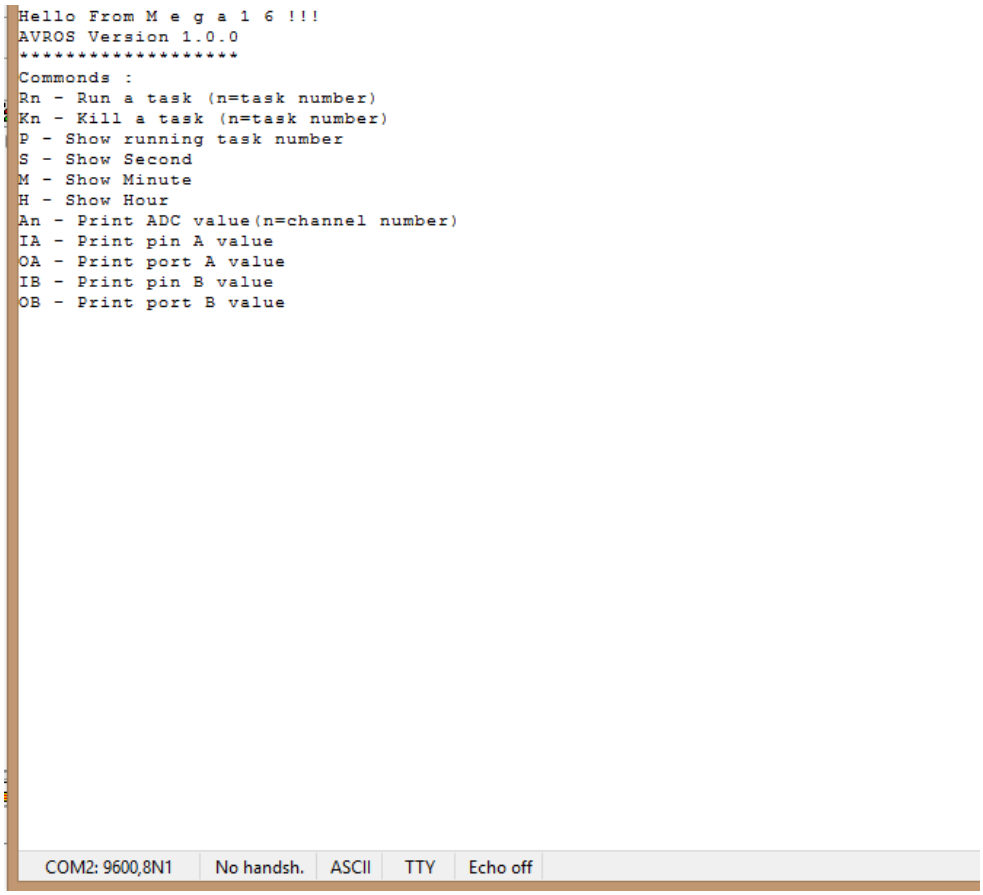
حمید رضا محرابیان - hr_mehrabian@yahoo.com

پروژه ی درس میکروکنترلر ، استاد کامرانی

این پروژه یک نوع دمو برای نمایش عمل مولتی تسکینگ است. در این پروژه از یک میکروکنترلر AVR مدل ATMEGA۱۶ استفاده شده است که قابلیت اتصال به کامپیوتر از طریق پورت سریال را دارد. کد این پروژه به طور کامل به زبان اسمبلی نوشته شده و تمام قسمت های آن از قبیل کنترل LCD ، کنترل رابط سریال و... به صورت کتابخانه هایی جداگانه مخصوص این پروژه به طور کامل نوشته شده است.

بعد از روشن کردن مدار اولین TASK برنامه که وظیفه ی گرفتن و اجرای دستورات از رابط سریال و کنترل سایر TASK ها را بر عهده دارد ، اجرا خواهد شد. این TASK همان KERNEL معروف در تمام سیستم عامل ها میباشد. این پروژه قابلیت اجرای حداکثر ٤ TASK دیگر را هم دارد و از آنجایی که این پروژه بیشتر جنبه ی نمایشی و آموزشی دارد ، این TASKها عملیات پیچیده ای انجام نمی دهند. با توجه به اینکه در سیستم عامل های دارای قابلیت MULTITASKING یکی از بزرگترین مزیت ها مدیریت رویدادها و رسیدگی به آنها بدون اختلال زمانی در برنامه ی جاری میباشد ، تلاش شده تا این ٤ TASK نیز رویداد محور باشند.

بعد از روشن کردن مدار اطلاعات مربوط به راهنمای دستورات از طریق پورت سریال به کامپیوتر ارسال میگردد :



دستورات فعلی مدار شاید کم باشند ، اما به راحتی قابل توسعه است. دستورات فعلی به شرح زیر است :

۱ – Rn برای اجرا (یا ادامه ی) یک TASK میباشد و n شماره ی TASK است (مثل R۲ که TASK شماره ی ۲ را اجرا میکند)

۲ – Kn برای توقف (Suspend) کردن یک TASK میباشد و n شماره ی TASK است.

