

حسن مودهائی

مجمع جامع سرلنتر AVR

سیستم کامپیوٹری نے اسے عبّت ایسا از ملیے۔ سیستم کامپیوٹری ہے : Embedded system :

در مردم ایتالیا هم می‌توانید لعبه‌های آنلاین بازی‌های آنلاین فوری داشته باشید.

طریق: تجهیزات منفعت، روپاند، بحثه ای ایز کش، لئه کی خواران، خودرو کی جدید و دسته ایلکترونیک

عکس دستگاهی نه از این سه استواره می‌ستد اندیعه سوابت، دستگاهی ملمس برداری، EEG، ECG

انلچ مردم، دیستراکس BTS، دریچ گز ATM، ترنز ABS، لسی هوا، درین گز ریچل

جایگزینی، GPS، یخچال، نایکر دریو و سیستم کنترل سازمانی BMS، پیغامات گیرندهای مورث

... POS

مارٹ پریاپرنس PC و سیم لئے سدھے

برنامه نویسی مجازی PC با استفاده از Application از سورچ کده توسط یک سیم کابل با استفاده از فرآوری های دارای

ماده ۱۱: نویسنده باید از برنامه نویسی از سمت افزار نیز دستیار اخراج را داشته باشد

سندھیاں نسبت افزایشی Firmware گویند

بسیار اینکم سیم کسی بقیه سده هم کوادر تجیهزات نهادی، صنعتی و نیز شعبه مور را سعاده فرار نموده باید

دارات دینامیکی مجهز نسبتی می باشند (Safety, Maintainability, Reliability)

Security ایمنی

سیستم آنلاین بازرسی موارد زیر را شامل نمایند است که بر بازرسی آن سیستم دارند

صرف نرخ ، حجم بزرگ ، زمان اجرا ، Power Consumption

از دلایل مزایه سیستم آنلاین میتوان به عملکرد کاربری ، عملکرد بینرال

User interface و کاربری ، Real time اساره نمود

برای میان رفته ای از PC عملکرد خواهی ای را میتواند

میتواند اینجا باعث داده ای از سیستم آنلاین نمایند بعده باعث راهنمایی نمود

راستا کاربری ای از LCD ، Touch Screen ، Keypad ، Push Button ای سیستم

راستا کاربری ای از میکروپردازه ، میکرو کنترلر و موس

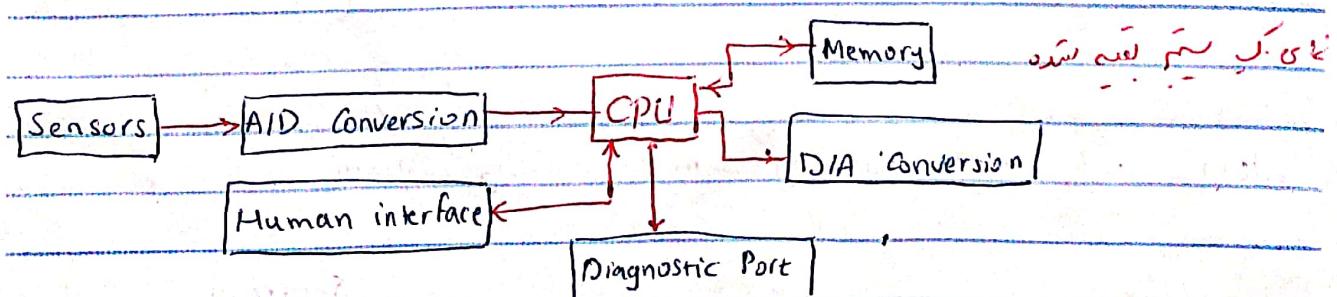
کاربری سیستم آنلاین ای از میکرو پرداز ، میکرو کنترلر

میکرو پرداز و بین بین تاکنترل سیستم آنلاین (از طبق نویز داعیل دستور) و خود زنگ بینرال

EMI : Electromagnetic Interface

نحوه ای ارتباط میکرو پرداز آنلاین با زیستی بینرال لازم است تا آنلاین از سیستم بینرال به سیستم

آنلاین داده ای از سیستم بینرال دستور دیگر که کار از این عمل را انجام دهد ترنسیستور ناسیه می شود



جی‌پی‌سی واحد گلکسی سیم تبیه شده و با ۱۰۰۰ مگابایت رام و پردازنده مرکزی ۱۰۰۰ مگاهرتزی

۵. باشد ملیح پردازی کی بود بر دریکی ستم نایمیوتیں تو سما این داده همین حی سردن

در گیر نمای. مگر هرگز نمای نمی بود زیرا هر توانید نقش داده که کوچک یا بزرگ نمای

وحدة المعالجة المركزية (Central Processing Unit)

برنامه های سیگنال ریجیستر Digital Signal Processing: DSP

پردازندگان بر حلاف پردازندگانی های هم تقدیره برای پردازش در زندگانی سه اند

میلیٹری ڈرلن ریپبلیک، پردازش صورت رئنگہ را با سرعت بای بی انجام دهد۔

✓ مکمل رئیسیتیو: یعنی نہ کسی کو ملک یا حکومت کے داری ایسا ایجاد کرنے کا اقرار جاننے

پاکت زیل لنج ۲ مدلیوں DAC و ADC و عنیه ریزتر لست Memory و CPU

• Controller : PLC : Programmable logic : دستگاه کنترل پردازی

فاینڈنگ‌های مخفی اسنادی سور و دارای ویژگی‌ی زیرا است.

۲۰۱۷-۱۸۱۷ تغییرات میان راه راهنمایی و تدریس خارج از کتاب در مدرسه ایمنی باشد

نخن و تفاوت‌های آن را در میان دو برادر اندیشه و لرزشی می‌بینیم: جان لیلیو اول نیست،

Subject :

Date :

FPGA: Field Programmable Gate Array

کاربرد اریتمتیک مابین بینای ریزی

ما تدریس می‌سینیم بی‌بینی بیوره و سکو سابل آرایی از نتیجه هستند می‌توان بررسی آنها

بعد نوع منع امداد رسانی را از بایی بیان دهانی کرد.

ASIC: Application-Specific Integrated Circuits

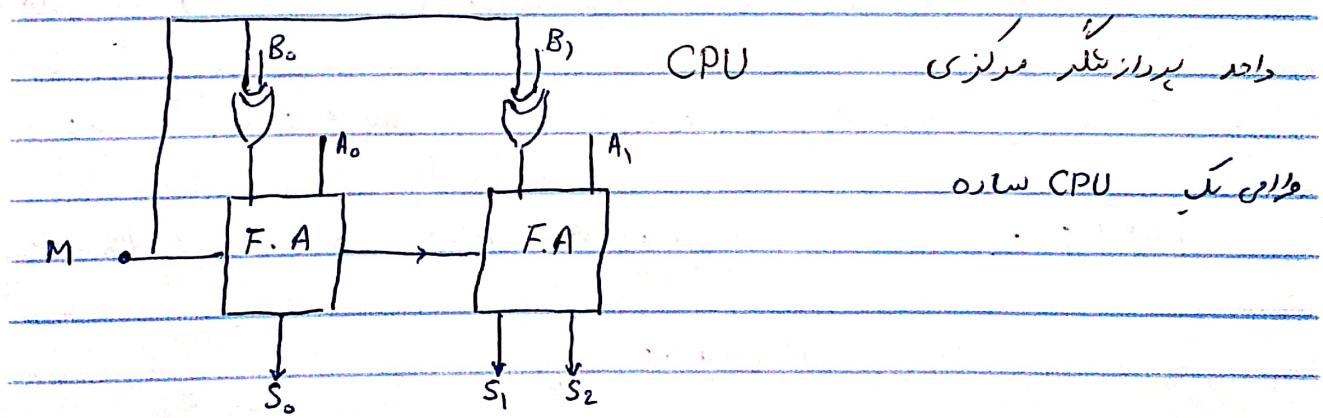
این مدارات بعین بعین کاربرد بسیار زیاد به صورت کاملاً سفارشی شد: برای هدف سفارشی

کارخانه سازند، تکمیل سکوند این سلسله باعث ماهنی هزینه می‌گشود خواهد شد.

بندهاین نسل داده لذتگیری تراویز ایک پردازند. ده سقوفه، پلیمر لذتگیری

و این سقوف ای بعین تیزابایی این می‌گشود لذتگیری آن معمول این معمول این معمول ASIC تراویز

می‌گشود تا هزینه محدود سهایی ماهنی یابد.



لقد عرض رکور
و این میانه
CPU

M	A ₁	A ₀	B ₁	B ₀	S ₂	S ₁	S ₀
---	----------------	----------------	----------------	----------------	----------------	----------------	----------------

if $M=0$ then $S=A+B$; if $M=1$ then $S=A-B$

آن چنین: داده اینست نتوسخا کمپیوتربول همچ واسطه ای این ای می‌گشود در این میانه

• MICRO®

۰۱ میلادی سنه است.

که باز زن سین شل بوره در فتح ایران در برگ نمی بزین سین نایبی، هادیت فرسایی است لذا از نیزه و دف بجای کدهای ۱۰۰ استفاده نمی شود. که در اینجا

باز زن این Assembly language چنین نویزد. زن این این بعده ترجمه باز

رجا تفسیه CPU نیست لذا از نیزه برنامه مترجم بخواهد که ترجمه باز این این

زن همین سورس است و خارج نماید.

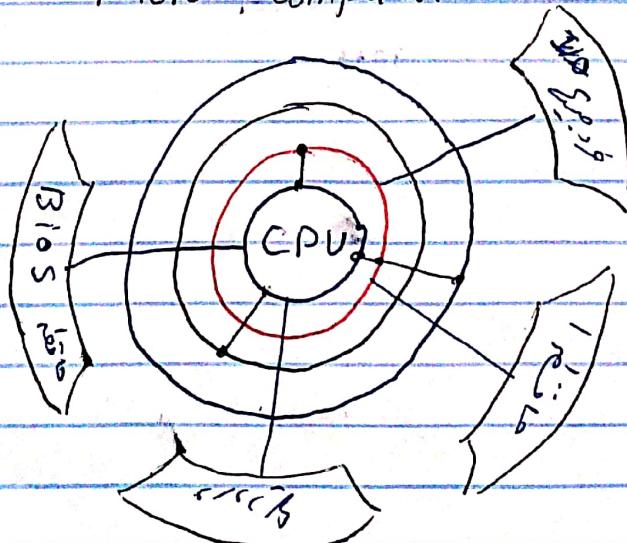
نکته: باز از این قدر دارد زن سین تا بدین سنت افزایی در داخل CPU مانند که آن کدرا

ایجاد کند با این نوع زن بزرگ نیزه سنت لزای CPU سی دی زن زن سی دی زن زن.

زن سین = زن این ، زن سی دی زن

زن سی دی زن = C# ، C++ ، C

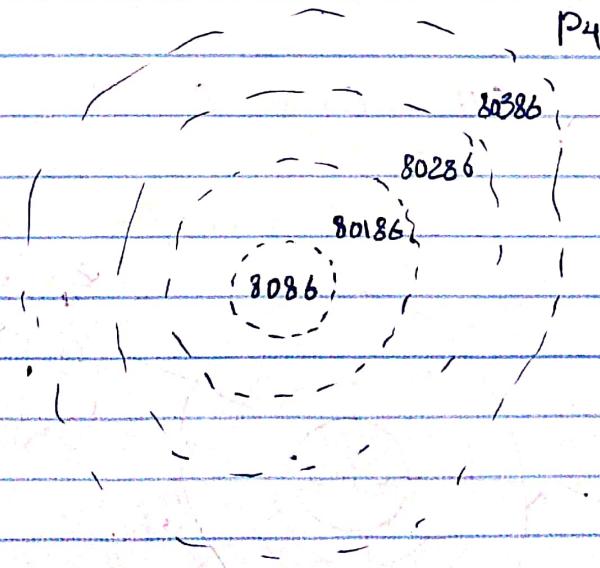
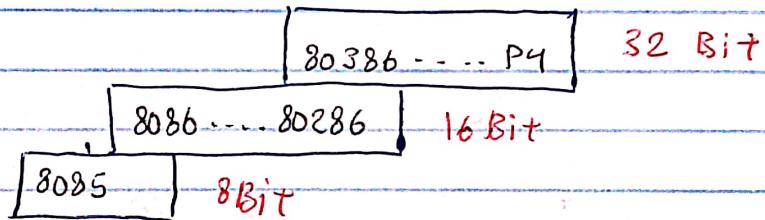
Micro Computer



تکلیر لتررها هاتند سیکر کاسیوئر های دارای CPU، حافظه، لرنرها و ورودی خروجی هی باشد
برای تفاوت نه در معنی سیکر کاسیوئر های حافظه هی تکلیر
نمایه در بنای نویسی سیکر لتررها نیز هی تران از زبان الگوی دیازنرایی سطح ۷۶۰ برایم
Famili

MCS51	8051	استواره در
Micro controller	AVR	Atmega 32
	PIC	16F 8877
	ARM	

سیکار کاسیوئرها مبنی بر IBM-PC



تران از زبان Basic، C، C++ برای نویسی سیکر لتررها استفاده میشود بسیار قدرتمندترین

استفاده میشود از آن استفاده خواهد شد.

ترانیی برنامه نویسی مقدماتی به زبان C

هر برنامه به زبان C حداقل ۱ تابع دارد که این تابع main نام دارد. هر برنامه C از این تابع آغاز می‌شود. نام تابع دیگر از این نام بعوستیم فرمانی دیگر نیستم فرمانی می‌شود.

عنصرهای تابعی ۱. identifier

عنصرهای مسائل نام تغیرها، توابع، برای ها و سایر اجزایی هستند که بوسیله ای برنامه نویس ایجاد شوند.

C در زبان Case Sensitive است بین معنی‌های بین دو کوچک و بزرگ تفاوت نمایل می‌باشد.

نیز این Number، number نیز این معنی است. نام اساسی از توانی که این کلمات کلیدی باشد.

انواع داده: داده معنی است بعضی تغیر یا تابع باشد بعدها می‌شوند در زبان C ۵ نوع داده دارند.

