نشریه علمی تخصصی مهندسی برق

پالس

PULSE

گاهنامه | شماره نهم| زمستان ۱۴۰۳

دانشگاه صنعتی همدان

فهرست مطالب

شناسنامه نشریـــــه

صاحب امتیاز: انجمن علمی مهندسی برق دانشگاه صنعتی همدان

دبیر: رضا صلح میرزایی

مدیر مسئول: محمد کسری صارمی

سردبیر نشریه: زهرا شاهسونی

ویراستار علمی و ادبی: نیما نوری

طراحی گرافیک و صفحه آرایی: نرگس طاوه ئی

نویسندگان: فاطمه حمیدی، تینا دهنوی، معین رسولی، زهرا شاهسونی، طاهره شایگانی، محمد کسری صارمی، رضا صلح میرزایی، حمید علیجانی، مهدی محمدی، محیا مرادی، پویا میرباقری، نیما نوری

نشانی: همدان، بلوار شهید فهمیده، خیابان مردم، دانشگاه صنعتی همدان، انجمن علمی مهندسی برق

صندوق پستی: ٦٥۱٥٥٥٧۹

شماره تلفن: ۰۸۱۳۸٤۱۱۳۱٤

این نشریه دارای شماره مجوز ۳٥/۸۹/ف/آ۱۷ در تاریخ ۱۳۸۹/۳/۱ از معاونت فرهنگی و اجتماعی است

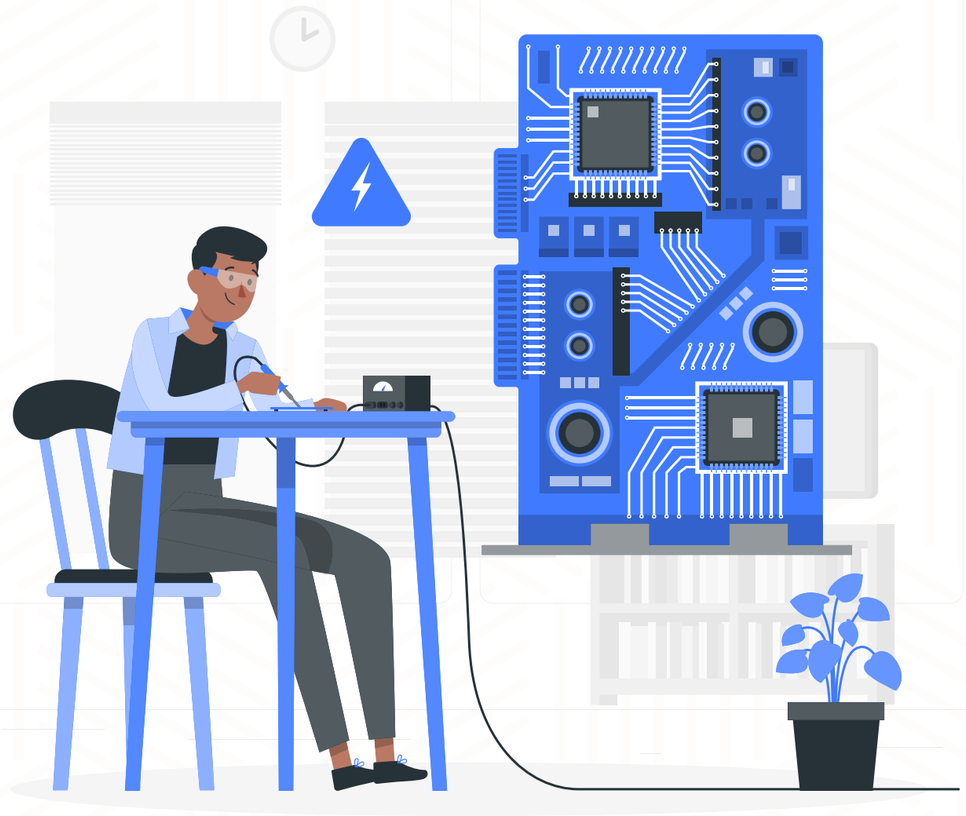
وبسایت: eee.hut.ac

پست الکترونیک انجمن: eehutac@gmail.com

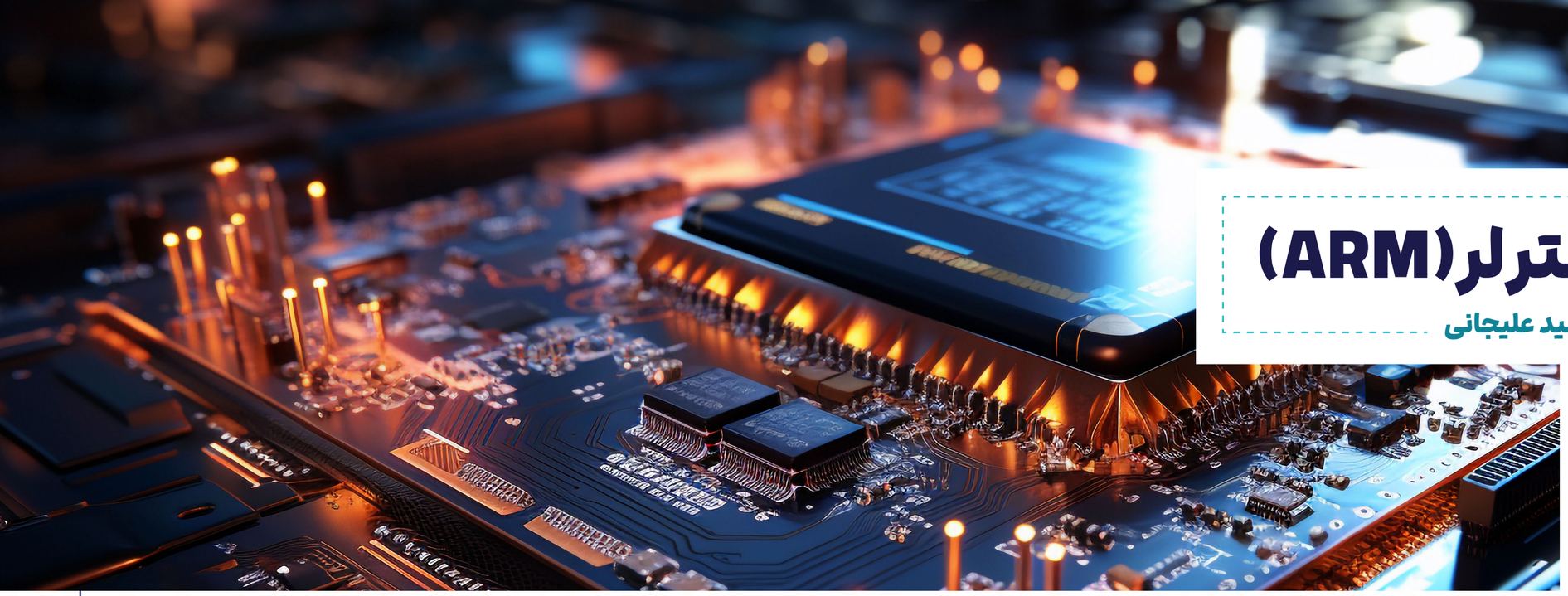
پست الکترونیک سردبیر: shzahra1822@gmail.com

آدرس اینستاگرام و تلگرام انجمن: eeea_hut@

آدرس کانال نشریه در تلگرام: https://t.me/pulse1400_HUT



بخش ۱:
الکترونیک



میکـــــرو کنترلر(ARM)

تینا دهنوی ، پویا میرباقری ، حمید علیجانی

میکروکنترلر چیست؟

میکروکنترلر یک مدار مجتمع[1] یا چیپ الکترونیکی برنامه پذیر است، که اطلاعات ورودی را جمع آوری کرده، آن را پردازش می کند و بر اساس اطلاعات جمع آوری شده عمل خاصی را انجام می دهد. این قطعات به گونه ای طراحی شده‌اند که بتوانند قطعات بزرگتر را بدون سیستم عامل پیچیده به سادگی کنترل کنند. از میکروکنترلر برای ساخت، کنترل و مانیتورینگ انواع سیستم های الکترونیکی استفاده می‌شود. به طورکلی میکروکنترلر از اجزای زیر تشکیل شده‌است.

واحد پردازش مرکزی[2]

واحد پردازش مرکزی (CPU) مغز میکروکنترلر است. CPU شامل یک واحد منطق حسابی[3] و یک واحد کنترل[4] است. یک CPU دستورات را برای انجام عملیات حسابی، منطقی و انتقال داده فراخوانی، رمزگشایی و اجرا می‌کند.

حافظه:

هر سیستم محاسباتی به نوع حافظه برنامه و داده نیاز دارد.

حافظه برنامه[5] (ROM) حاوی دستورالعمل هایی است که باید توسط CPU اجرا شوند.

حافظه داده[6] (RAM) برای ذخیره موقت داده‌ها در حین اجرای دستورالعمل هاست.

پورت های ورودی/خروجی (I/O)[7]:

رابطه میکروکنترلر با دنیای خارجی توسط پورت‌های ورودی/خروجی انجام می‌شود. دستگاه‌های ورودی مانند سوئیچ‌ها، صفحه‌کلیدها و غیره اطلاعاتی را در قالب داده‌های باینری از کاربر به CPU ارائه می‌کنند. سپس CPU با دریافت داده‌ها از دستگاه های ورودی، دستورالعمل های مناسب را اجرا کرده و به دستگاه‌های خروجی مانند LED، نمایشگر، چاپگر و غیره پاسخ می‌دهد.

تاریخچه:

اولین ریزپردازنده توسط فردریک فاجین یکی از مهندسان ایتالیایی حوزه الکترونیک و به کمک مهندسان اینتل تولید شد. این تراشه اینتل ۴ بیتی ۴۰۰۴ نام داشت. از این میکروکنترلر در کارهای محاسباتی و چاپی ساده استفاده می‌شود. پس از اینتل ۴۰۰۴ تراشه‌های ۴۰۴۰ و ۸ بیتی ۸۰۰۸ و ۸۰۸۰ روانه بازار شدند اما اینها علاوه بر هزینه زیاد و نداشتن صرفه اقتصادی نیازمند چندین تراشه خارجی بودند.

در سال ۱۹۷۴ شرکت TI[8] میکروکنترلر TMS1000 تک چیپی را تولید کرد که بسیار پیشرفته تر از نسخه‌های قبلی بود. در دهه ۹۰ میلادی با افزایش قابلیت‌ها، میکروکنترلرهایی با مصرف کمتر، قدرت بیشتر و قابلیت‌های گسترده تر معرفی شدند و ARM شروع به تاثیرگذاری در بازار میکروکنترلر کرد.

با ظهور اینترنت اشیاء تکنولوژی های جدید مانند BLE[9] و WiFi در میکروکنترلرها ادغام شدند تا امکان اتصال به شبکه را فراهم کنند و تقریبا در هر لحظه و مکان از زندگی روزمره ما حضور یابند.

8 Texas Instrument

9 Bluetooth Low Energy

چه چیزی معماری ARM را ارزشمند می کند؟

معماری‌های ARM رایج ترین طراحی الکترونیکی در جهان هستند، حتی اگر x86 در بازار سرور رایج‌تر باشد. معماری‌های ARM تقریبا در تمام طراحی‌های گوشی‌های هوشمند و همچنین در سایر دستگاه‌های کوچک موبایل و لپ‌تاپ‌ها استفاده می‌شود. تراشه‌های x86 برای بهینه سازی عملکرد طراحی شده‌اند. پردازنده‌های مبتنی بر ARM به گونه‌ای طراحی شده‌اند که هزینه را با اندازه‌های کوچکتر، مصرف انرژی کمتر، تولید حرارت کمتر، سرعت و به طور بالقوه عمر باتری بیشتر متعادل کنند. از آنجایی که Arm Holdings طراحی‌ها را می‌فروشد، نه سخت‌افزار، این به تولیدکنندگان سخت‌افزار اجازه می‌دهد تا ریزمعماری را مطابق با نیازهای خاص خود سفارشی سازی کنند و در عین حال اندازه کوچک، عملکرد بالا و کارایی انرژی را حفظ کنند. این هم مزایا و هم معایبی دارد، زیرا به این معناست، که سیستم عامل‌هایی مانند لینوکس، ویندوز و اندروید نیاز به پشتیبانی از طیف وسیع تری از سخت‌افزارها را دارند. این بدان معنا نیست که معماری‌های ARM فقط برای دستگاه‌های کوچک موبایل استفاده می‌شوند . یکی از سریع‌ترین ابررایانه‌های جهان، Fugaku که توسط Fujitsu و Riken طراحی شده است، از پردازنده ARM استفاده می‌کند. در مورد فوجیتسو، آنها تراشه ARM خود را برای ابرکامپیوتر خود طراحی کردند، اما ARM یک نمایه طراحی برای معماری های HPC نیز ارائه می‌دهد. به دلیل کاهش اندازه، مصرف انرژی کمتر و گرمای کمتر (کاهش نیاز به خنک‌کننده اضافی)، سازمان‌های بیشتری شروع به استفاده از سیستم‌های ARM برای ایجاد گره‌ها یا خوشه‌ها برای HPC و ناوگان ابری (مانند خدمات وب آمازون Graviton و Microsoft Azure می‌کنند.Arm compiler برای زنجیره ابزار لینوکس برای توسعه برنامه های HPC طراحی شده است. ارزیابی سازگاری بین برنامه‌های موجود، موارد استفاده و پردازنده‌های ARM هنگام ادغام آن‌ها بسیار مهم است. استفاده از معماری ARM به طراحان سخت افزار کنترل بیشتری بر طرح‌ها و عملکرد خود و همچنین کنترل بیشتری بر زنجیره تامین خود می دهد. این ترکیب کنترل و عملکرد در دستگاه‌های مصرف کننده کوچک و محیط‌های محاسباتی در مقیاس بزرگ به طور یکسان جذاب است.

طرح‌های اولیه RISC به طورکلی نوعی سیستم‌های آموزشی بودند و به‌طور خاص برای اجرای کامل طراحی نشده بودند. اصلی ترین آنها توانایی ارائه سریع وقفه‌ها بود که به ماشین‌ها اجازه می‌داد عملکرد ورودی/خروجی معقولی

1 Integrated Circuit
2 Central Processing Unit
3 Arithmetic Logic Unit
4 Control Unit
5 Read Only Memory
6 Random Access Memory
7 **Input/Output**

را ارائه دهند. طراحی ARM فضای آدرس فیزیکی خود را به ۶۴ مگابایت فضای آدرس پذیر محدود کرد که به ۲۶ بیت آدرس نیاز داشت. از آنجایی که دستورالعمل ها ۴ بایت (۳۲ بیت) بودند و باید روی مرزهای ۴ بایت تراز شوند، ۲ بیت پایینی یک آدرس دستورالعمل همیشه صفر بود. این بدان معناست که شمارنده برنامه[10] (PC) فقط باید ۲۴ بیت باشد و به آن اجازه می‌دهد تا به همراه پرچم‌های پردازنده هشت بیتی در یک ثبات ۳۲ بیتی ذخیره شود. این بدان معناست که با دریافت یک وقفه، کل وضعیت ماشین می‌تواند در یک عملیات ذخیره شود، در حالی که اگر رایانه شخصی یک مقدار کامل ۳۲ بیتی می‌بود، برای ذخیره رایانه و پرچم‌های وضعیت به عملیات جداگانه نیاز داشت. این تصمیم سربار وقفه را به نصف کاهش داد. تغییر دیگر و از مهمترین آنها از نظر عملکرد عملی در دنیای واقعی، اصلاح مجموعه دستورالعمل برای استفاده از DRAM حالت صفحه بود. حالت صفحه که اخیراً معرفی شد، به دسترسی‌های بعدی به حافظه اجازه داد که اگر تقریباً در یک مکان یا «صفحه» در تراشه DRAM قرار داشته باشند، دو برابر سریع‌تر اجرا شوند. طراحی برکلی حالت صفحه را در نظر نگرفت و با تمام حافظه به طور یکسان رفتار کرد. طراحی ARM دستورالعمل‌های ویژه دسترسی به حافظه بردار ما نند، "S-cycles" را اضافه کرد، که می‌توان از آنها برای پر کردن یا ذخیره چندین ثبات در یک صفحه با استفاده از حالت صفحه استفاده کرد. این کارایی حافظه را در زمانی که می‌توانستند مورد استفاده قرار دهند، دو برابر می‌کرد و به ویژه برای عملکرد گرافیکی مهم بود. طرح‌های RISC برکلی از پنجره‌های رجیستر برای کاهش تعداد ذخیره‌ها و بازیابی‌های ثبت‌شده در فراخوانی‌های رویه استفاده می‌کردند. طراحی ARM این را قبول نکرد. ویلسون مجموعه دستورالعمل‌ها را توسعه داد و شبیه‌سازی پردازنده را در بی‌بی‌سی بیسیک نوشت که روی بی‌بی‌سی میکرو با پردازنده دوم ۶۵۰۲ اجرا می‌شد. این امر مهندسان Acorn را متقاعد کرد که در مسیر درستی هستند. ویلسون با مدیر عامل شرکت Acorn، Hermann Hauser، تماس گرفت و درخواست منابع بیشتری کرد. هاوزر تایید خود را داد و تیم کوچکی را برای طراحی پردازنده واقعی بر اساس ISA ویلسون تشکیل داد. پروژه رسمی Acorn RISC

10 Program counter

ورود به دنیای میکروکنترلرها

میکروکنترلر AVR توسط شرکت Atmel در سال ۱۹۹۶ ساخته شده‌است. این میکروکنترلر مبتنی بر معماری مجموعه دستورالعمل RISC (ISA) است و به عنوان RISC مجازی پیشرفته نیز نامیده می‌شود. میکروکنترلر AT۹۰S۸۵۱۵ متعلق به خانواده AVR است. میکروکنترلر AVR محبوب ترین دسته میکروکنترلرها و همچنین جزو میکروکنترلرهای ارزان قیمت است. در بسیاری از برنامه‌های روباتیک از این میکروکنترلر استفاده می‌شود. خانواده زیادی از میکروکنترلرهایAVR مانند ATmega۸، ATmega۱۶، ATmega۳۲ و غیره وجود دارد. کاربردهای زیادی در AVR وجود دارد که در اتوماسیون خانگی، صفحه نمایش لمسی، دستگاه‌های پزشکی، صنعت اتومبیل و ... استفاده می‌شود.

میکرو کنترلرهای AVR دای ساختار داخلی متنوعی هستند، اما برخی از ویژگی‌های عمومی و اصلی آنها عبارتند از:

۱. پردازنده‌ی ۸ بیتی: بیشتر میکروکنترلرهای AVR، پردازنده‌های ۸ بیتی هستند که پردازشگر RISC است.

۲. حافظه: میکرو کنترلرهای AVR دارای حافظه داخلی برای ذخیره داده‌ها و برنامه‌های استفاده شده در عملکرد دستگاه هستند.

۳. تایمرها: این میکروکنترلرها دارای تایمرهای داخلی برای انجام کارهای زمانبندی و کنترلی هستند.

۴. پورت‌ها: AVR ها دارای پورت‌های ورودی و خروجی برای ارتباط با دیگر ماژول‌ها و سنسورهای خارجی هستند.

۵. مبدل آنالوگ به دیجیتال: برخی از میکرو کنترلرهای

AVR دارای مبدل آنالوگ به دیجیتال[11] (ADC) برای تبدیل سیگنال‌های آنالوگ به دیجیتال هستند.

۶. UART: برخی از AVR ها دارای ماژول UART برای ارتباط سریال با سایر دستگاه‌ها هستند.

۷. PWM: برنامه‌های تولید پالس[12] (PWM) برای کنترل خروجی‌های آنالوگ نیز از ویژگی‌های معمولی AVR ها هستند.

این ویژگی‌ها و ساختار داخلی میکرو کنترلرهای AVR آنها را برای کاربردهای الکترونیکی و کنترلی مناسب و محبوب کرده است.

انواع حافظه‌ها:

میکروکنترلرها دارای نواع مختلف حافظه‌های داخلی هستند که برای ذخیره داده‌های برنامه و داده‌های موقت استفاده می‌شوند. برخی از انواع حافظه‌های داخلی میکروکنترلرها عبارتند از:

. Flash Memory: این نوع حافظه برای ذخیره برنامه‌های اجرایی میکروکنترلر استفاده می‌شود. Flash memory برای ذخیره سازی کدهای برنامه استفاده می‌شود.

۲. حافظه SRAM برای ذخیره داده‌های موقت در حین اجرای برنامه استفاده می‌شود. این حافظه به سرعت بالا و دسترسی سریع به داده‌ها شناخته شده است.