۳ – P که تعداد TASK های در حال اجرا را نمایش میدهد.

٤ – S ثانیه زمانی که از روشن شدن مدار میگذرد را نمایش می دهد.

٥ – M دقیقه زمانی که از روشن شدن مدار میگذرد را نمایش می دهد.

٦ – H ساعت زمانی که از روشن شدن مدار میگذرد را نمایش می دهد.

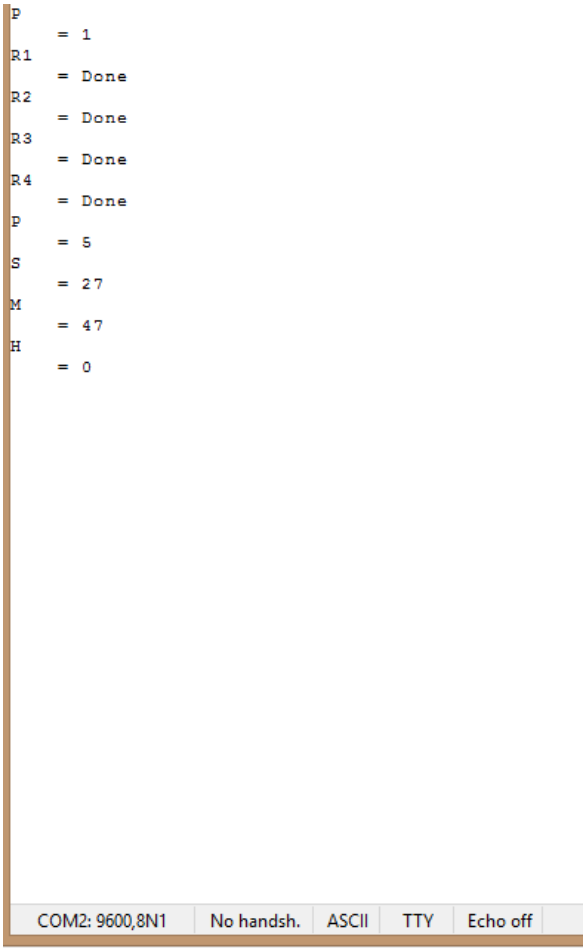
۷ – An مقدار آنالوگ موجود در مبدل آنالوگ به دیجیتال میکروکنترلر را نمایش میدهد که n شماره ی کانال است.

۸ – IA مقدار ورودی (رجیستر PIN) پورت A را نمایش میدهد.

۹ – OA مقدار خروجی (رجیستر PORT) پورت A را نمایش میدهد.

۱۰ – IB مقدار ورودی (رجیستر PIN) پورت B را نمایش میدهد.

۱۱ – OB مقدار خروجی (رجیستر PORT) پورت B را نمایش میدهد.



این پروژه دو منبع مشترک (LCD و مبدل آنالوگ به دیجیتال) بین TASK ها دارد که ممکن است چندین TASK به طور همزمان بخواهند از آنها استفاده کنند. به همین دلیل برای منابع مشترک از ساختار سمافرها استفاده شده است.

توضیح TASKها:

۰ – که همان KERNEL میباشد و همیشه در حال اجراست.

۱ – کنترل ۲ لامپ LED به صورت چشمک زن و صدای بوق

۲ – شمارنده ی ساده که با هر بار فشار کلید تعداد شمارش روی LCD نمایش داده میشود و ٤ عدد LED را که نشانگر عددی باینری متناظر با عدد شمارش است را روشن میکند

۳ – هر یک ثانیه یک بار دما را روی LCD نمایش میدهد.

٤ – کنترل شدت نور LED با استفاده از PWM و کلید فشاری و نمایش شدت نور بر روی LCD

پس به این ترتیب LCD بین TASKهای ۲ و ۳ و ٤ مشترک و ADC هم بین TASKهای ۰ و ۳ مشترک است.

شرح برنامه ی میکروکنترلر :

برنامه به صورت تکه تکه نوشته شده و هر قسمت به صورت کتابخانه وظایف خاصی انجام میدهد و توابع خاصی دارد:

ADC.INC :

توابع زیر را دارد :

۱ – GetADC : مقدار ADC را میخواند.

۲ – InitADC : تنظیمات ADC را انجام می دهد و رجیستر A ورودی آن است که شماره ی کانال ADC در آن قرار میگیرد.

AVROS.ASM :

این فایل نقطه ی شروع برنامه است و تنظیمات اولیه را برای LCD و ارتباط سریال و تایمر ها و راه اندازی اولین TASK که همان KERNEL باشد را انجام میدهد.

تایمر ۱ وظیفه ی جلو بردن زمان و تایمر ۰ وظیفه ی SwitchContext را بر عهده دارد.