دروی عدی طالع تجزیه وجود دارد

۱) bit: نوع داده Bit توانی که دارند در وضیعت صفر و ۱ را دارند

۲) char: برای تعدادی کاراکترها دارند ASCII کاراکترها ۸ بیت اینها دارند در صورتی

از این نوع داده برای تعدادی کاراکترها استفاده می‌شوند همچنان که این کاراکتر را در تغیر را بعوستیم

متغیر نسبت داده این کاراکتر را بل و علاوه نیز قبول می‌کنند از این عزیز نه عدد و مقادیر

۳) A' کاراکتر کاراکتر A هست

۴) int: همچنان عدد صحیح را ذیغه کنند

Currency : float (F)

ایس بارقت ساخته ایس فلوت (F) : double (d)

انواع داده های AVR در کامپایلر

نوع	Bits	اوجی	عملیاتی
bit	1	0, 1	
char	8	127 ... -128	
unsigned char	8	255 ... 0	
Signed char	8	127 ... -128	
int	16	32767 ... -32768	
short int	16	32767 ... -32768	
unsigned int	16	65535 ... 0	
long int	32	214783647 ... -2147483648	
unsigned long int	32	4294967295 ... 0	
Signed long int	32	2147483647 ... -2147483648	
float	32	$\pm 3.402 \times 10^{-38}$	$\pm 1.175 \times 10^{-38}$
double	32	$\pm 3.402 \times 10^{-38}$	$\pm 1.175 \times 10^{-38}$
Signed int	16	32767 ... -32768	

بینهوده ایس short, long, unsigned, Signed

کامپایلر می تواند تغیر رده

لئے در کامپایلر AVR ایس ایکن جو در در رک نماینده ایس char بیت مورن پیش فرض

عملیت در در نظر گرفته شون بین تنقدور می توانند از سید Project X Configure

کامپایلر نماینده ایس char is unsigned در کامپایلر C compiler

(a)

Subject :

Date :

تغییر :

نام متغیرها قبل از استفاده بسازند باید Declare مسوند

لقد نیز عدم تغییر ذ مقدار از [سیکوند کلاس ذ پرینت] نمایه نوع را در

char i; long int code; float pi=3.14; char ijk;

نکته: در صورتی بخواهیم در این بروج دریک تغییر باقلم تغذیه ازین بزرگی را که از کلیس

EEPROM char beta; کلیس EEPROM است اسعاره کرد

مقدار حافظه EEPROM حدوداً 100,000 بار نوشت و پایه کردن است و در صورتی از این مقدار

تجاهز سویر حافظه از کار نمی‌افتد.

استی با اینها تغییر عمومی (Static) و محلی (Local) و ای (Global)

ب محدودیتی که تغییر در آن ساخته شده است حزمه Scope لفته مسند تغییر می‌کوایند در دو حوزه

محلی ب عده اعلان سویر

main() تابع

تغییرهای عده: این نوع تغییرها در نام برنامه ساخته شده اند و مقدار

بدنه تابع main

آن مادر نام در این اجرای برنامه حقایقی بسازند در این شرایط همان

تغییرهای عده نیست

ب مقدار تغییر از داخل هر تابع راستی بسازد. تغییر Bit های اطلاعاتی را نمی‌گیرد. بدنه تابع ا

تغییر محلی از برخلاف متغیرهای عده ای، این تغییر تابع استفاده بوسیمه

تغییرهای عده نیست

نام تابع نیز ب مقدار تغییر محلی با اجرای آن) هم از حافظه و پرینت از کاری

بدنه تابع نیست

MICRO

Subject :

Date :

تحصیلی درده سده دین از شرح از تابع این مقنای آزاد می شود: این سقیرها در داخل تابع صورت پذیر

+ تغییر می کرد

تغییر می کرد: آن ریشه از سقیرها می باشد که لازم است بین از تابع ابر تابع تابع را فرض کند

باید با سیکوند اعلان (declare) سووند

F1 () {

int a; Local variable

static int b; Static variable

;

نکه: اگر بخواهیم یک سقیر را در آرین خاص از حافظه ram اعلان کنیم از @ لعبات زیر استفاده کنیم.

int a $\text{@} 0x80$; Hex تغییر مفهومی a را در آدرس 80 از حافظه SRAM کنیم

نکه: سیکوند را در AVR دلایل 32 رجیستر را داریم که هر یک دلیلی بیکاری هست که در آن دلیلی بیکاری رخواهی رجیسترها

باید باید بیکاری دلیلی است بین براین هست است سقیرها را نزول دسترسی به آن ها اعیانی دارند و تغییرات

برای دلیل درایل رجیسترها اعلان سیکوند برای این سقیرها از سیکوند register اسماه می شود.

register int a;

نکه: اگر آن آرین سقیر معرفی شود درین از رجیسترها R14 تا R1 نمی کرد به عنوان نکار سیکوند

نیز سقیر b را در رجیستر R10 اعلان می کند.

کلمه کلیه: کلیه بکار خواهد بود از سقیرها را به سقیر بجهت این دوین **MICRO**

(4)

Subject :

Date :

اگر در رجیستر های کاری ذخیره می شود در صورتی از کلمه volatile برای این تغییر اسقی دیگر

کامپیوکنیتی را در SRAM به بسته افتخاص فواهد دارد. آن را در رجیستر های کاری ذخیره نمی شود.

volatile int a;

ثوابت: ثابت مقداری است که در زن کامپیوکن جهش از لد برنامه را وجود دارد. حافظه flash منسوز

و در زمان اجرا عالی تغییر نیست. ثوابت از نوع عدد از مقال باشند 0XF3 (نمود 0XF3) ثوابت باشند

باشند 0b (نمود 0b01101) از این و آن ریکنندی بگارند. تعبیرات رسیکل تفسیری نمود.

لقریب تاب با مده کند

مکل

flash int CODE=1234; و const int CODE=1234;

const float PI=3.14;

نام: کوکب تری صورت بحصی استواره می شوند و داخل توابع کم اعدان نیست.

استواره از سیوکندهای زیری که نفع را دری اینست را تغییر کرد.

نوع دارد

منسوز

نام: 0b و 0X نیز بعنوان

unsigned integer

U

تغییر میکنند اعداد اتفاذه نیم

long integer

L

unsigned long integer

UL

float

F

بعضی از محفوظه برای در 99 تعبیرات پیش ذمی نوع را به کامپیوکن این اگر لازم باشد بگذار

تاریخی این رفتار مسوزی که این سیوکنده بعنوان تفسیره کردن.

99L

 MICRO

#define **رده همور پیش برد از نهادی**

با #define پیش برد از نهادی از جمله **Finish** هوانا نیز می باشد

با این بایت اینکه در این سورس و نویس بدل از کامپیل سدی برای مقدار بورد نظر چاپیزین می شود

با مفهوم میان در کد زیر هر جای برنامه از PI استفاده سورس باشد 3.1415 چاپیزین خواهد

#define PI 3.1415 **لیزیز چاپیزین می سوز**

#define LED_PORT PORTA

لهمه اینکه مقدار میان از رده همور #define می بینیم

نویسی دلخواه را دست تر برای این بقایت میان کسر رکوردهای دیگری سورس

لهمه برای تعریف تابع های سیگنال از کلمه کدی enum نیزی روان استفاده کرد

Array **کاری**

زین C از دو نوع دارای دو همراه ساخت یافته باشد که در نوع از آن های آن را باید

و ساختارها خستند. با استفاده از آن های هوانا نیز مقداری برای بیوئی از عاقله ایجاد کردند

که از میان نوع را در عین قابل بجزیه double float, int, char

char A[6];

char A[6] = {211, 'C', 165, 17, '1', 25};

int B[2][3];

int B[2][3] = {{1, 2, 3}, {4, 5, 6}};

تعریف کاری

لهمه الگوریتمی کاری برای Code Vision AVR این اعلان کاری با 8 بعدی بود دارد لایه برای این نهاد

MICRO

توصیه داشت نه حافظه داره سایر رکنتر لر محدود نهاده و ممکن نهاده فهم زیوری را با این نهاده اشغال نمود.

flash int B[2][3] = {{1,2,3},{4,5,6}} تعریف نمایه ای از آرایه

آرایه از A[0] مسُرد مسُرد A[n-1] خانه های اید لین ۷ ععنو دارد را این ععنو تعریف نموده در اندیش مفقر ترازی نمود.

لست

در زین ۷ برد خانه برد خانه زیگمی دیگر مانند بسیار فرعی داره هی رسم وجود ندارد. در این زبان رشته های متنی بوسیله می آرایه ای از کاراکترها ایجاد می شوند. تعبیان مثال خطا زیر یک رشته را ایجاد می کند

char str[10] تا ۹ کاراکتر را ذمیه نمود

در این و [۹] آرایه str کاراکتر رشته ای بسیار اماعل است (یک آرایه ۱۰ ععنو حداکثر می کند)

رشته ای و کاراکتر را ذمیه نمود null terminated بولن رشته ها مرتبط است

لکه در زین ۷ هدر رکنند بسیار ASCII صفر یا کاراکتر NULL نه بصورت ۱۰ ایست عایق داره

نه سکون یعنی هدر رکنند ۱۰ را قدر ریم که وکار نهیم ۱۰ به آفر ریم لفظ فرمی نمود

char str[10] = {'H', 'E', 'L', 'L', 'O', ' ', '0', ' ', ' ', ' '}

نه کاراکتر نه کاراکتر رشته به HELLO اضافه شد و بقیه عایق های رشته خانه ایست هم کوان بصورت نمی نماید تعریف

char str[10] = "HELLO", مفهوم

نه کاراکتر نه کاراکتر رشته به HELLO بسیار خانه هایم به ۱۰ و بقیه تا [۹] str خانه ای سی نماید

در صورتی که عدد داخل براکت حذف شود حامیابی نموده فوراً بقدر کارکرده ایجاد می شود

char str[] = "Hello"; را باید کارکردن 10 باینتر را نداشته باشیم لذت

استفاده از کلمه کلیدی const هنوز رکن را در حافظه Flash ذخیره کرد.

شکریتی دار کوئی نداشته باشیم

Const char str[] = "HELLO"; آن را در حافظه ذخیره کرد.

آنچه باعث می شود این است

ساختار یا کوکر هاست آرایه هنوز مسائل خوبی عضور از اینجا در میان float, int, char و وغایر

و رکن باشد با این تفاوت که برخلاف آرایه هایی به تسلیم بورن نوع داره ها کوکر نمی شود بلکه ساختار

با ایجاد الگویی برای تغیرها را ایجاد می کند لفظ بیان می کند، دستور زیر این را نمی شود

Struct measuredData {

 char minute;

 char hour; ← هر کدام از تغیرهای ساختار را که عضو member نام داشته باشد

 float value; ← داشته باشد

}

درین مثال نقطه تغیر ایجاد شده برای ساختار measuredData ایجاد شده و صحیح بیان از مفهوم

استوار آن اینکه این داده های داشت بنا بر این هنوز می توان این داده های را ایجاد نمود

Struct measuredData temperature;

Struct measuredData humidity;

در این حالت دو ساختار برابر باشند $\{ \text{temp} \text{, humidity} \}$ و $\{ \text{temp} \text{, value} \}$ که مجموعه ای از $\{ \text{temp} \text{, humidity} \}$ و $\{ \text{temp} \text{, value} \}$ است.

لکن اگر این دو ساختار را در یک struct قرار داریم برای رسمین باید این دو ساختار را در یک struct قرار داریم.

`temperature.minusC = 10;`

`temperature.hour = 15;`

`temperature.value = 27.5;`

آنچه این از این دو ساختار را متفاوت می کند این است که مجموع اندیازه های کاملاً متفاوت باشند.

Union

این نوع داده سلیم ساختار است با این تفاوت که در زمان اینها می توان سیمی از این نوع برخلاف ساختارها

تفاوتی برای نام اینها Union نیست بلکه مفهومی از آن تغییر نمی کند.

برای این داده از ساختار `union` استفاده می کنند که با این کلمه ممکن است این داده را در یک struct

`union Mycode`

{

`char a;`

`int b;`

`float c;`

}

از `union` استفاده می شود.

تفاوت تضمینی یا فنی باید تغییر از نوع Union برای با اندیزه های متفاوت باشد.

عنوانی این داده ای است که در این در این `union` دو ساختار متفاوت باشند.

عنوانی این داده ای است که در این `union` دو ساختار متفاوت باشند.

عنوانی این داده ای است که در این `union` دو ساختار متفاوت باشند.

`96. a = 126;`

`96. b = 1052;`

`96. c = 1245.83;`

در این داده دو ساختار متفاوت باشند.

Subject :

Date :

log

Under & Bitwise or Arithmetic عمليات بิตية أو حسابية في C C++

and assignment -> logical Relational ومتقاربة

رمز	عنوان	النوع	نتيجة	بيان
*	ضرب	3×2	6	
/	قسم	$5 / 2$	2.5	
+	جمع	$3 + 6$	9	
-	تفريح	$8 - 3$	5	
%	باقي	$10 \% 3$	1	مقدار بقى على رابد المجموع
&	AND	0xFF & 0x0F	0x00	
	OR	0x00 0x03	0x03	
^	XOR	0x0F ^ 0xFF	0xFF	
~	نكر	$\sim(0xFF)$	0x0F	
>>	زحف بيمين	$0xFF >> 4$	0x0F	
<<	زحف بيمين	$0xFF << 4$	0xFF	

مقدار بقى على رابد المجموع يساوى 1.5 و المقدار المخزن في int هو 2.5

float b = 2.5 (القيمة المخزنة في b هي 2.5) و المقدار المخزن في int a هو 2

نحوين . b يساوى 2.5 و a يساوى 2

Float a;

$a = (float)5 / 2$, $a = 2.5$

نحوين . a يساوى 2.5

نحوين . a يساوى 2.5 و b يساوى 2

(9)

Subject :

Date :

 $x = 10$ $x = 10$

سیوان یا جل

 $y = ++x;$ $y = x++;$ $n = y = 11$ $y = 11; n = 10$

بعد (sizeof) - این بزرگ نیز (اروپا تولیدی) نه تیوه (sizeof(char)) که خواهد بود.