۳. حافظه EEPROM برای ذخیره داده‌های دائمی استفاده می‌شود و می‌تواند توسط میکروکنترلر برنامه‌ریزی

11 Analog to Digital Converter

12 Pulse Width Modulation

شده و نوشته شود.

۴. Register File: هر میکروکنترلر AVR دارای یک فایل ثبت‌ها (register file) است که برای ذخیره داده‌ها و اطلاعات مرتبط با اجرای برنامه استفاده می‌شود.

۵. Special Function Registers (SFRs): این نوع حافظه برای اتصالات و وظایف خاص استفاده می‌شود که برای کنترل و مدیریت دستگاه‌های متصل به میکروکنترلر مورد نیاز است.

هر یک از این انواع حافظه داخلی میکروکنترلر AVR دارای وظایف و کاربردهای خاصی است و کنترلر AVR به صورت هوشمندانه این حافظه‌ها را برای ذخیره سازی و دسترسی به داده‌ها در طول اجرای برنامه مورد استفاده قرار می‌دهد.

میکروکنترلرهای AVR دای انواع مختلف حافظه‌های داخلی برای ذخیره داده‌های برنامهو داده‌های موقت هستند. در زیر به برخی از انواع حافظه‌های داخلی معمول در میکروکنترلرهای AVR اشاره شده است همراه با توضیحات کامل در مورد هرکدام:

Flash Memory در میکروکنترلرها برای ذخیره کدهای اجرایی پردازنده استفاده می‌شود. این نوع حافظه برنامه‌های کاربری و نیز توابع کدگذاری شده که توسط پردازنده اجرا می‌شوند را ذخیره می‌کند. Flash Memory برای ذخیره سازی دائمی برنامه‌ها و نرم‌افزارهای اجرایی استفاده می‌شود و حتی زمانی که تغییراتی اعمال نشود، داده‌ها حفظ خواهند شد.

SRAM برای ذخیره داده‌های موقت در حین اجرای برنامه استفاده می‌شود. این حافظه دسترسی سریع به داده‌ها و سرعت بالای خواندن و نوشتن دارد. SRAM به طور معمول برای ذخیره سازی متغیرها، میان‌نتیجه‌ها، وضعیت‌های فعلی و سایر داده‌های موقت برنامه مورد استفاده قرار می‌گیرد.

EEPROM برای ذخیره داده‌های دائمی استفاده می‌شود. این حافظه به طور قابل برنامه‌ریزی توسط میکروکنترلر می‌تواند نوشته و خوانده شود. داده‌های ذخیره شده در EEPROM حتی در صورت قطع برق به صورت پایدار حفظ می‌ماند.

هر میکروکنترلر دارای یک فایل ثبت‌ها (register file)

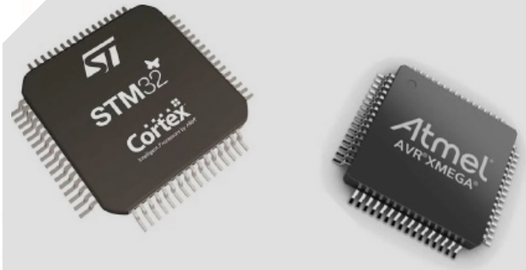
است که برای ذخیره داده‌ها و اطلاعات مرتبط با اجرای برنامه استفاده می‌شود. Register File شامل ثبت‌های عمومی، ثبت عملکرد، وضعیت اجرا، وضعیت‌های پردازشگر و دیگر اطلاعات لازم برای اجرای برنامه می‌باشد.این انواع حافظه‌های داخلی در میکروکنترلر AVR از دسته حافظه‌های اصلی می باشد که برای کنترل و اجرای برنامه‌ها و توابع مختلف استفاده می‌شوند.

یک واحد مهم در میکروکنترلر AVR نیز شامل ALU یا Arithmetic Logic Unit می‌باشد. این واحد مسئول انجام عملیات حسابی و منطقی بر روی داده‌های ورودی است. ALU از نوع واحدهای سخت‌افزاری است که عملات مرتبط با محاسبات و پردازش داده را انجام می‌دهد و نتایج را ارائه می‌دهد.

در میکروکنترلر AVR، ALU برای انجام عملیات اندازه‌گیری دقیق (مانند جمع، تفریق، ضرب و تقسیم)، عملیات منطقی (NOT، SHIFT، XOR، OR، AND و غیره) و دیگر عملیات پردازش داده‌ها استفاده می‌شود. ALU عموماً یک فرقه‌گریست‌ویاناگر(به طور خاص در مدارات منطقی خطی) و یک راور/بافنده(جمع) و اغلب چندین ساختار دیگر را اجرا می‌کند.

ALU یکی از قسمت‌های اصلی پردازنده‌ی مرکزی (CPU) است که وظیفه‌ی انجام عملیات محاسباتی و منطقی بر روی داده‌های ورودی را بر عهده دارد. این واحد از مهمترین بخش‌های پردازشگر است که واحد محاسباتی آن تعیین

تفاوت بین میکروکنترلرهای AVR و ARM



کننده‌ی قدرت و سرعت پردازش و اجرای برنامه‌هاست.

میکروکنترلرهای AVR و ARM دارای چندین تفاوت هستند که آنها را از یکدیگر متمایز می‌کند. در اینجا برخی از تفاوت‌های اصلی را بیان می‌کنیم.

معماری

میکروکنترلرهای ARM از معماری هاروارد با فضای حافظه مجزا برای دستورالعمل‌ها و داده‌های برنامه استفاده می‌کنند. این معماری زمان اجرای سریع‌تری را امکان پذیر می‌کند، اما به برنامه نویسی پیچیده‌تری نیاز دارد.در مقابل میکروکنترلرهای AVR از معماری Von Neumann با فضای حافظه یکپارچه برای دستورالعمل‌ها و داده‌های برنامه استفاده می‌کنند. این معماری برنامه‌نویسی ساده‌تری را امکان‌پذیر می‌کند، اما می‌تواند منجر به زمان اجرای کندتری شود.

مجموعه دستورالعمل (Instruction Set)

میکروکنترلرهای AVR یک مجموعه دستورالعمل ساده با تنها ۱۳۱ دستورالعمل دارند. این سادگی در مجموعه دستورالعمل، برنامه ریزی آنها را آسان می‌کند، اما عملکرد آنها را محدود می‌سازد.میکروکنترلرهای ARM دارای مجموعه دستورات پیچیده‌تری با بیش از ۵۰۰ دستورالعمل هستند. این پیچیدگی برنامه‌نویسی آن‌ها را دشوارتر می‌کند، اما امکان عملکرد بیشتری را برای این میکروکنترلرها فراهم می‌کند.

مصرف توان

به صورت کلی میکروکنترلرهای AVR در مقایسه با میکروکنترلرهای ARM مصرف انرژی پایین‌تری دارند. این موضوع باعث می‌شود، آنها برای برنامه‌هایی که نیاز به مصرف انرژی کم دارند، ایده آل باشند.

در مقابل و به صورت کلی میکروکنترلرهای ARM در مقایسه با میکروکنترلرهای AVR مصرف انرژی بالاتری دارند. در نتیجه برای برنامه‌هایی که نیاز به عملکرد بالا یا عملکرد پیچیده دارند، ایده آل هستند. البته با پیشرفت پردازنده‌های ARM امروزه مدل‌هایی از این نوع میکروکنترلر وارد بازار شده‌اند، که دارای مصرف توان بسیار کمی هستند.

هزینه

میکروکنترلرهای AVR عموماً ارزانتر از میکروکنترلرهای ARM هستند. در نتیجه برای برنامه‌های حساس به هزینه یا برنامه‌هایی که به عملکرد ساده نیاز دارند، ایده آل می‌باشند.

میکروکنترلرهای ARM عموماً گرانتر از میکروکنترلرهای AVR هستند. این ویژگی آنها را برای برنامه‌هایی که نیاز به عملکرد بالا یا عملکرد پیچیده‌تری دارند، ایده آل می‌کند.

تفاوت از نظر برنامه نویسی

میکروکنترلر ARM از زبان‌های برنامه نویسی زیادی چون پایتون و اسمبلی پشتیبانی می‌کند. این مدار از چارچوب‌های توسعه مختلف پشتیبانی می‌کند، که برای توسعه دهندگان برنامه نویسی را راحت‌تر می‌کند. برنامه ریزی میکروکنترلر AVR هم معمولا با C/C++ انجام می‌شود. دستورهای آن نیز نسبت به ARM ساده‌تر بوده، که برای مبتدیان بهتر و کار را آسان‌تر می‌کند.

حافظه

دیگر تفاوت میکروکنترلر AVR با میکروکنترلر ARM از نظر حافظه است. ARM حافظه بیشتری دارد که شامل حافظه FLASH و حافظه داده می‌شود. این حافظه‌ها می‌توانند ساختارهای داده‌ای پیچیده و بزرگتری را در خود جا دهند. معمولا ظرفیت حافظه نوع AVR کمتر بوده و برای برنامه‌های کوچک مناسب هستند.

پذیرش صنعت

هر دوی این مدارها در پیشرفت صنعت نقش مهمی دارند. هر کدام از آن‌ها در دستگاه‌های اینترنت اشیا، اتوماسیون صنعتی، لوازم الکترونیکی مصرفی، سیستم‌های خودرو و ... استفاده می‌شوند. نوع AVR در تجهیزات ساده بیشترین کاربرد را دارد، اما نوع ARM در تجهیزات پیشرفته بیشتر استفاده می‌شود.

عملکرد زمان واقعی

میکروکنترلر ARM در ارائه عملکرد زمان واقعی بهتر عمل می‌کند. این مدار ویژگی‌هایی مثل مدیریت دقیق وقفه، کنترل کننده‌های وقفه تو در تو و اجرای دستورهای قطعی دارد که به سبب آن برای برنامه‌های بی درنگ مناسب است، اما نوع AVR در مقابل ARM از نظر مدیریت زمان بندی دقیق و وقفه محدودیت‌هایی دارند.

اکوسیستم و ابزار توسعه

میکروکنترلر ARM یک اکوسیستمی با ابزارهای توسعه گسترده و کتابخانه‌های نرم افزاری تثبیت شده است. نوع AVR اکوسیستم اختصاصی با منابع و ابزارهای توسعه دارند و برای برنامه نویسی از IDE قدرتمندی چون ATMEL STUDIO استفاده می‌کنند. در حالی که پردازنده ARM از ابزارهای توسعه خود ARM وIDE های قدرتمندی مثل KELI MDK EMBEDDED WORKBENCH، IAR پشتیبانی می‌کند؛ بنابراین می‌توان گفت دیگر تفاوت میکروکنترلر AVR با میکروکنترلر ARM از نظر اکوسیستم و ابزار توسعه است.

حضور متفاوت در بازار

میکروکنترلر ARM در همه جای دنیا در بازار بیشتر حضور داشته و کاربرد وسیعی در وسایل خانگی و صنایع دارد. اما در حال حاضر در بازار ایران میکروکنترلرهای AVR بیشتر مورد استفاده هستند. نوع نیاز تولید کنندگان از دلایلی است که باعث حضور متفاوت پردازنده‌ها در بازار شده است.

کارایی

به طور کلی قدرت میکروکنترلر ARM نسبت به نوع AVR بیشتر بوده و عملکرد بالاتری را ارائه می‌دهد. مدار ARM در پیکربندی‌هایی مثل CORTEX_M۰، CORTEX_M۳، CORTEX_M۷ و CORTEX_M۴ در دسترس بوده‌است. همچنین قابلیت‌های محاسباتی، سطوح مختلف عملکرد و سرعت را ارائه می‌دهند. مدار AVR معمولا دارای معماری ۸ یا ۱۶ بیتی بوده و در کاربردهای ساده‌تری از آن‌ها استفاده می‌شود؛ همچنین در مقابل نوع ARM عملکرد کمتری دارند، ولی باز هم در وسایل خانگی و صنعتی و کاربردهایی که داده با سرعت پایین‌تری منتقل می‌شود، به کار می‌رود.

Bus Matrixدر میکروکنترلر های ARM یک ساختار معماری است که برای مدیریت دسترسی به حافظه و دستگاه‌های ورودی و خروجی به کنترلر مورد استفاده قرار می‌گیرد. این ساختار به میکروکنترلر امکان می‌دهد تا چندین واحد پردازشگری به صورت موازی به حافظه و دستگاه‌های ورودی و خروجی دسترسی پیدا کنند. در میکروکنترلرهای ARM، از Bus Matrix برای مدیریت دسترسی موازی به حافظه اصلی، حافظه‌های فلش، SRAM و دیگر دستگاه‌های ارتباطی مانند I۲C، SPI، UART و غیره استفاده می‌شود. در این ساختار، اطلاعات به صورت پارامترها از منابع مختلف به پردازشگر منتقل می‌شوند و برعکس. Bus Matrix امکان برنامه ریزی و مدیریت دسترسی برای واحدهای پردازشگری مختلف را فراهم می‌کند و میکروکنترلر را قادر به انجام چندین عملیات همزمان در زمان واقعی می‌کند. این ساختار به بهره‌وری و کارایی میکروکنترلر کمک می‌کند و از تداخل و اشکال در دسترسی به منابع جلوگیری می‌کند. در میکروکنترلرهای ARM دو اصلی‌ترین رابط‌های ارتباطی بین پردازنده مرکزی (CPU) و سایر واحدها مانند حافظه و دستگاه‌های ورودی و خروجی شامل AHB (Advanced) High-performance Bus و APB (Advanced Peripheral Bus) هستند.

AHB یک استاندارد صنعتی است که برای ارتباط با حافظه‌ها و دستگاه‌های پرسرعت مورد استفاده قرار می‌گیرد. AHB یک رابط مستقیم و سریع بین پردازنده مرکزی (CPU) و واحدهای پرسرعت مانند حافظه RAM، حافظه فلش، واحدهای DMA و دیگر اجزای سخت‌افزاری است. این رابط اجازه می‌دهد تا داده‌ها با سرعت بالا از و به CPU منتقل شوند و عملکرد سیستم بهبود یابد. از این دسته از رابط برای دستگاه‌های با نیاز به سرعت بالا مانند پردازشگرهای چندهسته‌ای و واحدهای پردازش داده‌های بزرگ استفاده می‌شود.

APB یک رابط مخصوص زیرسیستم‌ها و دستگاه‌های جانبی کنترلر است که به طور معمول در این جزئیات استفاده می‌شود. این رابط به CPU اجازه می‌دهد با دستگاه‌های جانبی مانند تایمرها، I۲C، SPI، UARTs و سایر واحدهای کنترلی ارتباط برقرار کند. APB معمولاً با سرعت پایین‌تری نسبت به AHB عمل می‌کند و به تبادل داده‌ها با دستگاه‌های کنترلی و ورودی/خروجی کاربری که نیاز به سرعت پایین‌تری دارند، کمک می‌کند. در کل، AHB برای ارتباط با واحدهای پرسرعت و APB برای ارتباط با دستگاه‌های جانبی با سرعت نسبتاً کمتر مورد استفاده قرار می‌گیرد. این دو رابط با همکاری از هم، امکان ارتباط و ارتباط موثر بین پردازنده و سایر واحدها را فراهم می‌آورند.

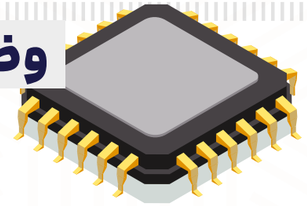
DMA یا دسترسی مستقیم به حافظه[۱۳] یک تکنیک است که برای انتقال داده به‌طور مستقیم بین دستگاه‌های ورودی/خروجی (I/O devices) و حافظه استفاده می‌شود بدون نیاز به دخالت مستقیم از سوی واحد پردازنده مرکزی (CPU). این تکنیک به میکروکنترلر امکان می‌دهد تا داده‌ها را بین حافظه و دستگاه‌های مختلف منتقل کند، در حالی که واحد CPU تمرکز خود را بر روی اجرای کد برنامه حفظ می‌کند. با استفاده از DMA، میکروکنترلر می‌تواند

13 Direct Access Memory

داده‌ها را بدون نیاز به مشارکت مستقیم CPU و به صورت موازی منتقل کند. این بهبود عملکرد میکروکنترلر ، زمانی که نیاز به انتقال داده‌های بزرگ یا در زمان‌های مشخصی از زمان رخ می‌دهد را افزایش می‌دهد.

DMA در میکروکنترلرها، به طور گسترده‌ای برای انتقال داده‌ها بین حافظه RAM، پورت‌های ورودی/خروجی (مانند ADC، I2C، SPI، UART و DAC) و دیگر دستگاه‌های جانبیاستفاده می‌شود. این تکنیک به توسعه‌دهندگان اجازه می‌دهد که عملیات انتقال داده‌ها را از CPU جدا کرده و به‌طور مؤثرتر با منابع سیستم میکروکنترلر مدیریت کنند.

وظایف اصلی DMA میکروکنترلرها



انتقال داده‌های از ورودی به حافظه (دریافت داده): DMA قابلیت انتقال داده‌هایی که توسط دستگاه‌های ورودی مختلف به میکروکنترلر وارد می‌شوند را دارا است. مثال‌هایی از دستگاه‌های ورودی شامل سنسورها، کارتهای حافظه، پورت‌های ارتباطی مثل UART یا SPI و غیره می‌باشند.

انتقال داده‌ها از حافظه به دستگاه‌های خروجی (ارسال داده): DMA می‌تواند انتقال داده از حافظه به دستگاه‌های خروجی مانند پورت‌های ارتباطی (UART، SPI، I2C، DAC، PWM) و دیگر دستگاه‌های جانبی را پشتیبانی کند.

انتقال داده‌ها درون حافظه: DMA همچنین قادر است به منظور انتقال داده‌ها بین مناطق مختلف حافظه (مثل انتقال داده بین حافظه RAM و حافظه فلش) در میکروکنترلر عمل کند. انتقال داده‌های بین دستگاه‌های پردازشی و حافظه: DMA می‌تواند داده‌ها را بین دستگاه‌های پردازشی مختلف یا بین دستگاه‌های پردازشی و حافظه منتقل کند.

به طور کلی، DMA در میکروکنترلرها برای بهبود عملکرد و کارایی سیستم با کاهش بار CPU و افزایش سرعت انتقال داده‌ها بین دستگاه‌های مختلف و حافظه استفاده می‌شود.

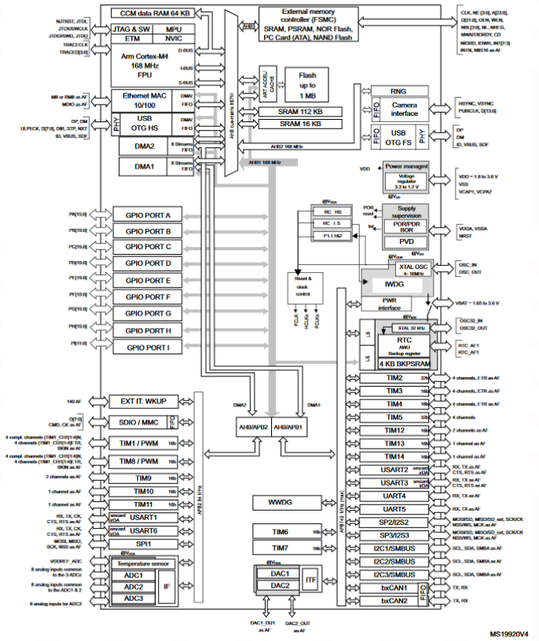
خانواده میکروکنترلرای STM32 از شرکت STMicroelectronics یکی از محبوب‌ترین و پرکاربردترین خانواده‌های میکروکنترلرها در صنعت الکترونیک است. این خانواده از میکروکنترلرها بر اساس معماری ARM Cortex-M ساخته شده است و انواع مختلفی از میکروکنترلرها با ویژگی‌ها و قابلیت‌های مختلف را ارائه می‌دهد.

یکی از میکروکنترلرهای این خانواده مدل STM-32F407VG است. در زیر مشخصات این میکروکنترلر را می‌بینید:

STM32F407VG:

- معماری: ARM Cortex-M4
- فرکانس پردازنده: ۱۶۸ MHz
- حافظه فلش: ۱ MB
- حافظه RAM: ۱۹۲ KB
- واحد محاسباتی: DSP نرخ نمونه برداری ۱۷۲ MSPS
- واحدهای ارتباطی: USART، SPI، I2C، USB OTG
- تایمرها: ۱۲ عدد تایمر ۱۶ بیتی
- ADC: ۱۲-bit ۲۴ کانال
- PWM: تا ۱۴ کانال
- کانال‌های DMA: ۱۶ عدد
- کلاک‌ها: دارای فاز و تایمینگ پیکربندی پذیر
- ولتاژ کاری: ۱٫۸ تا ۳٫۶ ولت
- پورت‌های IO: ۸۲ عدد

این میکروکنترلر یکی از مدل‌های پرکاربرد خانواده STM32 است که از ویژگی‌ها و قابلیت‌های بالایی برخوردار است و در بسیاری از پروژه‌های صنعتی و الکترونیکی استفاده می‌شود. این مدل همچنین دارای امکانات بیشماری است که برای توسعه‌دهندگان و طراحان، امکان پیاده‌سازی برنامه‌های پیچیده به راحتی فراهم می‌آورد.