CLOCK.INC :

این فایل فقط یک تابع دارد که در زمان OVERFLOW شدن تایمر ۱ اجرا میگردد و وظیفه ی جلوبردن زمان را بر عهده دارد.

LCD.ASM :

توابع زیر را دارد :

۱ – LCD_PUTF : رشته ای را که در حافظه ی فلش قرار دارد و آدرس شروع آن در رجیستر Z قرار دارد را روی LCD نمایش میدهد.

۲ – LCD_PUTS : رشته ای را که در حافظه ی SRAM قرار دارد و آدرس شروع آن در رجیستر X قرار دارد را روی LCD نمایش میدهد.

۳ – LCD_INIT : وظیفه ی راه اندازی اولیه ی LCD را بر عهده دارد.

٤ – LCD_SEND_DATA : داده ای را که در رجیستر A قرار دارد را به LCD ارسال میکند.

۵ - LCD_SEND_COMMAND : دستوری را که در رجیستر A قرار دارد را به LCD ارسال میکند.

٦ – LCD_READ : داده را از LCD میخواند و در رجیستر A قرار میدهد.

۷ – LCD_WAIT : تا صفر شدن فلگ BUSY_FLAG منتظر میماند.

۸ – LCD_HOME : اشاره گر را به خط اول و کارکتر اول انتقال میدهد.

۹ - LCD_CLS : LCD را پاک میکند.

۱۰ – LCD_L۲ : اشاره گر را به اول خط دوم منتقل میکند.

۱۱ – LCD_CSR : انتقال اشاره گر ۱ واحد به راست

۱۲ - LCD_CSL : انتقال اشاره گر ۱ واحد به چپ

۱۳ – LCD_TSR : انتقال کل داده های LCD ۱ واحد به راست

۱٤ – LCD_TSL : انتقال کل داده های LCD ۱ واحد به چپ

۱۵ – LCD_PUTC : قرار دادن کارکتر موجود در رجیستر A بر روی LCD

ماکروی وقفه LDELAY نیز در این کتابخانه قرار دارد.

: SERIAL.INC

توابع زیر را دارد:

۱ - USART_Init : که وظیفه ی راه اندازی اولیه ی ارتباط سریال را دارد.

۲ - USART_Recive : یک بایت را از رابط سریال دریافت میکند و در رجیستر A قرار میدهد.

۳ - USART_Send : داده ی موجود در رجیستر A را به رابط سریال میفرستد.

٤ - USART_Scan : رشته ای را تا رسیدن به کارکتر Enter از ورودی رابط سریال میگیرد و در جایی که آدرس آن در رجیستر X قرار دارد بار گذاری میکند.

۵ - USART_Print : رشته ای را که در حافظه ی SRAM قرار دارد و آدرس شروع آن در رجیستر X قرار دارد را به خروجی رابط سریال میفرستد.

٦ - USART_Printf : رشته ای را که در حافظه ی فلش قرار دارد و آدرس شروع آن در رجیستر Z قرار دارد را به خروجی رابط سریال میفرستد.

: PWM.INC

توابع زیر را دارد:

۱ – pwminit : وظیفه ی راه اندازی اولیه ی PWM را بر عهده دارد.

۲ – pwmset : تنظیم طول پالس PWM

: STDLIB.INC

این فایل کتابخانه ای برای تعاریف اصلی و بعضی از توابع کلی میباشد.

توابع آن به شرح زیر است :

۱ – strcmp : مقایسه ی دو رشته

۲ – itoa : که وظیفه ی تبدیل عدد موجود در رجیسترهای A و B (دو بایت) را دارد و رشته ی تبدیل شده را در آدرس X قرار میدهد.

۳ - Wait_Unlock : تا زمان آزاد شدن منبع منتظر میماند (سمافر).

٤ – SetLock : قفل کردن یک منبع که شناسه ی آن در رجستر A قرار دارد.

۵ – SetUnlock : باز کردن قفل منبعی که شناسه ی آن در رجستر A قرار دارد.

PROCESS.INC :

۱ – SwitchContext : وظیفه ی انتقال و سویچ کردن TASKها را بر عهده دارد.

۲ – KillProc : توقف TASKی که PID آن در رجیستر A قرار دارد.

۳ – InitProc : راه اندازی اولیه و بارگذاری TASKی که آدرس شروع آن در رجیستر X قرار دارد.

٤ – LoadABCDY : بار گذاری رجیسترهای اصلی برای هر TASK

۵ – SaveABCDY : ذخیره ی رجیسترهای اصلی برای هر TASK

KERNEL.ASM :

این فایل حاوی کد هسته است و اولین TASKی است که اجرا میشود.

APP۱.ASM :

برنامه ی TASK ۱ در این فایل قرار دارد.

APP۲.ASM :

برنامه ی TASK ۲ در این فایل قرار دارد.

APP۳.ASM :

برنامه ی TASK ۳ در این فایل قرار دارد.

APP٤.ASM :

برنامه ی TASK ٤ در این فایل قرار دارد.