عمل ماتریسی را بخواهیم

عمل	معنی	دل	نتیجه	نحو
>	بزرگتر	2 > 3	F	Boolean نوچ دراوه
<	کوچکتر	'n' < 'c'		
>=	بزرگتر مساوی	5 >= 5	T	دیگر عبارت دو مقادیر صفرها را False و اختیار
<=	کوچکتر مساوی	2.5 <= 4	T	
==	'A' == 'B'	0	F	علی صفر عبارت True خواهد بود
!=	0, ول	2 != 3	T	
AND	منطقی	(2 > 3) & & (7 != 3)	F	
OR	منطقی	('a' < 'z') (10)	T	
!	تفیض	(4 < 5) !	F	

عمل ماتریسی از اینا برا

عمل	دل	عمل	دل
$a \leftarrow b$	$a = b$	اسا ب	=
$a = a * b$	$a * = b$	ضرب راسا ب	* =
$a = a / b$	$a / = b$	قسمی راسا ب	/ =
$a = a + b$	$a + = b$	جمع راسا ب	+ =
$a = a - b$	$a - = b$	تفاضل راسا ب	- =
$a = a b$	$a = b$	OR راسا ب	=
$a = a \& b$	$a \& = b$	اسا ب, AND	\& =
$a = a ^ b$	$a ^ = b$	XOR راسا ب	^ =
$n = y $	$(a > 0) ? y = a : y = -a$	آنت بیت راسا ب	$a = (value) ? n : y$

1C100 : 3C05

آفون بور عمله بارست

در پیورتید نتیجه نتیجه عبارت ۱ صحیح باشد عبارت ۲ اجرای آنده و غیر اینقدر عبارت ۳ اجرای خواهد بود.

СИЛЫ

نحویات برداشتی از انتشاریات

初中数学教材

توضیحات راجع به نیزه

11 * * * 2 , 0

```
#include <mega8.h> header files
File name
#include "header file name" header files
File name
interrupt [....] void .... (void)
{
```

رسن روایا به رفعه های نخال شد در داخل برنامه

2

return_type function_name(Parameter_list)

{

نوابع که در داخل مرنا به فراخوان گردید

1

تعریف تقدیرها سراسری از این ارزشیان تقدیرها در داخل
توابع ما و بر ترتیب و قوه استفاده شده بوراید تیار آمیزه شد که

1

10

main سرمه تا

17

رسوارات این مدت فعال نیستند و بعدها ز دسترسی **reset** نیز برداشته شوند. این اتفاقات از نظر امنیتی می‌باشند.

Table 11

3

منابع

~~OMICRO~~

1

عد تغیری در نصایی برخانه، حافظه ای را استعمال کنند این حافظه می‌آرسد درین اسکاره مقدار دری

آن را در نمود دنیه های نشان می‌کنند مثلاً اسکاره ترکیب زیر است:

ج) `unsigned char *value;`

void main()
{
char b, *a, c;
b = 0xF1; // b = 241
a = &b // a = Address of b
c = *a // c = 241
}

هر دو دستورات ایجاد شده اند:

شیوه بردازندگی نهیت از کامپایلر است که قبل از شروع برنامه اصلی باید

شیوه کوئی کامپایلری نیست که بردازندگی این دستورات را بین

این دستورات بدان رجوع به فایلی که در آن دستورات در آن دستورات

کلیدی تعریف شده اند به کاربر برای این دستورات این دستورات را در این دستورات

نایاب نمایند این دستورات در Codevision درین شیوه سیر برخانه ای دستورات دستورات دستورات

`#include <filename.h>`

`#include "Address/file-name.h"`

Subject :

Date :

#undef

برای این کار که #define نمایند بگذارند ریسون لغزیدن نمایند

تعریف نمایند #define pi 3.14

برای این کار #undef pi

امانه درون کرده ای اینها برای تایپ

کد هایان با استفاده از ریسورچ ایمبل را در هر قسم از برنامه خود

#asm

اصنایع خالی :

nop

nop

#endasm

#asm ("sei") /* enable interrupt */ سر کل سر

لایم

تکانه سرچ یعنی اینی تابع که اجرا می شود (اگر main بسیار تراجم در دن این یعنی یک فایل از این

و شکنند (به بزر تراجم دفعه)

Return-type function-name (Parameter-list)

شیل کلی تعریف ۲۰

{

Body of the function.

}

در تابع فوق return-type خوب یعنی اگر تابع مترجم نمایش باند نمایش برخی آن باید void بود

سیو در این بگرداندن مقدار تابع از ریسور ایمبل return سیو در این فایل void باید اضافی شود

int sub (int a, int b)

↓

Return a+b;

}

:

↓

Void main()

{

int c;

c = sub (5, 15);

بینی ریسور نیست. شل

 MICRO®

Subject :

Date : _____

```
#include <header.h>
int Var1, Var2, Var3;
int Func1(int, int);
void Func2(int, int);
void main()
{
    Var1 = Func1(1, 5);
    Func2(Var1, 0, Var2);
    Var3 = Var2 + Var1;
    while(1);
}
int Func1(int value1, int
{
    return value1 + value2 + 5;
}
void Func2(int value1, int
{
    value2 = value1 + 2 - 1;
}
```

کلمه `float` در زبان C به دو صورت هستند که ترتیب که ترسناک است و ترتیبی که بروزدیده شده ترتیب `(1)` است. دستور `#include <math.h>` برای این کار از کاربر نیاز ندارد. باید استفاده از ترتیب `(2)` که بجا نهاده شود. این کار را در کد زیر مشاهده کنید.

```
#include <math.h>
Void main()
{
    float pi=3.14, b;
    b = sin(pi/2);
}
```

• MICRO°

دستورات کنترلی
for (تکراریت و سرطان چشم و مقدار کاری ایندیک) For (i=۰; i<۱۰; i++)

```

    {
        بین
    }
  
```

while (تکراریت) while (۱) while (۰)

```

do {
    int a=۰, b;
    do {
        a++;
    } while (a<۱۰)
  
```

if (تکراریت) if (۱۰) کاربرد
 {
 بین
 }

کاربرد if لیکن اگر تو در تو نه نمایش داده شود

موجودیت اولیه داشتند بین اینها نمایش داده شدند

لیکن از خواندن آن نیزیں گاهی داشتند
elseif (تکراریت) elseif (۳) if (۳) elseif (۲) if (۲) elseif (۱) if (۱)

یکبار گرفته شود و میراث خواندنی برای اینها بود

else { دستورات غیر معمولی }

Switch Case

از این ساختار برای تعمیم تریوی هندگانه برای مس مقادیر مختلف یک عبارت استفاده می شود

switch (عبارت) {

لعل همان عبارت زیر تعریف می گردد.

Case 1

ذ مقادیر 1

فرستادن 1

break;

case 2

ذ مقادیر 2

فرستادن 2

break;

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

ذین دستور ممکن است switch (عبارت)

فرستادن 1

فرستادن 2

فرستادن 3

فرستادن 4

فرستادن 5

فرستادن 6

فرستادن 7

فرستادن 8

فرستادن 9

فرستادن 10

فرستادن 11

فرستادن 12

فرستادن 13

فرستادن 14

فرستادن 15

فرستادن 16

فرستادن 17

فرستادن 18

فرستادن 19

فرستادن 20

فرستادن 21

فرستادن 22

فرستادن 23

فرستادن 24

فرستادن 25

فرستادن 26

فرستادن 27

فرستادن 28

فرستادن 29

فرستادن 30

فرستادن 31

فرستادن 32

فرستادن 33

فرستادن 34

فرستادن 35

switch (p1)

{

case 1:

i = 10;

break;

case 2: case 3: case 3:

i = 20;

break;

case 5:

i = 30;

break;

default:

i = 0;

• MICRO° }

Subject :

Dudu 3

break : جب فریم بدلن سکه از حلقة استفاده می شود و هرگاه برنامه به این رسمت برسد از حلقة خارج شده و راه برگردانه از نیوی فلجه ارائه می یابد.

Continue : این رسمت نیز نهاد در حلقة های استفاده می شود با رسیدن برنامه به این رسمت از بقیه رسمت های ایجاد شده دور بگذرد حلقة آغاز می شود یعنی برنامه به این رسمت دستورات تا ایمان حلقة صرف نظر شده دور بگذرد حلقة آغاز می شود یعنی برنامه به این رسمت دستورات تا ایمان حلقة صرف نظر شده دور بگذرد.

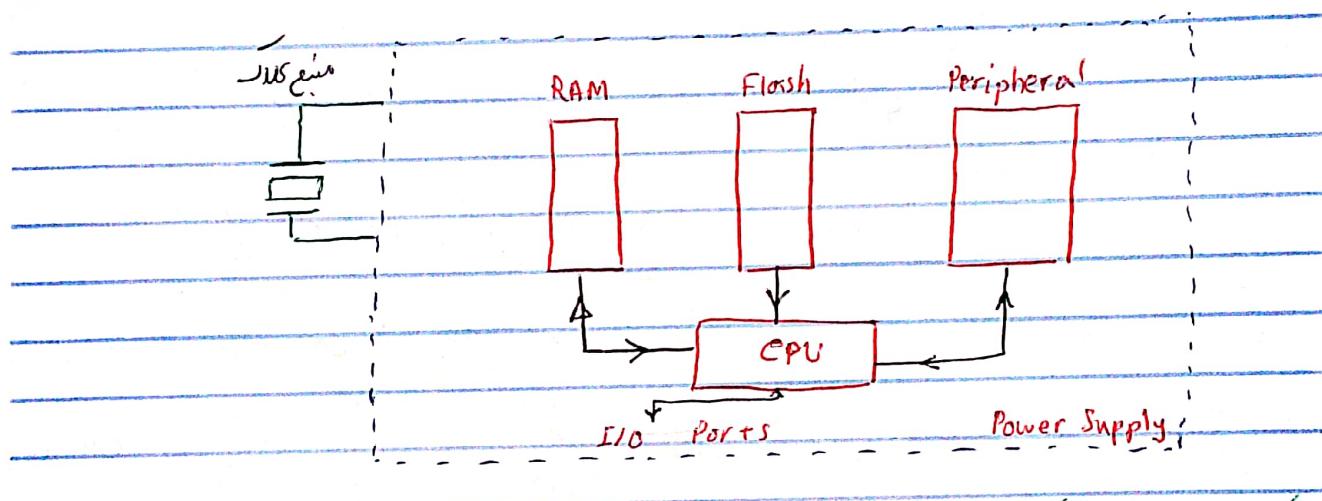
goto : از این سور برای پیش بگیر برای (سفاره) سور برای کم می خواهی پیش بگیری

نے رام نے بگانہ ایسی :

بيان طراحی میکرستم تینر بر AVR

مکرر لینتر را نوعی صادر یقین است که برخلاف ریز پردازنده بتعاقب بین این نیاز دارد این رسیده بیشتر خالی سه داشت که نسبت کاربرهای لینتری بوده دباقیت یا سین در لینترین است. افزایش آن در زیر ریده هسته ای هسته ای رسید تغذیه، مدارهای مکرر بتوسان کرده رنام افزا آن هزه، هسته ای سیستم برنا مه می پردازد

حافظه برنا، رسیده CPU حفایه حفایه ای سیستم

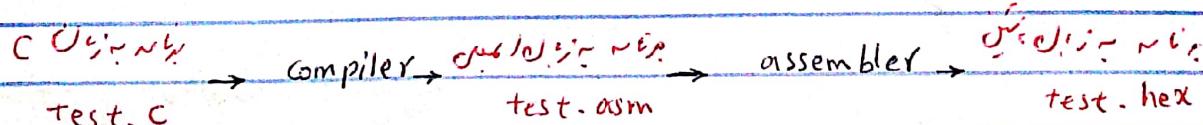


لایه در سورس میکر لینتر

محصول کار برنا نویسین ایکسی Flash or ROM Firmware میکر لینتر ایکسی و تغییر نمیکند، برای نویسند

تغذیه این برنا رسیده CPU حفایه حفایه ای کردن

روزه رسیده برای رسیده ای کردن



جیلیم میکر لینتر فلاش روم را برای ذخیره نمود.

تکمیلی سرمهی ۵ نی AVR

LSP: In System Programming

۱) استفاده از موس

Hypopituitarism

۱۲. بُرْنَارْدِزْ بُرْنَارْدِزْ بُرْنَارْدِزْ بُرْنَارْدِزْ بُرْنَارْدِزْ

HRSP ! عوادلی و رفاقت نیز نیز میگذرد (۱۵)

JTAG (en, jez ien) (E)

1) جنریٹر Boot loader میں دوسرے طرح

متاسیم بین المسمی ایزیزی سطح بالا

زیرا یہ زبان دنیا کے سطح بالا ہے۔ (لیکن)

۱۱) زبان اصلیه بیردزه: زبان های سطح عالی بازساختی در نیات بروجایه سنت افراز از زرگریدن بینایی نوین

بابیاری از بزرگیان سطح یا زین بدلگیری می‌کند. این مدل‌بیان کوچک‌ترین زدن ترسیمی برای رهگذر راه

ولذنیم هنرمند ایران آن را مامن می‌ردد.

(۲) برنامه نویسی ساخت یافته زبان های سطح بالا بطور معمم از ریسکرات استفاده، حلقة ایستگاه دوام

سُرْجِلَة

۱۵. اشغال‌زدایی Debugging

۲) ذهنیه سازی در رازیک راردها: زبان از نوع رله سنتی می‌باشد که آرای و ساختار نتیجه‌گیرنده

۲) تواضع لسانی از

۱۴۰۴ سیستم ریال تیک

۷) اسکاره از RTOS

کیا زو تیکس مسکر بیاره از سیستم های تپیه سه عملکردن رنگ آن هاست راهنمای عمل بیاره می باشد که برای این هنر رفیع ای که لازم است تا بخوبی روز طالع آن سیستم بی رنگ عمل کند اسکاره از آن سیستم بخوبی رنگ باشد RTOS است مینیمی AVR تباره داریکان سیزد بیان C بار و جرودر.

۸) گرافیکی و اسکاره کاربری

میزدیز نیل اسکاره بین سیستم

۱۱) دیش نیم برنامه، ۱۲) سریت ایمیزی بالاتر ۱۳) توان معرفی نیز

اسکاره با سریت لیکسی خانه AVR

Automotive AVR

lighting AVR

USB AVR

AVR Z-link

mega AVR

XMEGA

CAN AVR

Smart Battery AVR

AT90S (منبع شده)

LCD AVR

tiny AVR

tiny AVR ۶۰۰

کیمی اسکاره این خانواره سیم آن ۴۸ یا ۴۰ یا ۳۲ یا ۱۶ یا ۸ گیگابایت رهایقی

۵۱۲ بیت بین میز SRAM

mega AVR ۶۰۰

۵۱۲ بین میز SRAM

MICRO

مکالمه هشتم قسمات این سری سین 28 تا 100 یا بیش دارا هستند

آشنازی ! CPU سرور لنز لرهاي AVR

ساختار میکروکنترولر AVR به گونه ای است که بیست و سه بروت درونی از تقاضی ریزتر ناچی عمل می کند
و بیست و سه بروت 32 بیت ها قدرتی بینام های R0 تا R31 است که از کافی فیزیکی بزرگی از CPU
بوده و در تنظیم سریت ریزرسن به آن می باشد.