شکل ۱ شماتیک ساختار STM32F407VG

برنامه‌های مورد نیاز برای کار با پردازنده‌های ARM

1_برنامه STM32 cube mx

2_برنامه keil

STM32cubemx

STM32CubeMX نرم افزاری گرافیکی است، که به شما امکان می دهد، میکروکنترلرها و میکروپروسسورها STM32 را به راحتی پیکربندی کنید و کد C اولیه سازی مربوطه را برای هسته Arm Cortex-M از طریق یک فرآیند گام به گام تولید کنید. گام اول شامل انتخاب یک

میکروکنترلر STM32 یا میکروپروسسور STMicroelectronics است. برای میکروپروسسورها، گام دوم به شما امکان می دهدGPIO ها و پیکربندی ساعت را برای کل سیستم پیکربندی کنید و به صورت تعاملی پین ها را به هسته Arm Cortex-M یاCortexA اختصاص دهید.

با این برنامه کد نویسی ابتدایی پردازنده مورد نظر انجام می گیرد. کدهای مورد نیاز برای راه اندازی برنامه keil با توجه به کارایی مورد نیاز و پین‌های مشخص شده ، توسط برنامه نوشته شده و نیازی به کد نویسی توسط برنامه نویس در این برنامه نمی‌باشد .

پردازنده‌های مختلفی در این برنامه وجود دارد، که هرکدام دارای قابلیت‌های جداگانه می باشد. در این مقاله به بررسی پردازنده STM32F103C8T6 یا همان blue pill می پردازید .

میکروکنترلر STM32F103C8T6 به علت قیمت پایین و در دسترس بودن، برای شروع کار با میکروکنترلرهای STM گزینه مناسبی است.

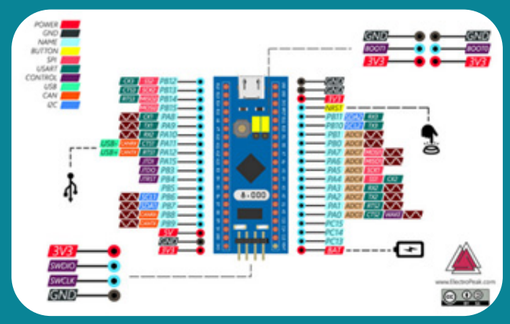
میکروکنترلر STM32F103C8T6 دارای هسته پردازنده 32 بیتی ARM Cortex M3 است، که توسط شرکت STMicroelectronics ساخته شده است و با فرکانس 72 مگاهرتز کار می‌کند. این میکروکنترلر 64 کیلوبایت حافظه FLASH روی تراشه را فراهم می‌کند. میکروکنترلر STM32F103C8T6 دارای 37 پین ورودی/خروجی و پروتکل های ارتباطی I2C, SPI UART/USART می‌باشد .

مشخصات کلی این میکروکنترلر به شرح زیر است:

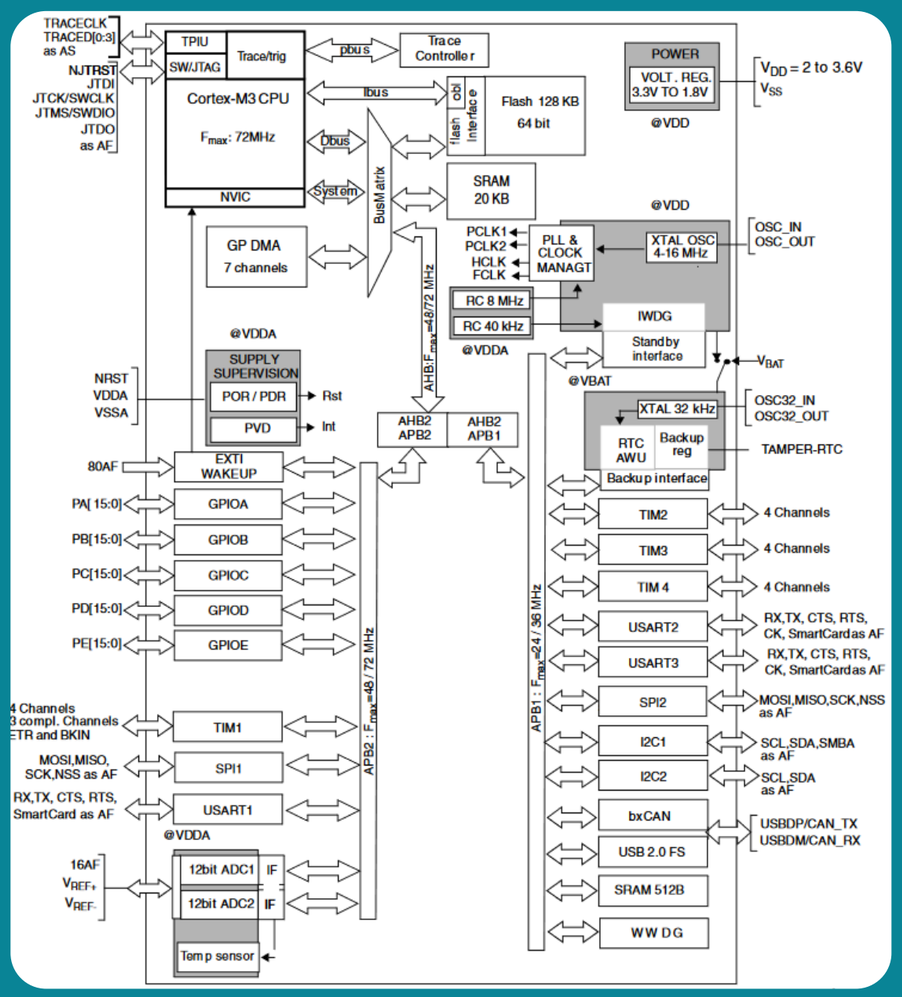
* میکروکنترلر 32 بیتی از خانواده CortexM3
* دارای 64 کیلوبایت حافظه فلش
* دارای 20 کیلوبایت حافظه SRAM
* حداکثر سرعت پردازش 72 مگاهرتز
* دارای 37 پایه GPIO
* دارای 12 کانال PWM
* دارای 10 کانال ADC 12 بیتی
* پشتیبانی از دو واحد I2C و دو واحد SPI و سه واحد UART
* دارای 3 تایمر 16 بیتی
* دارای یک واحد پشتیبانی از پروتکل Can 2.0

این میکروکنترلر بصورت ماژولار و آماده استفاده در دسترس می‌باشد.

بر روی هر پین این پردازنده ، می توان فانکشن‌های متفاوت ایجاد کرده و از آن بهره برد. در ادامه به بررسی چند قابلیت مهم از این پردازنده خواهیم پرداخت.



شکل ۲ شماتیک پین‌های میکروکنترلر STM32F103C8T6



شکل ۳ شماتیک ساختار داخلی F103C8T6

GPIO PIN_1

پین ورودی خروجی عمومی[14] یک رابط استاندارد برای ورودی و خروجی دیجیتال است، که در میکروکنترلر ها یافت می‌شود. این پین ها قادر به کنترل اجزای خارجی مانند روتورها و فرستنده‌های IR (خروجی) و همچنین دریافت داده‌ها از ماژول‌های حسگر و سوئیچ‌ها هستند.

RCC PIN_2

در میکروکنترلرها معمولاً به دو روش کلاک را تأمین می‌کنند. یک روش این است، که کل مدار به صورت خارجی کنار میکروکنترلر قرار داده می‌شود و روش دیگر اینکه فقط بخشی از این مدار، کنار میکروکنترلر قرار می‌گیرد، چون بخش دیگر این مدار به‌صورت داخلی درون خود میکروکنترلر وجود دارد.

درروش دوم فقط کریستال و دو خازن به‌صورت خارجی کنار میکروکنترلر قرار می‌گیرند و بقیه مدار درون میکروکنترلر قرار دارد. بخشی که در خارج مدار قرار می‌گیرد، فیدبک مدار نامیده می‌شود.

وقتی کل مدار به‌صورت یک ماژول کنار میکروکنترلر قرار داده می‌شود اصطلاحاً می‌گویند از اسیلاتور استفاده کرده ایم و وقتی بخشی از مدار کنار میکروکنترلر قرارداده می‌شود اصطلاحاً می‌گویند از کریستال استفاده کرده ایم.

ADC_3

یکی از کاربردی‌ترین ماژول‌هایی که در بسیاری از امبدد سیستم‌ها از آن استفاده می‌شود، مبدل آنالوگ به دیجیتال (ADC) است. این مبدل می‌تواند مقدار ولتاژ آنالوگ را از سنسورهای مختلفی مانند دما، جریان، سنجش میزان شیب و .. بخواند و آن را به مقدار معادل دیجیتال تبدیل کند.

EXTI_4

وقفه خارجی[15] یکی از ویژگی‌های اضافه شده به‌GPIO ها در حالت ورودی است، که باعث می‌شود میکروکنترلر روال فرآیندهایش را متوقف کند و به تغییرات اعمالی در پین‌های ورودی توسط event ها یا trigger ها پاسخ فوری بدهد.

Timer-5

Timer ابزاری است، که زمان را برای ما می‌سنجد، و سنجش زمان می‌تواند با استفاده از سیستم‌ها و روش‌های مختلفی ساخته شود. مثلاً زمان می‌تواند با استفاده از یک سیستم مکانیکی، پنوماتیکی و یا الکترونیکی ساخته شود. اما روش ساختی که در این مقاله مدنظر ما است، روش الکترونیکی و مشخصاً الکترونیک دیجیتال است. ساعت مچی نمونه‌ی خوبی از یک سیستم الکترونیکی-مکانیکی که زمان را برای ما می‌سنجد و معمولاً مبنای عملکرد آن بر اساس یک کریستال کوارتز است.

در میکروکنترلر هم معمولاً با استفاده از کریستال یا یک روش مشابه دیگر، یک سیگنال کلاک تولید می‌شود و ما با استفاده از همین سیگنال کلاک، زمان را می‌سنجیم.

14 General purpose Input/Output

15 External Interrupt

بخش برنامه نویسی

Keil MDK-ARM

نرم افزاری که به شما کمک می‌کند تا برنامه‌های کاربردی تعبیه شده برای دستگاه‌های مبتنی بر پردازنده-ARM Cor tex-M ایجاد کنید MDK . یک سیستم توسعه قدرتمند و در عین حال آسان برای یادگیری و استفاده است . این نرم افزار شامل MDK-Core و بسته‌های نرم افزاری است، که بر اساس نیاز برنامه شما قابل دانلود و نصب است

. ابزار MDK شامل تمام اجزایی است که برای ایجاد، ساخت و اشکال زدایی یک برنامه کاربردی تعبیه شده برای دستگاه‌های میکروکنترلر مبتنی بر ARM نیاز دارید .MDK-Core از IDE و دیباگر اصلی Keil µVision با پشتیبانی پیشرو برای دستگاه‌های میکروکنترلر مبتنی بر پردازنده CortexM از جمله معماری جدید ARMv8-M تشکیل شده است.

ویژگی نرم افزار Keil MDK-ARM :

- پشتیبانی کامل از سری ARM9 ، ARM7 ، M و ...
- پشتیبانی از زبان‌های برنامه نویسی C/C++
- دارای uVision 5 ، شبیه ساز و دیباگر
- امکان پشتیبانی از انواع پروتکل های شبکه‌ای و TCP/IP
- مجهز به کتابخانه کامل گرافیکی برای کمک در نوشتن برنامه
- دارای پروژه‌های آماده و گوناگون جهت یادگیری و آشنایی
- دارای نرم افزار استاندارد جهت کامپایل میکرو کنترلهای Cortex

پروگرام کردن وخطایابی میکرو کنترلرهای ARM

پروتکل JTAG (Joint Test Action Group) یک استاندارد ارتباطی است، که برای اتصال به میکروکنترلرها و همچنین برای برنامه‌ریزی و دیباگ نیزاستفاده می‌شود. این پروتکل از چهار سیگنال اصلی تشکیل شده است که به میکروکنترلر امکان فعالسازی، خواندن و نوشتن داده‌ها بر روی حافظه، و اجرای برنامه‌های دیباگ را می‌دهد.

از جهتی، پروتکل JTAG برای اتصال به شماتیک داخلی میکروکنترلر و اجازه دادن به مهندسان برای توسعه و دیباگ سیستم‌های الکترونیکی بسیار مفید است. اما به دلیل پیچیدگی بالای آن، برنامه‌ریزی و دیباگ از طریق JTAG برای برنامه نویسان کاربردی نیست و از ابزارها و پروتکل‌های دیگر برای این منظور استفاده می‌شود.

برای میکروکنترلرهای ARM نیز که از معماری Cortex-M و Cortex-A استفاده می‌کنند، پشتیبانی از پروتکل JTAG بسیار مهم است. این پروتکل برای برنامه‌ریزی، تست و دیباگ میکروکنترلرهای ARM استفاده می‌شود. با استفاده از پروتکل JTAG، توسعه‌دهندگان می‌توانند به نحو بهتری میکروکنترلرها را برنامه‌ریزی و دیباگ کرده و تست‌های لازم را انجام دهند تا اطمینان حاصل کنند که عملکرد میکروکنترلر به درستی انجام می‌شود. دیباگ کردن، به معنای شناسایی، ردیابی و رفع مشکلات و خطاهای نرم‌افزاری یا سخت‌افزاری در میکروکنترلرها است. وقتی که یک توسعه‌دهنده نرم‌افزار یا سخت‌افزار با یک میکروکنترلر کار می‌کند، ممکن است با خطاها و مشکلاتی مواجه شود که باعث اجرای نادرست برنامه شده و عملکرد ناپایدار می‌شود. در این موارد، فرایند دیباگ کردن اجازه می‌دهد تا توسعه‌دهنده این مشکلات و خطاها را شناسایی کرده، آنها را ردیابی کرده و با برطرف کردن آنها، عملکرد صحیح برنامه را بهبود بخشد.

در فرایند دیباگ کردن میکروکنترلرها، توسعه‌دهندگان ممکن است از ابزارهای دیباگ مختلف و تکنیک‌های متعددی استفاده کنند تا به مشکلات برنامه‌ی خود رسیدگی کنند. این ابزارها می‌توانند شامل ابزارهای سخت‌افزاری مانند JTAG و SWD یا نرم‌افزاری مانند Keil µVision، IAR Embedded Workbench و غیره باشند. بنابراین، دیباگ کردن میکروکنترلرها فرایندی است که توسعه‌دهندگان به کمک آن مشکلات و خطاها را در برنامه‌های نرم‌افزاری یا سخت‌افزاری که بر روی میکروکنترلرها اجرا می‌شوند، تشخیص داده و آنها را رفع می‌کنند تا بهبود عملکرد و اطمینان از صحت عملکرد میکروکنترلرها را دست یابند.

برای دیباگ کردن میکروکنترلرها، توسعه‌دهندگان باید از ابزارها و تکنیک‌های مناسبی استفاده کنند. برخی روش‌های معمول برای دیباگ کردن میکروکنترلرها به شرح زیر است:

استفاده از Debug Interface:

برای دیباگ کردن میکروکنترلرها، ابزارها و رابط‌های دیباگ مناسب به کار می‌روند. مثلاً از JTAG یا-

- SWD (Serial Wire Debug) برای اتصال به میکروکنترلرها استفاده می‌شود.

استفاده از Breakpoints

Breakpoints نقاط تعیین شده در کد برنامه هستند

که اجازه می‌دهند اجرای برنامه در آن نقطه متوقف شود. این ابزار به توسعه‌دهندگان کمک می‌کند تا مقدار متغیرها را بررسی کرده و مشکلات را شناسایی کنند.

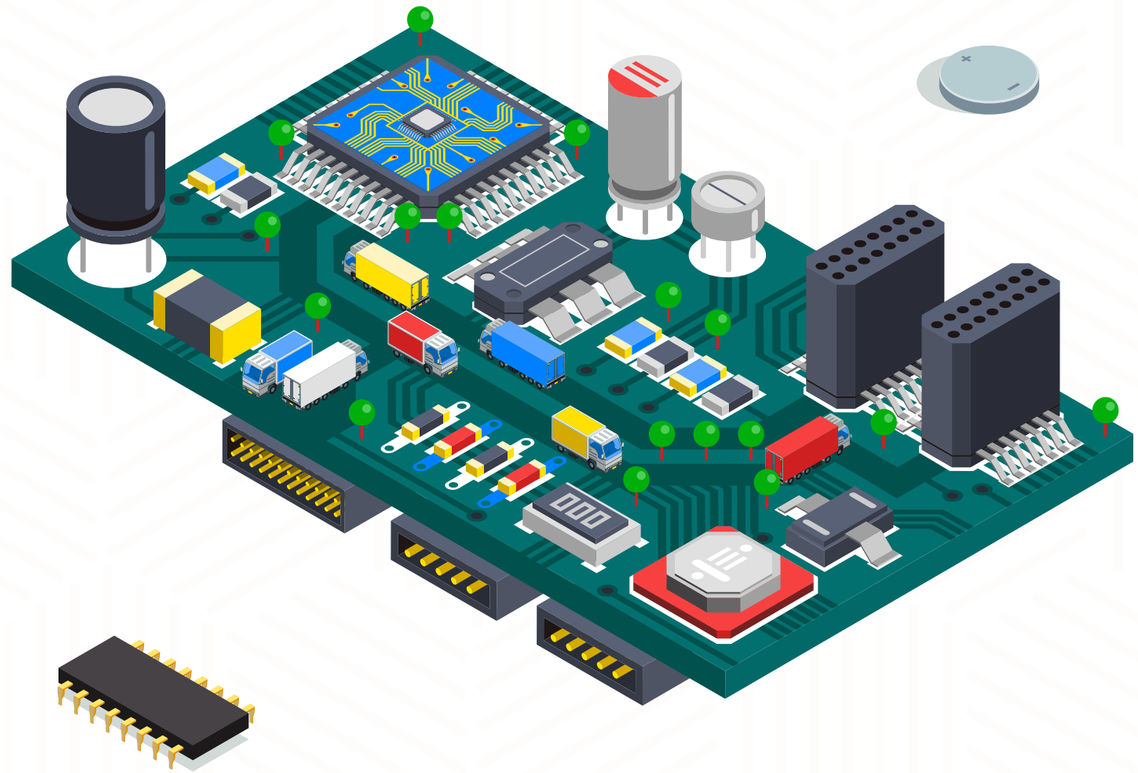
مشاهده مقادیر متغیرها: برخی از ابزارهای دیباگ امکان مشاهده مقادیر متغیرها در حین اجرای برنامه را فراهم می‌کنند که به توسعه‌دهندگان کمک می‌کند تا وضعیت برنامه را به صورت زنده مشاهده کرده و مشکلات را ردیابی کنند.

استفاده از نرم‌افزارهای ترمینال: استفاده از نرم‌افزارهای ترمینال می‌تواند به توسعه‌دهندگان کمک کند تا خروجی‌های سیستم را بررسی و مشکلات را ردیابی کنند.

استفاده از نرم‌افزارهای دیباگ: استفاده از نرم‌افزارهای دیباگ مانند IAR Embedded Workbench، Keil µVision، Atmel Studio و غیره نیز برای دیباگ کردن میکروکنترلرها بسیار مفید است. این نرم‌افزارها امکان برنامه‌ریزی، مانیتورینگ و دیباگ کد برنامه را فراهم می‌کنند.

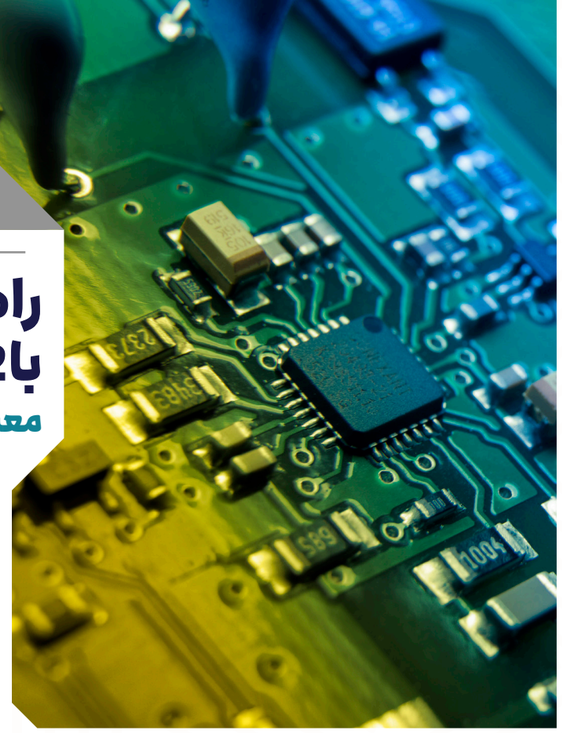
تست و نوشتن کد تستی: ایجاد تست‌های واحد و اجرای آنها می‌تواند به شناسایی و رفع خطاها کمک کند.

بازخورد و تجزیه و تحلیل خطاها: بعد از انجام دیباگ، تجزیه و تحلیل خطاها و بازخورد به توسعه‌دهنده بسیار مهم است تا بتواند از اشکالاتی که در حین دیباگ و رفع مشکلات به وجود آمده استفاده کرده و از آنها یاد بگیرد. با استفاده از این روش‌ها و ابزارها، توسعه‌دهندگان می‌توانند مشکلات و خطاهای برنامه‌های نرم‌افزاری و سخت‌افزاری میکروکنترلرها را شناسایی، ردیابی و رفع کنند و بهبود عملکرد و اطمینان از صحت عملکرد میکروکنترلرها را به دست آورند.



راه اندازی موتور DC با STM32

معین رسولی ، نیما نوری



برای اتصال یک موتور DC به میکروکنترلر STM-32F103C8T6 با قابلیت روشن و خاموش شدن و کنترل سرعت و جهت حرکت لازم است مراحل زیر را طی کنیم. این مدار توسط آی سی L298 که یک موتور درایور است طراحی و شبیه‌سازی شده است.

۱) تنظیمات STM32CubeMX

۲) کد های مرتبط با کنترل سخت افزار

۳) شبیه سازی مدار در پروتئوس یا ساخت مدار فیزیکی

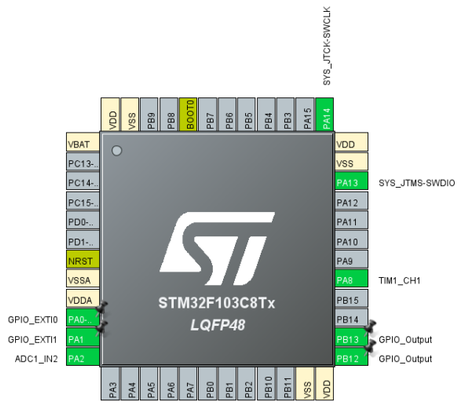
تنظیمات STM32CubeMX:

به طور کلی برای راه‌اندازی موتور نیازمند یک اختلاف ولتاژ هستیم. در صورتی که نیاز به کنترل سرعت و یا جهت نداشته باشیم می‌توانیم به صورت بسیار ساده یک پورت موتور را به قطب مثبت منبع تامین کننده مانند باتری و سر دیگر را به قطب منفی آن متصل کنیم؛ اما از آنجا که در اینجا هدف امکان کنترل جهت و سرعت چرخش موتور است ما تغییرات سرعت ایجاد شده بر روی موتور را با کمک موج PWM کنترل خواهیم کرد.

به ۲ کلید برای انجام موارد زیر نیاز داریم :

۱- روشن/خاموش

۲- جهت حرکت



شکل ۲-۱ پیکر بندی میکروکنترلر STM۳۲F۱۰۳C۸T۶

از سربرگ Pinout & Configuration و بخش Tim۱ مقدار Clock Source تایمر ۱ را روی Internal Clock قرار داده و سپس Channel ۱ را روی PWM Generation CH۱ قرار می دهیم.

باید توجه داشت به دلیل شبیه سازی در پروتئوس حتما باید کلاک سخت افزار به صورت داخلی و بر روی ۶۲ مگاهرتز تنظیم گردد.

پین های PA0 تا PA1 را جهت ساخت ۲ کلید کنترلی، روی حالت External interrupt یا همان GPIO_EXTIn قرار می‌دهیم. در سربرگ GPIO نوع حساسیت پین را مشخص کنید (Rising edge, falling edge) پین ها را از سربرگ GPIO حتما Pull Up یا Pull Down کنید. در سربرگ NVIC پین ها را Enable کنید و بسته به نیاز خود اولویت آنها را تعیین کنید. این کار باعث فعال شدن پین در کنترل کننده وقفه مرکزی می‌گردد. در ادامه برای کنترل خود موتور و فعال سازی آن باید پین‌های PB12 و PB13 را به صورت خروجی تنظیم

نموده تا حریکت موتور را به توان با این پین کنترل کرد. باید توجه داشت خروجی PWM باید به پایه ENA آی سی L298 برای کنترل سرعت متصل گردد. همچنین برای روشن و خاموش کردن موتور نیز باید هردو پین PB12 و PB13 را به IN1 و IIN2 آی سی متصل کرد. آی سی دو تغذیه مجزا دارد که تغذیه موتور بر روی ۲۴ ولت و تغذیه منطقی آن که برای پین‌ها استفاده می‌شود، بر روی ۵ ولت قرار داده شده‌اند. در شکل ۱-۱ پایه PA2 برای ورودی ADC که به پتانسومتر برای خوانش مقدار ولتاژ آن به منظور کنترل سرعت است، به کار گرفته شده است.

کدهای مرتبط با کنترل سخت

در شکل ۲-۲ خطوط ۵۲ تا ۵۷ پروتوتایپ‌های تابع های مورد استفاده می‌باشند. تابع‌های ۵۲ تا ۵۵ توسط خود نرم افزار کانفیگ شده‌اند. تابع‌های شماره ۵۶ و ۵۷ توسط خود ما به منظور کنترل سرعت و جهت موتور نوشته شده‌اند، که در ادامه آن‌ها را بررسی خواهیم کرد. خطوط بین ۶۷ تا ۷۸ بیانگر تابع رویداد وقفه می‌باشد. این تابع توسط خود شرکتST نوشته شده است و تنها نیاز به فراخوانی آن خارج از تابع main می‌باشد. در این تابع متغیر ۱۶ بیتی ورودی تابع توسط دو دستور شرطی if هنگامی که کلیدها فشرده می‌شوند، بررسی شده و عملیات مورد نظر را انجام می‌دهد. در خط ۶۴ دو متغیر برای ذخیره جهت موتور و روشن/خاموش کردن آن ایجاد کرده‌ایم. هنگام فشردن کلید در تابع وقفه، با توجه به کلید فشرده شده این متغیرها با مقدار ۱ جمع شده و در همان متغیر ذخیره می‌گردند.

```
52 void SystemClock_Config(void);
53 static void MX_GPIO_Init(void);
54 static void MX_TIM1_Init(void);
55 static void MX_ADC1_Init(void);
56 void speed_contnroll (uint32_t PWM_ADC_getvalue);
57 void set_direction (uint16_t Start_Stop, uint16_t left_right);
58 /* USER CODE BEGIN PFP */
59
60 /* USER CODE END PFP */
61
62 /* Private user code ---------------------------------------------------------*/
63 /* USER CODE BEGIN 0 */
64 uint16_t left_right_rotate=2 , start_stop=1 ;
65 uint32_t ADC_value ;
66
67 void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
68 {
69     if (GPIO_Pin == GPIO_PIN_0)
70             {
71         left_right_rotate=left_right_rotate+1;
72             }
73     if (GPIO_Pin == GPIO_PIN_1)
74             {
75         start_stop=start_stop+1;
76             }
77
78
```

شکل ۲-۲ دستورات ابتدایی در برگه main.c

با توجه به شکل ۲-۳ بین خطوط ۳۵۳ تا ۳۷۶ تابع اصلی نوشته شده است. در خط۳۵۴ ابتدا روشن و خاموش بودن موتور مورد بررسی قرار گرفته است. منطق این دستور از قاعده باقی‌مانده استفاده می‌کند. این منطق بدین

صورت عمل می‌کند که با فشردن کلید متغییر یک واحد افزایش می‌یابد. با افزایش متغییر عدد احتمالی فرد یا زوج می‌باشد. در صورت فرد بودن این تابع دو خروجی کنترلی موتور را ۰ کرده و موتور را در بین خطوط ۳۷۱ و ۳۷۲ خاموش می‌کند. اگر متغییر زوج باشد موتور شرط بعدی که از همان منطق تبعیت برای چرخش جهت موتور استفاده می‌کند. در بین خطوط ۳۵۸ تا ۳۶۵ می‌توان دید اگر متغییر تغییر جهت زوج باشد پین ۱۲ به صورت منطق ۱ و ۱۳ منطق ۰ به خود گرفته و موتور ساعتگرده می‌چرخد، در غیر این صورت منطق پین‌ها عوض شده و موتور پاد ساعت‌گرد خواهد چرخید.

```
351 void set_direction (uint16_t Start_Stop, uint16_t left_right)
352 {
353     //start_stop button GPIOB PIN_12
354     if (Start_Stop % 2 == 0)
355      {
356         if (left_right % 2 ==0)
357         {
358             HAL_GPIO_WritePin(GPIOB, GPIO_PIN_12 , 1);
359             HAL_GPIO_WritePin(GPIOB, GPIO_PIN_13 , 0);
360         }
361
362         else
363         {
364             HAL_GPIO_WritePin(GPIOB, GPIO_PIN_12 , 0);
365             HAL_GPIO_WritePin(GPIOB, GPIO_PIN_13 , 1);
366         }
367
368      }
369     else
370     {
371         HAL_GPIO_WritePin(GPIOB, GPIO_PIN_12 , 0);
372         HAL_GPIO_WritePin(GPIOB, GPIO_PIN_13 , 0);
373
374     }
375
376
377 }
```

شکل۲-۳ تابع چرخش موتور

با توجه به شکل ۲-۴ این تابع مقدار خوانده شده توسط واحد ADC که ولتاژ پوتانسیومتر را می‌خواند، دریافت کرده و در متغییر خط ۳۸۲ ذخیره می‌نماید. سپس متغییر دیگری که در خط ۳۸۲ ایجاد شده است، مقدار خام گرفته شده از ADC را به مقدار واقعی ولتاژ پتانسومتر تبدیل می‌نمایید. با توجه به این موضوع که ADC در مد ۱۲ بیتی پیکر بندی شده است و ۴۰۹۶=۱۲^۲ و همچنین ولتاژ مرجع ۳/۳ ولت برای ADC منطق فرمول ۳۸۳ را نمایان می‌کند. سپس در خط ۳۸۴ این مقدار به صورت پونتری در رجیستر ذخیره کننده دیوتی سایکل تایمر ۱ بارگزاری شده است.

```
380 void speed_contnroll (uint32_t PWM_ADC_getvalue)
381 {
382 uint32_t ADC_value = PWM_ADC_getvalue;
383 uint32_t ADC_integer_value = (ADC_value*3.3)/4095;
384 TIM1->CCR1 = (ADC_integer_value*100)/3.3;
385
386 }
387
```

شکل ۲-۴ تابع کنترل سرعت

در شکل شماره ۲-۵ متحویات درون تابع main نشان داده شده‌اند. در بین خطوط ۱۱۰ تا ۱۱۲ واحدهایی مانند ADC، GPIOو تایمر که توسط خود نرم افزار کانفیگ شده‌اند به صورت فراخوانی تابع فعال سازی می‌شوند. در بین خطوط ۱۱۶ تا ۱۱۹ واحدها با دستورات Init فعال سازی و سپس با دستورات Start کار خود را آغاز می‌نمایند، این دستورات از توابع HAL که نوشته توسط شرکت ST برای راه اندازی میکروکنترلر می‌باشد مورد استفاده قرار گرفته‌اند. در خطوط بین ۱۲۵ تا ۱۳۷ حلقه تکرار بی نهایت قرار دارد و دستورات این حلقه به صورت مدام تکرار خواند شد. در خط ۱۲۸ مقدار ADC دائما مورد خوانش قرار می‌گیرد و متغییر ADC_value مقدار جدید را بروزرسانی می‌کند. پس از تاخیر ۱۰۰ میلی ثانیه‌ای حالت موتور بررسی شده و سپس بعد از تاخیر۱۰۰ میلی ثانیه دیگر سرعت موتور تنظیم می‌گردد. و در نهایت ۱۰۰ میلی ثانیه تاخیر دیگر برای تکرار مجدد حلقه در نظر گرفته شده است.

شبیه سازی مدار به صورت کامل انجام شد و صحت عملکرد کدها تایید می‌گردد.

```
110   MX_GPIO_Init();
111   MX_TIM1_Init();
112   MX_ADC1_Init();
113   /* USER CODE BEGIN 2 */
114
115
116   HAL_TIM_Base_Init(&htim1);
117   HAL_TIM_PWM_Init(&htim1);
118   HAL_ADC_Start(&hadc1);
119   HAL_TIM_PWM_Start(&htim1, TIM_CHANNEL_1); //gpioa_pin8
120
121   /* USER CODE END 2 */
122
123   /* Infinite loop */
124   /* USER CODE BEGIN WHILE */
125   while (1)
126   {
127     /* USER CODE END WHILE */
128       ADC_value = HAL_ADC_GetValue( &hadc1) ;
129 HAL_Delay(100);
130 set_direction(start_stop, left_right_rotate);
131 HAL_Delay(100);
132 speed_contnroll(ADC_value);
133 HAL_Delay(100);
134     /* USER CODE BEGIN 3 */
135   }
136   /* USER CODE END 3 */
137 }
138
```

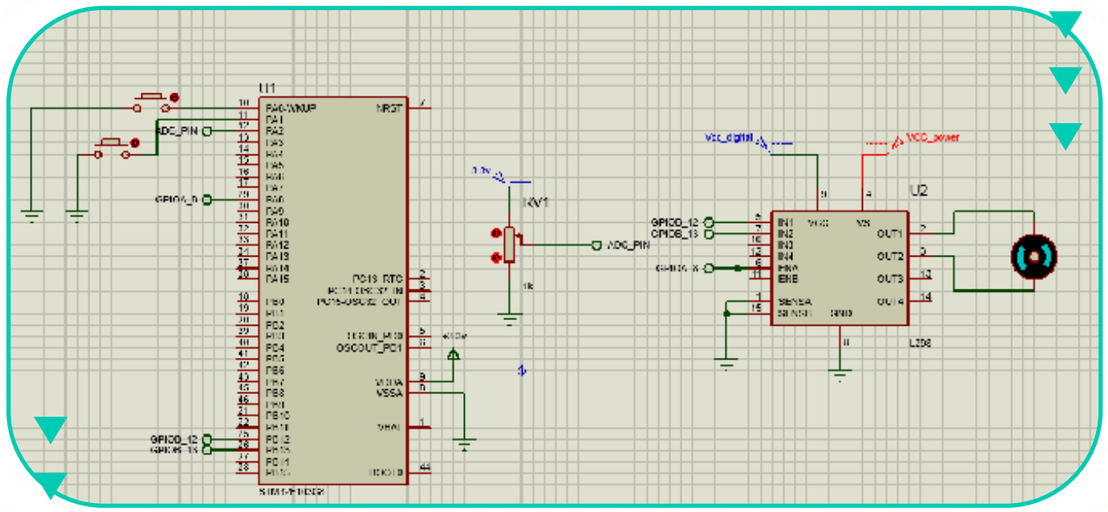
شکل ۲-۵ تابع اصلی main

شبیه سازی:

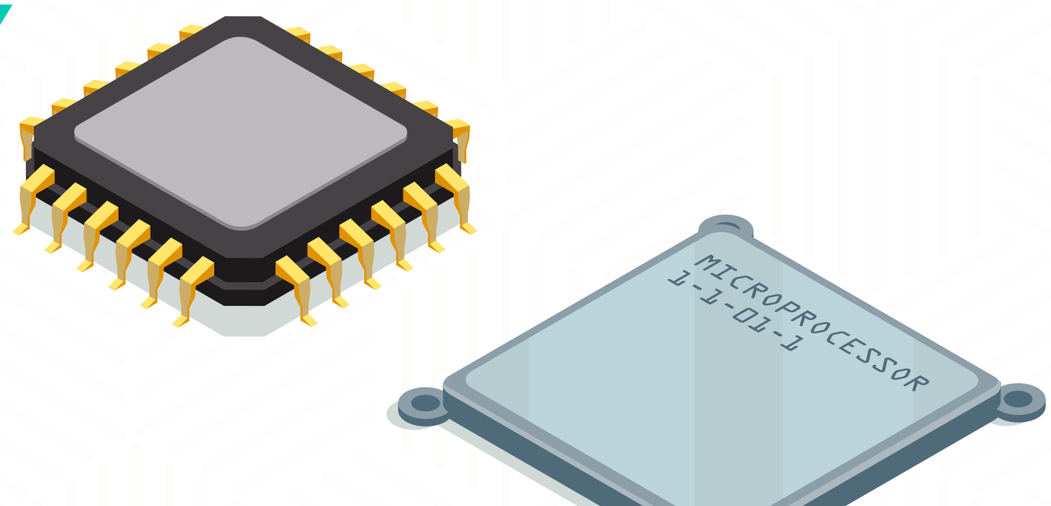
در شکل شماره ۲-۶ مدار شبیه‌سازی شده در محیط پروتئوس نمایش داده شده است. با توجه به پیکربندی‌ها دو کلید فعال به پایه‌های PA۰ و PA1 به صورت منطق 0 متصل گردیده‌اند. دو پین PB12 و PB13 نیز به پایه‌های 5

و 13 آی سی L298 متصل شده‌اند. با فشردن کلید دوم موتور خاموش و روشن شده و با فشردن کلید اول موتور تغییر جهت خواهد داد. در مرکز شکل یک پتانسومتر ۱ کیلو اهمی توسط ولتاژ ۳/۳ تغذیه شده و سپس به پایه PA8 متصل شده‌است.

با تغییر پتانسیومتر مقدار ولتاژ پایه ADC تغییر کرده و سپس سرعت خروجی متناسب با این مقدار عوض می‌شود. موتور نیز به زوج خروجی‌های اول آی سی و پین‌های ۲ و ۳ متصل شده است.



شکل شماره ۲-۶ مدار شبیه‌سازی شده در پروتئوس



ویندوز های مبتنی بر ARM

زهرا شاهسونی

ویندوزهای مبتنی بر ARM، نسخه‌هایی از ویندوزی هستند، که به طور خاص برای سخت افزار های ARM، مثل تراشه های Qualcomm Snapdragon)) طراحی شده‌اند. این ویندوزها می‌توانند بر روی انواع دستگاه‌ها، موبایل و همچنین رایانه‌های شخصی که، از معماری ARM برای اجرای سیستم عامل و برنامه‌های استفاده می کنند، پیاده سازی شوند. سیستم عامل ویندوز برای این تراشه‌ها بهینه شده است تا بتواند بهترین عملکرد را بدست آورد و بهره‌وری بالایی داشته باشد.

تاریخچه نسخه ویندوز آرم به چندین مرحله کلیدی تقسیم می شود:

- ویندوز RT اولین نسخه از ویندوز بود که برای پردازنده های آرم طراحی شد. این نسخه همراه با ویندوز 8 معرفی شد و به دلیل محدودیت‌های نرم افزاری و عدم پشتیبانی از برنامه‌های دسکتاپ سنتی موفقیت چندانی نداشت.
- ARMویندوز10 نسخه‌ای بود که امکان اجرای اپلیکیشن‌های معماری x86 را نیز فراهم می‌کرد و هدف آن ارائه دستگاه‌های سبک و کم مصرف با عمر باتری طولانی‌تر بود.
- ARM ویندوز11 جدیدترین نسخه‌ای است که، با بهبودهای بیشتری عرضه شده‌است. مانند پشتیبانی از نرم افزار بازی سازی Unity که به ویندوز ARM اضافه شد، تا به توسعه دهندگان اجازه دهد بازی‌های خود را مستقیما برای این پلتفرم توسعه دهند.

قابلیت های ARM ویندوز:

1.کارایی و مصرف انرژی: این پردازنده‌ها به دلیل مصرف انرژی کمتر، عمر باتری بیشتری دارند.

2.سبک و قابل حمل بودن: این دستگاه‌ها معمولا سبک تر و نازک‌تر هستند که آن‌ها را برای حمل و استفاده در مکان‌های مختلف بسیار مناسب می‌سازد.

3. امنیت بیشتر: به دلیل طراحی خاص، امنیت بیشتری در برابر برخی از حملات سایبری دارد.

4. شروع به کار سریع: این دستگاه ها معمولا زمان Startup سریع تری دارند و به سرعت آماده به کار می شوند.