رنی از ریز سرور لنز سرور لنز AVR RISC: Reduced Instruction Set Computer

برخلاف بردازدهای MCS-51 ماتریسی CISC: Complex Instruction Set Computer ای از

سریت روتول ریزرسن می باشد که تراویزی دارد.

خوب گفته فناوری AVR در SRAM

که بیش از کاربردهای SRAM استفاده دشته از آن بینان Heap و Stack است که گفته این

فناوری ماینر Codevision قابل تنقیم است.

Heap و Stack در SRAM می باشند و جو سیمی تراجم گفته

حافظه رنگی خواهد است نار ماینر است و می سود

حافظه نیازها برای راه اندازی سرور لنز را

بدهیزین مداری که آن ها را راه اندازی کرده ایم بگر بیننند یعنی سیمی کلید

Reset می باشد

Subject:

Date:

آشنایی با نویزهای فریز بدیت

نویزهایی که از میکروکنترلر AVR میباشد که این دستگاه بجزئیات بسیار بسیار

۱۰۴ صفحه کتاب مرجع باعث نویزهای فریز بدیت را به کار برخورد

آشنایی با نویزهای فریز بدیت

نیزهای فریز بدیت AVR Atmega32

۱۱ زوایر RC ۱۲ نوسان RC مارجی ۱۳ نیازهای خارجی ۱۴ کریستال کوارتز ۱۵ زوایر ناشر

سیستم ۱۶ ریزیل خویش بین ۱۷ نوسان ساز تایپ / کانتر غیر قدرتی

نیزهای سازهای را فرموده

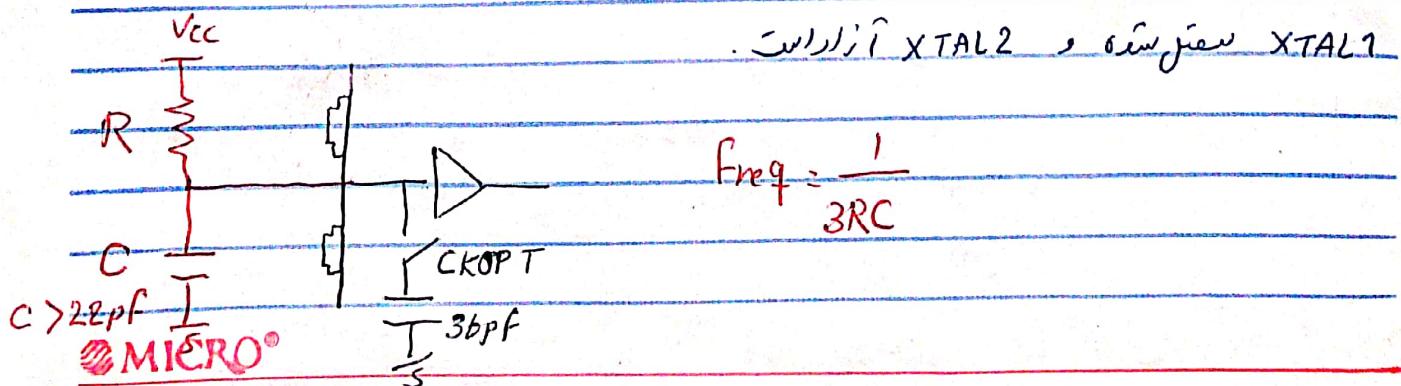
این سیستم ساده ترین و ارزان ترین منبع مولکولی میباشد.

نوسان ساز RC دارای در فرکانسی ۱، ۲، ۷، ۸ میباشد که قابل تنظیم است بدای را ۲۵

درجه سنتیگار ۵ مولت توسعا سریع سازنده کایلر، سرمه است در فرکانس 3 ± 3 درجه میباشد

نوسان RC مارجی

این مدار برای راه اندازی مطابق زیر به نیاز به نیاز دارد RC خارجی است. این مدار به عبارت

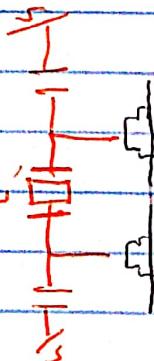


Subject :

Date :

پر کاری و در این بیان کار باید بوجه مشابه به بایدی اعمال نمود در این حالت

نیز CKSEL = 0000 بیهوده CKOPT



کرستیل کارنر یا زندگانی سریعه و

نریزی کارنر

کارنی انتشاره از اورت (Ort)

تعدادی درجه بین مکرر لذت بر دهنی خارج از طبقه بورت های درود/فرجه I/O سری دلخواه در اینجا ۳۰ میگوید

در مکرر لذت بر دهنی AVR میان بین مکرر لذت بر دهنی داشت.

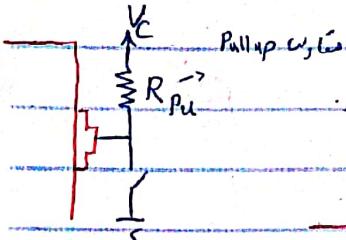
خطینه بین در ریخت درجه :

درینه سری هدین I/O در ریخت فرجه نیم سده باشد این تابیت را در این

جولان را دو بست حدایت آن اینها فرم Source ۰۱ بولان را فل Sink

خطینه بین در ریخت اورت

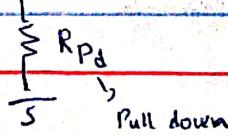
تعریف ماین سیستم دیوری با این میگردد که مکرر بین اند از زنی خارج صفر ۰۱ را دریافت نمایند



هر چنان تعریف میگردد که مکرر بین اند Pull-up میگردد که مکرر اعمال کند

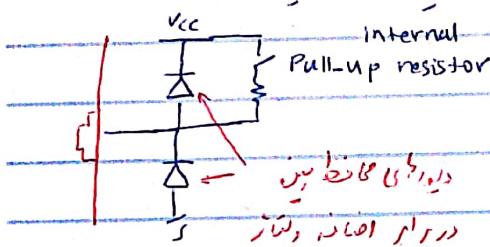
$$3.318 \times R_{pu} = R_{pd} < 10 \text{ k}\Omega$$

MICRO



Subject :

هرین سلریک تو اندر LED را بگردیم و میتوانیم نیاز بین از 200mA بودن نیست.



میتوانیم این سلریک تو اندر AVR

رجیت های در بین پورت ها

یکی از هاینها را رجیت های I/O نامیده میشوند اما درین رسمی با این اتفاق های مختلف سه افزار را به

برنامه می رهند از 64 بین رجیت I/O در تفهی ATmega32 12 رجیت مربوط به پورتین های 4 بین

سلریک تو اندر های بین ترتیب که معمولی دارای سه رجیت PINX، PDXR، DDRX بوده که

درینها نیز هم از صرف A, B, C, D است.

رجیت های PIN, PDXR, DDR

تمامین های پورت های در این اند و دری ای فرجه باشد بنابراین برای تنظیم کردن آنها

نیاز به رجیت I/O بود که برای تغییر رجیت های DDRX در این قیمت هست.

این بین سیت DDRXN. صفر بین این بین دیگر دارای بین این فرجه است.

بین هاینها از PortA فرجه بقیه و میتوانیم

نیاز در اینها لازم است تا فرجه داشت Pull-up با اینها میتوانند معرفی میکنند بین

لقد در این سیت PUD از رجیت SFIOR را کنند. (سیت های 2 از

خواهند از بین داشت و فرجه صفر بین بین های DDRX را فرجه میکنند هر کدامی از بین های دنگ

MICRO

ارطیق رجیستر تابع فواندن ایست PINX

نیمی از بندیکت نمایی باشد که در پورت A از سری نتیر Atmega32

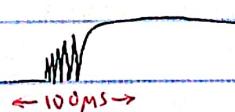
```
#include <mega32.h> .i 1,0
#include <delay.h>
#define LED_Port PORTA
#define LED-DDR DDRA
Void main()
{
    LED-DDR = 0xFF;
    while(1)
    {
        LED_Port = 0xFF;
        delay_ms(500);
        LED_Port = 0x00;
        delay_ms(500);
    }
}
```

دبلونسینگ / Debouncing

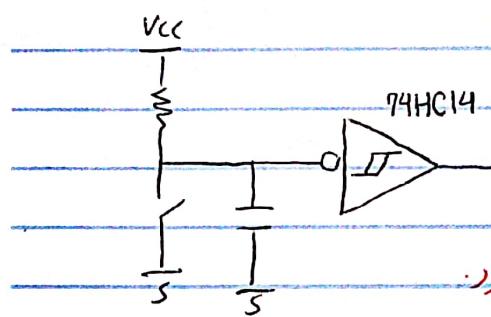
یوچیج ای معاشرین برصغیر آنچه استقر، می بود که تغییر ایمیل از 1 به 0 یا 0 به 1 نفعی نیافریده باشد

عنی لست . به عدالت خاصیت ارجاعی لست های آن هم چند سیم نانیه با برها سر رانفع رفته باشند این

لست های باندیز / Bounce



او سیکلی شدت افتراضی دو میلی ثانی افتراضی برای حل این سعی وجود دارد



دکتر مژون دهبونی Debonie مژون

در سیستم AVR، هر سین را در درایور دارایی که بازتر با اینهاات اینست

در سیستم AVR: هر سین در درجه دارایی θ بازتر با اینفراست اینست

در فوران نه مقادیر RC بسته از زنگ Bounce زنگ بسیار خارج از محدوده می باشد و در نتیجه

بررسی عملکرد مدار تائیر منزد از

برنامه نویسی کلیدی سه ده ب PBO و LED مصل سه ده ب

```
#include <mega32.h>
```

```
#include <delay.h>
```

```
#define CED_PORT_P8IRTB0 void io_init(void)
```

```
#define LED_ODR DDRB.0 };
```

```
#define key_port PORTD.0 CED_Port=0;
```

```
#define KEY_P0FT P1ND.0
```

адение key-DDR DDRD.0 key-DDR = 0,

```
void ro_init(volatile key_port_t key_port = 1);
```

```
void LED_toggle() {
```

```
void main(void) {
```

LED part = ~1 LED part;

```
while(1) { }
```

• **What is the difference between a primary and a secondary market?**

if (key == 0) {

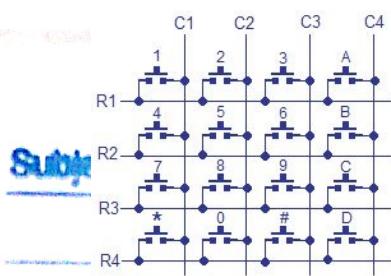
1. *What is the meaning of the following words?*

LED_top1c();

```
delay_ms(200);
```

1. *What is the relationship between the two main characters?*

 MICRO®

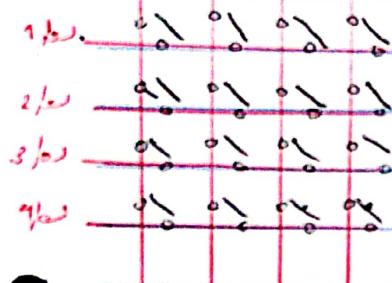


Date :

25/10/2022

لطفاً وصل كل مفتاح في المصفوفة بالصوارف المفتوحة اما فيكون له مفتاح ماي بعدي فيزالت الصوارف افتد

نحوه صوره متوجه بعدين



الله اعلم بالاجابة

مصفوفة 4x4 مفتاحي فلا مفتاح داريم لازم ان يكون 8 مفتاح قبل

مفتاح افتد

نقطهم المفتوحة لصوابون خارج در سطح نصف مفتر

نقطهم مفتوح ها لصوابون در درون اسما، اسما



$a =$ مفتاح مفتوح ها



نقطهم مفتوح ها لصوابون فوج در سطح نصف مفتر

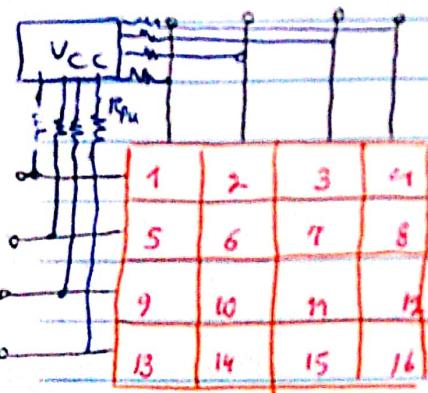
نقطهم مفتوحة لصوابون در درون اسما، اسما



$b =$ مفتاح مفتوح ها



از ترتيب a, b, a لاساس هم مفتوح ها



مفتاح مفتوح ها لصوابون 9x9 مفتاح مفتوح ها لصوابون

• MICRO°

(19)

Subject :

Date :

```
#define keypad_port PORTC
#define keypad_port_pin PFNC
#define keypad_DDR DDRC

Unsigned char keypad_read();
Unsigned char keypad_read();
{

    Unsigned char ScanCode, butnum;
    keypad_DDR = 0x0f;
    keypad_port = 0x0f;
    delay_us(1);
    ScanCode = keypad_port;
    keypad_DDR = 0x0f;
    keypad_Port = 0x0f;
    delay_us(5);
    ScanCode = ScanCode | keypad_port;
    if (ScanCode == 0xff) return 0;
    Switch (ScanCode)
    {
        Case 0xee: butnum = 1; break;
        case 0xed: butnum = 2; break;          ox7d = 1000110
        case 0xeb: butnum = 3; break;          ox7b = 1000101
        n  0xe7 = 11 4; 11 ;                 ox77 = 10001000
        oxde = 1101110                         default: butnum=0
        oxdd = 1101101                         }
        oxdb = 1101101                         return butnum;
        oxd7 = 1011000                         }
        oxbe = 1011110                         }
        Oxbd = 1011101                         }
        Oxbb = 1011011                         }
        Oxb7 = 1011100                         }
        ox7e = 10001110                        13
    }
```

١٢٣

در بین این هایی که در فعل های بدل بررسی کردیم هر یکی را داریم که ملکه ای افسن نامش باشد بود را بن حلقه، حلقه ای

• polling loop

تقریباً ATmega32 لجهز بـ 25 سینم رقمی است که است آن های عالی جدول ۱۹.۱ از شب می بندد

پرینٹر، EEPROM، ADC، USART، SPI، ورکریکو، 2x100 پیکسیل، و ریسٹ، و ریسٹ

• and for SPM \rightarrow TWL

بستھائی فعال سہیز رفع

سچھا جانے CPU بے (رخواست رتفعہ ای) یا سخ رہد آن لئے کہ رتفعہ سرور تصریف کان سوہہ بائسڈ دھور

رسئی مذکون هیچ سیع و تغه ای معال نیست. هد سیع و تغه هی ترازوست سبقاً معال یا نیز معال باشد. هر کدام

از رفته ساکن بیت فعال ساز امتحان دارند که در سورت استواره لازم است ترس طایر نباشد و سود.