5. اتصال دائمی به اینترنت: این قابلیت از طریق فناوری‌های مختلفی امکان پذیر است. یکی از این فناوری‌ها اینترنت اشیا باند باریک[1] (NB-IoT) است، که یک استاندارد کم مصرف و با برد بالا برای اتصال دستگاه‌های اینترنت اشیاء به شبکه‌های موبایل است. این فناوری به دستگاه‌ها اجازه می‌دهد تا با مصرف انرژی کم، داده‌های خود را به شبکه ارسال کنند و برای مدت طولانی تری بدون نیاز به تعویض باتری کار کنند.

1 Narrowband-IoT

علاوه بر این، دستگاه‌های مبتنی بر ARM می‌توانند از فناوری‌های دیگری مانند LTE Cat M1 و Wi-Fi نیز برای اتصال به اینترنت استفاده کنند. این فناوری‌ها به کاربران امکان می‌دهند تا همیشه آنلاین باشند و از خدمات اینترنتی بهره‌مند شوند.

6. اجرای برنامه‌های سنتی ویندوز: ویندوزهای مبتنی بر ARM، می‌توانند بسیاری از برنامه‌های سنتی ویندوز را از طریق اجرا بر روی شبیه سازی اجرا کنند. برای اجرای برنامه های x86 از تکنولوژی امولیشن[2] استفاده می‌شود. این تکنولوژی اجازه می‌دهد تا برنامه‌های x86 روی سیستم عامل ویندوز ARM اجرا شوند، حتی اگر نسخه‌ای از برنامه برای معماری ARM موجود نباشد. اما باید توجه داشت که اجرای برنامه‌ها ممکن است باعث کاهش عملکرد برنامه‌ها شود، زیرا امولیشن نیاز به پردازش اضافی دارد.

7. ARM64EC : این فناوری در ویندوز11 به توسعه دهندگان اجازه می‌دهد تا بدون بازنویسی کامل کد، تنها قسمت‌هایی از نرم افزارهای ARM را بهینه سازی کنند.

با توجه به پیشرفت‌های اخیر در اجرای نسخه‌ی کامل ویندوز ۱۰ روی پردازنده‌های ARM، آینده‌ی ویندوزها با این معماری وارد دگرگونی بسیار بزرگی از حال تا آینده پیش رو شده‌است. این پردازنده‌ها با مصرف کمتر انرژی، اتصال سریع‌تر به شبکه‌ها، و عمر باتری بیشتر، بهبود و نوآوری در رایانه‌های شخصی را به ارمغان می‌آورند.با اضافه شدن ویژگی‌های جدید در ویندوز 11، از جمله اجرای برنامه‌های اندروید، بهبود تجربه تماس‌های تصویری با هوش مصنوعی و قابلیت‌های مرتبط با تنظیمات باتری، کاربران می‌توانند از تجربه‌ی بهتری در استفاده از این پلتفرم بهره برند.

به طور خلاصه، شرکت آرم با توسعه نسخه ویندوز مخصوص خود با توانایی اجرای کامل در نسخه‌ی ویندوز ۱۰، قدم مهمی در جهت تحول رایانه‌های شخصی به سمت محصولات با عمر باتری بیشتر و اتصالات سریع‌تر برداشته‌است.

2 emulation

بخش ۲:

کنتـــــــــــرل

سیستم های کنترل

سیستم HMI محمد کسری صارمی

HMI مخفف عبارت Human Machine Interface میباشد و به معنای "رابط کاربری ماشین-انسان" میباشد.

HMI نوعی نمایشگر صنعتی هستند که می‌تواند با بقیه تجهیزات کنترلی، ارتباط برقرار کرده و تبادل اطلاعات را انجام دهد. واحد رابط انسان و ماشین یا همان HMI، یک دستگاه یا رابط گرافیکی است که ارتباط بین انسان و سیستم‌های اتوماسیونی مانند PLC را فراهم می‌کند. این دستگاه به کاربران امکان می‌دهد با استفاده از صفحه نمایش و واسط کاربری گرافیکی، فرآیندها و عملیات صنعتی را مشاهده، کنترل و مدیریت کنند.

HMI نوعی نمایشگر صنعتی هستند که می‌تواند با بقیه تجهیزات کنترلی، ارتباط برقرار کرده و تبادل اطلاعات را انجام دهد.

واحد رابط انسان و ماشین یا همان HMI، یک دستگاه یا رابط گرافیکی است که ارتباط بین انسان و سیستم‌های اتوماسیونی مانند PLC را فراهم می‌کند.

این دستگاه به کاربران امکان می‌دهد با استفاده از صفحه نمایش و واسط کاربری گرافیکی، فرآیندها و عملیات صنعتی را مشاهده، کنترل و مدیریت کنند.

از انواع مدل های HMI نیز میتوان به دو مدل تاچ(لمسی) و کیبوردی اشاره کرد.

از مزایای مهم و تاثیرگذار انتخاب hmi به عنوان یک سیستم نظارتی گرافیکی در تجهیزات اتوماسیون صنعتی می توان به موارد زیر اشاره کرد:

دارای صفحه لمسی و LCD

دارای پورت USB برای دریافت و ارسال برنامه‌ها

قابلیت اتصال به اینورتر، PLC و هر وسیله دارای پورت RS-۴۸۵

قابلیت نمایش اطلاعات به صورت نمودار و منحنی

ارتباط سریال از طریق دو پورت به صورت همزمان

شرکت های معروف زیادی دست به ساخت اچ ام آی به صورت تخصصی زدند، از جمله می توان به موارد زیر اشاره کرد:

دلتا (DELTA)

زیمنس (SIEMENS)

توشیبا (TOSHIBA)

ویگور (VIGOR)

فوجی (FUJI)

امرون (OMRON)

تفاوت بین PLC و HMI :

سیستم‌های کنترلی امروزی، در جهت کنترل بسیاری از تجهیزات مانند موتورها، دریچه ها و پمپ‌ها از PLC استفاده می‌کنند. اچ ام آی (HMI) وسیله ارتباطی بین کاربر و PLC می‌باشد. با نوشتن برنامه برای اچ ام آی، یک رابط گرافیکی در هر صفحه ایجاد می شود که امکان دریافت و ارسال دستورات به PLC را ببرای کاربر ایجاد می کند. تمامی موارد ایجاد شده مانند اعلام وضعیت و نمایش آلارم و ... به یک آدرس در PLC لینک می‌شود. به کمک HMI، کاربران می‌توانند با کارکرد PLC آشنا شوند، عملکرد آن را مشاهده کرده و به‌طور مستقیم دستورات خود را برای کنترل و مدیریت فرآیندها ارسال کنند. این ارتباط اساسی در صنایع مختلف از تولید گرفته تا خطوط تولید و سیستم‌های کنترلی پیچیده به کار می‌رود، به‌طوری که کاربران بتوانند به‌راحتی و کارآمدی با سیستم‌های PLC ارتباط برقرار کنند و فرآیندها را مدیریت کنند.

تفاوت بین scada و HMI :

این دو سیستم در ظاهر شباهت زیادی دارند اما هر کدام قابلیت و عملکرد متفاوتی دارد. سیستم های hmi و scada به ترتیب برای نمایش داده های خاص و جمع آوری استفاده می شوند. اسکادا و اچ ام آی در اتوماسیون صنعتی نقش بزرگی در کنار PLC ها ایفا می کنند. اچ ام آی به صورت گرافیکی یا بصری استفاده می شود که باعث درک بهتر و نظارت کارآمدتری می گردد. در سیستم های scada ظرفیت بسیار زیادی از اطلاعات را جمع آوری و ثبت می کنند و بر روی عملیات سیستم کنترل تمرکز دارند.

برنامه نویسی HMI :

برنامه نویسی HMI با اکثر زبان‌های برنامه نویسی دیگر متفاوت است. دلیل آن این است که HMI یک نمایش بصری از آنچه در طبقه تولید اتفاق می‌افتد است. بنابراین، برنامه‌نویسی واقعی HMI معمولاً بعنوان توسعه HMI شناخته می‌شود، زیرا بیشتر زمان صرف طراحی صفحه‌نمایش‌ها می‌شود تا کدنویسی به معنای سنتی تعریف شود. علاوه بر این، برنامه‌نویسی که ورودی و خروجی یک HMI را کنترل می‌کند، عموماً روی PLC قرار دارد و به برنامه‌نویس PLC بیشتر کنترل بر نحوه عملکرد HMI را می‌دهد. با این حال، هر دوی این عملکردها در اکثر امکانات با هم ترکیب می‌شوند و برنامه‌نویس PLC یا طرح‌بندی صفحه‌های HMI را ایجاد می‌کند یا به اندازه کافی با فرآیند آشنایی دارد تا نحوه اجرای برنامه‌نویسی HMI را تعیین کند.

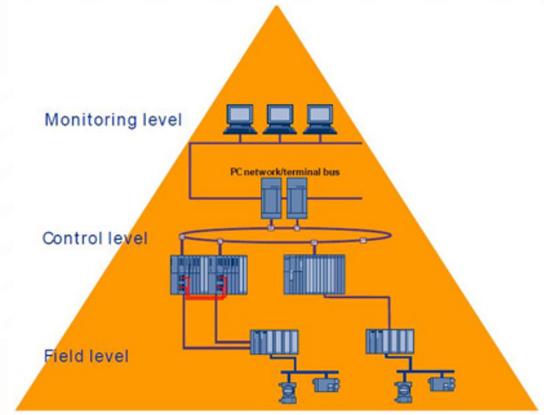
ابتدایی ترین HMI به اپراتور اجازه می‌دهد تا وضعیت فعلی یک فرآیند خاص را ببیند. برای لحظه ای تصور کنید که یک ماشین سنگ زنی دارید که می‌توانید با فشار دادن هر یک از دکمه‌ها شروع به کار کنید و متوقف کنید. یک HMI می‌تواند ایجاد شود تا یک نشانه بصری از وضعیت فعلی دستگاه ارائه دهد: متوقف یا در حال کار. با این حال، یک PLC بسته به نیاز عملیات می‌تواند اطلاعات بسیار بیشتری را از این دستگاه استخراج کند. بنابراین، HMI می‌تواند برای انتقال این اطلاعات به اپراتور مورد استفاده قرار گیرد و به او اجازه می‌دهد تا تصمیمات بهتری در رابطه با فرآیند بگیرد.

تصور کنید که همان ماشین سنگ زنی قادر به خراب شدن است. ممکن است بدلیل عدم نگهداری (سطح پایین روغن)، خرابی پیش بینی نشده (گیر در ورودی)، یا اشتباه اپراتور (درب ایمنی باز) متوقف شود. هر یک از این حالت‌های خطا توسط یک LED مستقیماً روی تابلو برق نمایش داده می‌شود. از آنجایی که ما این اطلاعات را در دسترس داریم، ممکن است انتخاب کنیم که آن را به یک HMI اضافه کنیم تا مکانیزم بازخورد دانه‌ای تری به اپراتور ارائه دهیم که خطای ذاتی را از بین می‌برد.

در نهایت، یک HMI می‌تواند شامل ویژگی‌های پیشرفته فرآیند مانند کنترل دسته‌ای، مدیریت دستور غذا، وضعیت خط و بسیاری موارد دیگر باشد. همانطور که HMIها قدرتمندتر شدند، از کنترل یک ماشین واحد به طرح‌های کنترلی در سطح کارخانه تبدیل شدند که گاهی اوقات بعنوان سیستم‌های کنترل نظارتی و جمع آوری داده یا SCADA توصیف می‌شوند. یک خط خاکستری وجود دارد که در آن یک HMI یک سیستم SCADA در نظر گرفته می‌شود، اما برای درک شما، و HMI یک خط تولید واحد را کنترل می‌کند. در مقابل، یک سیستم SCADA بر کل منطقه یا کل کارخانه تولید نظارت می‌کند.

در ادامه نیز هنگامی که با سیستم کنترلی Scada و PLC بیشتر اشنا شویم میتوانیم به تفاوت های HMI با انها بیشتر از قبل متوجه شویم.

سیستم DCS **طاهره شایگانی**



سیستم کنترل توزیع شده[1] (DCS) همانطور که ازاسمش پید است، به دلیل خاصیت توزیع پذیری تصور آن نسبت به بقیه سیستم‌ها سخت‌تر است. درواقع این سیستم دارای طراحی ویژه‌ای که متشکل از المان‌های کنترلی توزیع‌شده از نظرجغرافیایی درکارخانه یاحوزه کنترلی است. به زبان ساده این نوع سیستم یک نوع روش پیاده سازی سیستم کنترلی است نه یک تجهیز خاص . هر المان پردازشی ، ماشین‌های الکتریکی یا هرگروهی از ماشین‌ها به المان وسیله یک کنترل کننده مخصوص کنترل می‌شود. سیستم کنترل توزیع‌شده در اصل یعنی هر سیستمی که متشکل از تعداد زیادی کنترل کننده محلی است و ازطریق این شبکه‌ها ارتباطی مانند:

- شبکه کنترل[2]
- شبکه دستگاه[3]

به یکدیگر متصل می‌شوند .

این سیستم در مکانی کاربرد دارد که سیگنال‌ها به صورت آنالوگ هستند.از آنجایی که در این سیستم همه چیز به صورت آنالوگ قرار دارد (زیرا سیستم کنترلی بین چند کنترل کننده دریک اتاق کنترل قرارمی‌گیرد به آن سیستم کنترلی گفته می‌شود)به راحتی می‌توانیم آن را توسعه دهیم و یک تجهیز جدید بدون هیچ مشکلی به آن افزود.

مهم ترین کاربرد این سیستم این است که هرجا شاهد مقدار زیادی ورودی و خروجی در یک سیستم باشم می‌توانیم از این سیستم استفاده کنیم مثل نیروگاه های تولید برق و تولید مواد شیمیایی صنایع نفت گاز و تصویه آب و فاضلاب .

به طور تخصصی تر این سیستم مجموعه ای از کنترلرها با قابلیت پردازش بیش از یک حلقه است،که این کنترلرها باهم در ارتباط‌اند و می‌توانند به وسیله ورودی و خروجی‌ها بین ۱۰ تا ۱۰۰ حلقه را کنترل کنند.

اجزای این سیستم به صورت زیر می‌باشد.

- سطح زمینه[4]
- سطح کنترل [5]
- سطح نظارت[6]

نحوه عملکرد سیستم :

در این سیستم سنسورها برای دریافت داده‌ها و پردازش اطلاعات و انتقال داده‌ها به ماژول‌های ورودی وخروجی محلی استفاده می‌شوند. البته درآنجا نیز دستگاه‌های راه انداز به این محرک‌های ورودی/خروجی متصل می‌شوند. با وجود این اتصال پارامترهای پردازشی کاملا تحت مدیریت قرار می‌گیرند. در انجا داده‌های دریافت شده جمع آوری می‌گردند و از طریق پروتکل صنعتی فیلدباس به بخش کنترل پردازش ارسال شده، سپس داده‌های جمع آوری شده مجددا پردازش شده و مورد ارزیابی قرار می‌گردند. با توجه به منطق کنترلی که درکنترل کننده‌ها اجرا می‌شوند نتایج را ایجاد می‌کنند.

ساختار و معماری سیستم:

سطح۱_سطح میدانی [7]

تجهیزات میدانی مانند اجزای کنترلی نهایی مثل دریچه‌ها ، سنسورها ، گیج ها، موتورها و سوئیچ‌ها ... در این سطح گنجانده شده‌اند. ارتباط این سطح و سطح ۱ می‌تواند تقریبا از هر نوعی باشد. این موارد شامل اترنت صنعتی ، فیبر نوری و یا سایر پروتکل‌های ارتباطی اختصاصی باشد.

سطح۲_سطح میدانی [8]

تجهیزات میدانی مانند اجزای کنترلی نهایی مثل دریچه‌ها ، سنسورها ، گیج ها، موتورها و سوئیچ‌ها ... در این سطح گنجانده شده‌اند. ارتباط این سطح و سطح ۱ می‌تواند تقریبا از هر نوعی باشد. این موارد شامل اترنت صنعتی ، فیبر نوری و یا سایر پروتکل‌های ارتباطی اختصاصی باشد.

سطح۳_کنترل تولید[9]

این سطح که به نام مدیریت تولید شناخته می‌شود، به طور مستقیم به کنترل پردازش مرتبط نیست اما درر بررسی تولیدات و نظارت براهداف دخیل است. این سطح ممکن است شامل سرورها و کامپیوترهای بایگانی و ایستگاه‌های مهندسی باشد. ارتباط با سطح بالاتر(٤) معمولا اترنت صنعتی است. سرورها برای جمع‌آوری داده در سطح پردازنده استفاده می‌شوند و همچنین مسئول داده‌هایی هستند که بین سطوح ۲و٤ درکارخانه درجریان‌اند.

از کامپیوترهای بایگانی برای ذخیره داده‌های قدیمی استفاده می‌شود، که ممکن است برای پروژه‌های جدید نیز به کار برده شوند. ایستگاه‌های مهندسی برای ایجاد پروژه‌هایی که فرایند روی آنها انجام می‌گیرد استفاده می‌شوند.

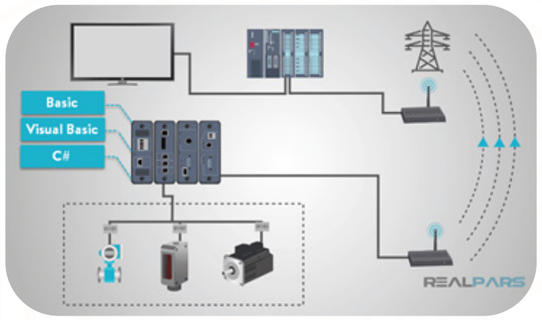
این امر شامل:

تنظیمات سخت افزاری_نمایش‌های گرافیکی برای تعامل با اپراتورها_مدیریت تمامی وظایف از طریق بسته‌های نرم افزاری نصب شده می‌شود.

سطح٤_زمان‌بندی تولید[10]

این سطح برنامه‌ریزی تولید نام دارد، که برنامه‌ریزی‌های اصلی وتعیین زمان‌بندی‌ها در فاز کلان، در این سطح انجام می‌شود.

سیستم RTU



پیشرفت تکنولوژی ارتباطات، راه های زیادی مانند مدباس RTU، به منظور اندازه گیری ،کنترل و مدیریت ایستگاه‌های راه دور طراحی و اجرا شدند. این سیستم در تمامی نیروگاه‌های کشور و در قسمت‌هایی از شرکت نفت، گاز، کانال‌ها و مخازن آب و فاضلاب، مترو و ...نصب می‌شود و باعث افزایش بهره وری در کل آن سیستم است. جمع آوری دیتا از مکان‌های موردنظر و ارسال آن‌ها به مرکز کنترل، خرابی‌های احتمالی در این نقاط را فوری تشخیص می‌دهد. به همین خاطر می‌توان گفت RTU در کاهش خطرات زیان بار نقش دارد. در این سیستم اطلاعات از یک نقطه کاری به صورت منقطع دریافت و جمع آوری می‌شود و سپس به صورت یکجا به مرکز کنترل تعریف شده ارسال می‌گردد.

RTU مخفف چیست؟

واحد پایانه کنترل از راه دور[11] (RTU) یا همان دیسپاچینگ است، که اطلاعات را جمع آوری می‌کند و جهت تحلیل به مرکز کنترل ارسال می‌کند. به علت استفاده از بردهای میکروپروسسوری یا ریزپردازنده‌ها در این پایانه، ورودی‌های RTU دردسته بندی‌های مشخصی قرار می گیرند.

RTU چیست؟

پایانه RTU، میکروپروسسوری است، که به منظور جمع‌آوری و پردازش اطلاعات درمکان‌های مختلف از آن استفاده می‌شود. مدباس RTU اطلاعات را از طریق سیستم‌های مخابراتی به ایستگاه‌های فیبرنوری ، ارسال می‌کند. همچنین RTU فرمان‌هایی را که از مرکز کنترل به پست ها ارسال می‌شود را هم دسته‌بندی کرده و به مکان مورد نظر ارسال می‌کند. تنوع داده‌های قابل پردازش درRTU زیاد است. ماژولRTU در نقاط مختلفی مثل کلیدها، سکسیونرها و آلارم‌ها و... ورودی‌های قابل اندازه گیری را دریافت کرده و فرمان‌های کنترلی مثل خاموش و روشن کردن، ارسال فرمان و... را به سیستم باز می‌گردانند.

ورودی‌های دیجیتالی مثل وضعیت کلیدها و یا قطع کننده‌های مربوط به تجهیزات ایستگاه‌های فشارقوی، وضعیت فشار روغن و سایز هشدارهای داخل پایانه می‌باشد. تفسیر و پردازش ورودی‌های دیجیتال در RTU متفاوت است، مانند:

- ورودی‌های دیجیتال یک بیتی
- ورودی‌های دیجتال دو بیتی
- ورودی‌های دیجیتال با ارزش بیش از دو بیتی
- ورودی‌های شمارنده پالس

طبیعتا خروجی‌های دیجیتال و آنالوگ فرمان‌های ارسالی به تجهیزات هم متفاوت هستند؛ مانند:

Trip / Close: به منظور تغییر کلیدها و یا قطع‌کننده سیستم‌ها به کار می‌رود.

lower / Raise: جهت تغییر زبانه ترانسفورماتورها و به صورت پالسی با زمان‌های متغییر؛ مورد استفاده قرارمی‌گیرد.

OF/ON : این فرمان، خروجی مورد نظر در DOTB را تا زمانی

1 Distributed Control System
2 ontrol Net
3 Device Net
4 Field Levels
5 Control Level
6 Monitoring Level
7 Plant
8 Plant
9 Production Control
10 Production Scheduling
11 Remote Terminal Unit

که فرمان جدیدی اعمال نشود، در دستور کار قرار می‌دهد. البته با توجه به قابل اطمینان نبودن، به ندرت استفاده می‌شود.

راه‌های ارتباط RTU به ایستگاه مرکزی:

درگاه واسط LAN، به کمک تجهیزات جانبی دیگر مانند مودم ADSL، فیبرنوری، مودم وایمکس، WiFi بی‌سیم، آنتن بشقاب و آنتن ماهواره‌ای، مدباس RTU را به ایستگاه مرکزی متصل می‌کند. معمولا نرم‌افزار جامعی برای هماهنگ کردن RTU با ایستگاه مرکزی وجود ندارد، اما به دلیل استفاده‌های متفاوت ماژولRTU در صنایع مختلف باعث شده که برنامه نویسان، نرم‌افزارهایی تخصصی در این زمینه طراحی و اجرا کنند. تیم آرشین نیز راهکارهای تخصصی خود را در ا ین زمینه به منظور کنترل و مانیتورینگ فرآیندها پیاده سازی نموده است. بنابراین RTU تنها در صنایع بزرگ و در اماکن خاص نظیر آب و تصفیه خانه‌ها استفاده نمی‌شود و کارایی بیشتری دارد.

دیاگرام سخت افزاری RTU به چه شکل است؟

RTU شامل چند پردازنده[12] (CPU) می باشد و به صورت سیستم های توزیع DO، DI، AI نمایش داده می‌شوند. واحد AI به صورت مستقل تا ۹۶ ورودی آنالوگ را دریافت و پردازش می‌کند. واحد DI نیز به شکل مجزا، تا ۱۹۲ سیگنال وضعیت کلیدها و اطلاعات وضعیتی دیگری مانند هشدارها را دریافت، پردازش و ارسال می نماید. واحد DO، هم می‌تواند ۶۴ عمل کنترل خروجی را انجام دهد. هر یک از این واحدها اطلاعات را به صورت لحظه‌ای دریافت و ذخیره می‌کنند. کنتاکت‌ها نیز بخش دیگری از سخت افزارRTU می‌باشند، که به دو نوع خشک و تر تقسیم می‌شوند.

۱. کنتاکت‌های نوع خشک: این کنتاکت‌ها مستقیما به RTU متصل می‌گردد و بدون ولتاژ هستند.

۲. کنتاکت‌های نوع تر: این کنتاکت‌ها از طریق منبع تغذیه به کارت‌های ترمینالی پایانه RTUمتصل می‌گردد و دارای ولتاژ می‌باشند.

از RTU جهت جمع آوری اطلاعات از حسگرها بهره گرفته‌شده و سپس داده را به IMTUارسال می‌نماید. ظرفیت ذخیره‌سازی RTU بسیار بالاست و به راحتی داده‌ها را ذخیره و منتقل می‌کند. اخیرا از RTU در واحدهای توسعه یافته سیستم‌های پیشرفته استفاده می‌گردد.از RTU در ایستگاه‌هایی که ارتباطات، مشکل‌تر است، استفاده می‌شود.

RTU در محل‌هایی با فاصله زیاد استفاده می‌شود و همچنین در محل‌های حساس مانند ایستگاه‌های پمپاژ و تانکرهای ذخیره سازی مورد استفاده قرار می‌گیرد.

نمونه‌گیری رادیویی ارتباطات RTU در یک شبکه گسترده باعث می‌شود، که در اندازه گیری‌های امنیتی با فعالیت‌ها هماهنگ باشد. RTUها از ارتباط بی‌سیم استفاده می‌کنند به همین دلیل برای محیط‌های بزرگ مناسب‌تر هستند.

سیستم IED

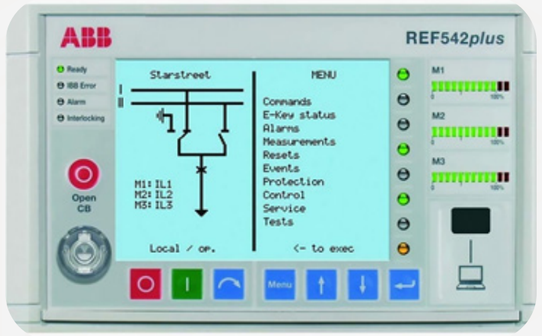
محیا مرادی

در دهه ۷۰ ميلادي، با پيدايش ميكرو پروسسورها، سازندگان تجهيزات (پستها) سعي كردند وسايل الكترومكانيكي را با وسايل نيمه‌هادي مجهز به ميكروپروسسور جايگزين كنند.

اين وسايل در صنعت به نام وسايل الكترونيكي هوشمند[13] (IED) شناخته شدند. IED قابليتها وتوانايي‌هاي اضافي به وسايل افزودند، نظير تشخيص خطا وچك كردن خودشان، داشتن رابط‌هاي مخابراتي و قابليت ذخيره داده‌ها و وقايع سيستم. همچنين IED ها باعث شدند تا وسايل تكراري، حذف شوند چون قابليت چندكار را داشتند. مجتمع كردن سيستم كنترل ايستگاهي به هم پيوستن تمامIED ها به يك سيستم كنترل مجتمع پست (ISCS) باعث كم شدن هزينه سيم كشي ،ارتباط ، نگهداري و بهره برداري مي‌شود و كيفيت برق و قابليت اطمينان آن را افزايش مي‌دهد. موضوع مهمي كه در مجتمع كردن IED در يك

12 Central Processing Unit

13 Intelligent Electronic Device



سيستم كنترل دستگاهي بايد مورد توجه قرارگيرد اين است كه بسياري ازIED ها تنها داراي يك پورت ارتباطي هستند و موقع ارسال فرمان توسط كاربر يا عامل به IED ، داده‌هاي ديگر براي IED قابل دسترس نيستند. اين وضعيت براي حالتي كه اين داده‌ها براي عمليات زمان حاضر لازم باشند، يك وضعيت بحراني است. سيستم بايد بتواند اين شرايط را تشخيص داده و به ديگر عاملان سيستم اعلام كند.درحال حاضر بسياري از سازندگان IEDمحصولات خود را با دو پورت (ورودي - خروجي) توليد مي‌كنند تا ازاين مشكل جلوگيري شود. كنترل از راه دور ايستگاه‌ها و تجهيزات از دهه ۱۹۶۰ شروع شد و در حدود دهه ۷۰ ، جايگزيني وسايل الكترومكانيكي با ابزارهاي نيمه‌هادي در مرحله ابتدايي و مقدماتي بود. يك طرح اتوماسيون پست، قبل از دهه ۹۰ به طور معمول شامل سه ناحيه عملياتي اصلي بود.كنترل نظارتي و جمع آوري داده‌ها (SCADA) كنترل پست شامل اندازه‌گيري و نمايش، حفاظت تجهيزات اتوماسيون مورد استفاده در هر يك از نواحي به طور عمده شامل وسايل الكترومكانيكي نظير وسايل اندازه‌گيري، رله‌ها و وسايل حفاظت، زمان سنج‌ها، شمارنده‌ها و وسايل نمايش آنالوگ و ديجيتال بود. سيستم‌هاي آنالوگ و ديجيتال اطلاعات اين سيستم‌ها را در محل وسايل و يا روي پانل‌هاي مدل سيستم نمايش مي‌دهند. همچنين در اين پانل‌ها سوئيچ‌هاي الكترومكانيكي قرار داشت كه اپراتورهاي پست براي كنترل براي نمايش تجهيزات مربوط به هر يك ازوسايل اوليه داخلي پست استفاده مي‌كردند. معمولا براى نمايش تجهيزات مربوط به هريک ازسه ناحيه عمليات اصلي قسمتي از پانل كنترل اختصاص داده شده‌بود.

با ظهور ريزپردازنده‌ها دردهه ۷۰ ، شرايط عوض شد. سازندگان تجهيزات پست‌ها جايگزيني وسايل الكترومكانيكي ساخت خود را با وسايل نيمه‌هادي شروع كردند. اين وسايل مبتني بر ريزپردازنده كه بعداً در صنعت به وسايل الكترونيكي هوشمند معروف شدند، مزاياي بسياری نسبت به وسايل قديمي داشتند. آن‌ها قابليت‌هاي اضافي نظير تشخيص خطا، خود چك كردن، توانايي ذخيره داده‌ها و ثبت وقايع، رابط‌هاي مخابراتي و واحد ورودي خروجي مجتمع با قابليت كنترل از راه دور داشتند. همچنين به خاطر اينكه چندين قابليت را مي توان در يك IEDفشرده ساخت، مي‌توان وسايل جانبي را حذف كرد. براي مثال، وقتي IED به يك ترانسفورماتور ولتاژ و جريان در مدار وصل است، اين وسيله مي‌تواند همزمان وظيفه حفاظت، اندازه گيري و كنترل از راه دور را به عهده بگيرد.

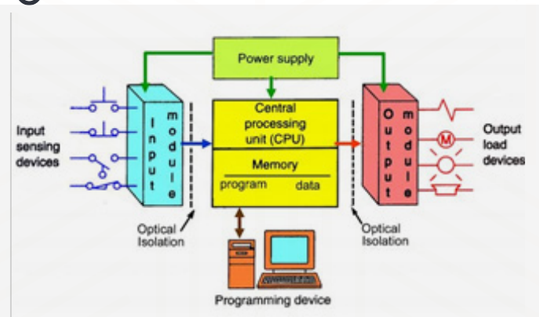
از امتيازات جالب توجه IED قابليت اطمينان، راحتي نگهداري و سرعت مشكل دهي و پيكربندي سيستم است.

در دهه ۷۰ و اوايل دهه ۸۰ كه اين وسايل عرضه شدند، به خاطر شك و ترديد در موردقابليت اطمينان آن‌ها و همچنين هزينه زياد، از آنها استقبال نشد. اما با كمتر شدن قيمت و پيشرفت در قابليت اطمينان و اضافه شدن قابليت‌ها، آنها بيشتر مورد پذيرش قرار گرفتند .

IEDاولين سطح فشرده سازي اتوماسيون است. اما حتي با استفاده گسترده از آن نيز تنها جزيره‌هايي از اتوماسيون در بين پست‌هاي مختلف پراكنده مي‌شوند. صرفه جويي بيشتر موقعي حاصل مي‌شود كه تمامIED ها در يك سيستم كنترل ايستگاه‌هاي متمركز قرار گيرند. تحقق سيستم‌هاي كنترل كاملا مجتمع، هزينه‌هاي سيم كشي، تعمير و نگهداري، مخابراتي وعملياتي را كاهش و كيفيت برق و قابليت اطمينان سيستم را افزايش مي‌دهد.

اگر چه اين مزايا ارزشمند است اما مجتمع كردن سيستم اتوماسيون ايستگاه‌ها (مثالً درآمريكاي شمالي) پيشرفت كمي داشته است و دليل عمده آن اين است رابط‌هاي سخت افزاري و پروتكل‌ها براي IED استاندارد نيستند. تعداد پروتكل‌ها برابر تعداد سازندگان وسايل و يا بلكه بيشتر، به خاطر اينكه توليدات يك كارخانه نيز اغلب پروتكل‌هاي مختلفي دارند.

يك راه حل براي اين مشكل نصب و برقراري يك gateway است، كه به عنوان يك سخت افزار ورابط پروتكل بين IED و يك شبكه عمل مي‌كند. gateway به شركت برق اجازه مي‌دهد تا با اجزاي يك شبكه و پروتكل ارتباطي مشترك، وسايل مختلف را با هم روي يك ايستگاه مجتمع كند. gateway به يك رابط فيزيكي بين IED و استانداردهاي الكتريكي شبكه و همچنين به يك مبدل پروتكل بين آنها است.



gatewayباعث مي‌شود تمامIED ها ازديدگاه شبكه مورد استفاده در پست، از نظر ارتباطي يكسان به نظر برسند.

از آن جا اندازه گيري‌هاي PML و يك PLC نوع Modicon را در سيستم كنترلي ايستگاهي خود نيز كار را پيچيده‌تر و مشكل‌تر كرده‌است. براي مثال ممكن است يك شركت بخواهد تعدادي رله حفاظت از نوع DEL ، رله‌هاي حفاظت فيدر از نوع ABB ، مونيتورهاي با كيفيت بالاي Multilim GE فرمت ASCLL كه توسط SEL پشتيباني مي‌شود، استفاده مي‌كند. رله‌هاي ABB و GE پروتكل 3.00DNP را مورد استفاده قرار مي‌دهند و اندازه گيري‌هاي PML نيز از همين پروتكل استفاده مي‌كنند. در حالي كه PLC براي ارتباط از پروتكل Modbus كه Modicon تهيه كرده‌است، استفاده مي‌كند. براي داشتن تمام ای IEDها و پروتكل‌هاي نامتجانس آن‌ها روي يك سكوي كامپيوتري، استفاده از درگاه بهترين راه حل است.

درگاه نه تنها به عنوان يك رابطه بين لايه فيزيكي شبكه محلي و درگاه‌هاي 485RS232/RS كه رويIED ها هستند عمل مي‌كند بلكه به عنوان يك مبدل پروتكل، پروتكل‌هاي خاص هر IEDرا مانند 3.0DNP SEL يا (Modbus) به پروتكل استاندارد مورد استفاده شبكه محلي نصب شده ترجمه مي‌كنند.

سیستم PLC

رضا صلح میرزایی

PLC در لاتين مخفف Programmable Logic Controller است كه به معني كنترل كننده منطقی قابل برنامه‌ريزی است. يعنی يک قطعه كنترل كننده كه برنامه پذير است و می‌توان برنامه‌های مختلف و متنوع كنترلی را بر روی آن نوشت. اما در اصل PLC چيست و چه كاربردی دارد؟اما قبل از ورود دقيقتر به پی ال سی، بهتر است نگاهی به كنترل در صنعت پيش از ظهور PLCها بيندازيم.

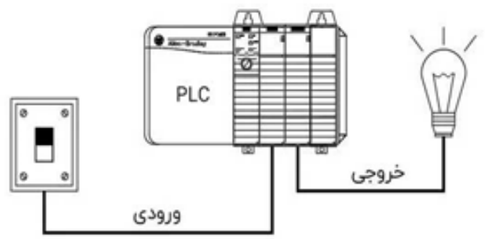
عمده سيستم‌های كنترلی كه در صنعت استفاده می‌شد، به صورت رله و بانک‌هايی از رله بود. جايی كه مجموعه‌ای از رله‌ها، وظيفه كنترل و خاموش/روشن كردن دستگاه‌های خروجی را به عهده داشتند. در برخی از اين سيستم‌ها تعداد رله‌ها اينقدر زياد می‌شد كه يک محفظه شامل رله و بانک رله تشكيل می‌داد.برخی از مشكلات بزرگ اين نوع سيستم ها ذاتاً به مكانيكی بودن رله‌ها برمی‌گشت. استهلاک در طول زمان بر اثر كليدزنی‌های فراوان باعث خرابی رله‌ها می‌شد.

يک عيب بزرگ ديگر اين سيستم‌ها اين بود كه اگر نقصی در سيستم پيش بيايد، تشخيص محل دقيق آن زمانبر و مشكل است. به عبارت ديگر در زمان نقص، دستگاهها در مدت زمان قابل توجهی بايد كاركرد خود را متوقف كنند تا مكان عيب تشخيص داده شده و رفع گردد.به همين علت افراد و شركت‌های مختلفی، مدتها روی سيستمی كار كردند كه اين مشكلات را برطرف كند؛ و در نهايت پی ال سی به عنوان جايگزين اين سيستم ارائه شد.

مهمترین مزیت PLC چیست؟

می‌توان گفت مهمترين نقطه قوت پی ال سی در مقابل سيستم‌های كنترل قديمی در اين است كه شما می‌توانيد پس از اينكه طرح مدنظر خود را برنامه‌ريزی كرديد، برگرديد و مجدداً برنامه و طرح خود را تغيير دهيد. اينكار هزينه كمی نسبت به سيستم‌های قديمی دارد و اين هزينه فقط در زمانی است كه برنامه‌نويس بايد برای تغيير برنامه خود صرف كند.

لامپی را در نظر بگيريد كه به يک كليد متصل شده است. لامپ دو حالت بيشتر ندارد، يا روشن است و يا خاموش:



شكل۳- ۱

به شما گفته شده است كه كاری كنيد تا وقتی كليد در حال وصل (ON) قرار گرفت، پس از ۳۰ ثانيه لامپ روشن شود. مسلماً در اين ساختار سيمی، پياده سازی چنين چيزی بسيار دشوار است.تنها راهی كه شايد بتوانيد به اين هدف دست پيدا كنيد اين است كه سيم‌كشی مدار خود را تا جايی زياد و پيچيده‌تر كنيد كه بتوانيد به اين تاخير مد نظر دست پيدا كنيد. البته باز هم در عمل اين كار بسيار دشوار است! بله برای يک تغيير جزيی، به دردسرهای بزرگی می‌افتيم.و اينجاست كه نقش كنترل كننده‌های منطقی قابل برنامه‌ريزی (PLC) مشهود و پررنگ می‌شود. به طوری كه نياز به هيچ سيم‌كشی اضافه‌ای در سيستم نداريم.تنها كافيست چند خط كد ساده به برنامه اضافه كنيم؛ با اين كاركرد كه كنترلر ۳۰ ثانيه تاخير در روشن كردن لامپ پس از فشردن كليد بيندازد. گذشته از اين با حضور پی ال سی، اكنون شما می‌توانيد چندين ورودی مختلف را دريافت كرده و در سمت خروجی نيز چندين دستگاه خروجی متفاوت را كنترل نماييد.

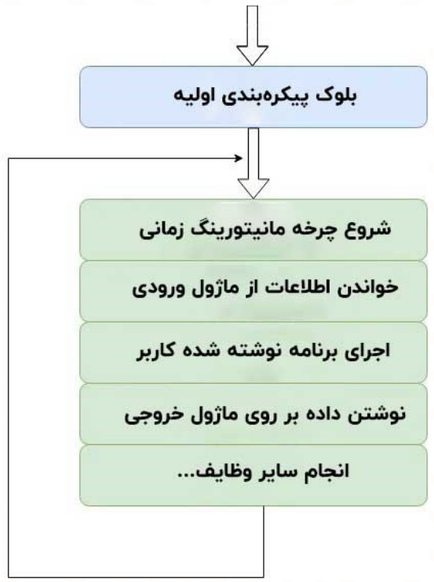
و اين مثال تنها يک نمونه بسيار ساده از كاركرد پی ال سی بود. يک PLC توانايی كنترل و اجرای فرآيندهای بسيار پيچيده‌تری را دارد.

نحوه عملکرد PLC :

در ابتدا و قبل از هر كاری بايد پيكره‌بندی اوليه PLC انجام شود. شناساندن نوع و شركت سازنده پی ال سی به نرم‌افزار، تعيين پورت‌های ورودی-خروجی و ... كه همه اين كارها معمولاً به صورت نرم‌افزاری انجام می‌شود.

حال به شرح مراحل مختلف اين چرخه می‌پردازيم:

- سيستم‌عامل در ابتدا رصد زمانی خود را شروع می‌كند و وارد اين چرخه می‌شود.
- CPU شروع به خواندن داده‌ها از ماژول ورودی می‌كند و وضعيت تمام ورودی‌ها را بررسی می‌كند.
- CPU در گام بعدی شروع به اجرای برنامه نوشته شده كاربر می‌كند. اين برنامه می‌تواند در منطق لدر (ladder) باشد و يا با هر زبان برنامه‌نويسی ديگر كه مبتنی بر PLC است، باشد.
- سپس CPU به سازوكارهای داخلی و وظايف ارتباطی خود می‌پردازد.
- بر اساس نتيجه برنامه اجرا شده، پردازنده آن داده را در ماژول خروجی می‌نويسد تا وضعيت تمام خروجی‌ها به روز شود.
- مراحل اين فرآيند ادامه پيدا می‌كند تا زمانی كه PLC در حالت اجرايی و عملكردی خودش قرار دارد.