نک سدن سیت نفای ساز امضا می برد و نفای سدن رفعه کافی بسته شده لازم است نک بیت خاص باشد ۱

نیز کسیدہ بائیڈی ایمیڈیا ایجنسی میں ایک حلقہ کے رہنماء تھے۔

SREG : I T H S V ~~W~~ N Z C

بیت ۱ یا فصل ساز یک مردمی رقص ها، اینکه آن استعداد از تابعیت رقص ها را تسلیم می کند. بدین معنی نه آن را صفر

باشد سیع سیع و تفاوں ناکارانیت فیکن اگر بیت قفال ساز افغانیس میں برو ٹا بیان نہ باشد، تو ای صرف

نیکردن بیت 1 درستور *Sei میان 1 کردن* و *کل میان یک کردن آن* رخداد دارد.

پذیره اجراء بر اساس ISK رقف باید ۳ سه ماهه زیر حزون برقرار بگرد.

۱- بیت نفال ساز یک یعنی رقفه ها با رسور 5d میگشند این است

۲۷- بیت فعل ساز در قسم موردنظر نمایند

۳- باعوی دارن رفدار مرد نظر پریام و قیمی آن روبراد نیست.

أمثلة على CPUs ذات المعمارية المترافق معها

آرس دستگاهی می‌باشد که CPU در حال اجرا آن است در شارژهای برقیهای از این می‌تواند با درخواست

وقتی این آرس در پیش از دنیه سده Push را پی برد آرس شفیع درین سی دن که سعدی ای

آرین: زیر نویسی‌ها را در اینجا رفع کنید و این را در ISR نمایم. درست را که باید باید دارن رفع کنید.

نظام سکو و قوی را نیز در این کیمی آن دستور بیزیست لزرفونه است ساریلین به این دستور آدرس

برنامه های تدوینی به همان نسبتی که از آن نسبت سهده بوده بازی تکرر.

نحوه بایریخ اینسان ISR . $I=0$ می باشد و به وقته که تبدیل شده سخن را در می سود بعد از آنها

۱۰) رشته‌ی ISR2 مورکه‌ی این ایکار رفع تورتولت می‌شود بعد از این‌جا ISR2 به

سے ۱۹۷۷

آشایی و سوئیت سرس رفته در ۶ میار

```
#define EXT_INT0 2      #define EXT_INT1 3      #define EXT_INT2 4
```

نرم افزار CodeVision AVR را برای نوشتن ISR ها می توانیم استفاده کنیم.

interrupt [نکاره سی برداردن فری] void تکریتین سرک رفته (void)
{
ISR سرکی.
}

سیکل سیکل آنکوو بے روچیل ۱۵۸ (۱۱)

```
#define ADC_INT 17
```

interrupt[ADC_INT] void ADC_Conversion(void);

مکالمہ

رئیسیہ نامہ

رنگه های خارج از ۷ بیتی های INT2 ، INT0 ، INT1 و میزون که عمل ATmega32 را نمی خواهد

(INT2/A1NO)	3 PB2	آن ها رسپلر ای نیز سینفون سنه است. رتفه کن خارجی نیست به سینفون
		۴ نوع
INT0	15 PD2	دریننگ موسیچی باشد نوع سینفون اعمال سنه میتواند کمپری با لاروند لبم
INT1	16 PD3	با سینفون روند آن و سطح صفر باشد که برسید و برنامه قابل تغییر است.

© MICRO®

نکته: رفته خارج در سکو میس باید بوره دهن گوشه میس بسطح باشد.

نکه نهم در راه اندازی رفته کی خارج

برای استفاده از مرکز از رفته های خارج باید استاد بکرین بست مرکز در رجیستر GICR آنرا

بله بست 7 6 5 4 3 2 1 0
GICR 0 INT1 INT0 INT2 - - - IVSEL IVCE فناز نزد

نکه 8 بکرین مرکز از بیوی 6، 5، 6، 7 رفته خارج بورز نظر را فعل خواهد کرد

فعی صفت رفته خارج صفر بسیار بست های ISC07، ISC00 از رجیستر MCUCR لین

7 6 5 9 3 2 1 0
MCUCR SE SM2 SM1 SM0 ISC11 ISC10 ISC01 ISC00 گردد.

انواع صفت رفته خارج صفر دی

ISC11 / ISC01 ISC10 / ISC00

0	0	0	INT0	سطح نفتی صفر درین	INT1
0	0	1	INT0	تغییر در سطح نفتی	INT1
1	1	0	INT0	لبه بین مرند در	INT1
1	1	1	INT0	لبه بالا مرند درین	INT1

نکه در صورتی که رفته میس باید تا زنگ سویی میس به قابل آنها

تیتر از نک سیل فلک راست رفته ایجاد نماید در صورتی رفته میس بسطح باشد باید باشد

ستراحت خارج سطح صورتی داشته باشد.

رفته خارج 2 شکل صورت میس باید راه اندازی می سوید در صورتی بست ISC2 از رجیستر

MCUCSR صفت بین مرند در صورت نک بیان لبه بالا مرند فرعی صفت رفته خارج
MICRO

لدر ۱ تعیین می‌کند حدایل زیر اعمال یا نیز به رتفه‌ی دو مارچی باید ۵۰ نانوگانتر باشد

7 6 5 4 3 2 1 0

MCUCSR jTD ISC2 - jTRF WDWF BWF .EXTRF PWF

C 0w

لیست نربرادر طبق فنالسازی رقص هادر رجیز GIER و نیکوں پیریم رقصه در رجیز GIER

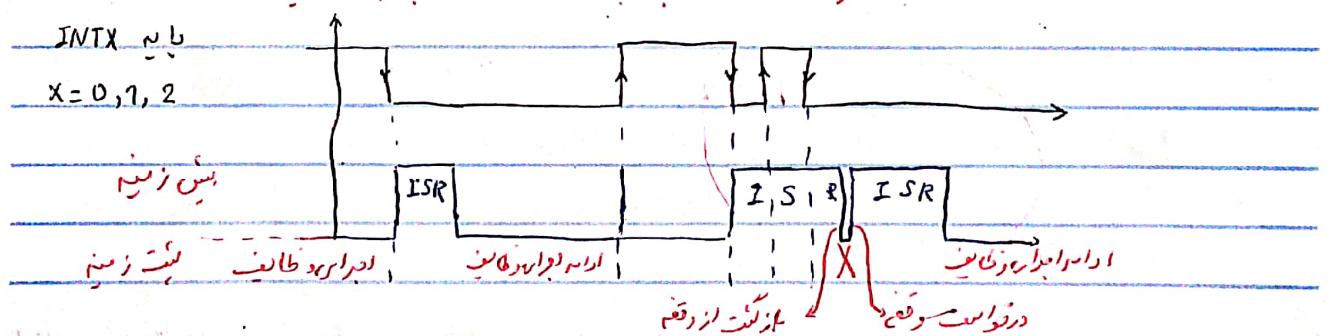
7 6 5 4 3 2 1 0

GIFR INTF1 INTF0 INTF2 - - - - -

اگر بیت نهاده از یک گروه رفعی (I) که باشد دیگر نهاده از گروه هم میشود در ICR نیز نباید

در این قدرت با ترتیب سیدن رفته بهم رقم مردم شنیده و رقم ایاری کلرود

نحوه علیر رفته و خارج به صورت اصلان به لد چیزی نزدیک



بِزَمْنِ اِنْتَهَىِ رَفْقَهُ خَارِجٌ لِعَبْرَتِ سَسَسَ بِلَبْرَىِ بِإِيْنِ لَوْنَدَهَ تَنْسِفَ لَسَدَهَ اِسَتَ بِاِعَالَ لَوْسَنَ لَبَرَىِ

پیش روئیده ISR اعتراف شده و مجدداً برنامه بسته بسیار سخت نهاده بازیگری کرده. این بیانات پایین روئیده

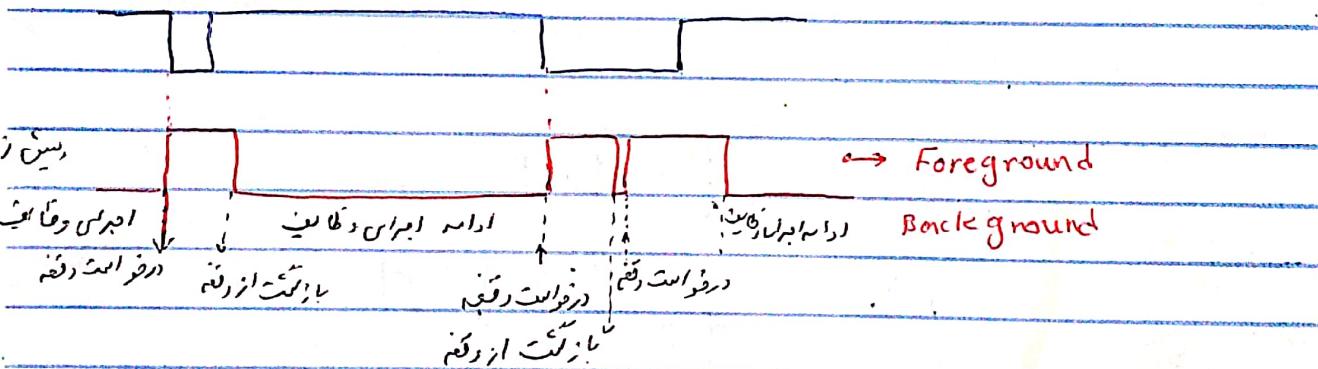
لعنی عبارت ISR این است که در این لحظه در میانه در ISR مسفل اینست که لبی میانه بین

رقمهای اعماق میگردید. از کنیای این که بیت I صفتیسته است ایکار راقمی تولید روش مدلسازی شده و در

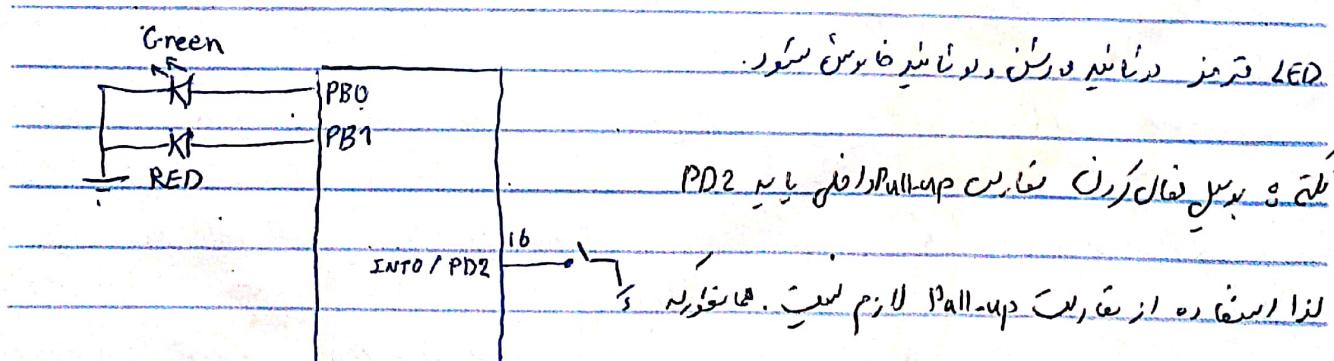
بعض وظائف خارجی تک در مادر باتخیر ISR بین I₂ سکه درینا به بسطح بین زینه

عزمی گردد. از آنها به که هنوز پرچم و قلمرو خارج یک است بینا نموده باشد ب ISR یعنی گرده ران را

رسانی زیر و تغیر خارجی تعبیرت می‌سازد به سطح تنظیم شده است با اعمال (رسانی سطح صفر و تغیر در خواست شده) ISR اجرا می‌شود این درست این است که برنامه در سطح هیئت زمینه در حال اجرا است در صورت که در مدت اعمال سطح هیئت برنامه دستگیری باشد باشد این سطح تا اجرای ISR ممکن خواهد بود در غیر این صورت رفع خارجی در خواست تحریک شد با اعمال سطح صفر بعد از برنامه مجدداً به ISR پیش کرده و در حلقه‌ی بازگشت از رفع سطح تغیر بین هموزن مفراست درستیم رفع خارجی در خواست سرمهد مجدداً ISR اجرا می‌شود.



نک) درست این LED 7 سینی یا زمین 7Hz رسانی و خلویه کرده مقام اعمال یا سینی روزه به INT0



هر رینگ آرینگ تعبیرت دروری پیکر بدن سرمهد باشد و بعد از خواست سرمهد دری بیت PORTxn و بیت متأثراً است Pull-up رسانی می‌لند مقادیری این متأثراً است افعال کرده و صفر آن را این افعال می‌لند سهولان

Jed

Subject :

Date :

```

#include <mega32.h>
#include <delay.h>

#define Green-LED PORTB.0
#define Green-LED-DDR DDRB.0
#define RED-LED PORTB.1
#define RED-LED-DDR DDRB.1
#define Tact-Switch PORTD.2
#define Tact-Switch-DDR DDRD.2

Void inti_ioc();
Void green-LED_toggle();
Void red-LED_on();

interrupt [EXT-INT0] void ext_isr(void)
{
    red-LED_on();                                RED-LED-DDR = 1;
    Tact-Switch-DDR = 0;                          Tact-Switch-Switch = 1;
}

Void main(void)
{
    // External Interrupt initialization
    inti_ioc();                                MCUCR = 0x40;  INT0ON  INTO 0;  MCUCSR = 0x00;
    #asm ("Sei")                                MCUCR = 0x02;  INTO Mode: Falling Edge
    while(1)                                    MCUCSR = 0x00;
    {
        green-LED_toggle();                      GIFR = 0x40;
        delay_ms(1000);                         }
        void green-LED_toggle()
        {
            Green-LED = ~Green-LED;
        }
}

Void inti_ioc()
{
    void red-LED_on()
    Green-LED-DDR = 1;
}

```

Subject :

Date :

```

RED_LED = 1;
delay_ms(2000);
RED_LED = 0;
delay_ms(2000);
}

```

زیل نیج / پا رنده

تذکر زیان نیج را رنده داریم ایستاده زیل نیج کار نمایم که تبدیل نیسانات که کار بازخان می‌باشد

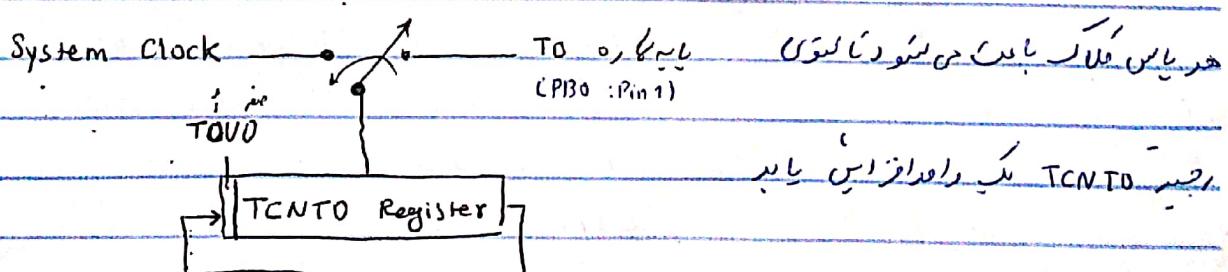
رنجف را می‌کاریم که رنده دو دیدگاه خارج را که کار دارد

ستمه زیل نیج / پا رنده ۸ بیت (0,2) ۲ زیل نیج / پا رنده ۸ بیت (1,1) ATmega32

از ۱۰

timer0 / Counter0

در شکل زیر نمایی می‌کنیم از داده زیل نیج صفر میگیریم و نتیجه AVR دیده می‌شود



آنچه میگذرد زیل نیج را که کار نمایم که کار رجیستر جدید بگیریم

تک راه رفاقت زیل نیج را که کار نمایم که کار رجیستر جدید بگیریم

TCCR0 : FOC0 WGM00 COM01 COM00 WGM01 CS02 CS01 CS00

Timer Counter Control Register 0

MICRO

تغییر بروز برای clock select . TCCR0:2

CS2	CS1	CS0	clock Source
0	0	0	بردن تاکر متوقف
0	0	1	شماریت بروز تیم
0	1	0	8/شماریت
0	1	1	1/64
1	0	0	1/256
1	0	1	1/1024
1	1	0	لبه ۳۲ بین ونده
1	1	1	لبه ۱۶ بین ونده

برایم مردیز زدن سنج/سما زند

تک TO50 بیت صفر از بیت TIFR ایست نه بیان رسانی تکرار TCCR0 بیت صادر کنی و غیره ای این

آن TO50 بیت TO50 که می سوزد بلطفت شمارکندر TO50 را بالا ببریم توکنی رسانی رفته

برایم زدن بیج صفر هست این را بایک کرد. اتفاقاً نه بیت صفر دارای این ایست که رعنی ریتکن هست همان تک TO50 بیج هاست

زدن بیج دیگر لغایت AVR شکر از بیج هست راه دهار این با اینکه بیت صفر هست باشد از آنجایی که زدن بیج

بیکانه یاری رفته باشیم راه اباجم رفه لذا درای ۴ رفیت کاری مختلفی نیست

Fast PWM ۱/۳ clear Timer on Compare Match: CTC - ۲ Normal - ۱

Phase Correct PWM - ۴

آن بیکانه زدن چهار رفیت کاری بارگذاری عدد نسبت در بین های WGMO[۱:۰]

اعمال بیکانه زدن - این بیکانه زدن TCCR0 Waveform Generation Mode

Subject :

Waveform Generation Mode

Date : clock selection

TCCR0 FOC0 WGM00 COM01 COM00 WGM01 CS02 CS01 CS00

WGM01~~000~~ WGM00 Timer0 Mode

0

0

Normal

0

1

PWM, Phase Correct

1

0

Clear Timer on Compare Match (CTC)

1

1

Fast PWM

TIFR OCF2 TOV2 ICF1 OCF1A OCF1B TOV1 OCF0 TOV0

دقتیت Normal

۱- در بزرگترین ایندکس سریلتر ATmega32 دقتیت Normal همچو زیر مجموعه ای اندک ای براحتی ترین

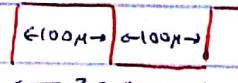
موضعی این رفتار را در دو حالت همان بروی کرد ۱) زمان سنج ایجاد نهاده عکس زدن سنج دکاری

در این حالت اندازه گیری ایجاد زدن و تکراری میس خارج و ۲) زمان سنج ریتم ریتم زدن باعث نهاده زدن

حالات ایجاد سفل سنج لغایت سنت افزایش

۲- از این ایجاد سنج ممکن است سریلتر ایجاد هر تری ایست یا سریع ایجاد هر تری ۵ کیلو هر تری بروی باید

$F = 5\text{kHz} \Rightarrow T = \frac{1}{5\text{kHz}} = 200\mu\text{s}$ تکراری PDI



در فرودیتی ممکن است نیز تریم نیز نهاده بر عذر و تغییر سریل برای ایجاد هر تری ۱ تری ایجاد هر تری به TCNT0

اعمال نموده شد در تغییر ناصلی هر ممکن برابر باشد سریل را نهاده است براحتی ایجاد سفل سنج حلقه برابر ۱۰۰

میکرو ثانی نیاز نداشت بین باید متادارلولی TCNT0 برابر ۱۵۶ باشد.

Subject :

Date :

```
#include <mega32.h>
```

```
Void delay_100ms();
```

```
Void init_io();
```

```
Void main(y,gid)
```

```
{
```

```
    init_io();
```

```
    while (1)
```

```
{
```

```
    PORTD.0 = 1;
```

```
    delay_100ms();
```

```
    PORTD.0 = 0;
```

```
    delay_100ms();
```

```
}
```

```
}
```

```
Void init_io()
```

```
{
```

```
    PORTD=0x00;
```

```
    DDRD=0x01;
```

```
}
```

```
Void delay_100ms()
```

```
{
```

```
    TCNT0 = 156;
```

تیکر در درایور تیکر که قسم کار سیم بیم، زن و مانع می شود.

بیت مقنن از رجیستر TIFR بزم TOVO است بنابراین TOVO

مالد تیکر این بزم خاصه بور دراین خطاب را می بینیم TOVO

که نیزه ایت سه طبقه while صنعت بور در درایور نهاده می شود.

که نیزه ایت سه طبقه while صنعت بور در درایور نهاده می شود.

```
// TIFR = TIFR | 0x01;
```

Subject:

Date:

```
Void delay_10ms()
{
    Unsigned char i=0;
    while (i<5)
    {
        TCNT0=6;           // Set for 250 counts
        TCCR0=0x03;        // Prescaler=64
        while ((TIFR&0x01)==0); // Wait for flag
        TCCR0=0;           // Stop the clock
        TIFR |=0x01;        // clear the flag
        i++;               // Increment loop index
    }
}
```

نیکی دقتہ سریز زمان بینج ایکسیوں

زمان بینی جو توان پیش از سریز کردن ترکیب کریں

تھوڑا بڑا اینٹا، باید بیت، جیسے، تیکنڈو ایکسیوں

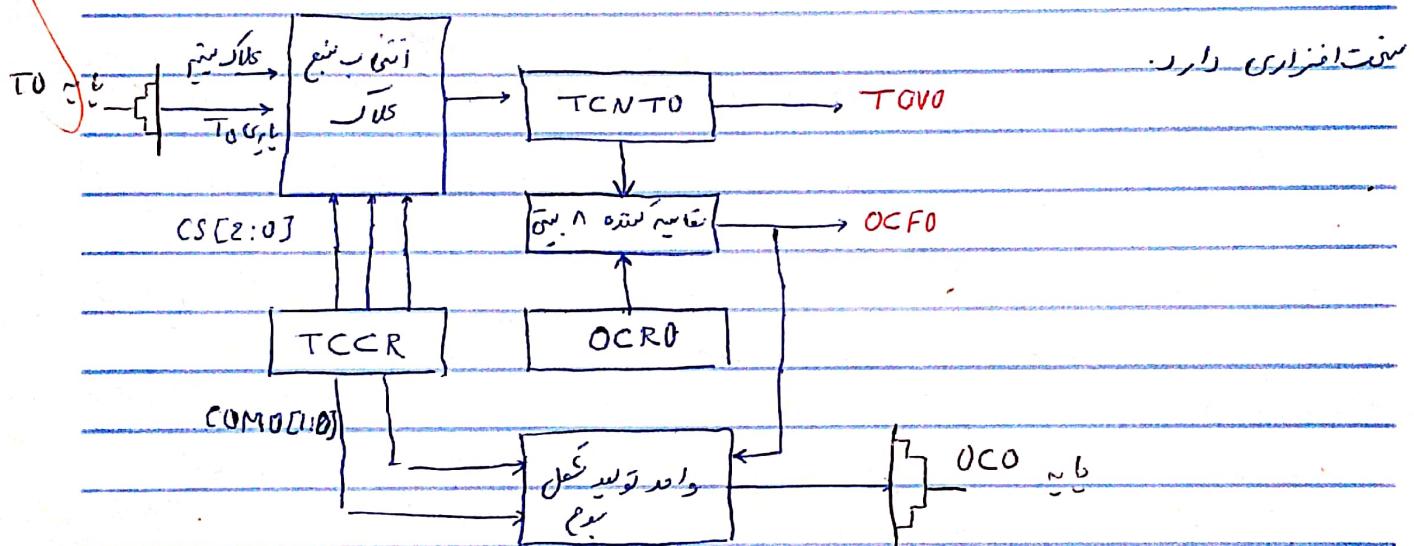
```
#include <mega32.h>           // اسکرپٹ کی دلیل
Void init_i0();
Interrupt [TIM0_OVF] Void timer0_ovf_isr(void)
{
    TCNT0=156;           // Set for 100 counts
    PORTD.0=~PORTD.0;    // PORTD=0x00;
}
Void main(void)
{
    init_i0();
    while (1)
        MICRO{}           // Enable global interrupt
};
```

برنامه صن ۲۴۶ آنچه نیز ۰-۹۹ بر روی $CC0$ باعث شد ۷۵ باینده.

آنچه میگذرد

آنچه میگذرد ۱) عکس زدن لمحه رساندن صفر را بین در این بین فرودیم تا ب یوری خالت ۲ پیروزیم

آنچه میگذرد که از نام این خالت مستقیم است، زدن لمحه رساندن تقابلیت مقایسه ای مقادیر $TCNT0$ را بین



آنچه میگذرد که از نام این خالت مستقیم است، زدن لمحه رساندن تقابلیت مقایسه ای مقادیر $TCNT0$ را بین

آنچه میگذرد که از نام این خالت مستقیم است، جیز $TCNT0$ در مدخل سایه سده و در صورت برایری

در مدخل لبیس $OCFO$ تا میگذرد احتمال برابری هایند که سرریز سریار یا هم بوره و آن را تطبیق نماییم

کوینز Compare Match

کوینز Compare Match

آنچه میگذرد که از نام این خالت مستقیم است، در مدخل سایه سده $OCFO$ در این برایی را اند ترکیب شکل برم و لکور چیز کردن

آنچه میگذرد با توجه به ترتیب این که از بینی $Compare match output mode$ = COMO[1:0]

Subject :

Date :

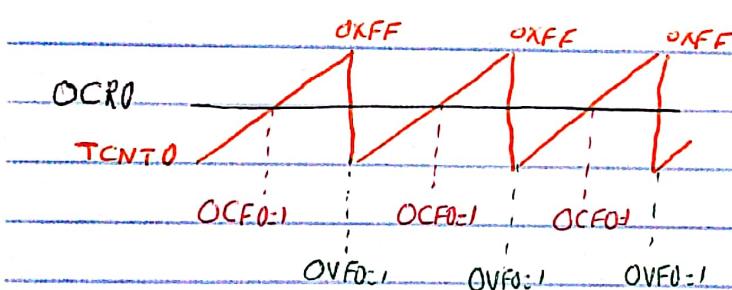
از رجیستر TCCR0 بوده درینت می‌کند سطح بسته باشد OCD

COM01	COM00	OCD	دقتیت می‌شوند
0	0	(I/O)	خیزش
0	1		در تبلیغ تا بر
1	0		در تبلیغ تا بر
1	1		در تبلیغ تا بر

نه و از راه دستیمی خودم می‌توانم به متغیر تولید شغل بروم اسفاره خود را درست CTC بسی

تولید شغل بروم های بازه من درگاه من سبب تر بوده و از رجیت Normal باز می‌ستم اسفاره عنوان شود.

۱۲) ایجاد رتفه می‌از درگاه سُنن



درینت ایجاد می‌کنیم OCRO، TCNT0، OCF0 (لینینت می‌ایم) می‌بینیم OCF0 که شده و می‌تواند رتفه در فرایت

سیگنال بین ترتیب با متادرهن نسبت به رجیستر OCRO می‌توان تأثیری رکواده نمود

از این ایجاد می‌توان در حالتی از این متادرهن دارن به TCNT0. این طریق کریم هر یک متادرهن

روی رارن ب CPU وقت TCNT0 را تلفی کردید این در فرایت رتفه در نکته تعلیق می‌ایم

لکی بودن بین TIMSK Output Compare Interrupt Enable از رجیستر OCIE0

لکی بودن بین آن ایت و همین میان بین دیگر میان این ایت کردن

Subject :

Date :

برای این پروژه دو اینترکت برای این سریزی رسم کنیم یعنی 1. در هنگام پرینت OCFO و 2. در هنگام پرینت OVF0.

```
#include <mega32.h>
```

```
Void init-io();
```

```
interrupt [TIM0_OVF] void timer0-ovf-isr(void)
```

```
{
```

1. طبقه 1

```
}
```

```
interrupt [TIM0_CMP] void timer0-comp-isr(void)
```

```
{
```

2. طبقه 2

```
}
```

```
Void main(void)
```

```
{
```

```
    init-io();
```

```
    while (1)
```

```
{
```

```
};
```

```
}
```

```
Void init-io()
```

```
{
```

TCNT0 = 1000000

OCRD = 1000000

TCCR0 = 0x02; // Prescale = 8 and

TIMSK = 0x03; // Enable OCFO, OVF0 interrupts

#asm ("sei"); // Enable global interrupt

```
}
```

Subject :

Date :

ردیفتی clear timer on compare match: CTC

اصلی ترین کاربرد CTC ابی ر تائینر در مکانیسم تیکریت سنت افزاری است. در این طالع باید

انواعی برای در ردیفتی CTC تیکریتی کاربردی داشت. در ردیفتی $WGMO[1,0] = 10$

کوکیتی از 256 تغییر داد در این طالع مقدار TOP در جای替ی $OCRO$ باشد که در بخلاف

ردیفتی Normal سیاره نه از تطبیق مقایسه از عدد $OCRO$ عبور نکرده باشد در حالت مقدار