شكل۳- ۲چرخه عملكرد PLC

ساختار PLC :

ساختار داخلی PLC به نحوی شبیه به ساختار کامپیوترها است. مهمترین بخش‌های درونی پی ال سی به شرح زیر است:

- ماژول منبع تغذیه
- واحد پردازشگر مرکزی (CPU)
- ماژول ورودی
- ماژول خروجی

ماژول منبع تغذیه:

ماژول منبع تغذیه وظیفه تامین برق و تغذیه مورد نیاز تمام بخشهای سیستم را به عهده دارد. این ماژول برق ورودی متناوب (AC) را به برق جریان مستقیم (DC) تبدیل می‌کند که مورد نیاز بخش‌هایی مثل CPU و ماژولهای ورودی و خروجی است.تغذیه اصلی اکثر پی ال سی ها، ۲۴ ولت دی سی است. PLCهای کمی هستند که نیاز به ولتاژی جز این مقدار داشته باشند.

ماژول CPU و حافظه:

ماژول CPU را می‌توان به نوعی مغز PLC دانست که وظیفه اجرای برنامه‌ها، و مدیریت وظایف محول شده به PLC را بر عهده دارد.ماژول سی پی یو یا همان پردازشگر مرکزی، شامل یک پردازنده اصلی و حافظه‌های ROM و RAM است. حافظه فقط خواندنی (ROM) شامل سیستم عامل، درایورهای مورد نیاز و برنامه‌های کاربردی است.از طرف دیگر حافظه دسترسی تصادفی (RAM) به منظور ذخیره برنامه‌ها و داده‌ها بر روی خود استفاده می‌شود.اگر بخواهیم این ماژول را با واحدهای سخت‌افزاری جایگزین کنیم، می‌توانیم آن را معادل مجموعه‌ای از رله‌ها، تایمرها و شمارنده‌های متصل به همدیگر در نظر بگیریم.اما دو نوع پردازنده معمولاً در PLC ها استفاده می‌شود:

۱. پردازنده یک بیتی

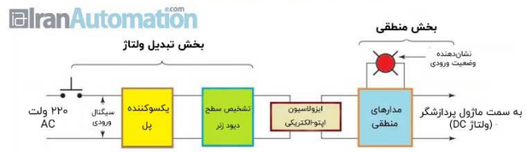
۲. پردازنده کلمه‌ای

پردازنده یک بیتی، برای کاربردهای ساده‌تر و منطقی (لاجیک) استفاده می‌شود. جایی که یک بیت برای داده به عنوان پردازش کافی باشد. ولی پردازنده کلمه‌ای، برای کاربردهای پیچیده‌تر مثل پردازش متن، داده‌های محاسباتی، کنترلی و...

به کار گرفته می‌شود.

به طور پیوسته، وضعیت مقادیر ورودی از دستگاه‌های متعدد ورودی را مانیتور می‌کند (دستگاه‌هایی مثل شتاب‌سنج، دماسنج، سنسورها و...)؛ داده ورودی تطبیق یافته را از ماژول ورودی می‌خواند و سپس بر اساس منطق نوشته شده در خود، خروجی مورد نظر را ایجاد کرده و به ماژول خروجی ارسال می‌کنند. این خروجی می‌تواند سیگنالی جهت روشن شدن موتور یا موارد مشابه باشد.

بلوک دیاگرام ماژول ورودی PLC



ماژول ورودی PLC :

اِعمال سیگنالهای فیزیکی نظیر دما، فشار، ارتعاش و... به پردازنده به دلیل ماهیتِ غیردیجیتال این سیگنالها، امکان‌پذیر نیست. ماژول CPU فقط منطق دیجیتال می‌فهمد و هر چیزی غیر از آن را نمی‌تواند درک کند!به همین دلیل ماژول‌های ورودی و خروجی طراحی شده‌اند تا بتوانند منطق دنیای بیرونِ PLC را به پردازشگر آن بفمانند.پس در حقیقت یکی از کاربردهای مهم ماژول ورودی و خروجی، تطبیق نوع سیگنال‌های دستگاههای ورودی با پی ال سی است. مثلاً اگر قرار باشد وضعیت دماسنج بررسی شده و دمای اعلامی آن به پی ال سی ارسال شود، نوع سیگنال آن باید از نوع آنالوگ به دیجیتال تبدیل شود.مبدل آنالوگ به دیجیتال در ماژول ورودی قرار دارد و سیگنال دماسنج را قابل فهم و پردازش برای CPU می‌کند. اساساً یکی از مهمترین مزایای استفاده از ماژول‌های ورودی و خروجی ایجاد ایزولاسیون بین پی ال سی و دنیای بیرون است. قرار نیست مشکلات احتمالی (نویز، سیگنالهای ناخواسته و...) که در خارج از کنترلر وجود دارد، روی آن اثری بگذارد. در تصویر زیر می‌توانید ساختار ساده‌ای از ماژول‌های ورودی را ملاحظه کنید.

پس ماژول ورودی PLC، چهار وظیفه اصلی دارد:

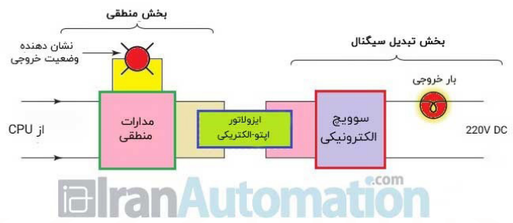
۱. دریافت سیگنالها از دستگاههای ورودی در سطح ولتاژ ۲۲۰ ولت AC

۲. تبدیل سیگنالهای ورودی به سطح ۵ ولت DC که قابل استفاده برای ماژول CPU باشد.

۳. ایزوله کردن PLC از نوسانات دستگاههای ورودی

۴. ارسال سیگنال نهایی با سطح ولتاژ ۵ ولت DC به خروجی (ماژول پردازشگر)

ماژول خروجی



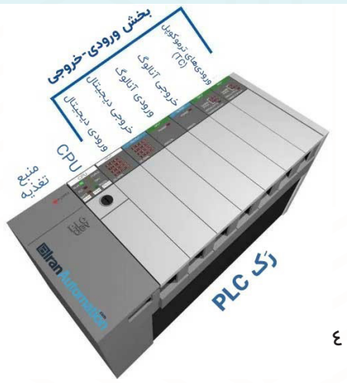
شکل۳- ۳

ماژول خروجی PLC:

عملکرد ماژول خروجی پی ال سی نیز مشابه ماژول ورودی است که به صورت عکس آن عمل می‌کند. این ماژول با دو بخش پردازنده و بار خروجی در ارتباط است. در ابتدا بخش مدار منطقی را داریم و سپس بخش مربوط به تبدیل سطح ولتاژ به دنبال آن می‌آید.

در این ساختار، وقتی که سیگنال منطقی ۱ یا High از سمت CPU به این ماژول ارسال می‌شود، LED روشن شده و نور آن به فتوترانزیستور موجود در ایزولاتور تابیده و باعث می‌شود تا این ترانزیستور در ناحیه هدایت خود قرار بگیرد. سپس یک پالس به گیت ترایاک (که در بخش سوویچ الکترونیکی) وجود دارد ارسال می‌شود. و در نهایت برق AC برای بار خروجی آماده می‌شود.

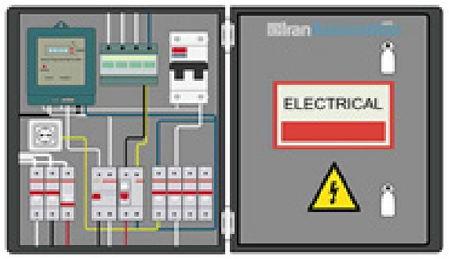
در برخی از ساختارها، رک یا قفسه‌ای که کل سیستم در آن قرار می‌گیرد را نیز بخشی از ساختار پی ال سی می‌دانند. البته رک در پی ال سی‌های ماژولار کاربرد دارد.



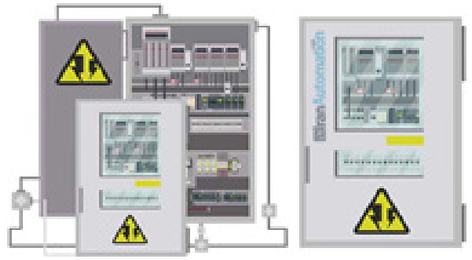
شکل۳- ٤

در این ساختار، وقتی که سیگنال منطقی ۱ یا High از سمت CPU به این ماژول ارسال می‌شود، LED روشن شده و نور آن به فتوترانزیستور موجود در ایزولاتور تابیده و باعث می‌شود تا این ترانزیستور در ناحیه هدایت خود قرار بگیرد. سپس یک پالس به گیت ترایاک (که در بخش سوویچ الکترونیکی) وجود دارد ارسال می‌شود. و در نهایت برق AC برای بار خروجی آماده می‌شود.

در برخی از ساختارها، رک یا قفسه‌ای که کل سیستم در آن قرار می‌گیرد را نیز بخشی از ساختار پی ال سی می‌دانند. البته رک در پی ال سی‌های ماژولار کاربرد دارد.



تابلو برق



تابلو PLC

شکل۳- ۵

تابلو PLC:

- تابلو برق یک محفظه است که دارای اجزای برچسب گذاری شده شامل تجهیزات قطع و وصل کردن مثل کلیدها، تجهیزات کنترل، حفاظت، رله‌ها و.... است. این تابلو همچنین اتصالات بین اجزا، متناسب با ساختار درونی تابلو است؛ اگر به این تابلو پی ال سی اضافه کنیم، تابلو برق پی ال سی یا پنلِ کنترل PLC پدید می‌آید. تابلو یا پنل (Panel) پی ال سی می‌تواند هر فرآیند را کنترل کرده و داده‌ها را در هرکجا و به هرصورت که شما به آن نیاز دارید فراهم کند. از مزایای این تابلو می توان به سرعت بالا در عملکرد ، انعطاف‌پذیری

در تغییر منطق برنامه به دلیل نرم‌افزاری بودن ساختار،قابلیت اطمینان بالاتر به دلیل عدم وجود قطعات متحرک ، مصرف برق پایین و ... اشاره کرد.

سازندگان و برندهای PLC:

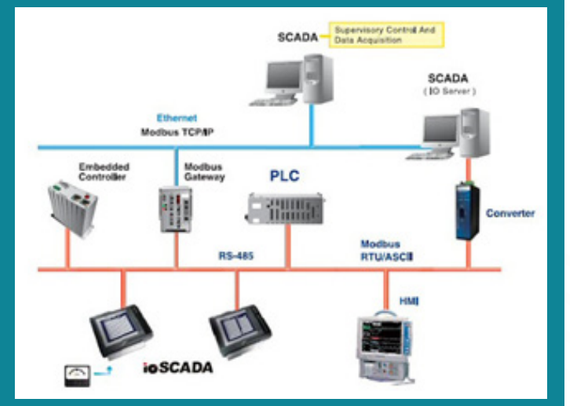
می‌توان گفت تقریباً همه برندهایی که در حوزه اتوماسیون صنعتی فعال هستند، در حال تولید و توسعه PLCهای برند خود هستند. اما برخی از این برندها به خوبی توانسته‌اند گوی رقابت را از بقیه ربوده و پیشتاز این عرصه در اتوماسیون صنعتی شوند. برندهایی مثل زیمنس، میتسوبیشی، ABB و اشنایدر را می‌توانیم از این دست بدانیم. اما بقیه برندها عمدتاً روی ساخت و تولید PLC و سایر تجهیزات اتوماسیون صنعتی تمرکز کرده اند؛ از بین آنها می توان به Fatek، Keyence، Omron، Yokagawa، و حتی محصولات plc دلتا اشاره کرد که در ایران نیز به دلیل قیمت مناسب، پرطرفدار شده است.

نگهداری و تعمیرات PLC:

نگهداری و تعمیر یکی از فرآیندهای حیاتی هر صنعتی است که مسئولیت آن به تیمها یا گروههایی حرفه ای و آموزش دیده در این حوزه واگذار می‌شود.برای اینکه این فرآیند به بهترین شکل ممکن و با کمترین خطا صورت گیرد، از نرم افزار نگهداری و تعمیرات (CMMS) به همین منظور استفاده می‌شود. اما آیا این فرآیند برای PLCها هم پراهمیت است؟اشاره کردیم که پی ال سی، یکی از قطعات حیاتی دنیای اتوماسیون و صنایع هستند. بروز هر گونه خرابی در این تجهیز می‌تواند باعث صدمات جبران ناپذیری شود و هزینه‌های زیادی را تحمیل کند. پس نگهداری صحیح از PLCها و قرار دادن آنها در برنامه نگهداری و تعمیر صنعت و اجرای آن، شما را از بروز اتفاقات ناخوشایند بعدی در امان نگه می‌دارد.

سیستم SCADA

مهدی محمدی



Scada در لاتین مخفف supervisory control and data acquisition” است که در فارسی میتوان ان را معادل «سامانه نظارت،کنترل و جمع آوری داده ها» در نظر گرفت . سیستمی را در نظر بگیرید که قرار است در یک منطقه یا فضای صنعتی، کلیه فرآیندهای صنعتی را در همان مکان و یا از راه دور کنترل کند. این سیستم میخواهد کلیه داده های سنسورها، ولوها، محرکها و تجهیزات صنعتی را به شکل Real-Time (بلادرنگ) جمع آوری کرده، آنها را مانیتور نموده (مانیتورینگ صنعتی) و بر اساس پارامترهای تعریف شده در خود، تصمیمات خاصی را برای کنترل صحیح کلیه بخشها بگیرد. چنین سیستمی به عنوان سیستم اسکادا شناخته می شود . در حالت کلی یک سیستم اسکادا مجموعه ای از بخشهای سخت افزاری و نرم افزاری است که امکان نظارت و کنترل یک سری فرآیندها و اقدامات صنعتی را در اختیار ما میگذارد. همانطور که در بخش های قبل گفته شد نرم افزار HMI یا واسط ماشین و انسان ، در سیستم اسکادا وظیفه دارد تا تعامل با دستگاه های میدانی مانند پمپ ها ، محرک ها ، موتور ها ، سنسور ها و ... را ساده تر کند .سیستم اسکادا یک ویژگی مهم دیگر نیز در بخش نرم افزاری خود دارد و آن نیز، توانایی ثبت دقیق داده ها به منظور نگهداری در تاریخچه داده ها و امکان رجوع آینده در صورت وقوع مشکلات است .

دو بخش مهم و جدا نشدنی سیستم های اسکادا PLC ها و RTU ها هستند . این دوبخش به نوعی پردازشگر هایی هستند که با دیوایس های میدانی مانند پمپ ها و HMI ها ارتباط برقرار میکنند . به گونه ای که داده های ارتباطی از این پردازشگر ها به سمت کامپیوتر های SCADA مسیر یابی و سوق داده میشوند . در این کامپیوتر ها ، نرم افزار های اسکادا وقایع را مدیریت کرده و داده ها را به اپراتور نمایش میدهند تا او بتواند این وقایع را تحلیل کند و به کنترل و اقدامات مناسب را انجام

نیازی به حضور پرسنل در طی فرایند مهم صنعتی اجباری نباشد .

امکانات سیستم اسکادا :

- **سیستم اسکادا این امکان را به ما میدهد که فرایند های صنعتی را به صورت محلی یا در مناطق دور افتاده کنترل کنیم .**
- **نظارت ، جمع آوری و پردازش های زمان واقعی را داشته باشیم .**
- **به طور مستقیم با تجهیزاتی مانند سنسور ها ، ولو ، پمپ ، موتور و موارد دیگر از طریق نرم افزار واسط انسان و ماشین (HMI) ارتباط برقرار کنیم .**
- **وقایع را در یک پرونده ثبت ، ضبط کنیم .**
- **عدم نیاز به حضور پرسنل در طی فرایند صنعتی.**
- **ثبت دقیق و نگهداری داده و امکان رجوع به آنها در آینده .**

اجزا سیستم اسکادا :

در حالت کلی میتوان گفت که هر سیستم اسکادا تشکیل شده از چند زیر سیستم است .

- **یک دستگاه (Apparatus) نمایشگر که توسط اپراتور انسانی مورد استفاده قرار میگیرد و تمام داده های فرایند برای اپراتور به نمایش در خواهد امد .**
- **یک سیستم نظارتی که وظیفه آن جمع اوری اطلاعات و داده از تمام فرایند است .**
- **واحد های ترمینال راه دور (RTU) که به سنسور های سیستم متصل هستند سیگنال های سنسور ها را به داده های دیجیتال تبدیل میکنند .**

فلسفه عملکرد اسکادا :

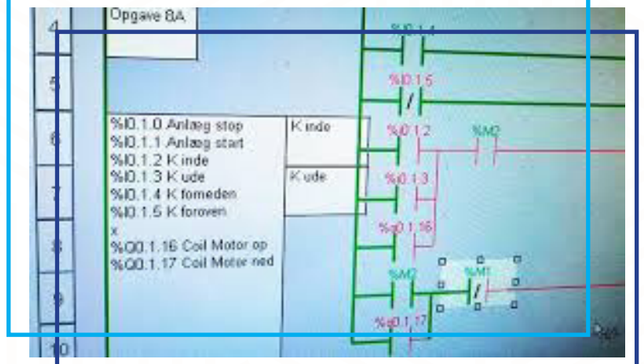
در برخی از سیستم‌ها، هزینه ناشی از خرابی سیستم کنترل بسیار بالا است و یا حتی نتایج بسیار ناگواری را به دنبال دارد. به همین دلیل در برخی از سیستم‌های اسکادا، سخت‌افزارها در برابر تحمل مقدار بالای دما، فشار و ولتاژ و ارتعاش مقاوم می‌شوند. اما در برخی از کاربردهای بحرانی، با استفاده از کانال‌های مخابراتی و سخت‌افزارهای اضافه، قابلیت اطمینان سیستم ارتقا داده می‌شود. در این سیستم‌ها، زمانی که یکی از بخش‌های سیستم آسیب ببیند و یا در عملکرد خود دچار مشکل شود، می‌تواند به سرعت شناسایی شود و سپس عملکرد آن به صورت اتوماتیک به سخت‌افزار پشتیبان انتقال داده شود. عمل جایگزینی باید بدون اخلال در کنترل فرایند انجام گیرد.

سیستم اسکادا چگونه کار میکند؟

همانطور که گفته شد سیستم SCADA مجموعه ای از قطعات سخت افزاری و نرم افزاری است . این مجموعه از اجزا با داده های real-time جمع‌آوری شده از دستگاه های واقع در صنعت همچون پمپ ها و ترنسمیتر ها شروع به کار میکند .نیازی نیست که این اجزا از فروشنده خاصی خریداری شوند ، تنها نیاز است تا یک پروتکل ارتباطی داشته باشند که پردازنده بتواند از ان استفاده کند ؛ سپس داده های جمع آوری شده از دستگاه های میدانی به پردازنده هایی مانند PLC ارسال میشوند. داده ها از پردازنده ، به سیستمی از دستگاه شبکه توزیع میشوند . این دستگاه ها ممکن است HMI ، کامپیوتر های کاربر نهایی یا سرور ها باشند . در HMI و کامپیوتر های کاربر نهایی امکان نمایش گرافیکی عملیات به منظور تعاملات اپراتور و کارهایی نظیر روشن کردن پمپ ها و باز کردن ولو ها وجود دارد . این داده ها همچنین ممکن است که برای تجزیه و تحلیل و تولید کارخانه و عیب یابی مشکلات استفاده شوند .

برنامه نویسی PLC

رضا صلح میرزایی



برنامه نویسی PLC وظیفه مهمی در طراحی و پیاده‌سازی برنامه کنترلی مورد نیاز شما در محیط صنعتی یا فضای مورد نظر شما است. یک برنامه PLC شامل مجموعه‌ای از دستورالعمل ها به صورت متنی یا گرافیکی است که نشان دهنده منطقی است که برای برنامه های صنعتی خاص در زمان واقعی اجرا مورد نیاز است.

چه نوع زبان برنامه نویسی برای پی ال سی ها استفاده می شود؟

منطق نردبانی (Ladder Logic) رایج‌ترین زبان برنامه نویسی است که برای کنترل کننده های منطقی قابل برنامه ریزی (PLC) در دنیا استفاده می شود. البته زبانها و منطقهای دیگری نیز برای برنامه نویسی PLC وجود دارد که در زیر لیست آنها را مشاهده می‌کنید:

- Instruction List
- Function Block Diagram
- Structured Text
- Sequential Function Charts

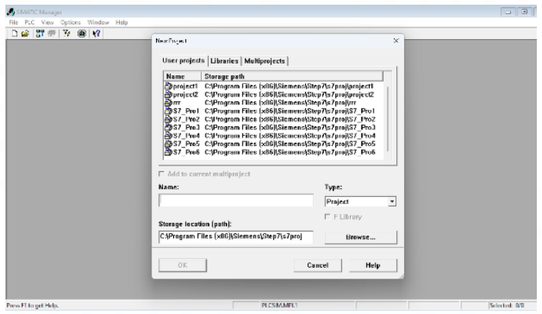
همه موارد فوق زبان های برنامه نویسی مفیدی هستند و بسته به کاربرد ممکن است مناسب تر از منطق نردبانی باشند.

راه اندازی یک موتور سه فاز به صورت چپگرد و راست گرد با زبان لدر در PLC Siemens:

نکات اساسی قبل از نوشتن برنامه PLC به زبان نردبانی نمودار (LD)،:

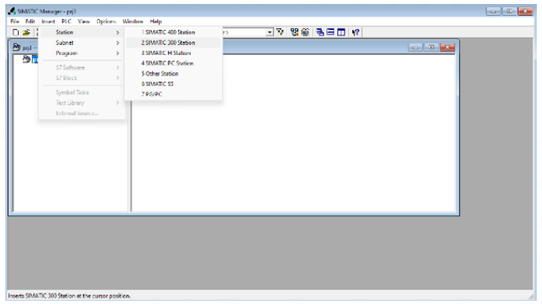
- مفهوم اصلی یک گیت منطقی در LD
- قوانین برنامه نویسی PLC
- دستورالعمل های برنامه نویسی PLC
- نحوه آدرس دهی در دستورالعمل های PLC

ابتدا وارد نرم افزار SIMATIC Manager میشویم.



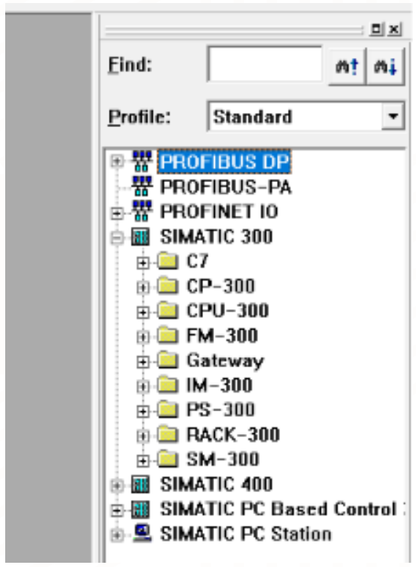
شکل٤- ١

New Project => نام ,آدرس , مسیرفایل پروژه



شکل٤- ٢

sation=> مدل و نوع پروگرامر



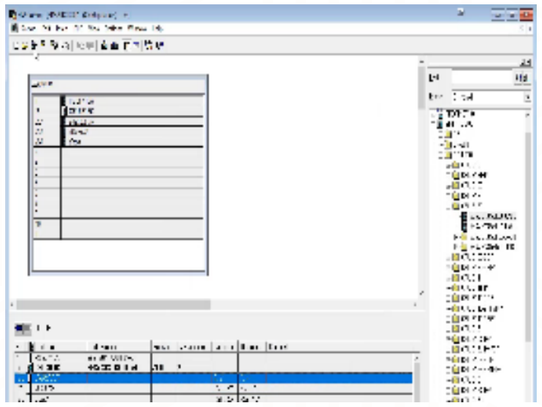
شکل٤- ٣

کاتالوگ تجهیزات=(Hardware Catalog) رک ، منبع تغذیه، cpuو...

Rail <= RACK

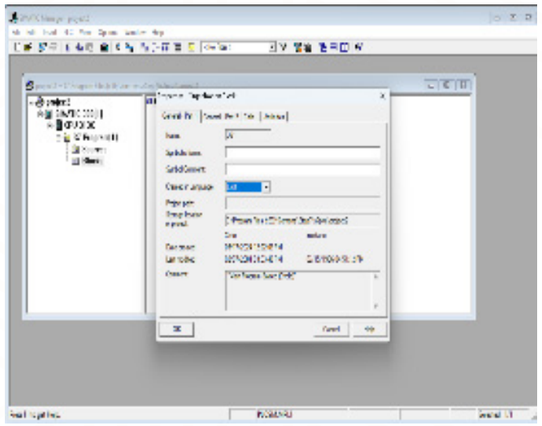
Ps-300 = منبع تغذیه

CPU =CPU-300 Digital Input (DI)و{ DO) {Digital Output



شکل٤- ٤

تنظیمات ورودی خروجی و آدرس دهی

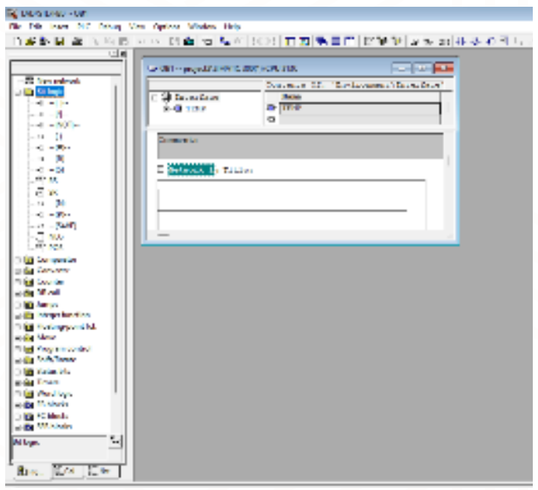


شکل٤- ٥

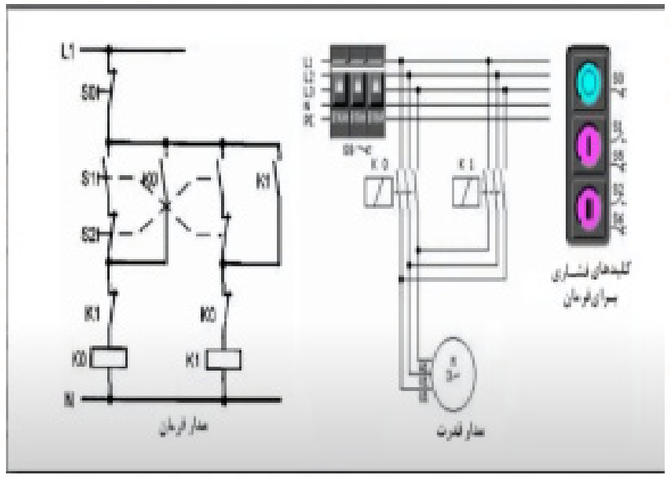
انتخاب زبان => Blocks -->OB1 -->LAD

دستورات عملیات منطقی روی بیت، عملیاتی را روی یک بیت انجام می¬دهند. برای دسترسی از قسمت Program Elements در برنامه LAD / STL / FBD پوشه ی Bit Logic.

Address	Data Type	Area
<Bit>	Bool	I,Q,M,L,D



شکل٤- ٦

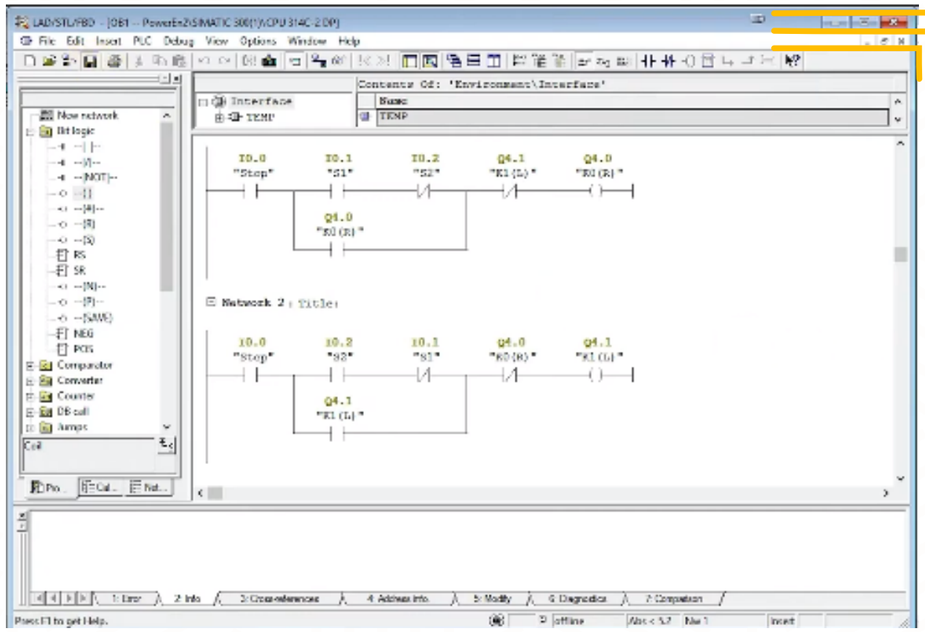


شکل٤- ۷ مدار فرمان راه اندازی یک موتورسه فاز به صورت چپگرد و راست گرد

S0 = Stop

S1= Right

S2= Left



شکل٤- ۸

بخش ویژه

ربات صنعتی دلتا

زهرا شاهسونی

ربات دلتا یک ربات صنعتی بازو ـ موازی است. ربات‌های دلتا ساختاری سه پایه دارند، که شامل یک پایه اصلی و سه بازوی حرکتی هستند. این بازوها از مفاصل متحرکی تشکیل شده‌اند که امکان حرکت چند محوری را فراهم می‌کنند. این نوع ربات ها به صورت نصب شده در بالا یا سقف طراحی شده‌اند و موتورها در قسمت پایه قرار می‌گیرند. به دلیل ساختار سه پایه و قابلیت حرکت سریع و قدرتمند خود، برای کاربردهای صنعتی و تولیدی مورد استفاده قرار می‌گیرند.

شرکت‌های تولید کننده:

ABB Robotics

KUKA Robotics

Fanuc

Yaskawa Electric Corporation

کاربرد ها:

این ربات‌ها به دلیل سرعت و دقت بالای خود برای بسته بندی، مرتب سازی و جابجایی در صنایع داروسازی، غذایی و نوشیدنی، بسته بندی، خرده فروشی، آرایشی و بهداشتی و پزشکی استفاده می‌شوند.

همچنین به دلیل طراحی خاص در برخی صنایع مورد استفاده قرار نمی‌گیرند.در صنایع سنگین و ساختمانی به دلیل نیاز به جابجایی و حمل بارهای سنگین، ربات‌های دلتا که برای کارهای دقیق و سبک طراحی شده‌اند، در این صنایع کاربرد ندارند.

در صنایع معدن به دلیل شرایط سخت و محیط‌های خشن، ربات‌های دلتا که برای محیط‌های تمیز و کنترل‌شده طراحی شده‌اند، در این صنایع استفاده نمی‌شوند.در صنایع کشاورزی نیز به دلیل نیاز به کار در محیط‌های باز و متغیر، ربات‌های دلتا که برای کار در محیط‌های ثابت و کنترل‌شده طراحی شده‌اند، در این صنایع کاربرد ندارند. در صنایع نظامی به دلیل نیاز به ربات‌های مقاوم و چندمنظوره، ربات‌های دلتا که برای کارهای خاص و دقیق طراحی شده‌اند، در این صنایع استفاده نمی شوند.

OMRON

Specifications

Delta robot XXL

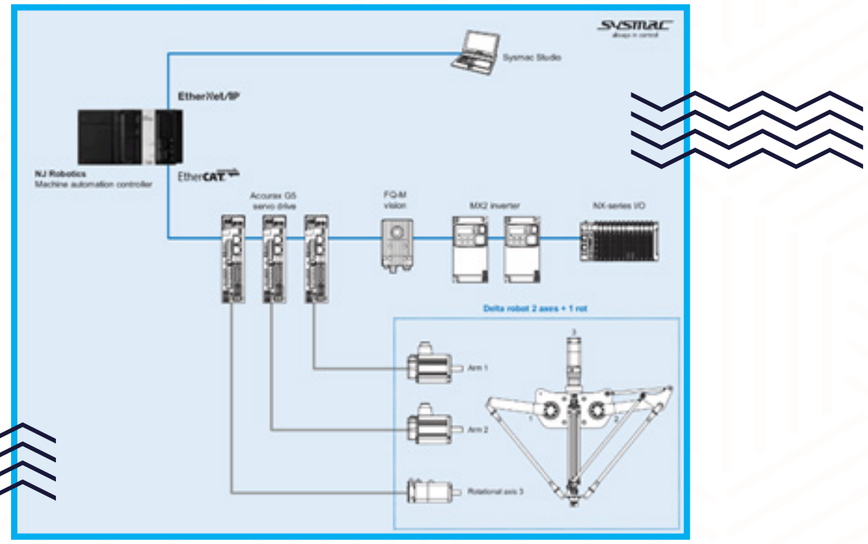
Model			CR_UGD21500_R
Working volume	X axis (stroke)		1,500 mm
	Z axis (stroke)*1		347 mm (maximum 1,500 mm)
	θ axis (rotation angle)		±180 deg (default setting, it can be changed)
Servo motor	Arm 1, 2	Model	R88M-K3K030C-BS2
		Capacity	3000 W
	Rotational axis 3	Model	R88M-K40030T-BS2
		Capacity	400 W
Repeatability*2	X, Z axis		±0.4 mm
	θ axis		±0.2 deg

شکل ۵-۱ کاتالوگ ربات دلتا محصول شرکت اومرون

هر رباتی با درجات آزادی مشخص می‌شود ، به صورت ساده یعتی هر مفصل یک درجه.

برای هر مفصل باید یکه موتور در نظر گرفت. در این قسمت توضیح داده شده که زاویه θ=180° است.

یعنی هر بازو که روی یک مفصل که همان سروو موتور است نصب شده، ۱۸۰ درجه می تواند جابجا شود. بازوی ۱ و ۲ هر کدام از سروو موتور با توان ۳۰۰۰ وات هستند.قسمت اند افکتور یا همان انتهایی از یک سروو دیگر با توان ۴۰۰ وات استفاده شده است. تکرار پذیری یکی از مشخصات ربات‌ها است یعنی اینکه بتواند کار یک انسان را انجام بدهد و در تکرار کار دچار خطا نشود، هر ربات برای رسیدن به هدف درصد خطایی دارد که شعاع حرکتی این خطاها را حساب می‌کنند. در این مدل خطای تکرار پذیری ±0.4 mm است.

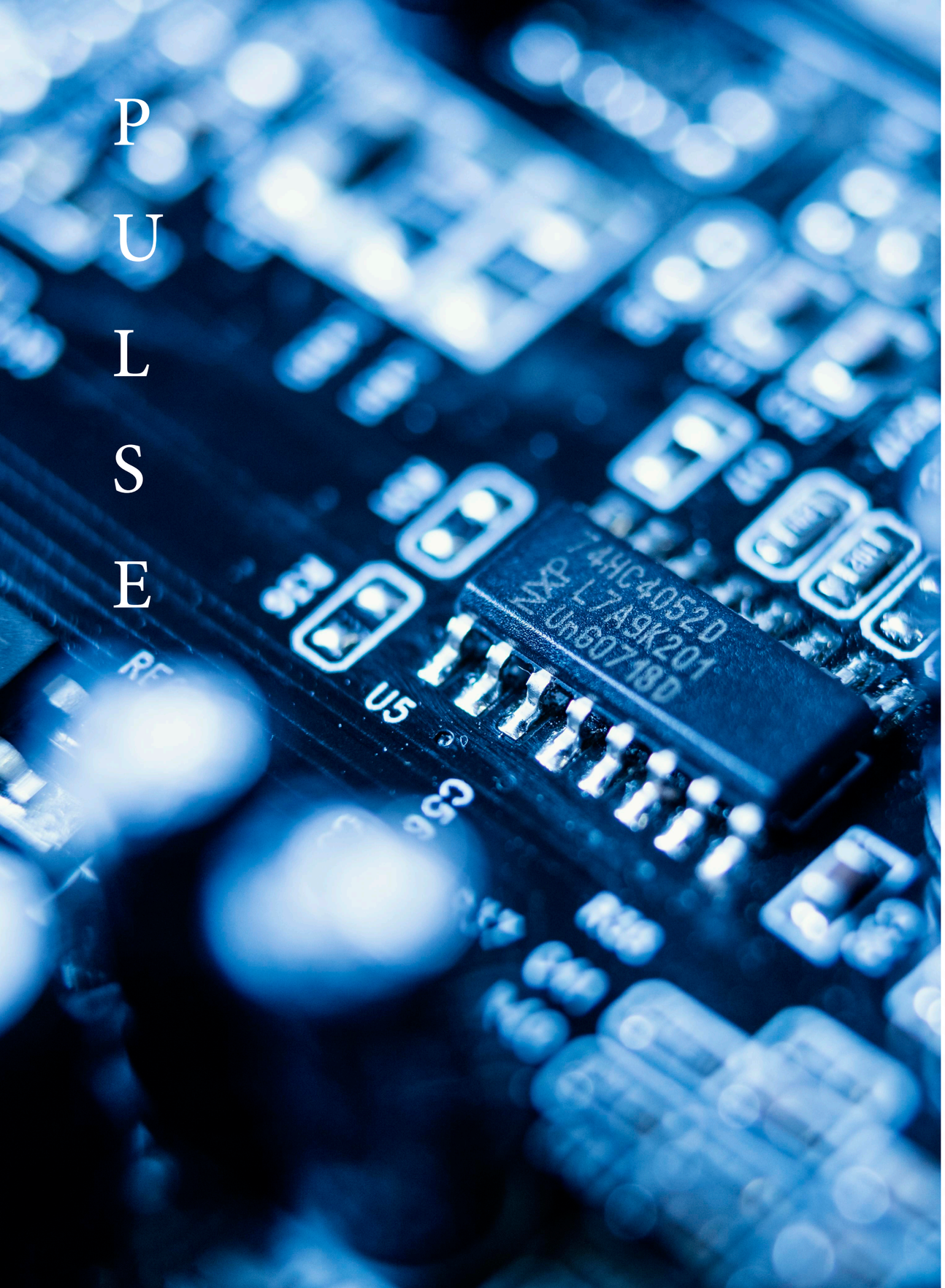


شکل ۵-۲ شکبه کنترلی ربات دلتا

برای کنترل ربات های دلتا از شرکت Omron، از کنترلرهای NJ و NX استفاده می‌شود. در این دو کنترلر بسیار قدرتمند هر دو عملکرد PLC و Motion Control وجود دارد. PLC های قدیمی برای جایگزین تعداد زیادی کلید و رله که کنترل مدارهای الکتریکی را انجام می دادند طراحی شده‌اند. اما یک Motion Control، برای کنترل سروو موتور، استپر موتور، شیرهای پروپرشنال هیدرولیکی بسیار دقیق، استفاده می‌شود و همچنین برای کنترل پوزیشن، کنترل سرعت و گشتاور کاربرد دارد. (کاری که برای ربات می خواهیم)

در این سری، PLC و Motion Control را در یک پکیج طراحی کرده‌اند.

کنترلرهای سری NJ و NX می‌توانند تمامی عملکردهای PLC را از طریق ورودی و خروجی‌های دیجیتال و آنالوگ انجام داده و در همان لحظه می‌توانند عملکردهای Motion Control را با دقت و سرعت خیلی بالا انجام بدهند. مانند بسیاری از کنترلرهای سری NX، مدل NJ501 قابلیت IIOT با OPC UA را دارد و به راحتی می‌تواند عملکرد دیتا بیس را در خودش جا بدهد، قابلیت کنترل تا 64 محور را دارد که در کاتالوگ به عنوان axes داریم. نرم افزار کامپیوتری Sysmac، یک ماشین کنترل و اتوماسیون کارخانه است که برای برنامه نویسی سری NX/NJ از کنترلرها استفاده می‌شود.



منابع

https://www.redhat.com

https://en.wikipedia.org

https://www.arm.com

https://www.techtarget.com

https://www.zdnet.com

https://irandoc.ac.ir

https://civilica.com

https://daneshyari.com

https://evsint.com

https://ariantp.com

https://github.com

https://www.zoomit.ir

https://www.microsoft.com

ایران اتوماسیون، زیمنس کنترل،کالنجی،مهندس یار

کتاب plc مقدماتی مهندس اسدالله کاظمی

کتاب مهندسی کنترل اوگاتا

کتاب مهندسی سیستم های کنترل نایس