سیاره نه از تطبیق سیاره میگیرد. در این رسانیدن سیاره بیندار، TOP، ابریم $OCFO$ تغییب نمود

مکانیزمی است I_1 باشد

نیز سیاره در ردیفتی CTC: شکل کم نماید تا هر 100 میلی ثانیه، مخفی از بین میگیرد

```
#include <mega32.h>
```

```
Void init_io();
```

```
interrupt [TIMO_COMP] void timer0_comp_isr(void)
```

```
{
```

 طبیعی

```
}
```

```
Void main(void)
```

```
{
```

```
    init_io();
```

$TCCR0 = 0x0A$ // Prescaler=1 and Mode: CTC

$OCRO = 99$; // Set for 100 count

$TIMSK = 0x02$; // Enable OCFO interrupt

EAasm("sei"); // Enable global interrupt

```
}
```

```
}
```

```
Void init_io()
```

```
{
```

 MICRO®

تولید شلچ بیج در دقتیت CTC

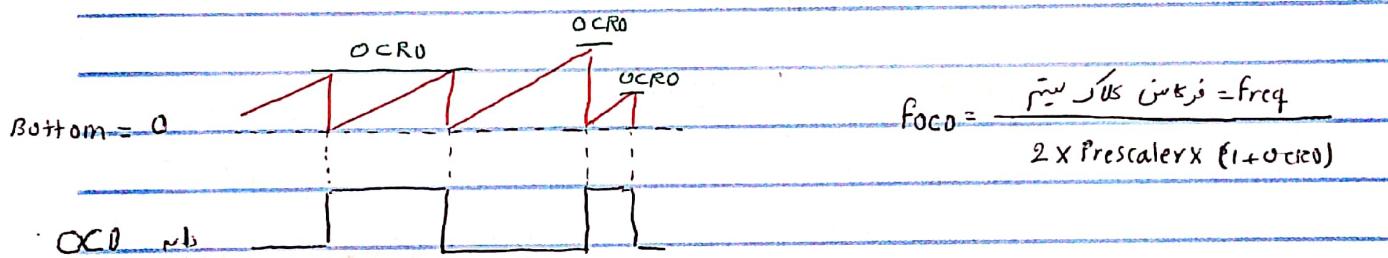
متداول از طریق تایر میکر بیم مرتبی در مدار از یاری های سری دایگار کرده ای این مار بایت (روتیر سد) CPU نیز

دھالیکی در میان تو سیستم بیج در دقتیت CTC نیز می دستیت لازم برای این تولید بیم نیز در مدار دو یونه

ریلیز آن می سکر راسمال این خواهد بود که شکن برای OCO با یکی از سه این ۹ در مقیمه ATmega32 گزوان

شلچ بیج ایجا ریکو را.

الگوریتم در دقتیت باشندگه در کمایی تطبیق مقایم باشی OCO COM0[1:0] میگزد



#include <mega32.h> بین مدارهای مذکور باید نیز OCO فرکانس ۵KHz بین ۱ یار کرده.

void init_io();

void main()

{ ایجا شلچ نمی شود از این ایجا شلچ نمی شود ایجا شلچ نمی شود

init_io();

while (1) میکارهی تولید در

}

}; خروجی کرن باشی OCO ایجا شلچ نمی شود

}

void init_io() درینه ایجورت شلچ بیج ایجا شلچ نمی شود بروی این نیز نیکو خواهد بود.

{

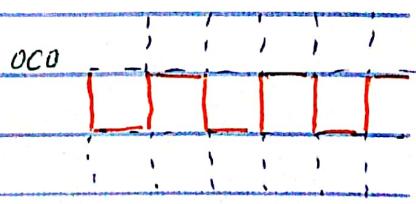
DDRB=0X08; //output OCO

TCCR0=0X1A; //Prescaler=8 and Mode: CTC

OCRO=99; //Toggle on compare match

MICRO®

Set for 100 count



آنالوگ باعث فرم Glitch در آن هنگام تغییر زمان

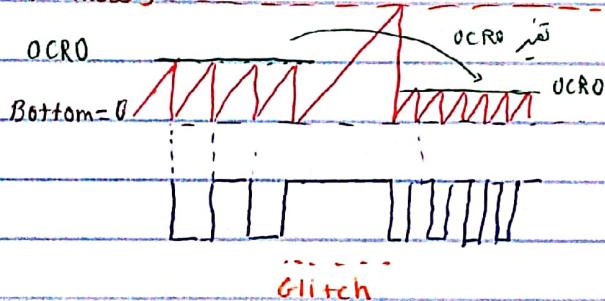
بایک تغییر زمان باید متدار OCRO تغییر یابد علیه است با تغییر OCRO باید مایه از زمان درین دوره را بخواهیم

OCRO را بخواهیم که معملاً درین لحظه کمترین تغییر زمان باید تردد بخوبی است بدلاً از متدار OCRO

در تفاصیل تبعیق مایه بخواهیم شارند برای اینکه متدار TCNTD در این حالت در کمترین

متدار خود را دارد. بجز این ممکن است این مسئله نیست.

MAX=255



آنالوگ باعث فرم مودولاسیون عرضی را می‌شود

کامپرهاز PWM و آنچنانکه اطلاعات در معرف فرستاده درین مورد تغییر عرضی می‌شود

که درین ترتیب در معرف تغییر نیز به ADC خواهد بود. آنچنانکه اطلاعات با نویز خود را خواهد

بود. ~~و اگر رنگ تدریجی نباشد~~ ~~و این بعده تغییر می‌شود~~ کامپرهاز می‌شوند

زمان بین صفر و تواند در درجه فسیمه Phase Correct PWM فرست PWM تولید می‌گاید.

در زول نسخه / سیستم صفر Fast PWM

بهای نسخه عورن زدن نسخه در این حالت لازم است تا پیست $WGMD[1:0]$ برابر ۱۱ باشد و عکس

پیکری COMD[1:0] در رعایت Fast PWM به قرار زیر است.

COMD1	COMD0	OC0	ردیفیت بین
0	0	غيرفعال (0) (معلومن)	
0	1	Fast PWM	برای زدن عورن : پایه زدن OC0 در ترتیب متعاقب با پایه زدن آن در TOP
1	0	Fast PWM	برای زدن آن در ترتیب متعاقب با پایه زدن OC0 در ترتیب متعاقب با پایه زدن آن در TOP
1	1		

در ردیفیت PWM غیرمعلومن شارند از MAX و BOTTOM ب صورت معمولی هستند و در رعایت ترتیب

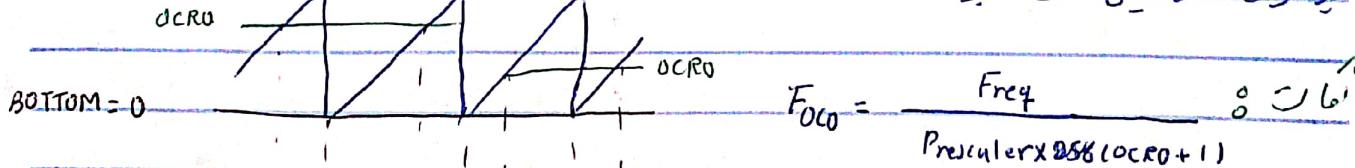
متاسیز ، پایین OC0 پاک شده را براید یا BOTTOM عبارت از سرور و در ردیفیت PWM معلومن

نیز شارند از MAX و BOTTOM هستند و در رعایت ترتیب متعاقب با پایین OC0 پاک شده را براید

معلومن نواهد بود . علت نا متعارفی PWM معلومن رغیر معلومن این است که در ردیفیت غیرمعلومن

بالاترین مقدار OCR0 بخنای پایس (زایش) یا تنه در ترتیب زدن رطیغه بالا میگردد و در ردیفیت PWM

غیرمعلومن مقدار OCR0 بخنای پایس (زایش) یا تنه در ترتیب زدن رطیغه بالا میگردد .



دستوری سفار OCR0 برابر با صفر سرور باشید .

OC0 = 0 سیستم صفر سرور باشید .

نیز شارنده پایس نوزن در خود را تولید نماید .

تازه ای مقدار OCR0 = 255 سرور بسته به ردیفیت [OC0[1:0] بین ۰ و ۱] مداره سفر را کنید .

مقدار 75% من تردد PWM با تردد 256 بـ، تردد 256 سرعت پین درون (طبقه 1)

برای این ایجاد کار لیم 8 تا هر 8 تا (OC0)

$$256 \times 0.75 = 192 \Rightarrow OCRO = 191$$

#include <mega32.h>

Void init-io();

Void main();

{

init-io();

while (1)

{

}

}

Void init-io()

{

DDRB = 0X08; // Output OC0

PORTB = 0X00; // OC0 = 0

TCCR0 = 0X6A; // Mode: non_inverting Fast PWM

OCRO = 191; // Duty Cycle: 75%

}

Phase Correct PWM درست

آندا بايد راهيت بـ OC0 و GM0[1:0] در وضعيت COM0[1:0] در وضعيت

COM01 COM00 OC0 درست

برونتيل به قرار عددي زير است . شير فعال (I/O) شد

برونتيل به زير است

برونتيل به زير است

برونتيل به زير است

MICRO®

برونتيل به زير است

آن راهي زير است : مقدار PWM

Subject :

Date :

در رفعتی PWM سرده و بریدن به MAX و BOTTOM از سرده و بریدن به

OCRO به صورت نزولی تا سرده و بریدن به MAX به صورت نزولی تا سرده و بریدن به

برای بروز سرور درین حالت درخواست بروز فرجه نصف رفعتی PWM باشد. اما مسئله بعده

فرجه نصف رفعتی درین سرور از زیر (Fast PWM) است در رفعتی PWM

برخلاف سرور درین سرور زدن بروز فرجه نصف رفعتی PWM باشد. اما سرور درین سرده و بریدن به MAX به سرده و بریدن به

کلی سرور در رفعتی PWM بازگشایی سرور در رفعتی PWM عنی نکویی

فرجه OC0 بیو سمت صفر بوره در صورتی MAX بازگشایی سرور فرجه آن بیو تبدیل

خواهد بود در بعد PWM نکویی دلخیختی خواهد بود

برای بروز PWM نکویی با قوانین تقدیم ۲ کیلوهرتز وزن رطیغی ۵۰۰

درین سرور بازگشایی OC0 ایجاد شد ۸ هر ثانیه

#include <mega32.h>

Void init_io();

DDRB = 0x08;

Void main(void)

PORTE = 0x00;

{

TCCR0 = 0x62; // Prescaler = 8 ; MODE1 Phase Correct PWM

init_io();

OCRO = 186; // Duty cycle = 75%.

while (1)

{

$T = 500\text{ms} \Rightarrow 75 \times 500\text{ms} = 375\text{ms} \Rightarrow \frac{375}{2} = 187.5 \Rightarrow OCRO = 186$

};

Phase correct PWM, Fast PWM در رفعتی های OCRO ۰٪ و ۱۰۰٪ رجیم

}

Void init_io()

OCRO = 0;

{

ان معتبر در بازگشایی سرده و بریدن سرده و بریدن به TOP معتبر بازگشایی

سئله نیست سرور درین ترتیب سئله فرجه ناسفارن و glitch رفعتی PWM

Subject :

Date :

FOCO: Force Output Compare

سیستم تغییر ابعادی فریم زمین لامپ مغناطیسی

تایزون یا آنک سیستم های رجیستر FOCO آنرا نیز علیرغم این بیت های فتووک است

آن در دقتی های خیر FOCO و اور توکنیک PNM ، CTC ، Normal می باشد

با این فریم (براسن تغییرات بیت های FOCO و OC0) چنین عبارت دیگر نمی شود

این بیت توسعه برنامه: عبارت باز این است که تغییر مقایسه ای در میان دو سیستم باید

رقمهای (یک رند) دو دقتی CTC باعث مغرسانه می شود TCNT0 کو احمد شد

تئک تائید آن تغییر دقتی می باشد مطابق با میانگین بیکاری COMO[1:0] است

LCD

ما در نوع کارالتری و گرافیکی بازار عرضه شده‌اند که نوع کارالتری آن به دو صورت سریع و سازی می‌باشد. نظریه LCD ماتریسی تراشه Hitachi HD44780 باشد. معمول در ساخت LCD صورت لایه سطحی قرار گرفته و با تراشه سارگار آن به سه بندی ارزان قیمت صورت COB (chip on Board) بوده است.

نامه: سه نماینده LCD ماتریسی می‌باشد: ۰.۶۴۶، ۰.۱۴۰، ۰.۱۰۰ و ۰.۰۷۰ متر را باشد.

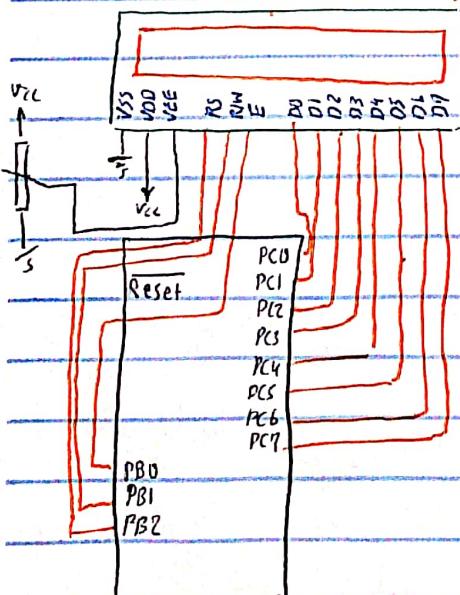
نظریه HD44780 دارای مادنیتی های زیر است

AC، بیت داده DR، بیت دستورالعمل IR، میکرو پریمیت BF، نماینده آدرس

character Generator ROM، CGROM، RAM، رمیکس کارالتری

CGRAM، RAM، RAM، RAM

نامه: دستورالعمل های تابع ابتداء از نظریه HD44780 در اینجا آورده شدند



امول ارتباطی درین صکور نظریه با نظریه LCD

اقریبی نظریه نظریه از این افراد می‌شود

```
#define LCD_DATA_PORT PORTC
```

```
#define LCD_DATA_DDR DDRC
```

```
#define LCD_DATA_PIN PINC
```

```
#define LCD_CONTROL_PORT PORTB
```

```
#define LCD_CONTROL_DDR DDRB
```

```
#define RS 2
```

```
#define RW 1
```

```
#define ENABLE 0
```

Subject :

Date :

۱۳۹۷

Set RS (۱) : طبقہ این تک رول بے RS کے رول دلٹسی بے رجیٹر دیسوار جعل ایس

Void setRS(void)

{
LCD_CONTROL_PORT |= (1 << RS); // LCD_CONTROL_PORT = LCD_CONTROL_PORT | 0b00000100;
delay_us(1);

پہلے جیسے RS کے سلسلہ میں دلٹسی، دلٹر نکل رہا تھا اور

لیکن اسی پڑیکے سیر کا نیا ایسا دلٹر تھا جو کہ RS کے سلسلہ میں دلٹر کر رہا

جعل ایس، طبقہ این تک صدر رول بے RS بے دلٹسی بے رجیٹر دادہ ایس

Void clearRS(void)

{
LCD_CONTROL_PORT &= ~ (1 << RS); //
delay_us(1);
}

* LCD_CONTROL_PORT = LCD_CONTROL_PORT & 0b1111011 RS کو دیں جسے

clearE (۱)، setE (۱)، clearRW (۰)، setRW (۱)

Void SetRW(void)

{
LCD_CONTROL_PORT |= (1 << RW);
delay_us(1);
}

Void clearRW(void)

{
LCD_CONTROL_PORT &= ~ (1 << RW);
delay_us(1);
}

Void SetE(void)

{
LCD_CONTROL_PORT |= (1 << ENABLE);
delay_us(1);
}

Void clearE(void)

{
LCD_CONTROL_PORT &= ~ (1 << ENABLE);
delay_us(1);
}

• MICRO •

Subject :

Date :

LCD از این تا اسفاره باید رکورده باشد و LCDCommand

Void LCDCommand (& unsigned char command)

{

LCD_DATA_DDR = 0xFF;

clear RSC();

clear RW();

Set E();

delay_ns(1);

LCD_DATA_PORT = command;

clear E();

delay_ns(40);

}

: LCD باید ورود

Void LcdClear(void) : LCD را کاملاً پاک کردن (از این تا اسفاره باید رکورده باشد) LCDClear

{

LcdCommand(0b00000001);

delay_ms(2);

}

: LCD را کاملاً پاک کردن (از این تا اسفاره باید رکورده باشد) LCDHome

Void LcdHome(void)

{

LcdCommand(0b00000010);

delay_ms(2);

}

: LCD را کاملاً پاک کردن (از این تا اسفاره باید رکورده باشد) LCDOn

{

LcdCommand(0b00001100); : LCD را روشن کردن (از این تا اسفاره باید رکورده باشد) LCDOff

Subject :

Date :

```
Void lcdOff(void)
{
    LCDCommand(0b00001000);
}

Void lcdGotoxy(unsigned char x, unsigned char y)
{
    if ((x < 40) & & (y < 2))
    {
        LCDCommand(0b00001000);
        LCDData(0x00);
    }
}
```

برای راه اندازی رنگیتر از پیش LCD از این دستور استفاده می شود :

```
Void lcd_ready();           از این LCD برای دریافت اطلاعات
Void lcd_write_data(unsigned char data);
```

از این تابع برای مقدار دهندن درین بیت دوون LCD استفاده می شود این بیت برای نزد

تغییات LCD بیار چند برای ساده تغییات می شوند به جمله برویم برای هم غیر
دیگر

```
-lcd_ready();           // COM1 ریزیت LCD
-lcd_write_data(0x0E); // نماین LCD، نماین یا درین
```

عدد 0x0E کنیم لیکن کار بعلوون میگردد

0x05 میگاید لیکن نماین برایت 0x01 میگردد

• MICRO°

اسفار از این ایجاد LCD باینی برای راه اندازی

ب سایر Code Wizard AVR با مادربرد کنیجانی LCD و قیمت ابزار CodeVision AVR این ایجاد را بسیار راحت فایل را با LCD کارتری راه رعد بسیار از این را می توانیم

ترتیب این ایجاد باین LCD باید همان ترتیب نصیرت نصیرت زیر باشد.

Bit0: RS

بنابراین در این ایجاد از LCD نصیرت زیر خواهد بود.

Bit1: RD

Bit2: EN

#include <mega32.h>

Bit3: Free

#asm

Bit4: DB4

.equ a-LCD-port = 0x1B ; PortA A گیرنده 0x1B

Bit5: DB5

#endasm

Bit6: DB6

#include <lcd.h>

Bit7: DB7

void main(void) LCD ایجاد از و نه

{

lcd_init(1b); ۱۶ نتاریت LCD را ایجاد کند

while (1)

{

};

}

high level LCD Function

lcd_init(m); LCD شماره میتوان

lcd_gotoxy(n,m); ایجاد نمایش نمایش را ب ترتیب n, m راه کند و نمایش را ب

lcd_clear(); پر کردن صفحه کاری و ایجاد نمایش خالی ب می بینیم (0,0)

lcd_putchar(char c); نمایش کارکتر c را ب می بینیم فن

lcd_puts(char *str); نمایش را کند str را در حافظه RAM قرار دارد

lcd_putsfl(char flush *str); ~ ROM ~ ~ ~ ~ ~

lcd_putsel(char eeprom *str); ~ EEPROM ~ ~ ~ ~ ~

Subject :

Date :

LCD

نامه مدار سقیر بر در

د دفعه بای خانه مکار مقدار متفاوت دارد

۱) توابع استاندارد تبدیل دره برئه

اعن جرایع استاندارد تبدیل دره برئه در قالب stdlib.h دارد

void itoa (int n, char *str) بوسیله این تابع مقدار متفاوت از نوع int را به رشته تبدیل می کند

void ltoa (long int n, char *str) بوسیله این تابع مقدار متفاوت از نوع long int را به رشته تبدیل می کند

void ftoa (float n, unsigned char decimals, char *str)

بوسیله این تابع مقدار عدد اعشاری n را به رشته تبدیل می کند اگرچه لام این تابع مقدار

هم اعشاری را نمی نماید

کارکرد ftoa() و ftoa() می باشد که در اعمال اعشاری

1.2345e2 ≈ 123.45

نماش را در صور می نماید

int a=1200;

float b=123.456;

unsigned char str1[10];

ftoa(b,5,str1);

lcd_init(16);

lcd_puts(str1);

while (1)

{

lcd_gotoxy(0,0);

itoa(a,str1);

lcd_puts(str1);

lcd_gotoxy(0,1);

1200

123.45600

• MICRO®

برنامه زیر نمایه سازی string print formatted بعنوان sprintf

از فرمت رون در میان قابل خاص ب خروجی از مکانی خود نه در کد برنامه تیکر دنیز کرد (۱۰/۱۰)

sprintf

ارسال به فریم ورود

جواب

sprintf(str, "Temp is %d and fan is %P", 27, "off");

lcd_puts(str);

جواب: Temp is 27 and fan is off

sprintf(str, "Temp is %d C", temp);

lcd_puts(str);

Temp is 27 C

sprintf(str, "Temp is %X C", temp);

lcd_puts(str); Temp is 19 C

sprintf("Hour: %d, Minute: %d, Temp: %f, Fan: %P", 15, 23, 27.3, "off");
lcd_puts(str);

Hour: 15, Minute: 23, Temp: 27.300000, Fan: off

کد های فرمت بسته ای از سده بوسیله یا مم

عمر صدیح رسائل علالت دار
%i
%d

بردن حالت
%u

عده اعماق ب فرم خارعه
%o

عده اعماق
%E

عده هشتم اعماق
%f

عده هشتار رسائل بردن مللت با ورن روی
%x

عده هشتار رسائل بردن مللت با ورن روی
%c

رسانه ای از کارل کتر سار حافظ
%s

MICRO%P

Flash ~ ~ ~

ناین کارلتر %

Subject :

Date :

stop و start بچهار بیتی هستند

```
#include <mega32.h>
```

```
#include <delay.h>
```

```
#include <stdio.h>
```

```
#asm
```

```
:equ _lcd_port = 0x713 ; //PORTA
```

```
#endasm
```

```
#include <lcd.h>
```

```
bit start = 0
```

```
Void show_lcd(unsigned int);
```

```
Void main(Void)
```

```
{
```

```
unsigned int counter = 0;
```

```
PORTD = 0x00;
```

```
DDRD = 0x00;
```

```
lcd_init(16);
```

```
lcd_putsf("counter : 0");
```

```
lcd_gotoxy(0, 1);
```

```
lcd_putsf("top count 10000");
```

```
while (1)
```

```
{
```

```
if (PIND.0 == 1)
```

```
{
```

```
start = 1;
```

```
}
```

```
if (PIND.1 == 1)
```

```
{
```

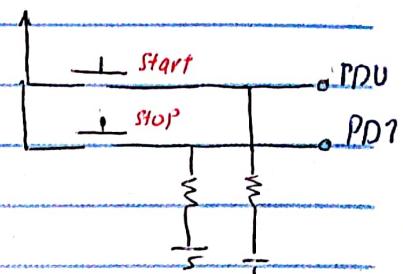
```
start = 0;
```

```
}
```

```
if (start == 1)
```

```
{
```

MICRO



Subject :

Date :

SFIOR ADTS2 ADTS1 ADTS0 - ACME PUD PSR2 PSR10 دستگاه کردن

ADTS2	ADTS1	ADTS0	سینه راه ایجاد
0	0	0	(ADIF) ADC بینی
0	0	1	(ACI) بینی مقایسه کننده آنلور
0	1	0	INTFO بینی رفته خارج کردن
0	1	1	OCFO بینی تطبیق نمایشی زدن سنج مفتر
1	0	0	TOV0 بینی سریز زدن سنج صفر
1	0	1	OCFB بینی تطبیق نمایشی زدن سنج مفتر
1	1	0	TOV1 بینی سریز زدن سنج تیز
1	1	1	ICF1 بینی درست شدن سنج مفتر

نحوه معرفی مسح از رجیستر ADCSRA

ADCSRA ADEN ADSC ADATE ADIF ADZE ADPS2 ADPS1 ADPS0

Free running mode

باید نیز بینی ADSC فعال بوده باید رنچ بینی ADIF بینی ADZE بینی ADPS0 بینی ADPS1 بینی ADPS2

سینه راه ایجاد موارد معرفی شده ای ADC

نحوه معرفی مسح از رجیستر ADCSRA

```

unsigned int read_adc(unsigned char adc_input, unsigned char vref_type)
{
    ADMUX = adc_input | vref_type;
    delay_us(10); // delay needed for the stabilization of the ADC input voltage
    ADCSRA |= 0x40; // Start the AD conversion
    while ((ADCSRA & 0x10) == 0); // wait for the AD conversion to complete
    ADCSRA |= 0x10; // clear ADIF
    return ADCW; // ADCW is 16 bit
}

```

MICRO

Subject:

Date:

SFICR ADTS2 ADTS1 ADTS0 - ACME PUD PSR2 PSR0

ADTS2 ADTS1 ADTS0

مخرج رقمي

0	0	0	(ADIF) ADC	مخرج رقمي
0	0	1	(ACI)	مخرج رقمي
0	1	0	INTFO	مخرج رقمي
0	1	1	OCFO	مخرج رقمي
1	0	0	TOV0	مخرج رقمي
1	0	1	OCFOB	مخرج رقمي
1	1	0	TOV01	مخرج رقمي
1	1	1	ICFI	مخرج رقمي

in ADCSRA

ADCSRA ADEN ADSC ADATE ADIF ADRE ADPS2 ADPS1 ADPS0

Free running mode

بواسطة ADIF يتحقق مخرج رقمي ، وبواسطة ADSC يتحقق رقمي

مخرج رقمي

ADCW <= 16 bit

unsigned int read_adc(unsigned char adc_input, unsigned char vref_type)

{

ADMUX = adc_input | vref_type;

delay_us(10); // Delay needed for the stabilization of the ADC input voltage

ADCSRA |= 0x40; // Start the AD conversion

while ((ADCSRA & 0x10) == 0); // wait for the AD conversion to complete

ADCSRA |= 0x10; // clear ADIF

return ADCW;

مخرج رقمي 16 bit من ADC

MICRO }

Subject :

Date :

Page:

مقدمة ملخص مفهوم ISR في ADC

interrupt [ADC_INT] void adc_isr(void)

{
 // Place your code here;
}

SR0 OR جلست

```
#include <mega8.h>
#include <delay.h>
#include <stdio.h> //getchar, putchar, puts, putsf, printf,
#include <stdlib.h> //atoi, atol, ltoa, ftoi, stof, rand,
asm
```

```
    .equ -lcd_port=0x18;
```

```
zendasm
```

```
#include <lcd.h>
```

```
#define ADC_VREF_TYPE 0x40;
```

```
unsigned int read_adc(unsigned char *adcl_input)
```

```
{
```

```
    ADMUX = adc_input | ADC_VREF_TYPE;
```

```
    ADCSRA1 = 0x40;
```

```
    while ((ADCSRA & 0x10) == 0);
```

```
    ADCSRA |= 0x10;
```

```
    return ADCW;
```

```
}
```

```
void main(void)
```

```
{
```

```
    float voltage1, voltage2, adc1, adc2;
```

```
    char str1[20], str2[20], str3[20], str4[20];
```

```
    ADCADMUX = ADC_VREF_TYPE;
```

MICRO

Subject :**Date :**

```
ADC.SRA = 0x86 ;  
lcd_init(16);  
while (1)  
{  
    adc1 = read_adc(0);  
    adc2 = read_adc(3);  
    Voltag1 = adc1 + 0.00488;  
    Voltag2 = adc2 + 0.00488;  
    lcd_clear();  
    lcd_gotoxy(0,0);  
    ftoa(Voltag1, 2, Str1);  
    sprintf(Str2, "Voltag1: %.5f", Str1);  
    lcd_puts(Str2);  
    lcd_gotoxy(0,1);  
    ftoa(Voltag2, 2, Str3);  
    lcd_puts(Str3);  
    sprintf(Str4, "Voltag2: %.5f", Str3);  
    lcd_puts(Str4);  
    delay_ms(100);  
}
```

```
show_lcd (counter);  
Counter++;  
delay_ms (2);  
if (Counter > 10000)  
{  
    Counter = 0;  
}  
}  
}  
}  
}  
Void show_lcd (unsigned int counter)  
{  
    char str[20];  
    lcd_cleur();  
    lcd_gotoxy (0,0);  
    sprintf (str, "Counter : %5u", Counter);  
    lcd_puts (str);  
    lcd_gotoxy (0, 11);  
    lcd_putsf ("top count 10000");  
    delay_ms (10);  
}
```